

# **PROYECTO FINAL DE INGENIERÍA**

## **APLICACIÓN WEB PARA LA PROVISIÓN Y MONITOREO DE EQUIPOS MESH INALÁMBRICOS**

**Rojas, Raúl – LU 83.922**  
**Ganá, Christian Gustavo – LU 82.596**  
Ingeniería en Informática

Tutor:  
**Rodriguez Duc, Fernando Alberto, UADE**

30 de octubre de 2014



**UADE**

**UNIVERSIDAD ARGENTINA DE LA EMPRESA**  
**FACULTAD DE INGENIERÍA Y CIENCIAS EXACTAS**

---

## **Agradecimientos**

A nuestras familias, por su apoyo y comprensión, y a nuestro tutor, el Ing. Fernando Rodriguez Duc, por el apoyo que nos brindó para realizar el trabajo.

## Resumen

Gracias a sus características de alta confiabilidad, facilidad de despliegue, y bajo costo de adquisición y operación, las redes mesh inalámbricas se presentan como una opción válida para brindar servicios de telefonía y acceso a Internet en empresas de telecomunicaciones rurales y para otras aplicaciones e industrias.

La implementación de estas tecnologías implica para las empresas desafíos técnicos y operativos, siendo uno de ellos la configuración de los equipos que se despliegan en el terreno. Este problema se vé incrementado en empresas pequeñas y/o rurales por las limitaciones de personal técnico capacitado disponible y por la existencia de equipamiento heterogéneo.

El objetivo del presente trabajo es el desarrollo de una herramienta de software para simplificar la configuración de equipos mesh inalámbricos, que cumpla dos premisas fundamentales: ser de bajo costo y de fácil utilización.

Aunque existen estándares que apuntan a dar solución a la problemática planteada, su implementación es de poca utilidad para empresas pequeñas, debido a la variedad del equipamiento utilizado, a que no todos los equipos soportan estos protocolos, y a que el reducido número de equipos instalados de cada modelo no justifica la inversión.

El desarrollo de la herramienta resultante se realizó como un proceso iterativo que facilitó una amplia evolución del producto final respecto del diseño original. Esta evolución permitió simplificar la herramienta final, reducir los requerimientos de equipamiento para el cliente y las necesidades de entrenamiento para el usuario, y por ende los costos de la solución, a la vez que permitió generar una arquitectura fácil de extender y mantener, gracias a que los principales componentes se encuentran centralizados en un servidor de configuración alojado en la nube, y a que el segmento que se utiliza en las instalaciones del cliente se entrega como un *computer appliance* dentro de un router programable de bajo costo.

También se obtuvo una experiencia positiva con el producto de software de VPN empleado para enlazar todos los equipos involucrados en la operación del sistema, porque posibilita que todas las comunicaciones se realicen en un contexto seguro y a la vez permite la comunicación en ambos sentidos de todos los componentes del sistema, inclusive de aquellos que se encuentran dentro de la intranet del cliente, y que para acceder al servidor de

configuración alojado en la nube deben atravesar proxies, NATs y firewalls.

Se presenta también como anexo del trabajo, la descripción de una prueba de campo de redes mesh realizada en una cooperativa telefónica para evaluar la factibilidad de implementación de esta tecnología.

## **Abstract**

### **Web based provisioning and monitoring software for wireless mesh equipment**

Due to its high reliability, ease of deployment and reduced total cost of ownership, wireless mesh networks are an applicable technology for providing telephone and Internet access services for rural telcos and other industries and applications.

Deployment of these technologies pose technical and operational challenges, among them the configuration of equipment to be deployed in the field. In small or rural companies this problem is aggravated by the shortage of skilled technical personnel and the heterogeneity of equipment.

The aim of this work is to develop a software utility that simplifies the configuration of wireless mesh equipment, capable of fulfilling two major goals: low cost and easy operation.

Although there are standards that address this issue, their implementation is not useful for small companies due to the mixed equipment used, not all of them supporting the same standard protocols and also because the small number of units of each model of equipment deployed does not justify the investment.

Development of the resulting solution was accomplished through an iterative process which enabled an ample evolution of the final product when compared to its original design. This evolution contributed to the simplicity of the final product, reduced operator equipment requirements and user training needs, thus lowering the total cost of ownership of the solution, and at the same time resulted in a easy to extend and maintain design, since its main components are centralized in a cloud hosted configuration server and the component that is deployed in the operator premises is provided as a computer appliance built into a low cost programmable router.

A positive experience was also gained with the VPN software used to link the components that participate in system operation, which makes all communications secure and at the same time enables two-way communication of every system component, even for those that reside in the operator intranet, who need to traverse proxies, NATs and firewalls in order to reach the configuration server.

It is also included, as an appendix, a description of a wireless mesh network field test conducted in a telco coop to evaluate the feasibility of deployment of this technology.

## Contenidos

Agradecimientos.....	ii
Resumen.....	iii
Abstract.....	v
Contenidos.....	vi
1. Introducción.....	1
2. Antecedentes.....	2
2.1. Estándares para configuración de hardware de telecomunicaciones.....	2
2.1.1. TR-069 o CWMP.....	2
2.1.1.1. Arquitectura básica.....	3
2.1.1.2. Configuración inicial.....	3
2.1.1.3. Capacidades del protocolo.....	4
2.1.1.4. Modelos de datos estándar.....	5
2.1.1.5. Ventajas.....	5
2.1.1.6. Nivel de adopción del estándar.....	6
2.1.2. SNMP.....	7
2.1.2.1. Diseño.....	7
2.1.2.2. Capacidades del protocolo.....	8
2.1.2.3. Modelos de datos estándar.....	8
2.1.2.4. Estado del estándar.....	9
2.2. Tecnología de redes mesh inalámbricas.....	9
2.2.1. Descripción general.....	9
2.2.2. Componentes de las redes Mesh inalámbricas.....	10
2.2.3. IEEE 802.11 y IEEE 802.11s.....	11
2.2.3.1. Definiciones de la norma 802.11s.....	12
2.2.4. Ventajas de la tecnología de redes mesh inalámbricas.....	14
2.2.5. Potencial de las redes mesh inalámbricas.....	15
2.2.6. Proyecto Serval.....	16
2.2.7. Uso de las redes mesh inalámbricas en Smart Grids.....	16
2.2.8. Las redes mesh en las cooperativas telefónicas en la Argentina.....	16
2.3. Prueba de red mesh en Cooperativa telefónica de Sarmiento, Chubut.....	17
2.3.1. Descripción de la cooperativa.....	17
2.3.2. Situación a solucionar (motivo por el que se ensayó el uso de mesh).....	18
2.3.3. Pruebas realizadas.....	18
3. Descripción del trabajo.....	18
3.1. Relevamiento inicial.....	18
3.1.1. Requerimientos y problemas a solucionar.....	18
3.1.2. Casos de uso.....	19
3.1.2.1. Caso de uso: Configurar un equipo.....	20
3.1.2.2. Caso de uso: Despachar un equipo.....	22
3.1.3. Características de los equipos a gestionar.....	23
3.1.3.1. Nodos cliente - Mesh Potato.....	23
3.1.3.2. Equipos Ubiquiti Nanostation M5.....	27
3.1.3.3. Módems ADSL Huawei EchoLife HG520s.....	27
3.1.4. Configuraciones a realizar en equipos Mesh Potato.....	28

3.1.5. Componentes principales.....	29
3.1.6. Secuencia de funcionamiento .....	30
3.1.7. Implementación .....	31
3.1.8. Selección del equipamiento a emplear .....	33
3.1.8.1. Router Ethernet programable .....	33
3.1.9. Selección de frameworks y lenguajes de programación.....	35
3.1.9.1. Para la aplicación Web de registro de equipos configurados.....	35
3.1.9.2. Para el desarrollo de los servicios de configuración de equipos.....	36
3.1.10. Selección de motor de base de datos.....	36
3.1.11. Selección del software de VPN a utilizar.....	37
3.2. Desarrollo de la aplicación y pruebas unitarias.....	38
3.3. Implementación del servicio en la nube.....	39
4. Metodología de Desarrollo.....	42
5. Pruebas Realizadas.....	43
5.1. Resultados del despliegue experimental realizado.....	43
6. Conclusiones.....	45
7. Bibliografía.....	47
8. Referencia de abreviaturas utilizadas.....	50
Anexo A – Descripción de las pruebas de red mesh realizadas.....	i
A.1. Configuración y equipos utilizados en la red mesh.....	i
A.2. Monitoreo realizado.....	i
A.2.1. Herramientas de monitoreo empleadas.....	i
A.2.1.1. SPUD (Simple Unified Dashboard for mesh networks).....	i
A.2.1.2. SmokePing.....	v
A.2.1.3. Cacti.....	v
A.2.1.4. Iperf.....	vi
A.2.1.5. SequireISP.....	vi
A.4. Conclusiones de la prueba .....	viii
Anexo B – Solución desarrollada.....	i
B.1. Aplicación web de registración.....	i
B.2. Servicio web de configuración .....	iii
archivos a copiar:.....	iv
B.3. Modelo de datos.....	iv

## 1. Introducción

La tecnología de redes mesh inalámbricas permite implementar rápidamente soluciones confiables y de bajo costo. Estas características hacen que sean una alternativa interesante para brindar servicio de telefonía y acceso a Internet en zonas de baja urbanización carentes de infraestructura de telecomunicaciones.

En este trabajo se pretende desarrollar una solución para un aspecto operativo que se presenta cuando se implementa una red de tecnología mesh, la configuración y registración de los equipos desplegados en el terreno.

El objetivo del trabajo es el desarrollo de una herramienta de software basada en la web, para simplificar la configuración de equipos mesh inalámbricos, para que una pequeña compañía telefónica de tipo cooperativa, pueda brindar servicios de telefonía y datos en una nueva urbanización que no cuenta con tendido de cables telefónicos.

Para poder sacar provecho de las ventajas potenciales que ofrece la tecnología de redes mesh, los equipos empleados deben poder ser configurados para ser usados al máximo de sus capacidades, y poder ser re-configurados para adaptarse a los cambios y mejoras que vaya necesitando la red con el paso del tiempo. Al mismo tiempo, la implementación, operación y el mantenimiento de la red debe mantenerse simple, inclusive cuando el equipamiento utilizado es heterogéneo por provenir de diferentes fabricantes y estar compuesto por diferentes modelos y diferentes versiones de cada modelo.

Por lo anterior, la herramienta a desarrollar debe satisfacer dos premisas fundamentales: ser de bajo costo, para que no incremente el costo total de adquisición de la solución mesh y pueda ser adquirida por compañías de bajo presupuesto, y ser de fácil utilización, para que pueda ser operada por personal sin conocimientos técnicos, debido a que en estas empresas los técnicos especializados habitualmente son recursos humanos escasos y costosos.

Asimismo, para que la solución pueda ser de bajo costo es necesario que la misma herramienta pueda ser aprovechada por múltiples empresas, lográndose de esta forma una escala que permita el desarrollo de adaptadores específicos para la configuración automática de diferentes tipos de equipos. Por ese motivo, la aplicación se plantea como un servicio a ser implementado en la web.

El trabajo comenzará con una descripción del estado actual de la tecnología mesh



inalámbrica y las ventajas potenciales de su utilización en cooperativas de telecomunicaciones en la Argentina, y en otras industrias que pueden beneficiarse con la misma tecnología.

También se presentará brevemente el estado del arte y un estándar existente para la problemática de configuración y mantenimiento de equipos terminales de cliente (CPE).

Luego se describirá el proceso de desarrollo de la herramienta, comenzando por presentar los requerimientos relevados, el diseño preliminar de la solución, su evolución a lo largo del proyecto, y la selección de los productos y tecnologías empleadas y su justificación. También se describirá la metodología empleada en el desarrollo del proyecto, y se presentarán los resultados obtenidos en las pruebas de operación realizadas.

Finalmente se presentarán las conclusiones obtenidas por el desarrollo del proyecto.

Se incluyen también, como anexos del presente trabajo, la descripción de una prueba de campo de redes mesh realizada en una cooperativa telefónica en relación con la situación que motiva el presente trabajo, presentando los resultados obtenidos y las conclusiones extraídas, e información de la herramienta desarrollada formada por fragmentos del código fuente desarrollado, captura de pantallas y del manual del usuario.

## **2. Antecedentes**

### **2.1. Estándares para configuración de hardware de telecomunicaciones**

#### **2.1.1. TR-069 o CWMP**

TR-069 es un protocolo de comunicación entre equipos desplegados en las instalaciones del cliente o CPE (por las siglas de su denominación en inglés: *Customer Premises Equipment*) y servidores de auto-configuración o ACS (por las siglas de su denominación en inglés: *Auto-Configuration Server*) (BLACKFORD y DIGDON, 2013).

Es un protocolo diseñado por el DSL Forum, siendo también conocido como CWMP (por las siglas de *CPE WAN Management Protocol*).

Inicialmente se diseñó para realizar el aprovisionamiento de equipos DSL, y en la actualidad está siendo utilizado para gestionar otros equipos de tipo CPE, tales como sintonizadores/decodificadores de televisión (conocidos como STB por sus siglas de *Set-Top Box*), *Femtocells* (puntos de acceso de telefonía celular para interiores), cable-módems, teléfonos IP, equipos gateway residenciales de PON (*Passive Optical Network*), cámaras de

video-vigilancia, etc.

### 2.1.1.1. Arquitectura básica

El protocolo define la existencia de un servidor de auto-configuración o ACS (por las siglas de su nombre en inglés *Auto-Configuration Server*) y de clientes, que son los equipos de tipo CPE a configurar.

