



**UNIVERSIDAD  
DE ANTIOQUIA**

# **AUTOMATIZACIÓN DEL PROCESO DE RECONEXIÓN DE USUARIOS EN FECHAS DE CORTE CON USO DE LA API DE MIKROTIK**

**Eduardo José Maya Rodríguez**

**Universidad de Antioquia  
Facultad de Ingeniería. Departamento de Ingeniería electrónica y  
telecomunicaciones  
Medellin, Antioquia  
2021**



AUTOMATIZACIÓN DEL PROCESO DE RECONEXIÓN DE USUARIOS EN FECHAS  
DE CORTE CON USO DE LA API DE MIKROTIK.

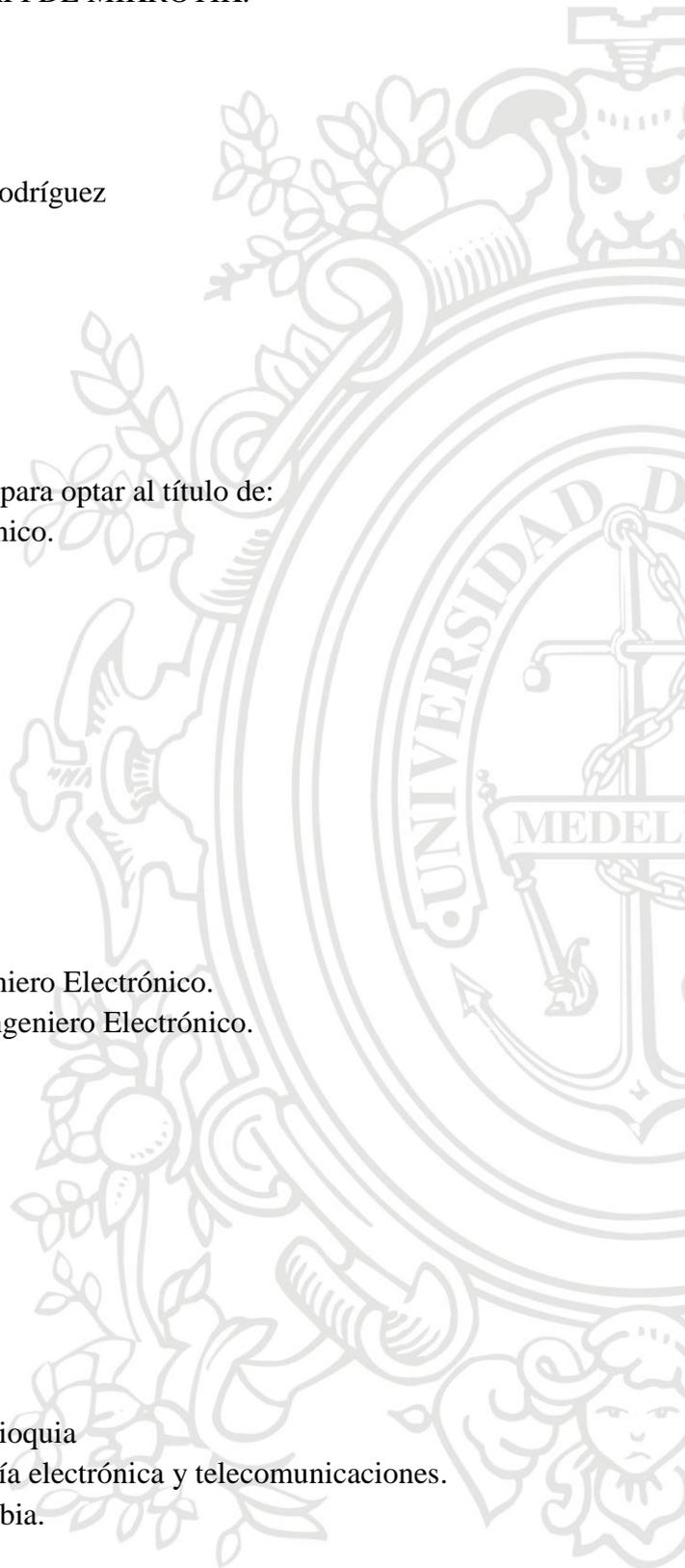
Eduardo José Maya Rodríguez

Informe de práctica como requisito para optar al título de:  
Ingeniero Electrónico.

Asesores.

Juan Pablo Urrea Duque. Ingeniero Electrónico.  
Jairo Hernán González Vallejo. Ingeniero Electrónico.

Universidad de Antioquia  
Facultad de Ingeniería, Departamento de ingeniería electrónica y telecomunicaciones.  
Medellín, Colombia.  
2021.



## Resumen

En la empresa ISANETWORKS ISP SAS, todos los meses se realizan suspensiones en las fechas de corte para aquellos usuarios que no hayan realizado el pago oportuno de la mensualidad del servicio de conexión a internet que tienen suscrito. Esta tarea es realizada por el equipo de auxiliares de sistemas, llevada a cabo de manera manual, lo cual, al ser una tarea repetitiva, tiende volverse tediosa debido al gran volumen de solicitudes de activación. Esto ocasiona en algunas oportunidades errores en los procedimientos, perjudicando la calidad del servicio prestado, retardos en atención al equipo de técnicos en campo y complicaciones en la recepción de llamadas para realizar soporte al cliente.

Por tal razón, se propuso como solución, la automatización del proceso de activación de usuarios. Esta labor de activación de usuarios es lograda con ayuda de un programa elaborado en el lenguaje Python usando librerías como *"librouters"* y *"routers\_api"*, llevando a cabo, de manera ágil la activación de los usuarios en el menor tiempo posible. Logrando que una tarea que puede tomar hasta una semana en ser completada de manera manual sea realizada en tan solo 20 minutos por medio de un script que ejecute la tarea de activación automática. Permitiendo que este tiempo sea liberado para cumplir la función de otras tareas como soporte a los equipos en campo para la realización de mantenimientos e instalaciones, brindar soporte técnico a los clientes que lo requieran ayuda vía telefónica entre otras labores que les sean conferidas al equipo de auxiliares de sistemas de la oficina.

## Introducción

Actualmente la necesidad de mantenerse conectados por medio de un punto de acceso a internet es algo que ha ido escalando a lo largo del tiempo hasta convertirse en una prioridad. Para los habitantes del departamento del Putumayo y el sur del departamento del Huila no es la excepción, aún más debido a la presión ejercida por los últimos acontecimientos globales. La mayoría de las actividades sean laborales o educativas se han volcado a un entorno virtual en el cual es requerida una conexión a internet estable y rápida.

Por tal razón, ISANETWORKS comienza operaciones en el año 2011 en el municipio de Villagarzón (Putumayo), ofreciendo servicio de internet por medio de sistemas de radio enlace. Para el año 2013 se constituye legalmente como ISANETWORKS ISP SAS. Para el año 2016, ISANETWORKS ISP SAS estableció su oficina en el municipio de Mocoa (Putumayo).

Con el establecimiento de su nueva sede en el municipio de Mocoa, ISANETWORKS ISP SAS cambió su tecnología de radio enlace por una red de conexión por fibra óptica, con tecnología FTTH (Fiber To The Home) y GPON (Gigabit-Capable Passive Optical Network) ampliando su cobertura en los municipios de Villagarzón y Mocoa en el departamento del Putumayo y Pitalito en el departamento del Huila, teniendo un total de 4000 usuarios distribuidos entre las 3 ciudades.

En los 6 primeros días de cada mes se realizan suspensiones del servicio en las fechas de corte de servicio para aquellos usuarios que no hayan realizado el pago oportuno de la mensualidad del servicio de conexión a internet que tienen suscrito con la empresa. Durante estos días aumenta considerablemente el volumen de usuarios que se acercan a realizar el pago con el fin de que su servicio no se vea interrumpido. Dicha labor se realiza de manera manual por parte del equipo de auxiliares de sistemas. Para efectuar esta tarea el equipo de ingeniería recibe como único dato el recibo de pago de la mensualidad en la cual se tienen datos como nombres completos y número de documento de identidad del usuario que realiza el pago y alguna anotación especial como: solicitud de cambio de contraseña, si el usuario tiene más de una línea en servicio, entre otras.

El proceso de activación era tedioso, debido a la realización de la activación manual de los usuarios en las 3 ciudades, ya que se vuelve una tarea sumamente repetitiva y voluminosa. Por tanto, este proceso puede presentar errores a la hora de usuarios, por ejemplo al activar usuarios que no han realizado el pago e información no existente en la base de datos (dirección IP que no le corresponde al usuario). Este tipo de errores afecta el tiempo de reactivación de servicio, aumento en los tiempos de respuesta a los equipos técnicos en campo que deben comunicarse con los auxiliares de sistemas para la resolución de problemas de conexión y retrasos en las instalaciones nuevas o traslados de usuarios.

Para dar solución a este problema se implementó un programa automatiza esta tarea, por medio del uso del servicio API presente en el sistema operativo RouterOS de los equipos MikroTik. El uso del servicio API se realiza con ayuda de un programa elaborado en lenguaje de programación Python. Al poner el programa en funcionamiento se lograron obtener buenos resultados en cuanto a disminución de tiempo de reactivación de los usuarios, y reducción en carga de trabajo para los auxiliares de sistemas puesto que una tarea que podría tomar hasta una semana para un solo auxiliar de sistemas es finalizada por completo en media jornada de trabajo. Logrando que ese tiempo disponible se enfoque a labores de atención a los equipos desplegados en campo para mantenimientos e instalaciones y atención a usuarios que requieran algún tipo de soporte técnico.



## Objetivos

- **General:**

Desarrollar una aplicación que permita la gestión de reconexión de los usuarios de la empresa ISANETWORKS ISP SAS durante las fechas de corte mediante el uso del servicio de API presente en los equipos MikroTik con el fin de mejorar la calidad en atención al usuario y soporte técnico en los servicios de telecomunicaciones prestados.

- **Específicos:**

- Identificar las deficiencias presentes en el proceso de reconexión de usuarios.
- Construir una base de datos que permita la administración ágil y eficiente de los usuarios a los cuales se les presta el servicio de telecomunicación.
- Desarrollar una aplicación en la cual se hará uso del servicio de API que permitan la integración del servicio con las aplicaciones de gestión presentes en la empresa.
- Automatizar la gestión de reconexión de los usuarios implementando una aplicación por medio del uso del servicio de API con el fin de mejorar los tiempos de reconexión de los usuarios en las fechas de corte.
- Evaluar la mejora en la calidad de servicio en la gestión de usuarios de la empresa ISANETWORKS ISP SAS con el desarrollo de la aplicación.