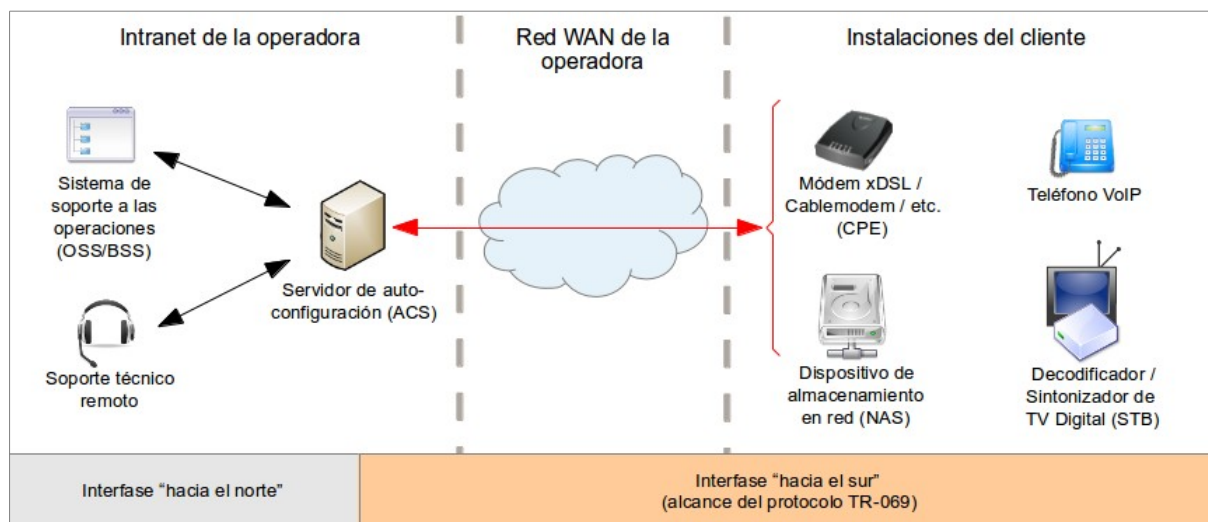


Figura 1: Esquema de conectividad del protocolo TR-069

Las comunicaciones se realizan en el marco de sesiones, que siempre son iniciadas por los equipos cliente, aunque está previsto un método que permite al servidor requerir al cliente el inicio de una sesión. Normalmente los clientes se configuran para que accedan periódicamente al servidor con el fin de obtener actualizaciones del firmware o de la configuración de la red.

Las órdenes y los datos son transmitidos como texto con formato XML/SOAP, sobre protocolo HTTP o HTTPS. El establecimiento de las sesiones y la autenticación se realizan mediante claves pre-acordadas entre los equipos. A su vez los equipos cliente pueden validar la autenticidad servidor ACS por medio de su certificado digital.

### 2.1.1.2. Configuración inicial

El protocolo prevé un mecanismo para que los equipos CPE adquieran en forma automática la configuración de acceso al servidor de auto-configuración del operador. Esto se logra mediante el uso de características estándar de los servidores DHCP.

En los requerimientos hechos por equipos clientes a servidores DHCP, los clientes pueden enviar el identificador de clase del fabricante (*vendor class identifier*) en la opción 60

de los mensajes. En base a esto, los servidores DHCP pueden devolver a los clientes información específica para ese identificador de clase, en la opción 43 de los mensajes.

El protocolo TR-069 define que en la opción 43 el servidor DHCP debe devolver la URL, nombre de usuario y contraseña, para que el equipo cliente pueda conectarse al servidor de auto-configuración (ACS).

### 2.1.1.3. Capacidades del protocolo

Lo que el protocolo permite hacer es:

- Al ACS sobre los CPE:
  - leer parámetros : esto permite consultar cualquier parámetro de la configuración del CPE, y también consultar el estado del CPE (la información de estado se expone en forma de parámetros);
  - modificar parámetros: modificar la configuración del CPE. Por ejemplo:
    - las credenciales para acceso a PPP, SIP y otros servicios (PERKOV L., 2014);
    - la configuración de servicios como DNS y DHCP;
    - la configuración de la red inalámbrica (Wi-Fi);
    - la configuración de rutas, firewall, y de calidad de servicio;
    - la configuración de bloqueo de acceso a sitios web, restricción de horario de acceso a sitios de juegos on-line, bloqueo de canales (en equipos STB), etc.;
  - reiniciar el CPE;
  - ordenar al CPE recuperar su configuración de fábrica;
  - ordenar al CPE que inicie la descarga de un archivo (ej. nueva imagen de firmware) ;
  - ordenar al CPE que envíe un archivo de log;
  - agendar una próxima sesión de acceso al ACS ;
- A los CPE sobre el ACS:
  - enviar log: para poder analizar fallas remotamente;
  - notificar la finalización de descarga de archivos: esto permite implementar lógica en el ACS para ejecutar tareas que impliquen varios pasos. Por ejemplo,

reiniciar el equipo una vez que se haya descargado un nuevo firmware para que éste se pueda instalar.

Más allá de las capacidades previstas por el protocolo, en la práctica las capacidades de las implementaciones que se utilicen dependerán de las opciones que estén soportadas en común en cada combinación de equipo cliente (CPE) y servidor (ACS).

#### **2.1.1.4. Modelos de datos estándar**

Además del protocolo TR-069, el Broadband Forum también define modelos de datos estándar para diferentes tipos de equipos CPE.

Estos modelos de datos son conjuntos de parámetros organizados jerárquicamente que definen objetos administrados accesibles a través del protocolo TR-069 para un tipo particular de dispositivo o servicio.

Estos modelos de datos se definen y publican mediante TRs (*Technical Reports*) adicionales. En la actualidad existen aproximadamente 20 esquemas definidos (PERKOV L., 2014).

Entre estos podemos mencionar:

- TR-104: para equipos de VoIP;
- TR-135: para equipos STB (*set-top box*);
- TR-140: para equipos de almacenamiento en red (NAS – *Network Attached Storage*);
- TR-196 ediciones 1 y 2: para equipos *Femtocell* (WALLS J., 2014).

#### **2.1.1.5. Ventajas**

La adopción del protocolo TR-069 tiene por un lado ventajas directas, tales como facilitar el despliegue y mantenimiento de CPEs, y reducir costos operativos gracias a la posibilidad de administrar y realizar diagnóstico y mantenimiento de estos equipos en forma remota.

Adicionalmente se observa que el uso del protocolo proporciona otras ventajas potenciales, relacionadas con la posibilidad de brindar más servicios de valor agregado al cliente sobre los CPE ya desplegados. Por ejemplo el uso de esta tecnología permite a las empresas ofrecer a sus clientes servicios de bloqueo de acceso a determinados recursos (ej. a contenido web o a canales de TV), poner límites de horario para el acceso a juegos en línea,

evitar el acceso de menores a redes sociales en horarios en que los padres no se encuentran presentes, administrar características avanzadas de los equipos CPE desplegados (ej. configurar calidad de servicio en routers de acceso a Internet, configurar NAT en routers, configurar seguridad en redes Wi-Fi), operar servicios de backup en forma remota, etc.

Además, el uso de los servicios de los servidores ACS no está limitado a la compañía operadora. También pueden implementarse portales que permitan al cliente acceder para administrar y configurar los servicios de sus CPE. Este enfoque tiene la ventaja de que permite implementar portales más amigables para el usuario final que los provistos por los fabricantes de los equipos, y además permite unificar la interface con equipos de diferentes fabricantes, lo que facilita el entrenamiento del personal de soporte de la operadora y mejora la experiencia del cliente cuando dispone de más de un CPE o cuando es necesario reemplazarle el CPE.

#### **2.1.1.6. Nivel de adopción del estándar**

Una estimación publicada a mediados de 2014 indica que existen 250 millones de equipos CPE en el mundo que están siendo administrados mediante este protocolo (WALLS J., 2014).

No obstante esto, aún se encuentran en operación muchos equipos que no soportan el protocolo, y hay fabricantes que aún no lo implementan en los equipos que producen.

En la actualidad existen o se encuentran en desarrollo diferentes implementaciones de aplicaciones para protocolo TR-069, tanto de aplicaciones cliente como de servidores de auto-configuración. Algunos son desarrollados en forma de software propietario y otros como proyectos de código abierto. Entre los primeros se encuentran los desarrollos de la empresa Axiros, que ofrece el stack completo de aplicaciones para soporte del protocolo TR-069, desde implementaciones de clientes TR-069 hasta servidores ACS. Entre los de código abierto se encuentra el proyecto de aplicación cliente *freecwmp* de licencia GPLv2 (freecwmp, 2014), y proyectos derivados, como EasyCwmp, que pueden instalarse en CPEs con sistema operativo OpenWRT y son portables a otros sistemas operativos basados en Linux). También existe el proyecto freeacs-ng que apunta a desarrollar un ACS publicado bajo licencia GPLv2 (freeacs-ng, 2014).

La existencia de opciones tanto de software libre como propietarias, permite presuponer que en el futuro más fabricantes de CPEs incluirán soporte de este protocolo en sus

productos.

### 2.1.2. SNMP

SNMP (*Simple Network Management Protocol*) es un protocolo para monitoreo y administración de equipos conectados en red. Puede ser usado para administrar cualquier tipo de equipo, como switches administrados, routers, UPSs, sensores de temperatura, servidores, impresoras, módems, etc.

SNMP funciona en la capa más alta del stack OSI, aplicación, y utiliza protocolo UDP.

#### 2.1.2.1. Diseño

**Administradores y agentes:** SNMP define dos actores posibles en la comunicación: administradores y agentes. Los primeros son servidores que corren aplicaciones capaces de monitorear y administrar dispositivos. Normalmente se los conoce como NMS, por sus siglas de *Network Management Station*. Los agentes son software que corre en los dispositivos administrados.

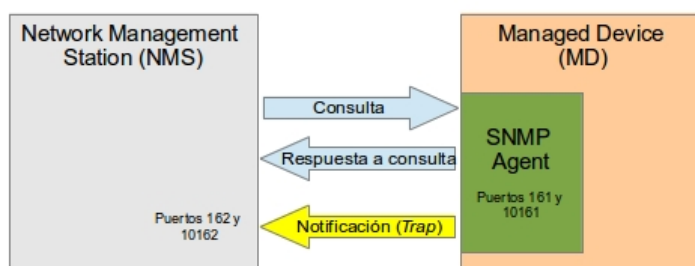


Figura 2: Comunicaciones posibles entre los actores definidos por el protocolo SNMP.

Los NMS se encargan de ejecutar consultas (*queries* en la terminología de SNMP) sobre los agentes de los dispositivos administrados para consultar información (ej. estado del dispositivo), cambiar valores de configuración o ejecutar acciones (ej. reiniciar el dispositivo). También tienen la capacidad de recibir notificaciones provenientes de los dispositivos administrados (llamadas *traps* en la terminología de SNMP).

Los agentes se encargan de satisfacer las consultas de los NMS, y notificar, mediante *traps*, a los NMS de determinados eventos (ej. la temperatura de un equipo supera un umbral establecido).

**Organización de los datos:** Debido a que el protocolo SNMP fue diseñado para administrar cualquier tipo de dispositivo, no incluye la definición de un modelo de datos

específico, sino que define elementos de datos básicos (llamados objetos) con los que se pueden formar bases de datos jerárquicas con estructura de árbol llamadas MIB (*Management Information Base*) con las que se puede modelar cualquier tipo de información. Además las MIB pueden ser un objeto dentro de otra MIB.

Está definido también un MIB general para todos los agentes, llamado MIB-II, que contiene información técnica y estadística básica aplicable a cualquier tipo de dispositivo.

También existen lineamientos para crear la estructura interna de los MIB, y un registro de números identificadores de empresa, para que cada empresa pueda disponer sus datos dentro de la estructura de un MIB sin riesgo de que existan colisiones de nombres de objetos.

Los objetos tienen un nombre que debe ser único dentro de la MIB. Este nombre puede contener números o por caracteres alfanuméricos, y está formado por la concatenación de toda la secuencia de nodos del árbol a recorrer, desde el raíz hasta la hoja dónde se encuentra ubicado el objeto.

(MAURO y SCHMIDT, 2005)

### **2.1.2.2. Capacidades del protocolo**

Lo que el protocolo permite hacer es:

- Al NMS sobre los dispositivos administrados:
  - consultar el valor de uno o más objetos;
  - modificar el valor de un objeto;
  - ejecutar tareas administrativas, tales como reiniciar el dispositivo;
- A los dispositivos administrados sobre el NMS:
  - enviar *traps* para notificar que se produjeron eventos.

### **2.1.2.3. Modelos de datos estándar**

Más allá de la norma que define SNMP, se han desarrollado otras normas, en forma de RFCs (*Request For Comments*), que definen MIBs específicos para determinadas clases de dispositivos, con el fin de estandarizarlos. Entre estos se encuentran los siguientes:

- RDBMS (*Relational Database Management System*) MIB (RFC 1697)
- RADIUS Authentication Server MIB (RFC 2619)
- DNS Server MIB (RFC 1611)

#### 2.1.2.4. Estado del estándar

Actualmente el protocolo SNMP es ampliamente utilizado, existiendo implementaciones de agentes para la mayor parte de los sistemas operativos que se utilizan como firmware.

No obstante esto, debido a las limitaciones de seguridad que tiene este protocolo, al menos en sus primeras versiones, hacen que en dispositivos críticos solamente se le habiliten las funciones de monitoreo, y no las de configuración.