## Marco Teórico

ISANETWORKS ISP SAS en su interés de formar parte de la demanda regional de acceso a internet, a lo largo de estos últimos años ha decidido actualizar toda su infraestructura de comunicación pasando de sistemas por radio enlace a conexiones de fibra óptica, exactamente tecnologías FTTH (Fiber To The Home), en conjunto con tecnologías de red GPON (Gigabit-Capable Passive Optical Network), con el objetivo de ampliar y mejorar los servicios ofrecidos para los habitantes de Mocoa y Villagarzón en el departamento del Putumayo y Pitalito en el departamento del Huila.

El funcionamiento en el cual se basa la tecnología FTTH bajo la arquitectura GPON permite enlazar un dispositivo óptico de nodo (ONT/ONU-Optical Node Termination/Unit) ubicada en el lugar de residencia del suscriptor del servicio a la central ubicada en las instalaciones de la empresa, donde, entre extremo y extremo se tienen una serie de elementos de multiplexación de la señal (Splitters) para abarcar el mayor área de conexión posible[1] (Figura 1).

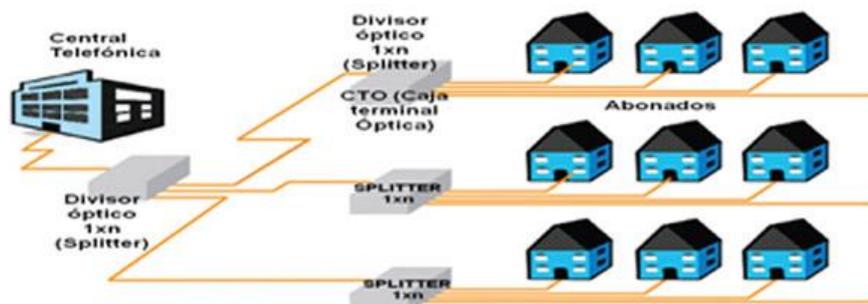


Figura 1. Arquitectura de conexión de la fibra bajo tecnología FTTH.

Las tecnologías GPON permiten alcanzar anchos de banda del orden de 2.4Gbps para descarga y 1.2Gbps para carga[2], con el cual se facilitó el desarrollo de la transmisión de voz (VoIP), datos (Internet) y televisión (IPTV), incluyendo el video bajo demanda (VOD). En cuanto a implementación de equipos físicos en la red por su premisa de ser pasivo, es decir, no requerir elementos que consuman energía, hace que su despliegue sea económico respecto a otras soluciones de transmisión óptica.



**Figura 2. Unidad Óptica de Nodo (ONT/ONU) que se instala en la residencia del usuario.**

En materia de seguridad, estas tecnologías cuentan con encriptación AES (Advanced Encryption Standard), norma establecida por la unión internacional de telecomunicaciones (ITU-International Telecommunication Union) conocida como el estándar G.984.1 [3], en el cual se establecen un conjunto de recomendaciones respecto a las técnicas de encapsulamiento de información, gestión y transporte de red, principalmente.

Para la administración de la red en la central, encargada de dar salida a internet a los usuarios, se emplean equipos que reciben la señal óptica proveniente de los equipos ONU/ONT (Figura 2) ubicados en las residencias o negocios de los clientes. Estos equipos son conocidos como OLT (Optical Line Termination).



**Figura 3. OLT SmartAX MA5800 X15.**

Esta OLT (Figura 3) es un elemento activo en la red, donde parten las fibras ópticas hacia los usuarios. Integra las funciones de switch Capa 2/Capa 3 en el sistema

GPON[4]. Su función principal es controlar desde la oficina central la información que se transmite a lo largo de la red (ODN-Optical Distribution Network).

A parte del OLT, también se tienen equipos de enrutamiento, encargados de la salida hacia internet y administración de la red implementada para la conexión de los usuarios, estos equipos son de la compañía MikroTik (Figura 4). MikroTik es una empresa Letona (Latvia) fundada en 1996, encargada del diseño de equipos de enrutamiento y software auxiliar diseñado para empresas proveedoras de servicios de internet (ISP) ofreciendo en su portafolio equipos a muy bajo costo con excelentes prestaciones, poseen una buena documentación en línea gracias a su wiki[5] y continuas actualizaciones del sistema operativo RouterOS, estos equipos tienen soporte continuo que se encuentra en su portal web, sin contar los numerosos foros y una comunidad activa presente para el mantenimiento de equipos.



**Figura 4. Imagen de muestra equipo MikroTik CloudCore CCR1072.**

RouterOS es el sistema operativo diseñado por MikroTik, cuyo kernel está basado en LINUX enfocado a la administración y trabajo en redes de comunicaciones para empresas ISP. Este sistema operativo se puede instalar en dispositivos de arquitectura x64 y x86[6].

Para complementar el equipo MikroTik y realizar la administración del router de manera más simple, MikroTik también desarrolló un programa llamado WinBox (Figura 5). Este programa está disponible solo para el sistema operativo Windows, aunque puede ser ejecutado en otras plataformas como Ubuntu y MacOS. Desde este programa es posible conocer las interfaces de red del equipo MikroTik que el equipo tenga. También es posible determinar direcciones de enrutamiento para redes de área local para el equipo, así mismo, descubrir otros equipos MikroTik que sean accesibles desde nuestra red. Para el caso de ISANETWORKS facilita el seguimiento de conexión de todos los usuarios que estén suscritos al servicio de internet, ajustar el plan que tienen contratado, verificar si cada ONT está enlazada a la red y crear reglas

de firewall para controlar el tráfico generado y proporcionar seguridad de los usuarios.

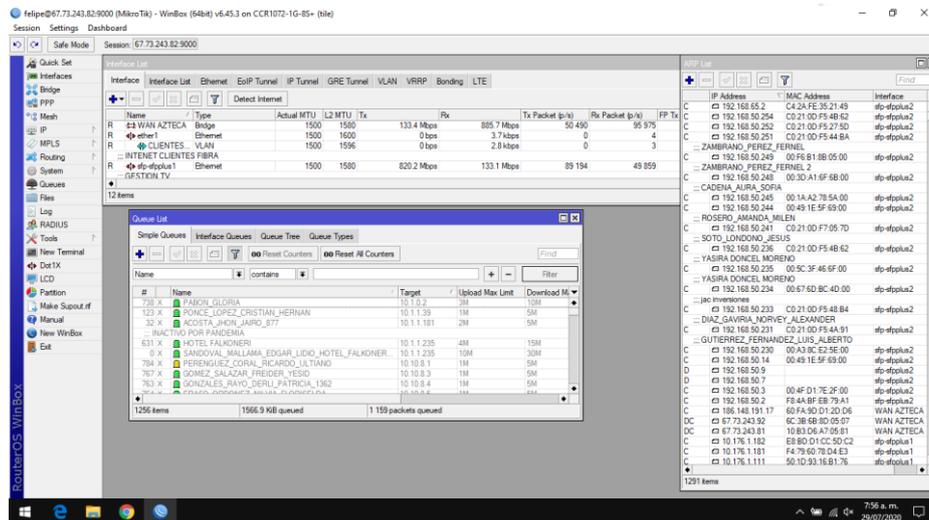


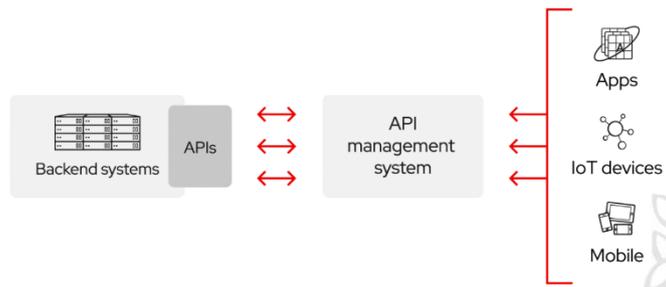
Figura 5. Interfaz de WinBox.

Dentro de los múltiples servicios ofrecidos por RouterOS, existe la posibilidad de emplear el servicio de API (Application Program Interface) para diferentes lenguajes de programación como JAVA, C++, PHP y Python, entre otros[7].

Una API es un conjunto de aplicaciones y protocolos para desarrollar e integrar software, por lo general muchos fabricantes permiten realizar este tipo de integraciones con el fin de lograr una solución que se ajuste a las necesidades de quien lo requiera.

Una API permite la comunicación entre productos y servicios sin la necesidad de conocer cómo están implementados en profundidad, lo que facilita un ahorro en tiempo y dinero. Las API se caracterizan por su flexibilidad y capacidad de adaptación para el diseño de herramientas y productos nuevos e innovadores.

Las API (Figura 6) ayudan al trabajo de un desarrollador de programas, ya que no debe crear código desde cero. Proveen al desarrollador el uso de funciones para la interacción con el sistema operativo con otros programas[8].



**Figura 6. Esquema de funcionamiento de una API.**



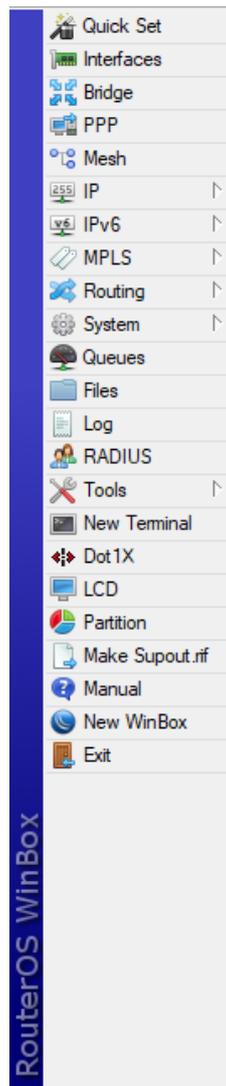
## **Metodología**

El desarrollo del proyecto consistió en 4 etapas, a fin de completar en su totalidad el cronograma propuesto para su realización. La primera etapa fue dedicada a la capacitación en funcionamiento y conocimientos de los equipos instalados en la oficina para la administración de la red. Finalizada esta etapa se realizó la segunda etapa, la cual consiste en definir cómo realizar la conexión del servicio API, es decir, bajo qué lenguaje de programación se va a realizar el desarrollo del proyecto y que herramientas se encontraban disponibles para el desarrollo. La tercera etapa consistió en el desarrollo del programa para la activación de usuarios y creación de una base de datos consistente de la cual se va a alimentar el programa para hacer control de los usuarios que fueron habilitados y realizar el proceso de validación. La etapa final es la comprobación y puesta a prueba del desarrollo elaborado para este proyecto. Todo este proceso es llevado a cabo en los equipos MikroTik de la oficina central, ubicada en el municipio de Mocoa (Putumayo).