## 2.2. Tecnología de redes *mesh* inalámbricas

### 2.2.1. Descripción general

Las redes malladas inalámbricas (o WMN por las siglas de su denominación en inglés: *Wireless Mesh Network*), son una tecnología que tiene el potencial de entregar acceso a Internet de banda ancha y telefonía, y brindar cobertura de red LAN en espacios amplios con bajo costo y rápido despliegue (Mojtaba et al, 2011). Se espera que permitan concretar el sueño de tener conectividad de red en todo momento y en todo lugar (AKYILDIZ y XUDONG, 2009).

Una red mesh (mallada) es una red dispuesta en una topología en la que cada nodo está conectado con otros nodos por más de un camino posible. Esta propiedad permite diseñar redes que tengan la capacidad de auto-arreglarse, lo que contribuye a aumentar la confiabilidad de la red. Por ejemplo, cuando el vínculo entre dos nodos falla, la comunicación puede seguir lográndose por caminos alternativo, a través de otros nodos.

Asimismo, las redes mesh pueden tener la capacidad de configurarse automáticamente, a partir de detectar los nodos vecinos con los que tiene conectividad y de detectar las novedades que se producen en la red, tales como la aparición o desaparición de nodos, o la adquisición o pérdida de conectividad entre nodos, y mediante diferentes algoritmos pueden determinar las rutas por las que deben enviar paquetes de datos para que lleguen a destino.

Estas características hacen que las redes mesh tengan la capacidad de auto-organizarse, auto-configurarse y auto-arreglarse, lo que permite lograr facilidad y rapidez de despliegue, facilidad de mantenimiento, bajo costo, alta escalabilidad y servicio de alta confiabilidad, a la vez que mejoran las características de capacidad, conectividad y resiliencia de la red (Yan et al, 2007).



Esta topología puede ser usada tanto en redes cableadas como inalámbricas, aunque en el caso de las redes cableadas, la cantidad de conductores necesarios para realizar las conexiones resulta muy alto, haciendo inviable su implementación por el costo de los conductores. Por ese motivo en la práctica esta topología encuentra su mayor aplicación en redes inalámbricas.

La capacidad de auto-organización de las redes mesh inalámbricas permite su utilización con nodos móviles, dando lugar a las llamadas redes ad hoc móviles (o MANETs por las siglas de *Mobile Ad Hoc Network*, sus denominación en inglés). Estas redes MANET pueden tener diferentes niveles de dinamismo, pudiendo estar integradas por nodos montados en vehículos con alta movilidad (llamadas *Vehicular Ad Hoc Networks* o VANETs), o conformadas por nodos relativamente estáticos, como computadoras de tipo laptop diseminadas en un campus. También se están desarrollando trabajos para implementar MANETs entre *smartphones*, que constituyen una alternativa económica al servicio de compañías de telefonía celular, o inclusive que pueden funcionar cuando ante situaciones de desastres naturales, como terremotos, las redes de celulares dejan de funcionar (ServalProject, 2014).

Los algoritmos que proveen inteligencia al encaminamiento de paquetes permiten a los nodos de redes mesh inalámbricas enviar paquetes hacia nodos que no se encuentran dentro de su radio de alcance directo. En esos casos los paquetes pueden llegar al nodo de destino en múltiples saltos (*hops*) pasando por distintos nodos intermedios. Esta característica es una gran ventaja potencial en términos de confiabilidad de las redes mesh frente a las redes inalámbricas tradicionales de un solo salto (Johnson et al, 2007).

### 2.2.2. Componentes de las redes Mesh inalámbricas

Las redes mesh inalámbricas están compuestas por tres tipos de nodos:

- **Nodos cliente:** son los dispositivos finales, usuarios de la red mesh, por ejemplo routers Wi-Fi que sirven como *hot spot*, equipos de usuario final como computadoras y *smartphones*, equipos compuestos que incluyen *hot spot* Wi-Fi e interface adaptadora para teléfono analógico (ATA por las siglas de su denominación en inglés *Analog Telephone Adapter*), medidores inteligentes, cámaras de vigilancia IP, sensores meteorológicos y de contaminación ambiental, etc. Estos nodos pueden ser fijos o móviles.

- **Nodos router:** son nodos que están presentes en la red para encaminar el tráfico de datos. Normalmente no se origina ni termina en ellos el tráfico de datos. Normalmente son fijos o tienen limitada movilidad, y son equipos de alta confiabilidad. La potencia de transmisión en estos nodos es baja en relación a los nodos inalámbricos tradicionales, en concordancia con la estrategia de alcanzar grandes distancias por medio de salto entre varios nodos, en lugar de transmisión directa al nodo de destino. Además los protocolos de control de acceso en estos nodos soportan múltiples canales y múltiples interfaces, para dar mayor ancho de banda a la red (al poder recibir y transmitir al mismo tiempo) (Mojtaba et al, 2011).
- **Nodos gateway:** los nodos gateway son nodos router que además incorporan enlace con la red jerárquica superior (ej. Internet o backbones), con acceso mediante una interface adicional y usando tecnologías que pueden ser inalámbricas (ej. WiMAX, redes de celulares y satelitales) o cableadas (fibra óptica, cable, DSL, PLC, etc.).

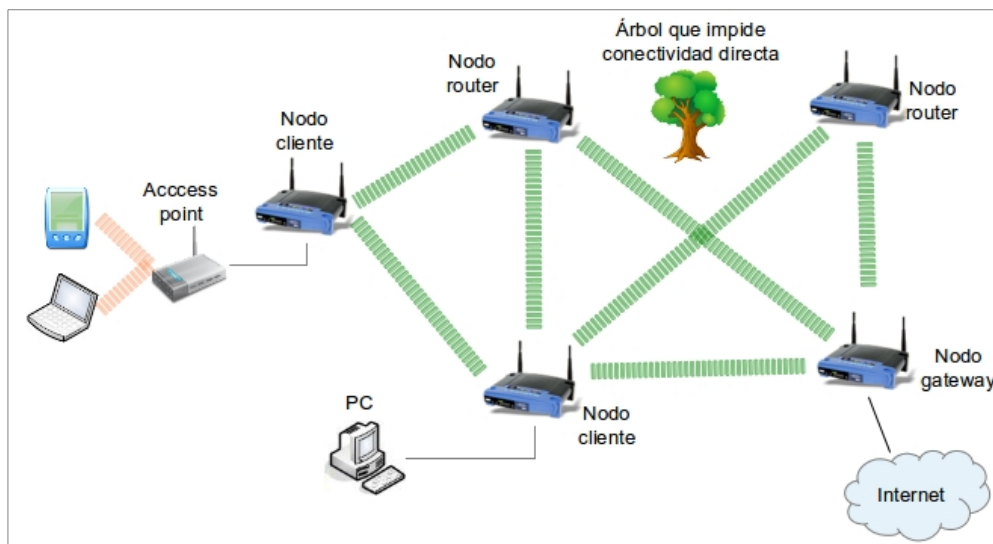


Figura 3: Esquema de una red mesh inalámbrica en la que se muestran los diferentes tipos de nodo que pueden componerla.

### 2.2.3. IEEE 802.11 y IEEE 802.11s

En general las redes mesh inalámbricas se implementan utilizando equipos de norma IEEE 802.11. De hecho, un factor que ha contribuido a la popularidad de las redes mesh inalámbricas es la amplia disponibilidad de equipos inalámbricos, en especial los equipos

WLAN de norma IEEE 802.11 (*MANGUES-BAFALLUY et al., 2009*).

IEEE 802.11 es un conjunto de normas que definen métodos de transmisión para redes LAN inalámbricas (o WLAN por las siglas de su denominación en inglés: *Wireless Local Area Network*). Cubren lo referido a dos primeras capas de las siete del modelo de referencia OSI de redes, lo relacionado con control acceso al medio (o MAC por las siglas de su denominación en inglés: *Media Access Control*) y con la capa física.

Las versiones más usadas de esta norma son las 802.11b, 802.11g y 802.11n, empleadas para brindar conectividad inalámbrica en hogares y oficinas.

Como resultado de la evolución de la tecnología mesh inalámbrica, se generó la especificación IEEE 802.11s, que es una extensión de la norma 802.11 para permitir que nodos inalámbricos se conecten entre ellos sin necesitar un punto de acceso (o AP por las siglas de su denominación en inglés: *Access Point*).

En las redes inalámbricas que usan un AP, la comunicación entre diferentes nodos de la red siempre pasa por el AP. La norma 802.11s permite la comunicación directa entre nodos de una red inalámbrica, sin necesidad de un AP. No obstante esto, la verdadera importancia de la norma 802.11s se presenta cuando existen múltiples nodos inalámbricos. Usando el estándar de redes mesh definido por 802.11s los nodos pueden formar una red de múltiples saltos (*hops* en inglés) dónde todos los enlaces entre los nodos son inalámbricos. Esto implica que no se requiere una infraestructura cableada para establecer la red. Esto es muy útil en escenarios dónde no se dispone de infraestructura. (The Linux Kernel Organization, 2014)

Un ejemplo de ello son las situaciones de catástrofe, dónde los equipos de salvamento pueden llevar equipos portátiles que pueden interconectarse para intercambiar información, inclusive entre nodos que no son alcanzables en forma directa entre sí. Un ejemplo puntual de esto es el objetivo del Proyecto Serval que se describe más adelante.

Otro ejemplo son los despliegues de computadoras portátiles realizados masivamente en el marco del programa OLPC (*One Laptop per Child*), dónde se utiliza esta tecnología para formar entre los equipos una red ad-hoc, que permite interconectar todos los equipos que se encuentran en la escuela con mínimos requerimientos de infraestructura de telecomunicaciones (Proyecto Ceibal, 2014).

### **2.2.3.1. Definiciones de la norma 802.11s**

A continuación se describen brevemente algunos de los elementos que define esta

norma:

**Tipos de nodos mesh:** IEEE 802.11s establece tres tipos de nodos que conforman las redes mesh inalámbricas:

- *Mesh Point* (MP): Los MP deben manejar un protocolo de administración entre pares (*Peer Link Management*) que es usado para descubrir nodos vecinos y mantener información acerca de ellos. El descubrimiento de nodos se limita a los que están dentro del alcance directo del MP.

Para comunicarse con nodos que están a más de un salto de distancia, los MP también soportan el protocolo híbrido de malla inalámbrica (o HWMP por las siglas de su denominación en inglés *Hybrid Wireless Mesh Protocol*).

- *Mesh Portal* (MPP): Una red mallada IEEE 802.11s puede ser utilizada para múltiples propósitos, siendo uno de ellos el proveer acceso a Internet. En esos casos, al menos uno de los nodos está conectado a Internet. Los usuarios conectados a la red mesh pueden acceder a Internet a través de esos MPP, que están conectados a la red mesh y a Internet al mismo tiempo. Para esto los MPP deben disponer de al menos dos interfaces de red interconectadas en forma de bridge para proveer la funcionalidad de gateway.
- *Mesh Access Point* (MAP): Un MAP es un AP tradicional aumentado con la funcionalidad mesh. De esta forma puede servir como AP y ser parte de la red mesh al mismo tiempo.

Todos los tipos de nodo deben tener la capacidad de encaminar tramas originadas en un nodo y destinadas a algún otro nodo de la red.

**Protocolo de administración entre pares:** el protocolo de administración entre pares se utiliza para descubrir y mantener la lista de nodos vecinos de cada nodo. Cuando un nodo inalámbrico se configura como MP, éste comienza a emitir señales a modo de balizas (*beacons*) para que el nuevo nodo pueda ser descubierto por los otros nodos que se encuentran dentro de su alcance.

**Protocolo híbrido de malla inalámbrica (HWMP):** “*HWMP es un protocolo de encaminamiento en redes malladas que combina la flexibilidad de encaminamiento bajo demanda con un árbol de topología generado proactivamente. La combinación de elementos reactivos y proactivos en HWMP permite lograr una óptima y eficiente selección de caminos en una amplia variedad de redes malladas (sean con o sin infraestructura)*” (Joshi A. et al.,

2006).

El modo de operación que utiliza depende de la configuración, aunque no son exclusivos, pudiendo ser utilizados en forma concurrente. En el modo bajo demanda se utiliza en situaciones dónde no hay ningún nodo en la malla configurado como nodo raíz, aunque en algunas circunstancias también es usado aunque haya un nodo raíz. El modo proactivo se basa en el uso de un árbol de topología construido previamente en base a mensajes enviados periódicamente por un nodo configurado como raíz.

#### 2.2.4. Ventajas de la tecnología de redes mesh inalámbricas

Las ventajas que tienen las redes mesh inalámbricas son:

**Confiabilidad y robustez:** gracias a que tienen la capacidad de manejar rutas redundantes para el encaminamiento de los paquetes de datos a transmitir, las fallas transitorias en un nodo intermedio, o la falta de conectividad por obstáculos (ej. crecimiento del follaje de árboles que interrumpen la línea de vista entre dos antenas) habitualmente se corrigen en forma automática mientras dura la falla, y vuelven a su estado original cuando cesa la causa de la falla. Esto permite implementar de manera natural redes sin puntos únicos de fallo (o SPOF por las siglas de su denominación en inglés: *Single Point of Failure*).

**Bajo costo de implementación:** son múltiples los factores que contribuyen a que el costo de implementación de redes mesh inalámbricas sea bajo. Entre estos podemos mencionar:

- como todas las redes inalámbricas, evitan el costo de cableado;
- habitualmente se implementan con hardware de bajo costo por ser de fabricación en masa, tal como los equipos Wi-Fi de protocolos IEEE 802.11 (Manoj et al, 2007) (FLICKENGER Rob et al., 2007) o los equipos con tecnología Zigbee (IEEE 802.15);
- la facilidad de despliegue contribuye a reducir costos de mano de obra y hardware;

**Bajo costo de operación:** cuando se las compara con el uso de servicios de comunicación de datos inalámbricos provistos por terceros, como las redes de telefonía celular o los enlaces satelitales, se obtiene un ahorro en costos fijos a lo largo de la vida útil de la red. Además las características de confiabilidad y robustez de esta tecnología ayudan a evitar penalizaciones y lucro cesante por indisponibilidad de la red, y a evitar costos operativos porque no es necesario atender fallas transitorias, y porque las fallas no transitorias

que no impiden el funcionamiento de la red pueden ser resueltas sin urgencia en los horarios de trabajo habitual del personal técnico.

**Facilidad de despliegue:** gracias a que las redes mesh tienen la habilidad de auto-descubrir a los nuevos nodos que se incorporan, el agregado de nodos en la red no requiere más configuración que la del nodo en sí. Todos los nodos cercanos de la red descubrirán al nuevo nodo y lo agregarán como ruta posible para el encaminamiento de paquetes, y a su vez el nuevo nodo descubrirá a los nodos existentes y los agregará en sus rutas posibles. Además los trabajos de instalación física de los nuevos nodos es simple, porque al ser amplias las áreas de cobertura de las redes existentes, los nuevos nodos se pueden ubicar en los lugares que resulten más convenientes (usando mástiles existentes y/o en lugares de fácil acceso). A su vez, como las antenas empleadas habitualmente no son direccionales y los nuevos nodos se pueden instalar de a uno por vez, no es necesario contar con dos equipos de trabajo coordinados en el terreno para poder realizar la instalación de nuevos nodos, como puede suceder con los enlaces inalámbricos punto a punto.

**Escalabilidad:** mientras que en las redes inalámbricas tradicionales, la performance de la red se reduce conforme aumenta el número de nodos, en las redes mesh inalámbricas la capacidad de transmisión mejora cuando se incorporan más nodos, gracias a que permiten hacer un mejor balanceo de carga y a disponer de más rutas alternativas para encaminar los paquetes (Mojtaba et al, 2011).

### 2.2.5. Potencial de las redes mesh inalámbricas

Por sus características, las redes inalámbricas en general, y especialmente las de topología mesh basadas en equipos de norma IEEE 802.11 son ideales para brindar servicios de telecomunicaciones en zonas rurales, de bajo nivel de desarrollo, y donde no existe presupuesto para implementar otras tecnologías (FLICKENGER Rob et al., 2007).

Por este motivo se han realizado muchas experiencias en el continente africano, en especial varios desarrollos realizados en Sudáfrica y otros realizados en Europa pero orientados a su utilización en África.

También pueden servir para cubrir porciones de ciudades dónde la demanda de tráfico no justifica la inversión de realizar un tendido de fibra óptica, o incluso para brindar servicio en asentamientos temporales. (Manges-Bafalluy et al., 2009)

### 2.2.6. Proyecto Serval

El proyecto Serval tiene como objetivo desarrollar un sistema de telecomunicaciones utilizando teléfonos celulares, capaz de funcionar fuera de la zona de cobertura de las redes de celulares. Está previsto para su uso en zonas rurales donde no hay cobertura de redes de celulares, y también en zonas donde, por catástrofes de cualquier tipo, las redes de celulares hayan quedado fuera de servicio.

Consiste en el desarrollo de un software que puede ser instalado en teléfonos celulares de bajo costo que tengan sistema operativo Android, con el que se implementa una red mesh móvil (MANET) que permite realizar comunicaciones por voz y mensajes de texto (ServalProject, 2014).

### 2.2.7. Uso de las redes mesh inalámbricas en *Smart Grids*

Por sus características, las redes mesh inalámbricas están encontrando aplicación en otras industrias, además de la de las telecomunicaciones. Un ejemplo es su uso para interconectar componentes que brindan inteligencia y control a las redes de transmisión y distribución de energía eléctrica (lo que se conoce bajo el nombre de *Smart Grid*) (KEMAL M., 2014). En particular las redes mesh se presentan como una alternativa para extender la cobertura de las redes de la compañía a toda su área de concesión, formando una FAN (*Field Area Network*), que puede ser empleada para interconectar los nodos concentradores de medidores inteligentes (AMR – *Automatic Meter Reading*) (FARIBA A., 2012), equipos de protección, tele-medición y tele-control. Inclusive algunos fabricantes proponen su utilización para la comunicación de vehículos y equipos de trabajo de la compañía dentro del área de cobertura de la FAN.

### 2.2.8. Las redes mesh en las cooperativas telefónicas en la Argentina

En la Argentina las compañías telefónicas en general, incluyendo a las cooperativas tienen sus tarifas de telefonía fija reguladas, con precios tope que se encuentran congelados desde el año 1999 (Comisión Nacional de Comunicaciones, 2014). A esto se suma, para el caso de las cooperativas rurales, el brindar servicio en zonas de baja densidad de población, lo que hace que los ingresos por facturación del servicio de telefonía fija sean bajos en relación a los costos de construcción y operación de las redes de telefonía.

Por este motivo las pequeñas empresas telefónicas rurales necesitan adoptar soluciones



de bajo costo, que les permitan captar nuevos clientes y proveer servicios de mayor precio, como acceso a Internet, para poder incrementar la facturación.

En ese contexto las redes mesh inalámbricas se presentan como una tecnología que les permite ampliar su zona de cobertura para servicios de telefonía e Internet, en forma rápida y con baja inversión y bajos costos operativos (MOTA E. et al., 2009).

Esta tecnología también puede ser empleada por cooperativas que se encuentran en zonas de explotación petrolera, para brindar servicio a los campamentos temporarios que emplazan habitualmente las compañías petroleras.

Por otro lado, el despliegue de redes inalámbricas de tecnología mesh implica para las compañías telefónicas la necesidad de disponer de personal técnico capacitado para poder realizar la implementación y el mantenimiento de estas redes.

Uno de los aspectos que requiere personal técnico capacitado es la configuración de los equipos mesh, que es compleja, propensa a errores, y la solución de estos errores una vez que los equipos se encuentran desplegados en el terreno resulta costosa. Por ende, es necesario disponer de soluciones que permitan automatizar en todo lo que sea posible estos trabajos, puesto que normalmente el personal técnico capacitado es escaso, y capacitar personal para hacerlo es costoso y no brinda otros beneficios.

### **2.3. Prueba de red mesh en Cooperativa telefónica de Sarmiento, Chubut**

#### **2.3.1. Descripción de la cooperativa**

La Sociedad Cooperativa de Provisión de Servicios Públicos de Sarmiento Ltda., es una cooperativa de servicios públicos que brinda servicio de distribución de energía eléctrica, telefonía y acceso a Internet, entre otros servicios, en la ciudad de Sarmiento, Provincia de Chubut, y alrededores. La ciudad se encuentra ubicada a unos 156 km de Comodoro Rivadavia, en dirección oeste, en el valle del Río Senguer, una zona de importante actividad de explotación petrolera.

Al 30 de junio de 2012 brindaba servicio de telefonía a 2421 usuarios, contando con centrales telefónicas analógicas para el servicio de telefonía en la mayor parte del área urbana, y centrales telefónicas y gateways de telefonía IP para algunos barrios y para las zonas rurales (Coopsar, 2014).



### **2.3.2. Situación a solucionar (motivo por el que se ensayó el uso de mesh)**

En la actualidad la Cooperativa tiene la necesidad de brindar servicio de telefonía e Internet a nuevas urbanizaciones, donde la densidad de población inicial es muy baja para justificar la inversión de realizar el cableado para implementar una red de telefonía convencional. La estrategia ideada consiste en brindar cobertura dentro de esa nueva urbanización en muy corto plazo y con baja inversión, mediante una red mesh inalámbrica, posponiendo, para cuando la densidad de población lo justifique, la construcción de una red cableada.

### **2.3.3. Pruebas realizadas**

Para verificar la posibilidad de emplear esta tecnología, la Cooperativa realizó en febrero de 2014 ensayos con tecnología mesh. En el Anexo “Anexo A – Descripción de las pruebas de red mesh realizadas” se describe la implementación de prueba realizada.

## **3. Descripción del trabajo**

### **3.1. Relevamiento inicial**

El trabajo comenzó un un relevamiento inicial realizado para conocer las necesidades reales de una cooperativa telefónica para realizar la configuración de sus equipos. A continuación se describen los resultados de este relevamiento.

#### **3.1.1. Requerimientos y problemas a solucionar**

Se desea que los nuevos equipos puedan ser configurados por personal poco calificado, por lo tanto el procedimiento de configuración debe simplificarse tanto como sea posible.

Se desea evitar la necesidad de tener que usar equipos laptop para realizar la configuración de los equipos, por la fragilidad de estos equipos, la corta autonomía con batería, y la posibilidad de que resulten utilizados para otros fines, causando ineficiencia y distracciones.

Preferiblemente los equipos deben poder ser configurados en el depósito, para agilizar los trabajos en el terreno y poder tener un mejor control y registro de la configuración que se le aplica a cada equipo.

Se desea que el equipamiento necesario para realizar la configuración de los equipos sea el mínimo necesario.

Los equipos que se utilizarán en la red mesh son los *Mesh Potato*. No obstante esto, la infraestructura que se desarrolle debe poder ser extendida para, en el futuro, poder configurar otros equipos, tanto para la red mesh como para otros usos. Por ejemplo, en el corto plazo se considera que puede ser aprovechado también para configurar los equipos utilizados en enlaces punto-a-punto y punto-multipunto, y también para configurar módems ADSL.

Dado que la red mesh será creada de antemano (los nodos router se desplegarán desde el inicio), solamente se necesita que la herramienta permita la configuración de nuevos nodos cliente. Eventualmente si se detectan zonas con mala cobertura o es necesario ampliar la zona de cobertura de la red mesh, los técnicos concurrirán al terreno a instalar nodos router adicionales o a reubicar los existentes.

Se requiere que el sistema registre automáticamente los identificadores utilizados en los equipos configurados, tales como dirección MAC del equipo, dirección IP asignada, y número telefónico asignado, para reducir el riesgo de que se pierda esta información, y a la vez asegurar un uso correcto de los stocks de identificadores disponibles (ej. números telefónicos).

Es necesario que el costo total de la solución a implementar sea bajo, puesto que la cantidad de equipos a configurar será reducida, lo que impide lograr un retorno en caso que la inversión realizada sea elevada. Además, para un número reducido de equipos, realizar la configuración en forma manual es una alternativa posible, contra la que la solución desarrollada debe competir.

### **3.1.2. Casos de uso**

A continuación se describen los dos principales casos de uso que describen la interacción del usuario con el sistema.

3.1.2.1. Caso de uso: Configurar un equipo

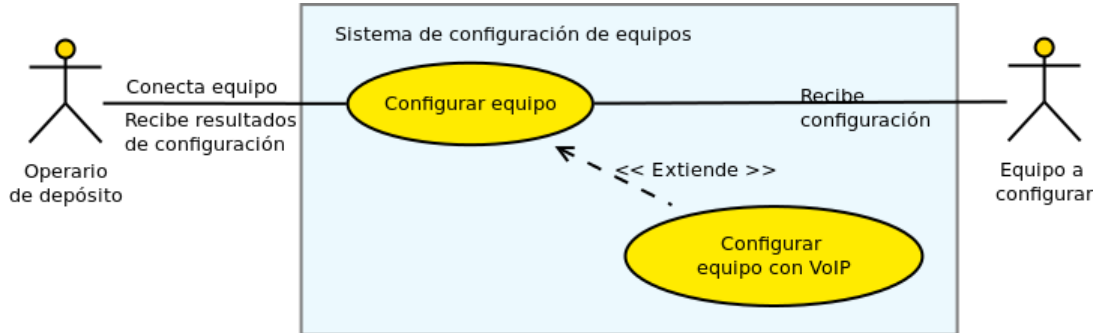


Figura 4: Diagrama de caso de uso "Configurar un equipo".