### **ETAPA 1: Capacitación en funcionamiento de los equipos MikroTik.**

Para la administración de los usuarios suscritos al servicio de internet se tienen los equipos MikroTik CCR1072[9]. En estos equipos se realizó una inducción sobre cómo funciona la red de internet de ISANETWORKS ISP SAS, donde es posible acceder a su configuración de manera gráfica con el software WinBox, propiedad del fabricante de los equipos MikroTik. El programa puede ser obtenido desde la página del fabricante y es de libre descarga.

Este programa permite ejecutar una serie de opciones tanto para mantenimiento del equipo como a nivel de administración de la red que se está configurando.



**Figura 7. Menú de opciones de WinBox.**

Para la parte de administración de la red principalmente se tienen las opciones IP y Queue. En el apartado de IP están las opciones más usadas para la administración:

- Elaboración direccionamientos permitidos para las diferentes interfaces físicas o lógicas (por medio de VLAN).
- Enrutamientos para dar salida a internet y alcanzar otras redes que estén definidas (Opción IP/Routes).
- Crear servidores DHCP para la obtención autónoma de direcciones IP.
- Definir los servidores DNS para la resolución de direcciones.
- crear Pool de direcciones para ser asignadas por el servidor DHCP.

ARP
Accounting
Addresses
Cloud
DHCP Client
DHCP Relay
DHCP Server
DNS
Firewall
Hotspot
IPsec
Kid Control
Neighbors
Packing
Pool
Routes
SMB
SNMP
Services
Settings
Socks
TFTP
Traffic Flow
UPnP
Web Proxy

**Figura 8. Funciones dentro de la opción IP.**

En la opción IP se encuentra la lista ARP (Address Resolution Protocol) en la cual se mapean (listan) todas las direcciones IP (Capa 3 Modelo OSI) y direcciones MAC (Capa 2 Modelo OSI) asociados a dispositivos de red (ONT) que están conectados en la red que administra el equipo MikroTik, lo que se traduce en los usuarios inscritos al servicio de internet. Cada dato presente en esta lista lo componen una dirección IP, una dirección MAC, la interfaz de precedencia y una etiqueta (tag) indicadora del estado en el cual se encuentra cada dirección IP asignada a cada usuario, las etiquetas tienen los siguientes valores mostrados en la tabla 1:

Tabla 1. Valores de la etiqueta para direcciones presentes en la lista ARP.

Etiqueta	Descripción
X	Deshabilitado (Disabled)
C	Completo (Complete)
D	Dinámico (Dynamic)
P	Publicada (Published)
H	DHCP

Es posible tener dentro de la lista ARP una combinación de estas etiquetas, donde la combinación más común es DC (Dynamic-Complete), tal y como se ve en la Figura 9.

	IP Address	MAC Address	Interface
C	186.148.191.17	60:FA:9D:D1:2D:D6	WAN AZTECA
DC	67.73.243.92	6C:3B:6B:8D:05:07	WAN AZTECA
DC	67.73.243.81	10:83:D6:A7:05:81	WAN AZTECA
C	10.176.1.182	E8:BD:D1:CC:5D:C2	sfp-sfpplus1
C	10.176.1.181	F4:79:6D:78:D4:E3	sfp-sfpplus1
C	10.176.1.111	50:1D:93:16:B1:76	sfp-sfpplus1
X	10.176.1.76	20:0B:C7:FC:60:35	sfp-sfpplus1
C	10.176.1.74	2C:97:B1:94:56:E1	sfp-sfpplus1
C	10.176.1.70	18:C5:8A:AF:CA:46	sfp-sfpplus1
C	10.176.1.69	18:C5:8A:BA:72:0B	sfp-sfpplus1
X	10.176.1.67	3C:78:43:87:F7:2D	sfp-sfpplus1
C	10.176.1.66	78:F5:57:30:CD:59	sfp-sfpplus1
C	10.176.1.65	BC:62:0E:22:37:6A	sfp-sfpplus1
X	10.176.1.64	D0:3E:5C:B9:82:54	sfp-sfpplus1
C	10.176.1.59	C4:47:3F:F5:B1:70	sfp-sfpplus1
C	10.176.1.58	C4:47:3F:F2:E7:9D	sfp-sfpplus1
C	10.176.1.57	54:39:DF:E4:8D:AB	sfp-sfpplus1
C	10.176.1.56	00:2E:C7:4A:9B:11	sfp-sfpplus1
C	10.176.1.53	80:41:26:B6:9C:44	sfp-sfpplus1
C	10.176.1.51	B4:15:13:03:65:D4	sfp-sfpplus1
C	10.176.1.49	B4:15:13:76:E7:21	sfp-sfpplus1
C	10.176.1.45	20:0B:C7:FC:0B:0A	sfp-sfpplus1
C	10.176.1.43	94:DB:DA:04:8E:A3	sfp-sfpplus1
C	10.176.1.41	F4:E5:F2:37:B9:97	sfp-sfpplus1
C	10.176.1.40	AC:85:3D:43:73:F4	sfp-sfpplus1
C	10.176.1.39	78:F5:57:33:DC:8C	sfp-sfpplus1
C	10.176.1.37	3C:78:43:87:94:F7	sfp-sfpplus1
C	10.176.1.36	20:0B:C7:FC:22:5C	sfp-sfpplus1
C	10.176.1.19	F4:4C:7F:BF:1A:44	sfp-sfpplus1
C	10.176.1.18	3C:78:43:63:BD:83	sfp-sfpplus1
C	10.176.1.15	2C:CF:58:8F:00:E0	sfp-sfpplus1
C	10.176.1.13	50:6F:77:D1:09:BA	sfp-sfpplus1
C	10.176.1.12	00:2E:C7:4F:49:76	sfp-sfpplus1
C	10.176.1.6	3C:E8:24:F5:D4:32	sfp-sfpplus1
C	10.176.1.2	80:41:26:AD:18:C6	sfp-sfpplus1
C	10.130.1.24	F4:E5:F2:37:B0:0B	sfp-sfpplus1

Figura 9. Lista ARP.

Dentro de la lista ARP es posible agregar direcciones IP de manera manual ingresando la Dirección IP, la dirección MAC del equipo y a que interfaz se asocia dicha dirección IP. Al ser agregada de manera manual pasa directo con etiqueta C. Como ejemplo se puede observar la Figura 10.

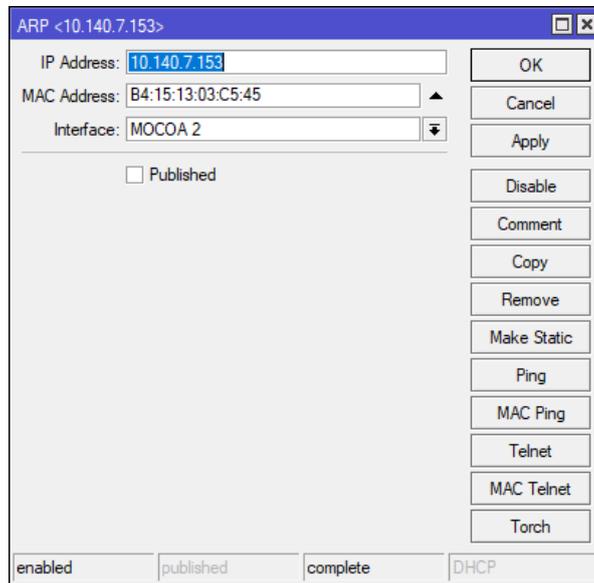


Figura 10. ARP usuario con IP 10.140.7.153.

También se define un perfil de ancho de banda y priorización de paquetes[10] asociado a las direcciones IP encontradas en la lista ARP. Esto se puede realizar gracias al apartado de Queue List (Figura 11), y en él se establece la velocidad de carga y descarga para cada usuario presente y que servicio tiene con la empresa. Esta función permite ver una gráfica de ancho de banda consumido y paquetes enviados y recibidos en tiempo real, lo que permite ver el comportamiento de la red de cada usuario y verificar en caso de fallas si existe problemas de conexión de parte del usuario o de parte de la empresa que presta el servicio.

#	Name	Target	Upload Max Limit	Download M.
738 X	PABON_GLORIA	10.1.0.2	3M	10M
123 X	PONCE_LOPEZ_CRISTIAN_HERNAN	10.1.1.39	1M	5M
32 X	ACOSTA_JHON_JAIRO_877	10.1.1.181	2M	5M
...	INACTIVO POR PANDEMIA			
631 X	HOTEL_FALKONERI	10.1.1.235	4M	15M
0 X	SANDOVAL_MALLAMA_EDGAR_LIDIO_HOTEL_FALKONER...	10.1.1.235	10M	30M
784 X	PERENGUEZ_CORAL_RICARDO_ULTIANO	10.10.8.1	1M	5M
767 X	GOMEZ_SALAZAR_FREIDER_YESID	10.10.8.3	1M	5M
763 X	GONZALES_RAYO_DERLI_PATRICIA_1362	10.10.8.4	1M	5M
764 X	ERASO_ODRONEZ_NILVA_FLORESDA	10.10.8.5	1M	5M

1256 items      2426.2 KiB queued      1 830 packets queued

Figura 11. Ejemplo de Queue List.

Cada Queue tiene como datos importantes nombre del usuario suscrito al servicio, dirección IP (Target), velocidad máxima de carga/descarga. Sin embargo, también es posible configurar más parámetros, como ráfaga de datos, tiempo que demora la ráfaga de datos. Estos datos adicionales permiten definir el servicio del cliente (Figura 12).

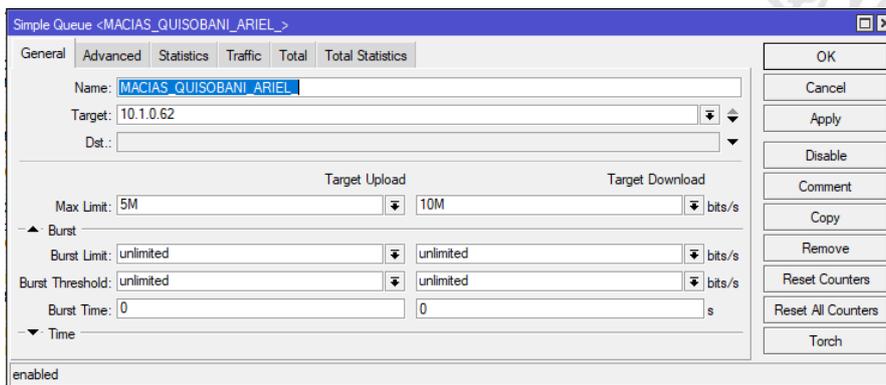


Figura 12. Ejemplo de Simple Queue.

A parte de configuraciones simples para cada usuario también se pueden establecer gráficos de consumo con el fin de ver en tiempo real la demanda de datos del usuario que está marcando desde la ONT ubicada en su residencia. Esta gráfica permite ver si el usuario tiene consumo y por ende posee servicio a internet activo (Figura 13).

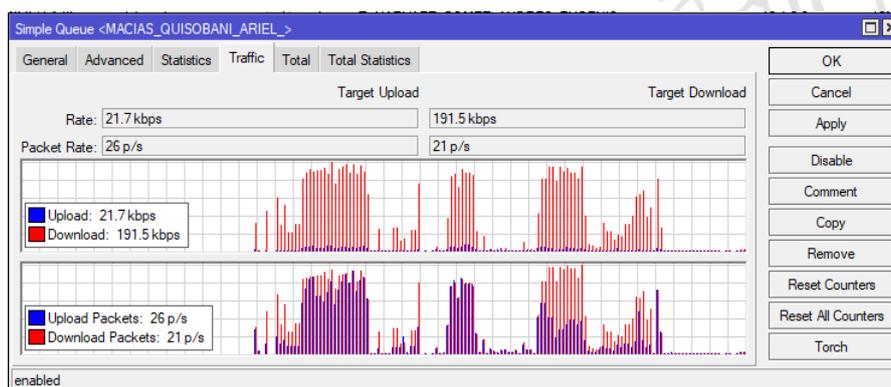


Figura 13. Visualización de tráfico de red de usuario de Simple Queue.

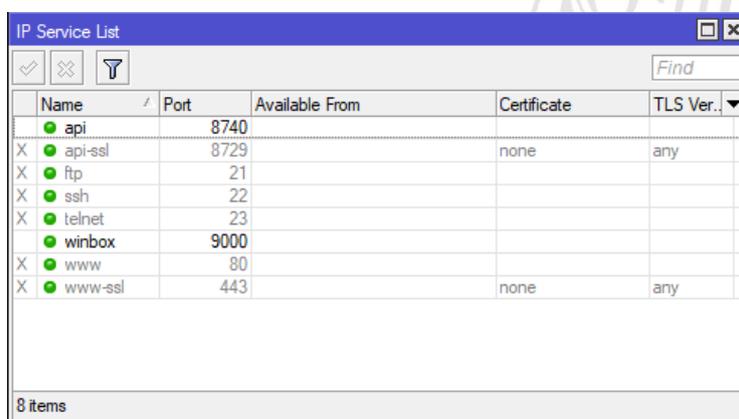
Cabe decir, que estos apartados también son accesibles gracias la programación por terminal, cuya herramienta se encuentra en WinBox, en la parte de terminal, ubicado en el menú principal definido en la figura 7.

## ETAPA 2: Conexión del servicio API.

Como se mencionó anteriormente, una API es un conjunto de aplicaciones y protocolos para desarrollar e integrar software. MikroTik permite la integración del sistema operativo RouterOS para configurar y administrar routers, API tiene una sintaxis cercana a una línea de interfaz de comandos (CLI-Command Line Interface).

La comunicación con el router es realizada por medio de la secuencia orden/respuesta, es decir, el usuario que quiere acceder a la terminal del equipo realiza una serie de órdenes y el equipo responde en la medida que estas órdenes son realizadas.

Para establecer el uso de API se recomienda tener una versión de RouterOS 3.X o superior. Para que se pueda establecer una comunicación y pueda ser visible desde WinBox para hacer pruebas de conexión, es importante que el servicio esté activo. Muchas veces por cuestiones de seguridad, expertos recomiendan que todo servicio que no se utilice debe ser deshabilitado para evitar ataques que comprometen la seguridad y la calidad de servicio prestado a los usuarios.

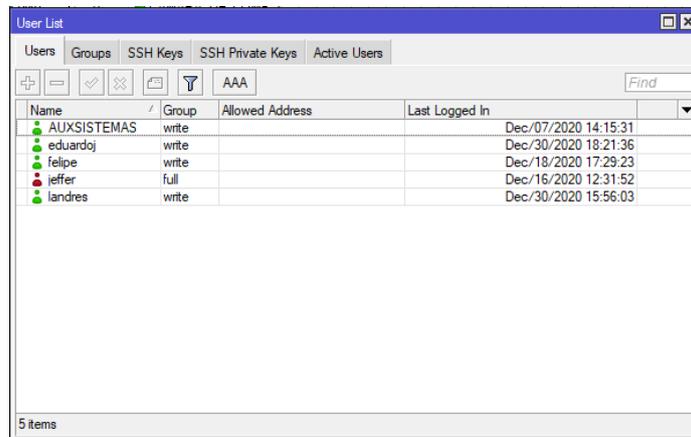


Name	Port	Available From	Certificate	TLS Ver..
api	8740			
X api-ssl	8729		none	any
X ftp	21			
X ssh	22			
X telnet	23			
winbox	9000			
X www	80			
X www-ssl	443		none	any

Figura 14. Lista de Servicios RouterOS.

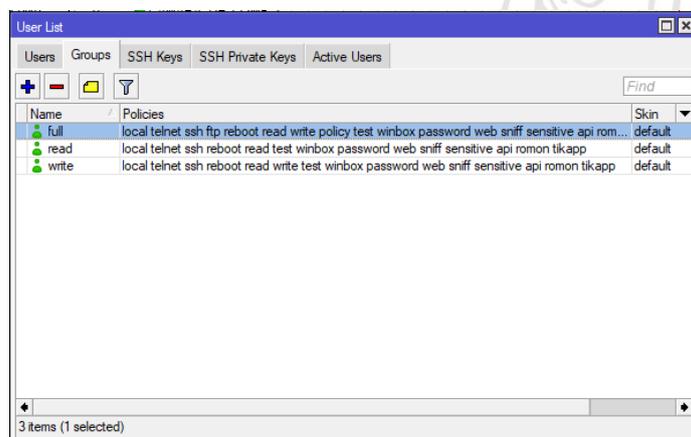
Como se puede notar, se debe definir un puerto de servicio, MikroTik define en este caso para API el puerto 8728, para el caso de Mocoa es el 8740, puerto definido por el administrador de la red. Adicionalmente se tiene también el servicio API-SSL (Application Programmable Interface - Secure Sockets Layer) cuyo puerto por defecto es 8729 y el servicio SSH (Secure Shell) definido en el puerto 22.

Para la conexión por medio del puerto de servicio API es importante tener un usuario y contraseña (Figura 15) para acceder a toda la información que tenga el equipo MikroTik con el fin de realizar la administración del equipo.



**Figura 15. Lista de Usuarios WinBox.**

Este usuario debe estar asociado a un grupo en el cual se tienen ciertos atributos para poder trabajar dentro de WinBox o el mismo servicio API. Dichos permisos pueden verse dentro de la opción Groups de la ventana user list (Figura 16).



**Figura 16. Lista de grupos definidos por el administrador.**

Para establecer conexión del servicio API, el usuario que se crea debe tener en el grupo que está asociado como principales atributos "read", "write" y "api" principalmente (Figura 17), puesto que si no tiene habilitado este atributo no le es posible acceder a ese servicio, y el equipo MikroTik simplemente no lo reconoce con los permisos suficientes para poder establecer la comunicación. Este usuario debe estar asociado a un grupo en el cual se tienen ciertos atributos para poder trabajar dentro de WinBox o el mismo servicio API.

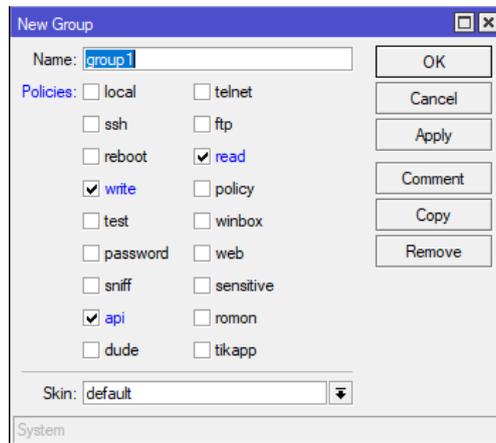


Figura 17. Lista de atributos para la creación de un nuevo grupo.

Para fines de desarrollo de este proyecto se decidió realizar la conexión por servicio API usando el lenguaje de programación Python, aunque es posible establecer desarrollos en otros lenguajes como JAVA o PHP.

Si bien es posible realizar un desarrollo bajo otros lenguajes, Python fue la opción elegida gracias a su sencillez a la hora de programar, así como las muchas integraciones con otras librerías ya sea para el manejo de datos, el desarrollo de interfaces gráficas y hasta la posibilidad de migrar hacia una aplicación web.

Principalmente, haciendo una búsqueda en la red se encontró al momento de hacer el desarrollo de la aplicación la obtención de dos librerías para la comunicación vía servicio API con los equipos MikroTik, estas librerías son:

- **routeros-api:** Esta librería desarrollada y mantenida actualmente por Jakub Goctawski[11] (@jgoclawski) y la compañía social wifi[12], es una librería de código libre, con distribución MIT para acceso de quien la requiera. Permite establecer una conexión sencilla para la conexión del servicio API, al hacer peticiones de datos al equipo MikroTik cuya respuesta es devuelta en forma de diccionario, lo que permite el manejo de datos de manera fácil. Es posible acceder a diferentes recursos del equipo MikroTik como lo es la lista ARP y el QUEUE.
- **librouteros:** Librería desarrollada por Łukasz Kostka[13], librería de código libre desarrollada para realizar comunicación del servicio API de MikroTik por medio de lenguaje Python. Como principales características tiene: respuesta de datos en forma de diccionarios, respeta la sintaxis de Python y especificación de comunicación por medio de puertos y dirección fuente.

De estas dos librerías encontradas fue empleada `routeros-api`, la razón por la cual fue desarrollado el proyecto con esta librería es la facilidad en el manejo y el corto tutorial encontrado en la documentación. Aunque, por otro lado, se realizaron pruebas con `librouteros` y el comportamiento es exactamente el mismo, retorna datos y ejecuta las mismas órdenes, siendo indiferente que librería se puede emplear para este proyecto.

La conexión para el servicio API se realiza inicialmente invocando la librería `routeros-api`, seguido del método `connect`. Este método (Figura 18) tiene como datos de entrada la dirección de host (Dirección IP del equipo MikroTik), el nombre de usuario definido en la base de datos del equipo MikroTik y su respectiva contraseña.

```
def connect(host, username, password):
    connection = routeros_api.RouterOsApiPool(host,
                                              username=username,
                                              password=password,
                                              port=8748,
                                              plaintext_login=True)

    api = connection.get_api()
    print('Conexión Establecida')
    return api, connection
```

**Figura 18. Método de conexión con el servicio API.**

Este método retorna un token asociado a la conexión del servicio API que será de utilidad para modificación de valores de activación (`disabled`) de los usuarios a los cuales les fue activado el servicio de internet. Además el método de conexión requiere de los siguientes parámetros adicionales:

- **port:** Puerto de conexión definido por el administrador de la red, para ejemplo ver Figura 14.
- **plaintext\_login:** Establece si los datos enviados hacia el equipo MikroTik son enviados en formato de texto plano. Parámetro exigido para versiones de RouterOS superiores a 6.43.