Título del caso de uso:	<b>Configurar un <i>equipo</i></b>
Terminología / Abreviaturas utilizadas:	<p><i>equipo</i>: equipo a ser configurado. Son equipos terminales que se instalan en el domicilio del usuario (CPEs). Pueden ser nuevos o usados que vuelven al depósito para su verificación y re-configuración antes de volver a ser utilizados.</p> <p><i>ACS</i>: servidor de auto-configuración. Servicio alojado en la nube que realiza la configuración y prueba remota de los <i>equipos</i>.</p> <p><i>router</i>: router dispuesto en el banco de trabajo del operario que sirve como interface entre el <i>equipo</i> y el servidor <i>ACS</i>.</p> <p><i>BD</i>: base de datos de activos técnicos, dónde se registra información de los <i>equipos</i> configurados.</p>
Objetivo:	Configurar el <i>equipo</i> para su utilización, asignarle sus identificaciones en la red (dirección IP y número telefónico cuando aplique), y registrarlo en la <i>BD</i> .
Alcance:	Sistema de configuración de equipos
Nivel:	Tarea primaria
Pre-condiciones:	<p>El <i>equipo</i> puede ser nuevo o usado.</p> <p>Exteriormente se observa que el <i>equipo</i> está en buenas condiciones.</p> <p>El <i>router</i> se encuentra conectado y encendido.</p>
Situación cuando finaliza exitosamente:	<p>El <i>equipo</i> queda registrado en la <i>BD</i> de activos técnicos.</p> <p>(Cuando corresponda) la dirección IP y el número telefónico asignados al <i>equipo</i> quedan marcados como usados.</p> <p>El sistema notifica al operador que la operación se completó exitosamente, informando el número de inventario correspondiente al <i>equipo</i> y el número telefónico asignado al <i>equipo</i> (cuando corresponda).</p>
Actores primarios:	Operario
Actores secundarios:	<i>equipo</i>
Disparador:	Conexión del <i>equipo</i> al <i>router</i>
Requerimientos de performance:	Se espera que la demora en la ejecución del caso no supere los 5 minutos.
Forma de comunicación con los actores:	Correo electrónico y voz sintetizada (para casos de <i>equipos</i> con telefonía VoIP)

Flujo normal	
Paso	Acción
1	El operario conecta el <i>equipo</i> al <i>router</i>
2	El <i>ACS</i> determina el tipo de <i>equipo</i> conectado y lo configura
3	El <i>ACS</i> notifica al operario el resultado de la configuración por correo electrónico

Extensiones al flujo normal		
Extensión	Paso extendido	Acción adicional
<i>Equipo</i> con telefonía VoIP	1	El operario conecta un teléfono analógico al <i>equipo</i>
<i>Equipo</i> con telefonía VoIP	3	El <i>ACS</i> hace sonar el teléfono analógico y dicta con voz sintetizada el número de teléfono asignado al <i>equipo</i> .
<i>Equipo</i> con telefonía VoIP	4	El operario registra en la caja del <i>equipo</i> el número de teléfono y la dirección IP asignada.

### 3.1.2.2. Caso de uso: Despachar un equipo

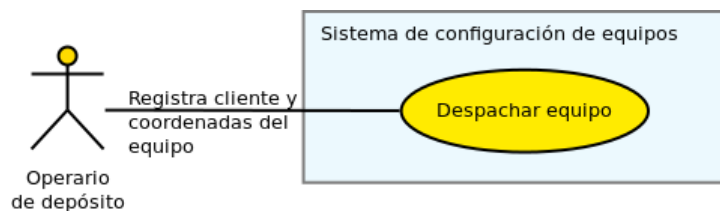


Figura 5: Diagrama de caso de uso "Despachar un equipo"

<b>Título:</b>	<b>Despachar un <i>equipo</i></b>
Terminología / Abreviaturas utilizadas:	<i>equipo</i> : equipo a ser despachado. Son equipos terminales que se instalan en el domicilio del usuario (CPEs). <i>BD</i> : base de datos de activos técnicos, dónde se registra información de los <i>equipos</i> configurados.
Objetivo:	Registrar en la <i>BD</i> los datos de utilización del <i>equipo</i> (cliente, dirección y coordenadas)
Alcance:	Sistema de configuración de equipos
Nivel:	Tarea secundaria
Pre-condiciones:	El <i>equipo</i> debe estar configurado
Situación cuando finaliza exitosamente:	Datos de utilización del <i>equipo</i> (cliente, dirección y coordenadas) registrados en <i>BD</i> .
Actores primarios:	Operario
Actores secundarios:	
Disparador:	Nuevo cliente o reemplazo de un <i>equipo</i> defectuoso.
Requerimientos de performance:	El caso se ejecutará en línea. La demora en ingreso al sistema, carga de datos y grabación no debería superar los 2 minutos.
Forma de comunicación con los actores:	Aplicación interactiva, por pantalla.

### 3.1.3. Características de los equipos a gestionar

#### 3.1.3.1. Nodos cliente - Mesh Potato

Se trata de equipos diseñados específicamente para brindar telefonía y datos mediante redes inalámbricas de tecnología IEEE 802.11 en zonas rurales. Son capaces de funcionar en redes inalámbricas tradicionales (punto-a-punto o punto-multipunto) o en redes mesh inalámbricas.

Fueron diseñados por la organización sudafricana Village Telco y su fabricación encargada a la empresa China, ATCOM. El diseño del hardware del Mesh Potato es abierto, siguiendo los ideales de los proyectos de software libre. Asimismo, el firmware de los Mesh Potato es software libre (Rowe D., 2009).

Combinan en el mismo equipo, una interface Wi-Fi, un puerto Ethernet y un adaptador para teléfono analógico (ATA por las siglas de su denominación en inglés *Analog Telephone Adapter*). La conjunción de todos estos elementos en un mismo equipo permite hacer más sencilla la instalación, reducir el consumo de energía eléctrica, reducir el costo de equipos, reducir la cantidad de cables y conexiones y reducir los posibles puntos de falla.

Se trata de equipos de construcción robusta, diseñados para poder ser desplegados en el exterior y fijados a cualquier tipo de estructura. Tienen antena incorporada, pero también disponen de un conector para utilizar una antena externa.

Pueden ser alimentados de energía eléctrica mediante la fuente de alimentación provista, o también por la conexión Ethernet (PoE por las siglas de su denominación en inglés: *Power over Ethernet*) o por la conexión telefónica (PoTL por las siglas de su denominación en inglés: *Power over Telephone Line*).



Figura 6: Vistas de equipo Mesh Potato

El firmware está compuesto por un sistema operativo OpenWRT, que es un sistema operativo basado en el kernel Linux desarrollada específicamente para ser usada como firmware, especialmente en routers.

Incluye el módulo `batman-adv`, que proporciona una interface de red virtual que permite encaminar tramas en redes mesh.

También incluye el software Asterisk, que permite:

- realizar llamadas telefónicas entre equipos pares dispuestos en la misma red;
- realizar llamadas telefónicas a la red pública accediendo a gateways mediante protocolo SIP (*Session Initiation Protocol*);
- implementar menús telefónicos mediante IVR (*Interactive Voice Response*) para configuración del equipo.

Los equipos Mesh Potato pueden funcionar como nodos router componentes de una red mesh inalámbrica, o como nodos clientes de una red Wi-Fi. En el primer caso pueden

comportarse como nodos gateway, conectándoles a su puerto Ethernet una conexión a otras redes (ej. Internet). En ambos casos pueden dar servicio de acceso a la red a otros equipos a través de su puerto Ethernet (Guillet , T.).



Tabla I: Especificaciones técnicas de los equipos Mesh Potato. Fuente: (Mesh Potato)

Plataforma de hardware:	Procesador:	MIPS 4k 180 Mhz
	Red inalámbrica:	Atheros AR2317 <i>System On a Chip</i>
	Puerto FXS:	Silicon labs
	Puertos LAN:	Uno, 10/100Mbit
	Memoria RAM:	16 MiB
	ROM:	EEPROM Flash Serial de 8MiB
Red inalámbrica:	Normas:	Normas: IEEE 802.11b/g
	Frecuencia de operación:	2.4 a 2.462GHz
	Tipo de antena:	Omnidireccional, interna, en circuito impreso
	Potencia de transmisión (EIRP):	1-24 Mbit - 20dBm 36-54 Mbit - 17dBm
Interfaces y Puertos:	LAN:	Uno, RJ-45
	FXS:	Uno, RJ-11
Firmware:	<ul style="list-style-type: none"> <li>• Sistema operativo <i>OpenWRT Kamikaze</i> (modificado)</li> <li>• Kernel Linux 2.26.3</li> <li>• Módulo de ruteo B.A.T.M.A.N. versión 0.3</li> <li>• <i>Asterisk</i> 1.4.11</li> </ul>	
Condiciones ambientales de operación	Humedad relativa ambiente:	hasta 95%
	Temperatura:	Entre -20° y +70° C.
Características eléctricas	Tipo de alimentación:	Corriente continua, voltaje variable
	Formas de alimentación:	Con adaptador de CA provisto <i>Power over Ethernet</i> (PoE) <i>Power over Telephone Line</i> (PoTL)
	Tensión de alimentación:	Entre 9 y 40 V de CC
	Potencia nominal:	2.5W
	Protecciones:	Sobre-voltaje, inversión de polaridad en alimentación de CC, descargas atmosféricas, electricidad estática
	Puertos protegidos:	Entrada de CC, RJ-11 y RJ-45
Características físicas	Gabinete:	Para exteriores, con protección UV, color beige.
	LEDs indicadores de estado:	Encendido, enlace en LAN, tráfico en LAN, Wi-Fi activa, etc.
	Ancho:	106 mm
	Espesor:	55 mm
	Altura:	228 mm
	Peso:	0,3 Kg

### 3.1.3.2. Equipos Ubiquiti Nanostation M5

Los equipos Ubiquiti Nanostation M5 son equipos terminales de cliente (CPE) inalámbricos para transmisión de datos que operan en la banda de 5 GHz.

Pueden ser utilizados para cumplir las siguientes funciones:

- *Access Point* (AP) para crear redes locales inalámbricas (WLAN) de gran alcance;
- repetidor de *Access Point*, para extender el alcance de una WLAN;
- *station*, para servir como nodo cliente en una WLAN;
- nodo (cualquier extremo), en un enlace punto-a-punto;
- nodo base o central en un enlace punto-multi-punto;
- nodo remoto o esclavo en un enlace punto-multi-punto;

Son equipos robustos, preparados para ser utilizados en el exterior, e incorporan una antena en frente del equipo.

Se alimentan mediante PoE (*Power over Ethernet*).

Incorporan el sistema operativo del fabricante llamado *airOS*, y su configuración se realiza utilizando interfaces gráficas provistas por el S.O.

No implementan el protocolo TR-069 y el SNMP solo en modo lectura.



Figura 7: Equipos Ubiquiti Nanostation M5

### 3.1.3.3. Módems ADSL Huawei EchoLife HG520s

Son módems ADSL que incorporan funcionalidad de router Wi-Fi.

Implementan los protocolos TR-069/CWMP y SNMP.



Figura 8: Módem ADSL Huawei EchoLife HG520s

### 3.1.4. Configuraciones a realizar en equipos Mesh Potato

Las configuraciones que se realizan en los equipos Mesh Potato para librarlos al servicio son:

- cambiar la contraseña de fábrica del usuario root;
- habilitar SSH en lugar de Telnet para su configuración;
- configurar la interface inalámbrica del equipo:
  - modo;
  - canal;
  - SSID y BSSID de la red mesh de la Cooperativa;
  - contraseña de la red inalámbrica;
  - potencia de transmisión;
  - etc;
- configurar los parámetros de protocolo de Internet del equipo: ej. dirección IP, máscara de sub-red, servidores DNS a utilizar, etc;
- configurar la interface Ethernet del equipo (modo *bridge*);
- configurar el cliente de telefonía VoIP para acceder a la central telefónica de la cooperativa por protocolo SIP;
- incorporar los datos del equipo al sistema de facturación.

La fase de diseño comenzó por realizar un análisis de los protocolos y las herramientas existentes para realizar configuración de equipos de red, con el fin de determinar la posibilidad de utilizar software y/o protocolos estándar.

Debido que los equipos Mesh Potato no implementan ningún estándar de

configuración, se evaluó la factibilidad y conveniencia de dotar a estos equipos con los agentes necesarios para conectarse a servicios estándar, pero esta opción se descartó porque implica reemplazar y mantener el firmware de los equipos.

Además, el uso de un protocolo estándar para configurar los equipos Mesh Potato no asegura que la solución resulte capaz de configurar cualquier tipo de equipo. Esto se debe a que no existe un protocolo común que sea implementado por todos los equipos analizados. El equipo principal, Mesh Potato, no implementa ninguno, y de los otros dos, uno implementa ambos protocolos y el restante no implementa ninguno. Además, los modelos de datos estándar de los protocolos existentes no siempre son suficientes para modelar cualquier equipo, en general requieren un cierto grado de adaptación para cada modelo.

Por estos motivos se decidió utilizar un enfoque *ad-hoc* para la solución, pero aprovechando lo observado en el análisis de los protocolos existentes, especialmente sus aspectos negativos, con el fin de evitar repetirlos, y también contemplando que en versiones futuras el producto pueda implementar alguno de estos protocolos, en caso que los equipos a configurar los implementen.

El proceso de diseño comenzó con evaluar las herramientas necesarias para poder realizar la configuración de los equipos Mesh Potato y los demás equipos que eventualmente será necesario manejar en versiones futuras del producto. Luego se avanzó en determinar la forma de integrarlas en una arquitectura que resultara óptima para satisfacer todos los requerimientos del proyecto.

Esta fase del desarrollo fue altamente iterativa, comenzando con un diseño que lograba solucionar todos los aspectos requeridos, que luego fue evolucionando en varias iteraciones para simplificarse y requerir una menor cantidad de pasos, involucrar una menor cantidad de componentes y tecnologías, y a la vez proporcionar puntos de extensión que permitan darle a la solución la flexibilidad necesaria para adaptarse para el uso en diferentes contextos y para configurar diferentes tipos de equipos.

### **3.1.5. Componentes principales**

En su versión final, el diseño de alto nivel involucra los siguientes componentes principales:

- el equipo a configurar;
- un equipo “adaptador”, que permite conectar cualquier tipo de equipo a

configurar con el servidor de configuración;

- un servidor de configuración. Este servidor es el encargado de determinar las características del equipo conectado para ser configurado, registrarlo en la base de datos de equipos configurados, y aplicarle la configuración correspondiente;
- una clase por cada tipo de equipo manejado, que implementa la lógica de reconocimiento del equipo conectado y también la lógica propia de su configuración.

Para permitir que el mismo sistema tenga la capacidad de tratar diferentes tipos de equipos, el servicio se diseñó siguiendo un patrón *strategy*, que permite que la lógica particular de la configuración de cada tipo de equipo sea implementada en la clase específica del tipo de equipo.