Al momento de realizar la conexión y que esta sea exitosa, la función establecida por la librería `routeros-api` no retorna ningún carácter o algún dato que muestre la conexión efectiva. Por otro lado, si se revisa el estado de usuarios activos en el programa WinBox es posible notar que usuario se conectó, por cual medio y cuando realizó dicha conexión, esto se ve en la Figura 19.

Name	At	From	Via	Group
eduardoj	Jan/03/2021 19:25:53	10.1.0.176	winbox	write
eduardoj	Jan/03/2021 19:26:44	10.1.0.176	api	write
jeffer	Oct/27/2020 14:44:17	192.168.99.254	telnet	full
jeffer	Oct/27/2020 14:44:19	192.168.99.254	telnet	full

**Figura 19. Lista de usuarios activos vista desde WinBox.**

Como se puede observar, la conexión desde el servicio API se realizó con éxito, y con ayuda de los atributos definidos en el group list, se pueden ejecutar las acciones que nos permitan los atributos como la lectura, escritura de datos en WinBox, acceso a recursos como lista ARP y Queue List, entre otros.

### **ETAPA 3: Desarrollo de la aplicación de activación de usuarios y construcción de base de datos.**

Antes de desarrollar el proyecto, el equipo de auxiliares de ingeniería debía realizar el proceso de corte y activación de usuarios, este proceso era llevado a cabo de la siguiente forma:

- De parte del equipo de caja y atención del usuario era entregada cierta cantidad de recibos de pago que eran realizados de manera física en la oficina, estos recibos de pago tenían información importante como número de cédula y nombres del titular de la línea de internet (Figura 20).
- Con el nombre completo del suscriptor del servicio se busca en la parte del Queue List del programa WinBox. Por ejemplo, se tiene al usuario Eduardo José Maya Rodríguez, la búsqueda en Queue List se realizaría de la forma **"MAYA\_RODRIGUEZ\_EDUARDO\_JOSE"** y se extrae la dirección IP (target) a la cual el usuario está asociado.
- En caso de que no estuviese el usuario en Queue List, se informaba a los compañeros de caja y atención al usuario para que se encuentre la dirección IP del usuario pidiéndoles información ya sea por el nombre o número de identificación del usuario (Cédula de ciudadanía o NIT en caso de ser una empresa) en la base de datos de usuarios que se maneja para la contabilidad de la empresa.

- Con la dirección IP encontrada se buscaba la dirección dentro de la lista ARP, en tal caso de estar deshabilitada (disabled), esta dirección IP tiene la etiqueta X (Disabled), se procede a cambiar la etiqueta, es decir dejarla en estado habilitada (enabled) para que la lista ARP le cambie la etiqueta a C (Completed).

Caja Aux Mocoa

**isanet** Nº. 900.593.821 - 9  
Actividad Económica ICA 6120  
No. Somos Grandes Contribuyentes  
No Somos Autoempresarios  
Regístrate Conigo **PRINCIPAL**  
Carrera 9 24 038 Vía Del Norte  
Tel: (8) 626641 - 311349113  
Mocoa - Pichincha  
www.isanetworks.co

**RECIBO DE CAJA**

Número: 0100044210	Fecha: 20201207	Página: 1
-----------------------	--------------------	--------------

Recibido de: 69008045	Razón social o nombre: FLOREZ ERAZO NUBIA YANETH	Apertura: 0100001055
Elaborado por: ayakelin	Concepto: Cancelación facturas credito	Destino del ingreso: 02001

Observaciones:

TIPO DE PAGO	CNT. MONETARIA	DETALLE	VALOR
EF - EFECTIVO			60,000.00
<b>TOTAL:</b>			<b>60,000.00</b>

**RELACION DE DOCUMENTOS**

ITEM	No. FACTURA	IDENTIFICACION	\$ CANCELADO	ANTICIPOS	TIPO AJUSTE	\$ AJUSTE	\$ NETO
1	FE00000277	69008045 FLOREZ ERAZO NUBIA YANI	80,000.00	0.00	1 DESCUENTG	20,000.00	60,000.00
<b>TOTALES</b>			<b>80,000.00</b>			<b>-20,000.00</b>	<b>60,000.00</b>

Aprobado por: \_\_\_\_\_ Revisado por: \_\_\_\_\_

Fecha: MA 1124866849 espacio al final  
Lp clave Baja plan 20MR

**ISANETWORKS**

Recibido Por: **MO**

**07 DIC 2020**

**CO**

**CA**

**CANCELADO**

NIT: 900593821-9

Figura 20. Ejemplo de recibo de pago con anotaciones.

Con el funcionamiento manual para la activación definido, para el desarrollo de una aplicación funcional que permita el uso del servicio API, fue importante definir un programa sobre el cual va a operar la librería routers\_api. La realización de la aplicación fue llevada a cabo con los siguientes paquetes para el completo desarrollo del proyecto:

- **Tkinter:** Librería dedicada a la programación de interfaces gráficas de usuario (GUI), que viene dentro del paquete de instalación básica de Python (no requiere instalación adicional) con la cual es posible diseñar una interfaz gráfica sobre la cual se ejecutará el programa de activación de usuarios que permite la programación de widgets como espacios de texto, botones, barras de desplazamiento, entre otros. Con base en funciones de eventos para la acción de los widgets que se definan en el programa.
- **Pandas:** Paquete de Python diseñado para el manejo rápido, flexible y sencillo de estructuras datos, empleado en analítica de datos y que será de utilidad para el manejo de datos de los usuarios a activar. Permite el uso de archivos

tipo hojas de cálculo (Excel) para la lectura de datos y así realizar la activación de los usuarios de manera automática.

- **Pymysql:** Librería para el manejo de la API para administración de bases de datos. Con esta librería se crearon la base de datos, las tablas con los datos que deben ser ordenados y la construcción de la base con los datos de los usuarios para la activación y modificación de valores de la base de datos creada.

Para elaborar el proyecto también se hace uso de un IDE (Integrated Development Environment), el IDE escogido es PyCharm, IDE diseñado para el desarrollo en lenguaje de programación Python, es un software de uso libre y gratuito, con una gran comunidad de desarrollo; enfocado principalmente para el desarrollo en este lenguaje de programación, soporta frameworks para desarrollo web como django o flask, herramientas científicas como matplotlib y NumPy. Se caracteriza también por el uso de tecnologías diferentes a Python, es decir permite desarrollar en otros lenguajes como JavaScript o Cython. Razón por la cual fue elegido para el desarrollo de este proyecto.

Continuando con el proceso de activación de usuarios desde el punto de vista de la comunicación con el servicio API, una vez conectado, es posible acceder a cualquier recurso que se tenga en el equipo. Los recursos disponibles se pueden ver en la figura 7, el método con el cual se pueden obtener datos de cualquier recurso es el siguiente:

```
api.get_resource('resource')
```

Donde la palabra "resource" es el recurso al cual se desea acceder, siendo así por ejemplo si se desea llegar hasta la información de enrutamiento el comando indicado sería "ip/routes" o "/ip/routes". En el caso de este proyecto los únicos recursos empleados son "ip/arp" para acceder a toda la lista ARP presente en el equipo MikroTik, y "queue/simple" o "/queue/simple" donde están todos los perfiles asociados a los usuarios activos en la red.

Acompañado del método anterior se tiene la siguiente orden:

```
List.get()
```

Esta orden me entrega un diccionario de Python en el cual se encuentran todos los datos pertenecientes al recurso al que se está accediendo por medio del comando "api.get\_resource("ip/arp")", por defecto está vacío, pero se pueden emplear datos específicos de búsqueda como

- id.
- address.
- mac-address.

- interface.
- published.
- invalid.
- DHCP.
- dynamic.
- complete.
- disabled.

```
[{'id': '*26FD',
  'address': '10.10.15.6',
  'mac-address': '50:6F:77:D1:09:D4',
  'interface': 'sfp-sfpplus2',
  'published': 'false',
  'invalid': 'false',
  'DHCP': 'false',
  'dynamic': 'false',
  'complete': 'true',
  'disabled': 'false'}]
```

**Figura 21. Diccionario resultado de la búsqueda de la IP 10.10.15.6 en recurso IP/ARP.**

Si se requiere obtener la información en la lista ARP de la dirección 10.10.15.6, el método que se debe usar es por ejemplo `"list.get(address='10.10.15.6')"`. Este devuelve un diccionario con todos los arreglos que coincidan con la dirección 10.10.15.6, aunque también se puede hacer una búsqueda con los demás parámetros presentados en la figura 21. Para el caso de la sintaxis, es importante escribir las separaciones como en el caso de `"mac-address"` como `"mac_address"`.

Todas estas palabras clave son obtenidas gracias al apartado de scripting y del uso de la terminal presente en la wiki de Mikrotik[5].

Para la modificación de valores, para este proyecto, por ejemplo, solo se desea cambiar el estado `"disabled"`, que en caso de que esté inactiva la dirección IP es de valor equivalente a `"true"` (verdadero), debe pasar a `"false"` (falso). La forma de realizar el cambio es el siguiente:

```
list.set(id="value", address="ip_address", disabled="false")
```

Donde `"list"` es la lista de datos que fueron traídos del recurso que se solicite es imperativo tener el valor de ID y `"disabled"` para realizar modificaciones, y la dirección IP es opcional, aunque por cuestiones de confirmación se ingresa para evitar fallos a la hora de activar por equivocación usuarios que todavía no realizan el pago del servicio.

Por otro lado, el Queue List, presenta los siguientes parámetros:

- id.
- name.
- target.
- parent.
- packet-marks.
- priority.
- queue.
- limit-at.
- max-limit.
- burst-limit.
- burst-threshold.
- burst-time.
- bucket-size.
- bytes.
- total-bytes.
- packets.
- total-packets.
- dropped.
- total-dropped.
- packet-rate.
- total-packet-rate.
- queued-packets.
- queued-bytes.
- invalid.
- dynamic.
- disabled.



```
[{'id': '*375',
  'name': 'PANTOJA_SALCEDO_ARMANDO_DE_JESUS_',
  'target': '10.10.15.6/32',
  'parent': 'none',
  'packet-marks': '',
  'priority': '8/8',
  'queue': 'default-small/default-small',
  'limit-at': '0/0',
  'max-limit': '10000000/20000000',
  'burst-limit': '0/0',
  'burst-threshold': '0/0',
  'burst-time': '0s/0s',
  'bucket-size': '0.1/0.1',
  'bytes': '799835221/1724627819',
  'total-bytes': '0',
  'packets': '4883401/3544876',
  'total-packets': '0',
  'dropped': '0/5545',
  'total-dropped': '0',
  'rate': '600/728',
  'total-rate': '0',
  'packet-rate': '1/1',
  'total-packet-rate': '0',
  'queued-packets': '0/0',
  'total-queued-packets': '0',
  'queued-bytes': '0/0',
  'total-queued-bytes': '0',
  'invalid': 'false',
  'dynamic': 'false',
  'disabled': 'false'}]
```

**Figura 22. Diccionario resultado de la búsqueda de la IP 10.10.15.6 en recurso Queue/Simple.**

Así mismo, de este recurso se toman principalmente los valores de ID para hacer modificaciones como en ancho de banda para carga y descarga de datos, nombre del usuario al cual está asociado el plan, conocer estadísticas de ese perfil, como paquetes transmitidos, paquetes caídos, velocidad de transferencia media, entre otras funciones. El método “set” permite realizar modificaciones tal y como se muestra en el caso del recurso “ip/arp”.

Con el funcionamiento de la comunicación por medio del servicio API, teniendo definido el comportamiento de la tarea de activación manual y las herramientas dispuestas para llevar a cabo el proyecto, se decide comenzar el desarrollo del aplicativo con el cual se automatiza la activación de los usuarios. Para llevar a cabo esta tarea se definió el siguiente algoritmo:

- Se validan los datos de conexión del servicio API, es decir, se debe ingresar al equipo CloudCore de MikroTik con el usuario y contraseña creados para tal fin y que se establezca una conexión exitosa al servicio. En el caso en que no sean válidos el sistema no hará nada, hasta que los datos válidos sean ingresados.
- Una vez establecida la conexión con el servicio API se debe tomar la dirección IP de cada usuario a activar.
- Una vez tomada la dirección IP debe ser validada tanto en el equipo MikroTik, es decir, que se encuentre en la lista ARP, como en la base de datos creada. La dirección IP nos permite tomar un valor de identificación (ID) que es necesario para realizar la modificación de valores como se muestra en[16].

Finalmente, en caso de existir tanto en la lista ARP como en la base de datos se da la orden de habilitación del servicio.

- En caso de que no exista la dirección IP, se notificará con un error que la IP solicitada no se puede activar.
- Este proceso se realiza hasta que no se tengan más usuarios por activar.

La estructura base del programa consiste en una ventana raíz, en la cual son desplegados widgets que están referenciados a funciones de tal forma que en la medida que se interactúa ya sea con un botón, un grupo de selección, un campo de texto o una tecla presionada, se realiza una o varias acciones por debajo de la interfaz gráfica que un usuario ve normalmente.

A continuación, se hará una descripción de las ventanas que se realizaron para el diseño de la aplicación:

- **Ventana de inicio de sesión:** Ventana inicial que ve el usuario que desea establecer comunicación con el equipo MikroTik. Consta de una imagen con el logo de la empresa, un formulario con los campos de host (Dirección IP del equipo MikroTik con el que se realizará la conexión), username (Nombre de usuario definido en el equipo MikroTik), password (Contraseña asociada al nombre de usuario almacenado en el equipo MikroTik) y un conjunto de botones de Inicio de sesión, y Salir (Figura 24).
- **Ventana principal para activación y desactivación de usuarios:** Ventana que se despliega una vez se realiza conexión con el equipo MikroTik. En esta ventana se tiene un conjunto de botones que permiten la activación o desactivación de usuarios, un campo de texto donde se ingresa la dirección IP a activar/desactivar, la opción de tomar un archivo excel para la activación masiva de usuarios, y muestra la cantidad de usuarios almacenados en la base de datos diseñada para la aplicación basada en los datos del equipo MikroTik (Figura 25).
- **Ventana de resultado de activación cuando se usan archivos Excel:** Esta ventana se genera cuando el programa termina de activar/desactivar con base en archivos Excel en los cuales hay direcciones IP. En ella se muestra la cantidad de direcciones IP activadas o desactivadas, el estado en el que se encuentra la IP luego de ser modificado el valor de disabled y si la dirección IP presenta un error como si no está en la base de datos, y una alerta de error de dato (Figura 26).

- **Ventanas emergentes:** Se diseñaron ventanas emergentes con el fin de indicarle al usuario la terminación de la tarea de manera exitosa cuando se realiza una activación, una ventana cuando hay un error con la activación o desactivación del usuario, y para finalizar sesión y cerrar el programa (Figura 27, Figura 28 y Figura 29).

Finalmente, con la puesta en marcha del programa es importante tener una base de datos adecuada de tal forma que permita llevar un control de los usuarios que son activados y que deben estar dentro del equipo MikroTik. Esta base de datos se realiza en MySQL, el cual es un gestor de base de datos de código libre, empleado en muchos proyectos gracias a su fácil manejo y es popular entre muchos desarrolladores. Los datos que van a ser almacenados en la base de datos de MySQL son una combinación de los datos que se pueden obtener de la lista ARP (Dirección IP, Dirección MAC, ID, Interfaz, y Estado activo/inactivo), y del Queue List (Nombre completo del usuario y Velocidad Carga/Descarga contratada). Un ejemplo de la tabla de la base de datos es la vista en la Figura 21.

ID	NAME	IP_ADDRESS	MAC_ADDRESS	SPEED	INTERFACE	DISABLED
*22	VARGAS_PEREZ_ROBINSON_	10.1.20.50	3C:78:43:87:F2:4D	13000000/25000000	video juegos	false
*23	MARULANDA_CRIALES_YERMAIN_	10.1.20.53	3C:78:43:63:C1:45	13000000/25000000	video juegos	false
*27	SOLARTE_LOPEZ_CARLOS_ANDRES	10.1.20.60	3C:78:43:87:99:FF	13000000/25000000	video juegos	false
*2D	DAZA_DIAZ_NANCY_LEIDY_4540	10.1.20.69	3C:E8:24:97:53:D5	13000000/25000000	video juegos	false
*2E	PORTLLA_ZAMBRANO_MARTHA_GECILIA	10.1.20.24	3C:78:43:87:F9:B8	10000000/20000000	video juegos	false
*35	ZUIN_CHANCHI_AURA_MARI	10.1.20.79	78:58:60:5F:96:21	10000000/20000000	video juegos	false
*3F	ACOSTA_VALLEJO_SARA_NATALIA	10.1.20.73	84:A9:C4:C7:4C:60	10000000/20000000	video juegos	False
*46	QUINTERO_MARIN_JOSSIE_STEBAN_	10.1.20.76	3C:78:43:87:A3:22	10000000/20000000	video juegos	False
*5B	BUITRAGO_ZAMBRANO_MIGUEL_ANGEL_comercial	10.1.20.80	48:FD:8E:B9:64:FA	13000000/25000000	video juegos	False
*5D	ZAMBRANO_TORO_WILLIAM_FERNANDO	10.1.20.81	A4:BE:2B:71:DE:ED	13000000/25000000	video juegos	False
*6A	FERNANDEZ_GONZALEZ_YENSER	10.1.20.85	3C:E8:24:0F:6A:60	13000000/25000000	video juegos	False
*AF	SANCHEZ_RIOS_LIBARDO	10.1.20.92	70:A8:E3:0C:24:CD	13000000/25000000	video juegos	False
*B1	GUAMANGA_GOMEZ_KEVIN_DANILO_2255	10.1.20.95	3C:E8:24:51:3B:E7	13000000/25000000	video juegos	False
*D8	ALVAREZ_CORDOBA_MARIO_ANTONIO_	10.1.20.97	50:6F:77:26:76:1D	13000000/25000000	video juegos	False
*DD	QUIROGA_BUCHELLY_JULIAN	10.1.20.98	24:4C:07:97:44:28	13000000/25000000	video juegos	False
*EA	ZAMUDIO_VEGA_ANA_JULIA	10.1.20.99	78:58:60:D1:5A:50	13000000/25000000	video juegos	False
*F7	IPS_SOLUCIONES_INTEGRALES_DE_TRANSITO_SAS_	10.1.20.102	3C:78:43:87:AA:CE	10000000/20000000	video juegos	false
*11F	MEDICIS_GARZON_JUAN_MANUEL	10.1.20.108	B4:15:13:03:C3:3D	13000000/25000000	video juegos	False
*14B	GUZMAN_RIOS_DAVID_SANTIAGO	10.1.20.111	78:F5:57:30:F9:C8	13000000/25000000	video juegos	False
*152	CHAVARRO_REYES_FERLEY	10.1.20.110	88:CF:98:D4:F3:F2	13000000/25000000	video juegos	False
*15B	CLAROS_VERA_FABIA	10.1.20.112	AC:85:3D:43:C3:1F	13000000/25000000	video juegos	False
*1E0	HERNANDEZ_LOIZA_BERNANDINA_	10.1.20.136	A0:08:6F:26:42:44	13000000/25000000	video juegos	False
*1E7	SILVA_LUCERO_NATALI_comer	10.1.20.125	3C:78:43:87:A8:37	15000000/30000000	video juegos	False
*1E9	ROJAS_ARIAS_JHONATAN	10.1.20.87	AC:85:3D:43:D4:08	13000000/25000000	video juegos	False
*1EB	GOMEZ_BISCUE_EYDER_ANDRES_	10.1.20.124	78:F5:57:30:CB:B9	13000000/25000000	video juegos	False

Figura 23. Tabla de usuarios de base de datos.

#### ETAPA 4: Comprobación de la aplicación y puesta a prueba.

Como parte de esta etapa se decide poner en funcionamiento la aplicación en un equipo de prueba MikroTik, en la cual hay una base de 96 direcciones IP, con las cuales es posible ejecutar pruebas sin afectar el rendimiento de la red principal de Mocoa.

Las pruebas consisten en lo siguiente:

- Establecimiento de conexión con el equipo MikroTik por medio del servicio API, observando que dicha comunicación se establezca y aparezca en la tabla de usuarios activos (Figura 19).
- Desactivación individual de una dirección IP perteneciente a la lista ARP presente en ese equipo MikroTik y tiempo de respuesta del equipo.
- Activación individual de la dirección IP anteriormente desactivada y verificación de tiempo de respuesta.
- Desactivación de un grupo de direcciones IP, provistas por un archivo Excel, comprobar funcionamiento de la tarea y tiempo de respuesta de la tarea.
- Activación de un grupo de direcciones IP, provistas por un archivo Excel, comprobar funcionamiento de la tarea y tiempo de respuesta de la tarea.
- Revisión de posibles fallas en la interfaz gráfica que afecten el comportamiento de la aplicación.