Además, como parte de la solución se incluye una aplicación que sirve para mantener la base de datos de equipos configurados, el stock de direcciones IP y números telefónicos disponibles para asignar, y para realizar consultas sobre los equipos configurados.

### 3.1.6. Secuencia de funcionamiento

Antes de realizar la configuración Personal Técnico define números de Teléfono y direcciones disponibles.

El proceso de configuración comienza con la conexión física del equipo a configurar al equipo adaptador. El servicio de monitoreo que corre en el adaptador, detecta la existencia de un equipo conectado y como resultado dispara el servicio de configuración en el servidor de configuración.

El servicio de configuración comienza por determinar el tipo de equipo conectado, ejecutando para eso el método de reconocimiento de cada una de las clases definidas, y luego ejecuta el método de configuración de la clase correspondiente al tipo de equipo detectado.

El método de configuración aplica en el equipo a configurar todos los ajustes necesarios, registra en la base de datos la información correspondiente al equipo configurado y los números asignados, y finalmente devuelve el control al servicio de configuración, que notifica al cliente la finalización del proceso mediante el envío de un correo electrónico.

En el caso particular de la configuración de equipos Mesh Potato, además de conectarse al equipo adaptador, puede conectarse también a un teléfono analógico. Luego, el método de configuración, como paso final realiza una llamada al teléfono analógico, dictando

al usuario, con voz sintetizada, el número de teléfono asignado al equipo, lo que evita la necesidad de que el usuario necesite disponer de acceso al correo electrónico.

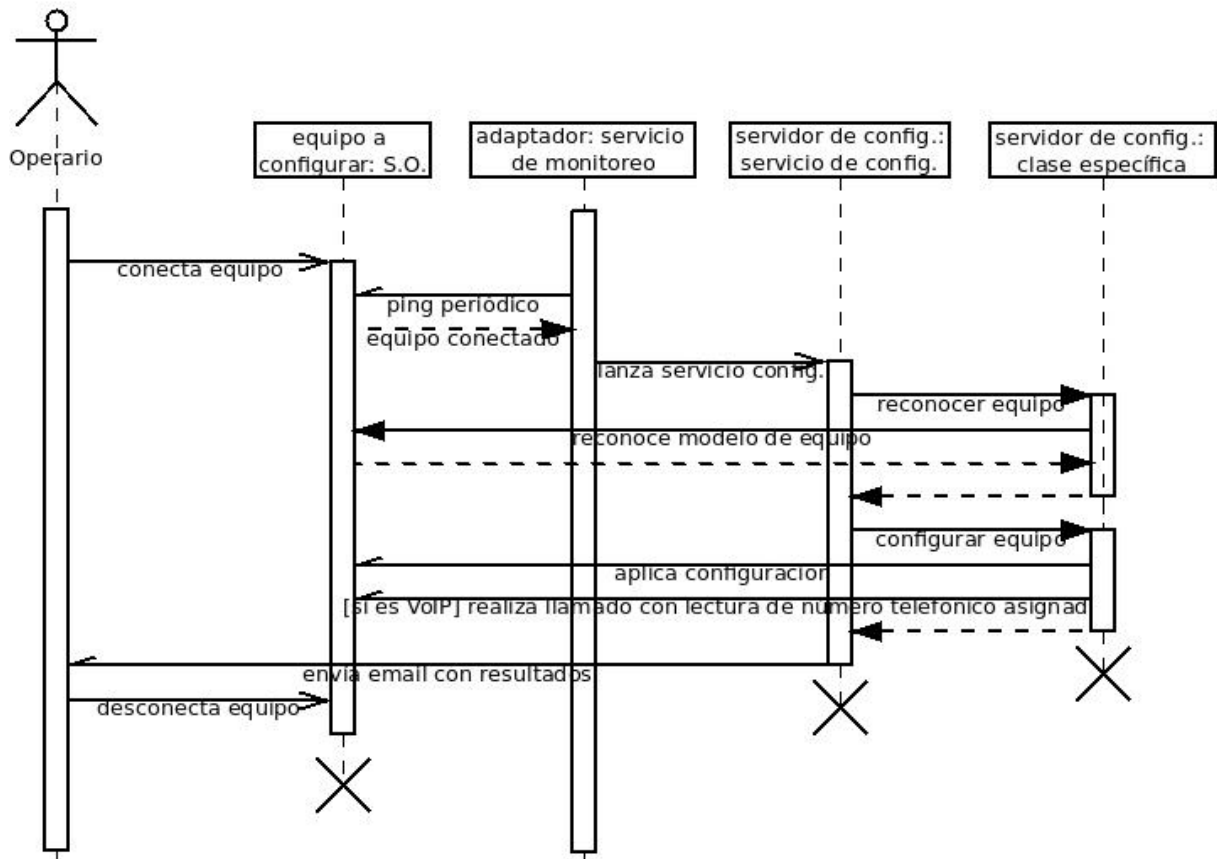


Figura 9: Diagrama de secuencia de operaciones del proceso de configuración automática de equipos.

### 3.1.7. Implementación

Para facilitar el soporte y mantenimiento del sistema se decidió darle una arquitectura distribuida, con un servidor de configuración centralizado, y segmentos adaptadores que se despliegan en las instalaciones del cliente.

A pesar de tratarse de un sistema distribuido, se procuró dotarlo de ciertas características propias de los denominados *computer appliance*, para facilitar su despliegue y puesta en servicio en las instalaciones del cliente, y a la vez reducir las necesidades de soporte y mantenimiento del sistema (Smith B. et al. 2007).

Para la implementación del servidor de configuración centralizado, se decidió utilizar un servidor privado virtual (o VPS, por las siglas de su denominación en inglés: *Virtual Private Server*) alojado en la nube.

Para servir como “adaptadores” en las instalaciones del cliente, se decidió utilizar un

router programable. Inicialmente se consideró emplear para esta función un equipo tipo PC, pero luego se pudo determinar que un router programable podía cumplir la misma función, a una fracción del costo, y a la vez lograr apegarse más al concepto de *appliance*.

Además el modelo de router elegido permite simplificar la operación, puesto que tiene la capacidad de brindar alimentación eléctrica al equipo a configurar, reduciéndose de esta forma la cantidad de conexiones a realizar.

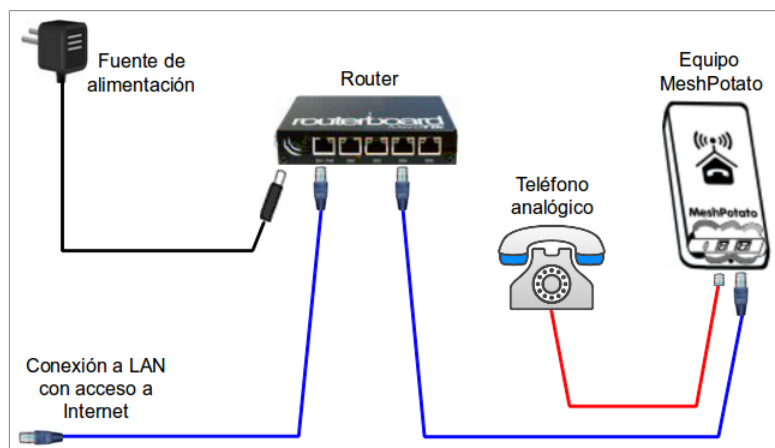


Figura 10: Esquema de conectividad física del router y equipo Mesh Potato a configurar

El servicio de configuración se implementó como un servicio web, lo que permite limitar el desarrollo a la lógica propia de la aplicación, quedando las cuestiones técnicas propias de la implementación del servicio a cargo de la infraestructura provista por el servidor web.

Para facilitar la comunicación entre los diferentes componentes que se encuentran distribuidos en diferentes redes, se decidió implementar una VPN, que permite disponer de una conexión bidireccional entre cualquiera de los nodos, como si se encontraran conectados a una misma red local.

Además, el uso de la VPN hace que todas las comunicaciones sean seguras, evita exponer funcionalidad del servidor de configuración a Internet, y facilita la salida de los equipos desplegados en la intranet del cliente hacia Internet resolviendo los problemas que se presentan al tener que pasar a través de *proxies*, *firewalls* y NATs.

El servicio de VPN se implementa en el servidor de configuración, y los routers se despliegan con el cliente de VPN configurado de forma que se conecten automáticamente a la VPN en cuanto tengan acceso a Internet, usando para autenticación un certificado digital

generado específicamente para el router. Esto permite que, en caso que el certificado digital del router resulte expuesto, sólo es necesario revocar el acceso a la VPN para ese certificado, lo que incrementa el nivel de seguridad del conjunto, y facilita el mantenimiento del sistema.

Por último, la base de datos de equipos instalados se alojó en el mismo servidor de configuración, junto con la aplicación de mantenimiento y consulta de la base de datos, que se implementó como una aplicación web.

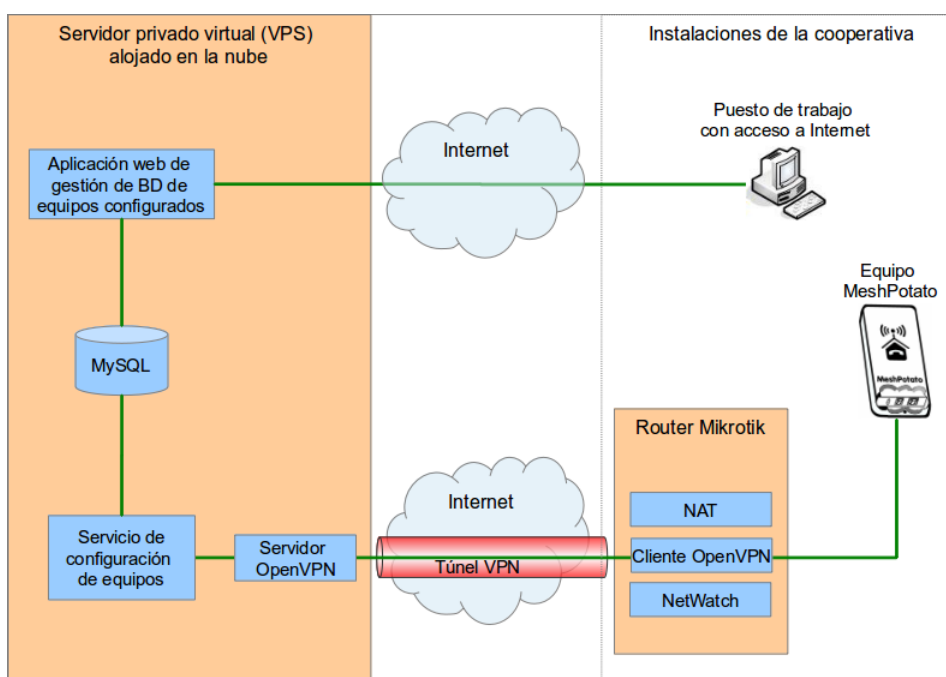


Figura 11: Esquema de conectividad de red entre los diferentes módulos de la solución.

### 3.1.8. Selección del equipamiento a emplear

#### 3.1.8.1. Router Ethernet programable

Para implementar la solución diseñada se necesitó utilizar un router ethernet que reuniera las siguientes características:

- ser de bajo costo;
- ser liviano, pequeño, robusto y de fácil instalación, para poder distribuirlo enviándolo por correo al cliente, que una vez en las instalaciones del cliente no ocupe demasiado espacio de trabajo, pueda ser conectado fácilmente, y que soporte años de utilización;



- ser extensible para poder implementar la funcionalidad deseada;
- que el firmware fuera de código abierto para poder hacerle cambios en caso de ser necesario, o que pudiera ser reemplazado por un sistema operativo *Open Source* para routers como OpenWrt y otros;
- que pudiera adquirirse fácilmente en el mercado local.

Al realizar este análisis se encontró el router **MikroTik RouterBOARD RB750UP** que cumple con todos estos requisitos y además dispone de una amplia base instalada y muchos recursos de soporte comunitarios.

Prueba de la versatilidad y robustez de los equipos de esta marca es que son equipos empleados en enlaces de bajo costo para llevar Internet a zonas remotas de África (Longobardi et al., 2009) (Bartalesi et al, 2009).

También tienen la ventaja de que el servidor DHCP que implementan maneja la funcionalidad requerida por el protocolo CWMP para el acceso inicial de los equipos CPE al servidor ACS, lo que facilita implementar el protocolo CWMP en versiones futuras del producto. Esto consiste en poder entregar los parámetros para acceder al ACS (ej. URL, nombre de usuario y contraseña) en el dato correspondiente al código 43 – *vendor specific information* correspondiente a la cada clase de equipo informada por el dispositivo en el valor del código 60 – *vendor class identifier*.

Otra característica secundaria pero de mucha utilidad de este modelo es que puede ser alimentado mediante PoE (*Power over Ethernet*) pasivo, y además puede alimentar a los equipos que se conecten a él (admite hasta 500mA en cada puerto ethernet, con el nivel de tensión que se le conecte en la entrada de CC, que puede ser de hasta 28VCC), siendo ambas funcionalidades muy útiles para simplificar la instalación del router o la conexión de equipos a configurar al router.

El software de los routers MikroTik es el MikroTik RouterOS, un sistema operativo basado en el kernel Linux, que puede ser usado tanto en el hardware propietario MikroTik como en computadores de arquitectura x86. Este sistema operativo incluye utilidades, tales como firewall, cliente y servidor de VPN, diferentes protocolos de encaminamiento de paquetes, gestión del ancho de banda y calidad de servicio, funcionalidad de punto de acceso inalámbrico (*wireless access point*), web proxy, etc. También incluye herramientas de red tales como Ping, Traceroute, Telnet, SSH, etc. (<http://www.mikrotik-routeros.net/>, 2014)

Algunas de estas utilidades y herramientas fueron utilizadas al implementar la

funcionalidad requerida por este proyecto.



Figura 12: Router MikroTik RouterBOARD RB750UP

Además, el hardware de los equipos MikroTik tiene la capacidad de ser extendido mediante puertos de expansión Mini PCI y puertos USB. De esta forma se puede dotar a los routers MikroTik con interfaces inalámbricas IEEE 802.11, interfaces 3G y 4G (LTE), tarjetas de memoria flash, conectarle impresoras, etc.

### 3.1.9. Selección de frameworks y lenguajes de programación

#### 3.1.9.1. Para la aplicación Web de registro de equipos configurados

Para el desarrollo de la aplicación de registro de equipos se buscó utilizar un framework de desarrollo web que cumpliera con las siguientes premisas:

- que incluyera infraestructura básica común a cualquier aplicación, como control de acceso de usuarios, gestión de permisos y manejo de sesión;
- que requiriera integrar una mínima cantidad de herramientas para realizar desarrollo y para el despliegue de la aplicación;
- que permitiera el desarrollo rápido de aplicaciones orientadas a la gestión de datos, en especial disponiendo de:
  - facilidades para la generación automática de funciones de alta, baja y modificación (o CRUD por las siglas de su denominación en inglés: *create, read, update and delete*);
  - soporte para diseño Model-View-Controller;
  - capa de abstracción de acceso a base de datos;
- que permitiera implementar y consumir servicios web con arquitectura de estilo REST (RESTful APIs), en especial para permitir la integración de esta aplicación con los sistemas de control de stock y facturación (*billing*) existentes en las

compañías telefónicas;

- ser software libre, de uso gratuito y código abierto;
- que empleara Python como lenguaje de desarrollo, por múltiples motivos:
  - para aprovechar la amplia variedad de módulos disponibles que brindan acceso (*binding*) a librerías avanzadas escritas en otros lenguajes como C y C++;
  - para aprovechar las facilidades del lenguaje Python, como su limpieza sintáctica, su fuerte orientación a objetos, y la facilidad de trabajar con un lenguaje tipado dinámicamente (DALY L., 2007);
  - para aprovechar la familiaridad de los autores con el lenguaje.

Como resultado de esta búsqueda surgió el framework `web2py`, un framework para desarrollo rápido de aplicaciones web orientadas a base de datos (BERTOLINA M., 2012), que reúne todas estas características.

Además `web2py` tiene la ventaja de ser uno de los frameworks soportados por la plataforma de servicios de computación en la nube de Google (*Google App Engine*), lo que eventualmente permitiría desplegar la aplicación como un servicio dentro de esta infraestructura.

### 3.1.9.2. Para el desarrollo de los servicios de configuración de equipos

Para el reconocimiento y la configuración de los equipos Mesh Potato se usan scripts desarrollados en lenguaje Python y Bash.

En ellos se hace uso de diferentes herramientas, entre las que se encuentran:

- `scp` - *Secure Copy*: utilizado para transferir archivos de configuración desde y hacia el Mesh Potato;
- `wget` - *World Wide Web Get*: utilizado para ejecutar servicios web en el servidor de configuración;
- `sed` - *Stream editor*: utilizado para completar archivos de configuración con la información específica de un nuevo equipo a configurar.

### 3.1.10. Selección de motor de base de datos

Para alojar los datos de equipos configurados se necesitó utilizar un servidor de base de datos. Para esto hubo que seleccionar un motor de base de datos. Las alternativas evaluadas y las características correspondientes a cada una fueron las siguientes:

- MySQL: de entre los RDBMS evaluados, es el que tiene la mayor base instalada, y la mayor facilidad de configuración y despliegue. Además reúne características de performance y seguridad adecuadas para las necesidades del proyecto (SCHWARTZ B. et al., 2012);
- PostgreSQL: de entre los RDBMS evaluados, es el más completo. Ofrece un muy buen nivel de seguridad y performance, y además tiene las mejores facilidades de extensión y lenguajes para programación de procedimientos almacenados. Por otro lado, su instalación y mantenimiento es más compleja que la de MySQL.
- SQLite: no es un RDBMS, sino una librería de gestión de base de datos. Sus capacidades y performance son adecuadas para el uso que se necesita. Además su despliegue y mantenimiento es muy sencillo por estar toda la base de datos contenida en un sólo archivo. Por otro lado tiene limitaciones importantes, entre ellas el no soportar cambios concurrentes y no ser accesible a través de la red.

Como resultado de este análisis, se optó por utilizar MySQL. SQLite se descartó por las limitaciones que presenta, mientras que MySQL se eligió por sobre PostgreSQL por presentar mayor facilidad de despliegue y mantenimiento, y porque las potenciales ventajas de PostgreSQL no son necesarias para este proyecto.

A la vez, el uso de MySQL permitió unificar tecnologías, porque es el RDBMS empleado por la herramienta de monitoreo de red mesh utilizada.

### 3.1.11. Selección del software de VPN a utilizar

Como herramienta para crear la red privada virtual (o VPN por las siglas de su denominación en inglés: *Virtual Private Network*), se decidió utilizar OpenVPN.

OpenVPN es una aplicación gratuita y de código abierto (*Open Source*), publicada mediante licencia GPL (*GNU General Public License*), que permite crear, mediante técnicas de VPN, túneles encriptados entre computadoras, los que pueden ser de tipo punto-a-punto o punto-multipunto.

OpenVPN tiene muchas ventajas en relación a otros productos similares o a otras técnicas para implementar VPNs, siendo las siguientes las más relevantes para el uso que se le da en el presente trabajo:

- Los túneles que realiza OpenVPN pueden pasar a través de casi cualquier *firewall*. Mientras se disponga de acceso a Internet, y se pueda acceder a sitios web que

usen protocolo HTTPS, OpenVPN deberá funcionar;

- Los túneles que realiza OpenVPN pueden pasar a través de traductores de direcciones de red (NATs por las siglas de su denominación en inglés: *Network Address Translator*), lo que permite que tanto el cliente como el servidor puedan funcionar desde intranets, con direcciones IP privadas;
- Tiene la capacidad de utilizar proxies. Puede ser configurado para utilizar protocolos TCP o UDP, y para ejecutarse como un servidor o como un cliente. Cuando funciona como servidor simplemente espera hasta que un cliente solicita establecer una conexión, mientras que cuando funciona como cliente trata de establecer una conexión de acuerdo con su configuración;
- El servidor puede ser configurado para que “empuje” algunas opciones de configuración de red a los clientes que se conectan, incluyendo direcciones IP, rutas y otras opciones de conexión;
- Alta flexibilidad con muchas posibilidades de implementar *scripts* que pueden ser usados para diferentes propósitos;
- No requiere el uso de direcciones IP fijas en ninguno de los extremos del túnel.

(Comañici, 2009)

### 3.2. Desarrollo de la aplicación y pruebas unitarias

El desarrollo del trabajo se caracterizó por la necesidad de idear mecanismos que permitieran simplificar y reducir la infraestructura necesaria para realizar la configuración de equipos Mesh Potato, mantener bajo el acoplamiento de la solución con las características de estos equipos, de manera que pudiera ser utilizada también para la configuración de equipos de otros tipos, y mantener bajo el número de posibles puntos de fallo en la solución.

Otro aspecto que presentó desafíos fue el reducir las necesidades de intervención del usuario en el proceso de configuración de los equipos.

Algunos aspectos que se fueron simplificando a lo largo de las iteraciones realizadas son los siguientes:

- se evitó la necesidad de usar correo electrónico para notificar al usuario de la finalización exitosa de la configuración de equipos Mesh Potato y notificar el número telefónico asignado al equipo. En su reemplazo se implementó el llamado al teléfono analógico y dictado por voz sintetizada del número

telefónico asignado. A la vez este mecanismo proporciona un punto más de verificación del correcto funcionamiento del equipo;

- se evitó la necesidad de emplear un equipo de tipo PC como adaptador para la conexión de equipos a configurar, reemplazándose por un router que tiene un costo muy inferior, y a la vez proporciona una solución más efectiva, por tratarse de un equipo más robusto, completamente de estado sólido y con menor cantidad de puntos de falla, que ocupa menos espacio, causa menos distracciones, y que puede ser enviado por correo al lugar de uso;
- se redujo el número de conexiones a realizar al configurar equipos, mediante la alimentación de los Mesh Potato vía PoE;
- se simplificó el servicio de configuración de equipos que corre en el servidor de aplicaciones, por dos causas:
  - porque al incorporar el router se pudo usar la utilidad `netwatch` para detectar la conexión de nuevos equipos, evitándose la necesidad de desarrollar nuevamente esa lógica;
  - porque se implementó como un servicio web, de forma que se evita desarrollar toda la lógica para que la aplicación se comporte como servicio, puesto que se utiliza la proporcionada por el servidor web que corre en el servidor de configuración.

Las pruebas de funcionamiento del sistema se realizaron de manera tradicional, ejecutando los procesos de configuración y verificando los resultados volcados en el equipo Mesh Potato.

Para simular las condiciones reales de utilización (ej. elevada latencia por el uso de redes WAN, lo que aumenta el riesgo de que se presenten condiciones de carrera que impidan que los pasos del proceso de configuración se ejecuten en el orden en que fueron diseñados), los objetos *mock* creados para realizar el desarrollo dirigido por casos de prueba se desplegaron en el servidor de configuración para realizar las pruebas unitarias.

Por el mismo motivo, las pruebas de integración se realizaron reproduciendo completamente en el laboratorio, la infraestructura a emplear en las cooperativas telefónicas.

### 3.3. Implementación del servicio en la nube

Para la implementación del servicio en la nube se necesitó contar con un servicio que

satisficiera las siguientes necesidades (en orden de importancia descendiente):

- a) que permita el mayor nivel de acceso posible a los recursos del equipo, con la posibilidad de instalar cualquier sistema operativo y sobre éste, cualquier software;
- b) que el tipo de servicio sea ofrecido por otros proveedores, de manera que exista la posibilidad de mudar el servicio a otro sitio en caso de ser necesario;
- c) que el proveedor asegure una alta disponibilidad del servicio, tenga buena reputación, y perspectiva de soportar el servicio por un largo plazo;
- d) que el servicio sea de bajo costo, en especial el costo fijo mensual, puesto que es un servicio que deberá estar disponible por mucho tiempo.
- e) que la implementación y el mantenimiento del servicio requieran poco esfuerzo, tanto en aspectos técnicos como administrativos.

Otros aspectos como la capacidad del equipo y del acceso a Internet, que en general son relevantes al evaluar estos servicios, son secundarios para esta aplicación, puesto que los requerimientos son muy bajos y son cubiertos en exceso por los planes más básicos que se ofrecen.

La solución ideal para solucionar el aspecto a) es utilizar un servidor dedicado. Dentro de este campo hay dos alternativas, servidores físicos y servidores virtuales (o VPS por las siglas de su denominación en inglés: *Virtual Private Server*). Debido a que la capacidad de computación y memoria requeridos para esta aplicación son muy bajos en relación a las capacidades de cualquier equipo actual, la alternativa lógica es utilizar un servidor virtual. A la misma conclusión se llega fácilmente cuando se evalúa el costo de una y otra alternativa.

De entre los servicios VPS que se comercializan actualmente se identifican dos grandes opciones.

Los implementados mediante virtualización a nivel de sistema operativo, que son también conocidos como “contenerización”. Algunos ejemplos de este tipo de virtualización son OpenVZ y FreeBSD Jail.

La otra opción la conforman los implementados mediante virtualización a nivel de hardware, que disponen de un módulo llamado Virtual Machine Monitor (VMM) o Hypervisor sobre el que se crean y corren las máquinas virtuales. Algunos ejemplos de software que permiten implementar este tipo de virtualización son: Linux KVM, VMware Workstation, Xen, Microsoft Virtual PC y VirtualBox.

Los primeros están limitados a ejecutar el mismo sistema operativo que la máquina física, por lo tanto para satisfacer completamente el primer aspecto es necesario seleccionar un VPS implementado mediante virtualización a nivel de hardware.

Actualmente existen varios proveedores que ofrecen VPS basadas en virtualización de hardware, lo que satisface el aspecto b). Algunos de ellos son: DigitalOcean (<http://www.digitalocean.com>), RamNode (<http://www.ramnode.com/>), BuyVM (<http://buyvm.net/>).

Entre ellos algunos ofrecen acuerdos de nivel de servicio (o SLA las siglas de su denominación en inglés: *Service Level Agreement*) comprometiendo niveles de disponibilidad de 99.99% (aproximadamente 4,32 minutos de indisponibilidad por mes), aunque en los planes de menor costo la penalización por incumplimiento se limita a no cobrar el servicio por el tiempo que dura la indisponibilidad. Para satisfacer el aspecto c), en servicios de bajo costo, es de esperar que el cumplimiento del nivel de servicio sea impuesto más por la necesidad de mantener una buena reputación y retener clientes, dada la competencia existente, que por penalizaciones directas.

Con respecto al aspecto d), costo del servicio, se pueden contratar servicios de VPS de bajo costo, con precios desde 3,5 o 5 dólares estadounidenses por mes dependiendo del proveedor.

Finalmente, con respecto el aspecto e), la carga de trabajo que implica implementar y mantener el servicio, desde el punto de vista técnico, al tratarse de una máquina virtual, el esfuerzo dependerá de como se instale y configure esa máquina. Desde el punto de vista administrativo todos los proveedores ofrecen servicios similares, pagos con tarjeta de crédito, PayPal o servicios similares, facturación en línea, y la contratación del servicio se realiza en línea en cuestión de minutos.