## Resultados y análisis

Como resultado principal se entrega una interfaz gráfica completamente funcional para que el equipo de auxiliares de sistemas pueda hacer uso de la aplicación. La interfaz fue diseñada con base en los diseños planteados en la metodología, con lo cual se entrega el diseño final de la siguiente forma:

- La ventana de inicio de sesión mostrada en la figura 24 donde el usuario del servicio API puede conectarse a los equipos MikroTik de la oficina haciendo uso de la dirección IP (Host) del MikroTik, el usuario (Username) y contraseña (Password) suministrados por el administrador de la red.



**Figura 24. Interfaz para el ingreso a MikroTik.**

- La ventana principal en la cual se interactúa con el servicio API que fue diseñada para habilitar usuarios y adicionalmente deshabilitar usuarios si así se requiere. Esta ventana principal ofrece la opción de habilitar usuarios por medio de su dirección IP (un solo usuario) o la facilidad de habilitar usuarios por medio de un archivo extra (archivo Excel). Se puede notar también que hay una parte en la que es posible ver el estado de los usuarios presentes en la base de datos y conocer su estado actual validado, en conjunto con el equipo administrador. Esta ventana se muestra en la figura 25 (MikroTik).

API ISANET: eduardoj@190.216.198.27

Activar usuarios     Desactivar usuarios  
 Dirección IP:   
   

Cerrar Sesión

Item	ID	Nombre Usuario	Dirección IP	Dirección MAC	Velocidad Carga/Descarga	Interfaz	Deshabilitado
74	*CESD	LUGO_MALAVE_FRAY_LUIS_4647	10.1.20.30	F4E5F2:37:F4A7	10000000/20000000	video juegos	False
75	*D0A	BOLANOS_SORRIGUEZ_DIEGO_FERNANDO_1012034	10.1.20.34	F4E5F2:38:8122	10000000/20000000	video juegos	False
76	*D0B4	POTOSI_POTOSI_CESAR_AUGUSTO	10.1.20.32	2C:97:81:64:00:18	10000000/20000000	video juegos	False
77	*D651	INVERSIONES_VALARC	10.1.20.33	6B:C:4E:3F:4D:2E	13000000/25000000	video juegos	False
78	*D672	LICEO_VICTORIA_REGIA_	10.1.20.40	8B:A5:8D:14:6A:38	25000000/40000000	video juegos	False
79	*D6DF	MUNOZ_MOLANO_JHON_EDWARD_	10.1.20.40	F4:79:60:7B:C0:04	18000000/35000000	video juegos	False
80	*E108	CLINICA_AYUNAN_	10.1.20.66	8B:A5:8D:14:80:90	55000000/55000000	video juegos	False
81	*E6A6	ALARCON_PERDOMO_NARCEY_FABIO_	10.1.20.34	AC:85:3D:5B:27:27	10000000/20000000	video juegos	False
82	*E7FF	MARIN_GRASS_LAUDY_DAHIANA_	10.1.20.35	F4E5F2:38:88:73	13000000/25000000	video juegos	False
83	*E953	MANZULLO_GUTIERREZ_HAROLD_BLAZIMI	10.1.20.83	8A:15:13:77:48:CF	13000000/25000000	video juegos	False

Figura 25. Ventana Principal de la aplicación.

- La ventana de resultados, en la cual para el caso de uso de archivos Excel, es posible conocer el estado de activación del usuario una vez haya finalizado la tarea. En caso de no encontrar la dirección IP en la base de datos y en el MikroTik está arrojando en lugar del estado de habilitado o deshabilitado, un campo vacío tanto en el estado de habilitación como en el campo de ID, producto de que esa dirección IP no está en la base de datos y por lo tanto no se puede activar (Figura 26).

API ISANET: Activar

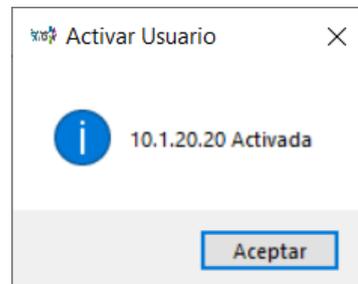
Estado Usuarios

Item	ID	Dirección IP	Disabled
4	*80A	10.1.20.19	false
5		10.120.45.6	
6	*11B04	10.1.20.20	false
7	*12AF1	10.1.20.21	false
8	*1391D	10.1.20.22	false
9	*B594	10.1.20.23	false
10	*2E	10.1.20.24	false
11		10.120.45.5	
12		10.120.45.7	
13		10.120.45.8	

Cerrar

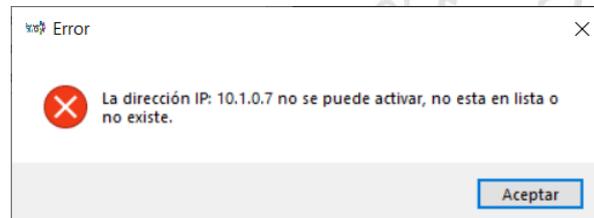
Figura 26. Ventana emergente de finalización de tarea para activación de usuarios desde archivos Excel.

- La ventana emergente, mostrada en la figura 27, muestra información cuando se habilita de manera correcta un usuario de manera sencilla.



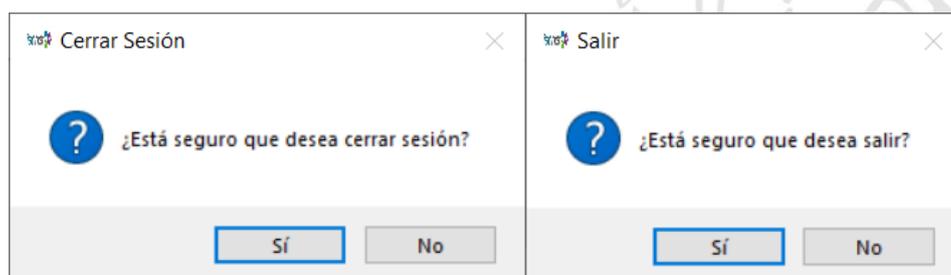
**Figura 27. Ventana de información de dirección IP habilitada.**

- La ventana emergente de la figura 28 indica un error en caso de que la habilitación presente falla. La falla por lo general radica en que la dirección IP del usuario no está presente en la base de datos.



**Figura 28. ventana de información en caso de error.**

- Las ventanas emergentes para cerrar sesión y salir del programa se muestran en la figura 29.



**Figura 29. Ventanas de decisión para cerrar sesión (Izquierda) y cerrar programa (Derecha).**

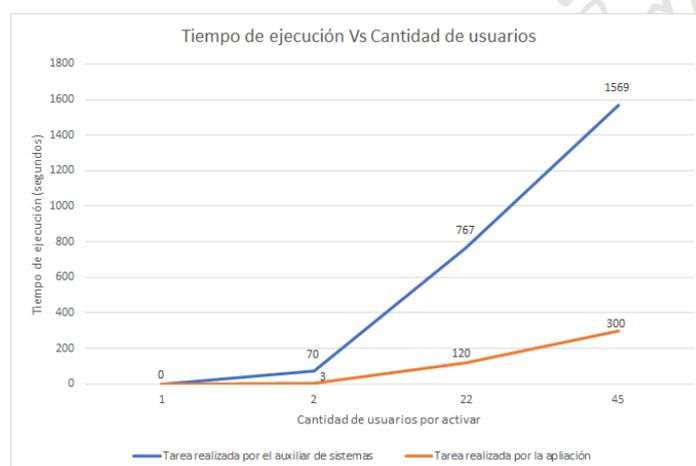
Con la interfaz gráfica lista para el uso por parte del equipo de auxiliares de sistemas se realizan pruebas para comprobar el comportamiento en la actividad de activación. Para las pruebas se decidió usar usuarios de la base de datos de la interfaz “videojuegos”, en esta interfaz se encuentran un total de 96 usuarios.

Las pruebas realizadas compararan el tiempo que toma activar cierta cantidad de usuarios de la manera en que se realizaba la tarea inicialmente, tarea ejecutada por un auxiliar de sistemas, frente al tiempo que le toma a la aplicación ejecutar el proceso de activación, los resultados son mostrados en la tabla 2:

**Tabla 2. Resultados en tiempo obtenidos en las pruebas realizadas para la aplicación.**

Cantidad de usuarios por activar	Tiempo realizado por auxiliar de sistemas (segundos)	Tiempo realizado por aplicación (segundos)
1	34.85	1.48
2	70	3
22	767 (12 minutos Aprox.)	120 (2 minutos)
45	1569 (26 minutos Aprox.)	300 (5 minutos)

Como resultado de las pruebas y con base en los datos suministrados por la tabla, es posible trazar una gráfica para ver los resultados, visualizados en la figura 28:



**Figura 30. Gráfico del tiempo de ejecución de la tarea de activación de usuarios ejecutada por un auxiliar de sistemas frente a la aplicación con servicio API.**

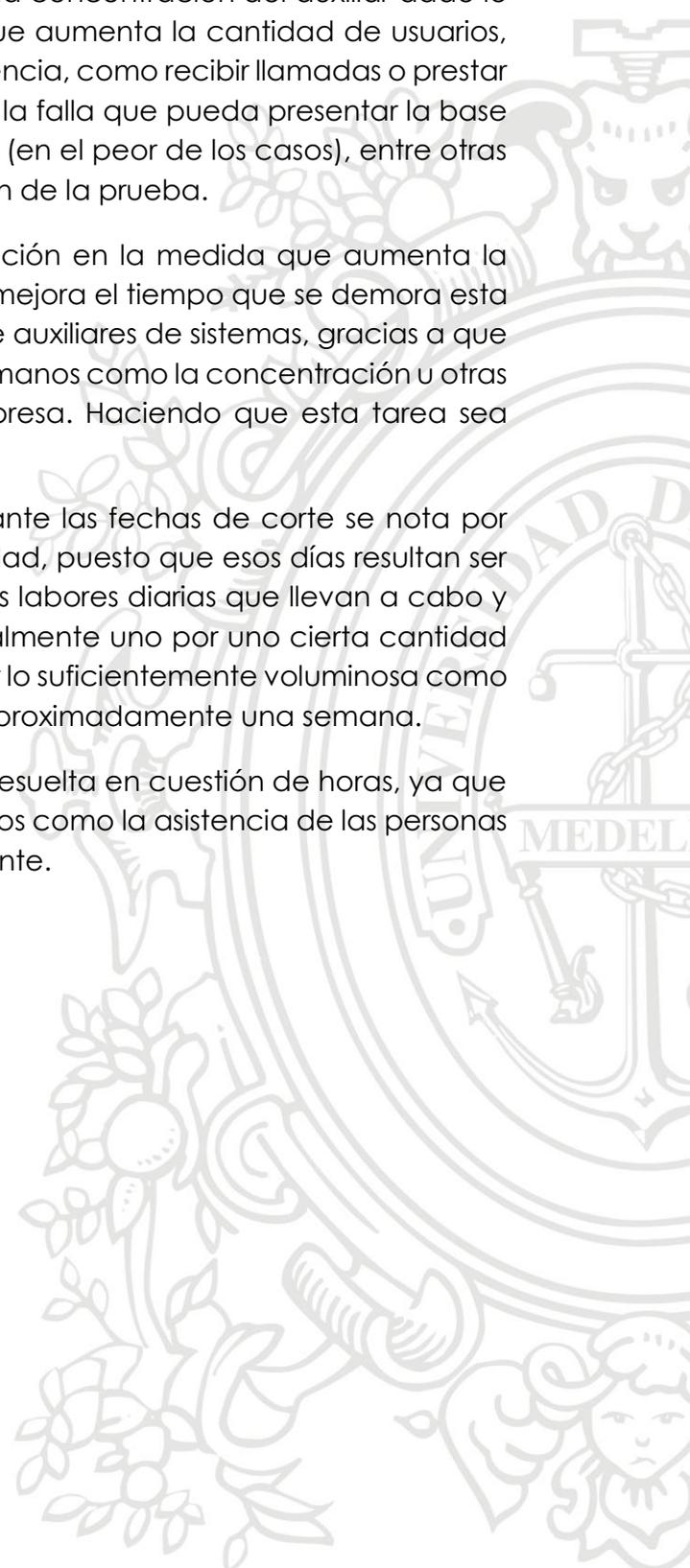
Como se puede notar de la Figura 28, en la medida que aumenta la cantidad de usuarios por activar, la tarea realizada por parte del auxiliar de sistemas toma un

mayor tiempo en completarse. Esta demora es atribuida a múltiples factores que pueden afectar a quien realiza la tarea, como la concentración del auxiliar dado lo tedioso que se vuelve la tarea en la medida que aumenta la cantidad de usuarios, la suspensión de la tarea por atender a otra urgencia, como recibir llamadas o prestar servicio a los equipos desplegados en campo y la falla que pueda presentar la base de datos en cuanto a los usuarios que se tenga (en el peor de los casos), entre otras causas que se encontraron durante la ejecución de la prueba.

Por otro lado, el comportamiento de la aplicación en la medida que aumenta la cantidad de usuarios tiende a ser lineal, lo que mejora el tiempo que se demora esta tarea ejecutada por una persona del equipo de auxiliares de sistemas, gracias a que la aplicación no se ve afectada por factores humanos como la concentración u otras tareas que deben ser desarrolladas en la empresa. Haciendo que esta tarea sea cumplida a cabalidad en menos tiempo.

Al poner en funcionamiento la aplicación durante las fechas de corte se nota por parte de los auxiliares cierto grado de tranquilidad, puesto que esos días resultan ser los mas agotadores para el equipo debido a las labores diarias que llevan a cabo y la tarea adicional que era la de activar manualmente uno por uno cierta cantidad de usuarios, que varían cada mes, pero logra ser lo suficientemente voluminosa como para ocupar gran tiempo de trabajo durante aproximadamente una semana.

Implementado el proyecto la tarea pasa a ser resuelta en cuestión de horas, ya que esta labor también depende de factores externos como la asistencia de las personas que se acercan a pagar su servicio principalmente.



## Conclusiones

- El uso del servicio API permite desarrollar tareas simples sin necesidad de conocer de forma detallada el funcionamiento de un programa. El servicio ejerce la función de traductor a fin de que se realice una tarea por medio de lenguajes desconocidos entre ellos pero que buscan resolver una tarea en común, como en el caso de la activación de usuarios para la empresa ISANETWORKS ISP SAS.
- La solución de tareas como las de este proyecto permite superar etapas que para empresas en crecimiento como ISANETWORKS ISP SAS, que buscan ser referentes en la región, puedan alcanzar y proponer mayores y mejores objetivos en búsqueda del desarrollo de los territorios donde opera como lo son el departamento del Huila y el Putumayo.
- El servicio API también permite el desarrollo en otras áreas o recursos disponibles en WinBox. También es posible crear usuarios nuevos, realizar funciones de redes como la creación de direccionamiento o administración de redes. No existen limitantes para el desarrollo de nuevas tareas en el proyecto y permite la integración fácil con otras API, ya que el manejo de datos resulta más sencillo que usando otros recursos disponibles en la plataforma.
- MikroTik demuestra tener herramientas interesantes para ofrecer soluciones tecnológicas de bajo costo y buen mantenimiento, resultando de gran utilidad para pequeñas y medianas ISP. Gracias a su software de administración, activa comunidad en línea, y demás objetos que permitan la integración de servicios como lo es API con soluciones que se pueden desarrollar bajo el marco de software libre a fin de ser un marco de referencia para personas que están en el mundo de la telemática.
- Es importante recalcar la función de los auxiliares de sistemas de la empresa, y comunicación de manera acertada con ellos permitió la realización de un programa que se ajuste a las necesidades del equipo, con el objetivo de mejorar la calidad del servicio para los usuarios y los tiempos de atención para los técnicos desplegados en campo.
- Mantener una base de datos consistente es importante para este tipo de proyectos, ya que sin esta parte no sería posible la administración de los usuarios que adquieren un servicio con la empresa, y perjudica la experiencia personal de la empresa con cada uno de sus clientes.

- El trabajo conjunto entre áreas, en este caso la telemática y el desarrollo de aplicaciones para automatización de procesos buscan fortalecer continuamente las debilidades de cada área y mejorar aquellas en las que cada campo se destaca.
- De los resultados obtenidos en el proyecto se puede ver claramente que una tarea por más sencilla que sea, en la medida que esta aumenta considerablemente en volumen, termina tomando más tiempo en ser completada si se realiza de forma manual, lo que afecta negativamente al rendimiento del proceso. Labores de automatización sencillas permiten optimizar el tiempo que toma una tarea en ser completada, observando que es posible completar la activación de usuarios en fracciones de tiempo cercanos entre el 16% y 20% del tiempo dedicado a ese tipo de tareas si se realizan de manera manual por medio de una persona.



## Referencias Bibliográficas

1. ¿Cómo funciona una conexión de fibra GPON y FTTH?, naseros.com. Disponible en: <https://naseros.com/2017/03/13/como-funciona-una-conexion-de-fibra-gpon-y-ftth/>.
2. What is (Gigabit Passive Optical Networks) GPON, linkedin.com. Disponible en: <https://www.linkedin.com/pulse/what-gigabit-passive-optical-networks-gpon-sayed-qaisar-shah#:~:text=GPON%20stands%20for%20Gigabit%20Passive,multiple%20homes%20and%20small%20businesses.>
3. INTERNATIONAL TELECOMMUNICATION UNION. Gigabit-Capable Passive Optical Networks (GPON): General characteristics. G.984.1. Geneve, Switzerland. 2009. 43p.
4. Análisis de PON: ¿Qué es OLT, ONT y ODN?, medium.com. Disponible en: <https://xxamin1314.medium.com/an%C3%A1lisis-de-pon-qu%C3%A9-es-olt-onu-ont-y-odn-8e78eb25e4bb>
5. Manual: API, wiki.mikrotik.com. Disponible en: <https://wiki.mikrotik.com/wiki/Manual:API>.
6. MONGE, Adrián. Desarrollo de una aplicación para la monitorización de puntos de acceso MikroTik. Trabajo de grado ingeniero de tecnologías de la información. Cantabria, España. Universidad de Cantabria. Escuela técnica superior de ingenieros industriales y de telecomunicación. 2017. 66p.
7. ¿Qué es una API?, redhat.com. Disponible en: <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces#:~:text=Una%20API%20es%20un%20conjunto,de%20saber%20c%C3%B3mo%20est%C3%A1n%20implementados.>
8. MikroTik products: Routerboard CCR1072-1G-8S+ technical specifications. mikrotik.com. Disponible en: <https://mikrotik.com/product/CCR1072-1G-8Splus>.
9. Manual: Queue, wiki.mikrotik.com. Disponible en: <https://wiki.mikrotik.com/wiki/Manual:Queue>
10. SANTACRUZ, Edwin Jacinto. Diseño de un Sistema de Control de Equipo Wireless MikroTik para la Gestión y Administración de Clientes en una Red WISP. Trabajo de grado Magíster en telecomunicaciones. Guayaquil, Ecuador. Universidad Católica de Santiago de Guayaquil. Departamento de posgrados. 2019. 73p.
11. GOCLAWSKI, Jakub. Github.com. Disponible en: <https://github.com/jgoclawski>
12. Social WiFi RouterOS-api, github.com. Disponible en: <https://github.com/socialwifi/RouterOS-api>

13. Librouteros, github.com Disponible en: <https://github.com/luqasz/librouteros>
14. PRIETO, Eduardo José. Implementación de un Hotspot y una aplicación web para su gestión. Trabajo de grado ingeniero técnico informático de gestión. Leganés, España. Universidad Carlos III de Madrid. Facultad de Ingeniería. 2015. 90p.
15. STOISOV, G. y RANGELOV, V. One implementation of API interface for RouterOS. Plodvid University. 2016. 5p.
16. AGILAR, J. QUEZADA, R. y GARCIA, K. Aplicación Javapara el control de RB MikroTik en empresas proveedoras de servicio de internet. Revista Ciencia UNEMI. Pp161-169. 2018.
17. Introduction librouteros, readthedocs.io. Disponible en: <https://librouteros.readthedocs.io/en/latest/introduction.html>
18. Graphical User Interfaces with Tk, docs.python.org. Disponible en: <https://docs.python.org/3/library/tk.html>
19. Pandas, pypi.org. Disponible en: <https://pypi.org/project/pandas/>
20. ¿Qué es MySQL?: Características y ventajas, openwebinars.net. Disponible en: <https://openwebinars.net/blog/que-es-mysql/>
21. KOMARUDIN, Ahmad Rosid. MikroTik Automation Using Scripting, SSH & API. Webinar. 41p. Disponible en: [https://mum.mikrotik.com/presentations/ID18/presentation\\_5587\\_1540244493.pdf](https://mum.mikrotik.com/presentations/ID18/presentation_5587_1540244493.pdf)
22. Lección 2 ¿Qué es la programación orientada a eventos?, contenidos.sucerman.com. Disponible en: <http://contenidos.sucerman.com/nivel3/dispositivos/unidad1/leccion2.html>
23. ¿Qué es MySQL?, hostinger.es. Disponible en: <https://www.hostinger.es/tutoriales/que-es-mysql/>