Como resultado del análisis de todos los aspectos mencionados se seleccionó el servicio de DigitalOcean.

Utilizando las herramientas provistas por el proveedor se creó una máquina virtual con sistema operativo GNU/Linux de distribución Debian, sobre la que se instalaron las herramientas necesarias para correr la aplicación utilizando la herramienta de gestión de paquetes Apt (*Advanced Package Tool*) propia de esta distribución (Hertzog R. et al. 2013).

Para asegurar mínimos riesgos de seguridad y mantener baja la carga de trabajo de mantenimiento del servidor, se siguieron recomendaciones y buenas prácticas de



administración propias del sistema operativo utilizado. Por ejemplo, se tomaron las siguientes medidas:

- se instaló una versión estable y reciente del sistema operativo, que aún recibe actualizaciones de seguridad;
- se concentró el log de OpenVPN, en el `syslog` para aprovechar su configuración de auto-rotación;
- se configuró `mysql-log-rotate` para la auto-rotación de logs de MySQL;
- se le implementó un *firewall*, usando `ufw`, configurado para admitir sólo las conexiones entrantes a OpenVPN y al servidor web Apache;
- se instalaron solamente los servicios necesarios para soportar la aplicación desarrollada.

Queda pendiente la implementación de otros recursos necesarios para asegurar la alta disponibilidad en un sistema productivo, tales como la implementación de rutinas de backup, la descarga automática y aviso de actualizaciones críticas, etc.

A las medidas adoptadas para favorecer la seguridad y disponibilidad del servicio se suman las que implementa el proveedor de servicio de hosting, puesto que también tiene interés en evitar riesgos, dado que las fallas y los ataques (ej. de denegación de servicio) que pueda sufrir una VPS alojada, pueden repercutir indirectamente en el servicio que brindan a los restantes clientes que tienen VPSs alojadas en el mismo servidor físico o en el mismo *data center*.

#### 4. Metodología de Desarrollo

El desarrollo del proyecto se llevó adelante utilizando algunas prácticas propias de metodologías ágiles que resultaron adecuadas a las características del proyecto. Entre estas se encuentran:

- casos de uso: como guía para la captura y documentación de los requerimientos;
- desarrollo iterativo e incremental (IID por *Iterative Incremental Development*);
- *refactoring*.

A pesar de que el proyecto se desarrolló a distancia, aquellas prácticas que implican participación del usuario pudieron ser aplicadas.

El proyecto comenzó con un relevamiento tradicional, *in situ*, de las necesidades del cliente y luego entró en un proceso iterativo de diseño y validación empírica de las soluciones diseñadas mediante el uso de prototipos. En las siguientes iteraciones, las soluciones probadas en forma de prototipos se fueron incorporando al producto.

En cada iteración se procuró simplificar el diseño, siguiendo la premisa de implementar “*The simplest thing that could possibly work*” (BECK Kent, 2007). Este enfoque se utilizó tanto para la construcción del software como para el diseño de la interacción del usuario con el sistema.

Debido que se trata de una arquitectura distribuida, fue necesario implementar objetos simulados (*mocks*) y scripts de prueba unitaria para poder agilizar el proceso de desarrollo y prueba. Para simular la latencia existente en las condiciones reales de funcionamiento del sistema se aprovechó el servidor VPS en la nube, creando los objetos *mock* en este equipo.

## 5. Pruebas Realizadas

Más allá de las pruebas de laboratorio realizadas durante el desarrollo de la solución, se realizó una implementación experimental en la Cooperativa de Sarmiento, Chubut, que se describe a continuación.

No se necesitó realizar pruebas de stress/performance, porque el volumen de trabajo y de datos es muy bajo, y la demora del proceso de configuración automática es despreciable en comparación con la demora que implican los trabajos de manipulación y conexión de los equipos a configurar.

### 5.1. Resultados del despliegue experimental realizado

Para verificar el funcionamiento de la solución se solicitó la colaboración del personal de la Cooperativa de Sarmiento para realizar la configuración de algunos equipos. Para ello se envió a la Cooperativa un router MikroTik configurado, y se coordinó para que personal del depósito realizara los trabajos de configuración sobre equipos Mesh Potato que se encuentran a prueba en la Cooperativa.

Como resultado de esta prueba se detectó la conveniencia de implementar el llamado al equipo Mesh Potato una vez finalizada la configuración, para que sirva de aviso al usuario de que la configuración se encuentra completada, y al mismo tiempo para evitar la necesidad de que acceda al correo electrónico, puesto que no dispone de ese servicio en el puesto de

trabajo. Más allá de esto las pruebas resultaron satisfactorias, aún siendo realizadas por personal que no conoce los equipos Mesh Potato.

## 6. Conclusiones

El enfoque utilizado permitió que la solución evolucione desde su diseño inicial hasta la versión implementada, lográndose simplificar la operatoria y reducir los requerimientos de equipamiento y conectividad necesarios para realizar el aprovisionamiento de nuevos equipos.

La investigación de estándares existentes para usos comparables a los objetivos del trabajo permitió aprovechar algunos conceptos ya desarrollados, aunque no pudieron ser implementados porque los equipos a configurar no los implementan.

Las experiencias obtenidas con las pruebas realizadas permiten confirmar los beneficios que brindan los siguientes conceptos y herramientas que fueron empleadas en el desarrollo de la solución:

- el uso de un router programable como adaptador para conectar diferentes tipos de equipos de red a un sistema distribuido;
- el uso de servicios de VPS (Virtual Private Server) para desplegar aplicaciones de bajo costo en la nube;
- el software OpenVPN para integrar componentes de un sistema distribuido de manera simple y segura.

Se observa un potencial para aplicaciones de este tipo, en especial si el servicio de configuración es capaz de manejar una mayor diversidad de equipos a configurar, tales como:

- módems ADSL para compañías telefónicas;
- equipos inalámbricos para enlaces punto-a-punto o punto-multipunto (no mesh) para compañías telefónicas e ISPs;
- módems y equipos STB (*set-top box*) para empresas de televisión por cable/ISP;
- medidores inteligentes y nodos concentradores de tecnología mesh (para distribuidoras eléctricas).

Asimismo se identificaron múltiples aspectos en los que la solución puede evolucionar para brindar mejor información a los usuarios e integrar sistemas para evitar carga de datos redundante. Algunas de estas funcionalidades son:

- manejar un polígono geográfico que identifique las zonas que tienen cobertura asegurada de red mesh o de red inalámbrica punto-multipunto;

- manejar una zona exterior del polígono geográfico de zona de cobertura de redes inalámbricas para tomar requerimientos de conexión sujetos a verificación de factibilidad;
- incorporar facilidades para interrogar el estado de los diversos equipos desde una aplicación centralizada para ser utilizada por el primer nivel de soporte.

## 7. Bibliografía

AKYILDIZ Ian F. y XUDONG Wang. Wireless Mesh Networks. John Wiley & Sons, Ltd, 2009. ISBN 978-0-470-03256-5.

BARTALESI, Raphael et al. SaharaWi-Fi Project. Low-cost full wireless connections over long distances in desert environment, 2009.

BECK Kent. Extreme Programming Explained: Embrace change. First Edition, 1999. 137 p. ISBN 078-5342616415.

BERTOLINA, Martín. Netconf Element Management System – Design and Implementation. Tesis. Universidad Nacional de La Plata, Fac. de Informática, Julio de 2012.

BLACKFORD, John y DIGDON Mike. TR-069 CPE WAN Management Protocol. Issue: 1 Amendment 5. November 2013. The Broadband Forum.

Comănici Oana. ZCOV - A Zero-config Solution For OpenVPN. DIPLOMA THESIS. 2009, Bucharest.

Comisión Nacional de Comunicaciones, 2014 [en línea]. [consulta 22 sep. 2014]. <[http://www.cnc.gov.ar/ciudadanos/telefonía\\_fija/tarifas.asp](http://www.cnc.gov.ar/ciudadanos/telefonía_fija/tarifas.asp)>.

Cooperativa de provisión de servicios públicos de Sarmiento Ltda. (Coopsar), 2014 [en línea]. [consulta 1 sep. 2014]. <<http://www.coopsar.com.ar/>>.

DALY, Liza. Next-Generation Web Frameworks in Python. O'Reilly Media, Inc., 2007, 43 p. ISBN 978-0-596-51371-9.

Fariba Aalamifar, VIABILITY OF POWERLINE COMMUNICATION FOR SMART GRID REALIZATION, Kingston, Ontario, Canada, 2012.

FLICKENGER Rob et al. Wireless Networking in the Developing World. A practical guide to planning and building low-cost telecommunications infrastructure. Second Edition. Hacker Friendly LLC, 2007.

freeacs-ng [en línea]. [consulta 9 oct. 2014]. <<http://freeacs-ng.org/>>.

freecwmp [en línea]. [consulta 9 oct. 2014]. <<http://freecwmp.org/>>.

GUILLET, T. Village Telco - Small Enterprise/Campus Network - SECN-2.0 - User Guide. 43 p..

HERTZOG, Raphaël y MAS, Roland. The Debian Administrator's Handbook. 2013. 462 p. ISBN: 979-10-91414-03-6.

JOHNSON, David et al. Building a Rural Wireless Mesh Network: A do-it-yourself guide to planning and building a Freifunk based mesh network. Version 0.8. South Africa: Meraka Institute, 2007, 44 p.

Joshi, Avinash et al. IEEE 802.11-06/1778r1 - IEEE P802.11 - Wireless LANs - HWMP Specification. IEEE. 2006. 26 p..

KEMAL, Mohammed Seifu. Smart Grid Networks: Analysis of Timing Requirements for Data Aggregation and Control in Smart Grids. Masters Thesis. Aalborg University, Dinamarca, 2014.

LONGOBARDI, Federico et al. BOULSAT PROJECT. I° Congresso Nazionale CUCS sulla Cooperazione Universitaria, Pavia, 17-18 Giugno 2009.

MANGUES-BAFALLUY J. et al. Virtual Home Region Multi-hash Location Management Service (VIMLOC) for Large-Scale Wireless Mesh Networks. En FUNABIKI, Nobuo. Wireless Mesh Networks. InTech, 2011, p. 209-236. ISBN 978-953-307-519-8.

Mangues-Bafalluy Josep et al. VIMLOC location management in wireless meshes: Experimental performance evaluation and comparison. Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Barcelona – Spain, 2009.

MANOJ B.S. et al. WIRELESS MESH NETWORKS: ISSUES AND SOLUTIONS. En YAN ZHANG et al. WIRELESS MESH NETWORKING: Architectures, Protocols and Standards. 1a. ed. New York: Auerbach Publications, 2007, p. 3-48. ISBN 978-0-8493-7399-2.

MAURO, Douglas y SCHMIDT, Kevin. Essential SNMP. 2a ed. Sebastopol California: O'Reilly Media, 2005. 464 p. ISBN 978-0-596-00840-6.

Mesh Potato [en línea]. [consulta 28 sep. 2014]. <<http://villagetelco.org/mesh-potato/>>.

mikrotik-routeros.net, 2014 [en línea]. [consulta 3 oct. 2014]. <<http://www.mikrotik-routeros.net/routeros.aspx>>.

MOJTABA Seyedzadegan et al. Wireless Mesh Networks: WMN Overview, WMN Architecture. 2011 International Conference on Communication Engineering and Networks. IPCSIT vol.19 (2011) © (2011) IACSIT Press, Singapore.

MOTA E. et al. Pursuing Credibility in Performance Evaluation of VoIP Over Wireless Mesh Networks. En FUNABIKI, Nobuo. Wireless Mesh Networks. InTech, 2011, p. 185-208. ISBN 978-953-307-519-8.

PERKOV, Luka. ISP's black box: provisioning behind the scenes. [en línea]. Presentación. December 27, 2012. <[http://events.ccc.de/congress/2012/Fahrplan/attachments/2227\\_29c3\\_cwmp.pdf](http://events.ccc.de/congress/2012/Fahrplan/attachments/2227_29c3_cwmp.pdf)>. [consulta 3 oct. 2014].

Proyecto Ceibal [en línea]. [consulta 9 oct. 2014]. <<http://www.ceibal.edu.uy/UserFiles/P0001/Image/contenidos/pdfs/areatecnica/queeslaredmalamesh.pdf>>.

ROWE, David. Linux Journal [en línea]. 01 dic. 2009. [consulta 28 sep. 2014]. <<http://www.linuxjournal.com/magazine/mesh-potato>>.

SCHWARTZ Baron et al. High Performance MySQL. 3ra. ed. Sebastopol California: O'Reilly Media, Inc., 2012. 726 p. ISBN 978-1-449-31428-6.

Serval Project [en línea]. [consulta 11 sep. 2014] <<http://www.servalproject.org/>>.

SMITH, Bob et al. Linux Appliance Design. 1a. ed. San Francisco California: No Starch Press Inc., 2007. 356 p. ISBN 978-1593271404.

The Linux Kernel Organization [en línea]. [consulta 20 sep. 2014].  
<<http://wireless.kernel.org/en/developers/Documentation/ieee80211/802.11s>>.

WALLS, Jason. TR-069: A Brief Overview. [en línea]. Presentación. June 26, 2014.  
<[http://www.ieee1904.org/events/workshop\\_june2014/s3\\_walls\\_tr069.pdf](http://www.ieee1904.org/events/workshop_june2014/s3_walls_tr069.pdf)>. [consulta 3 oct. 2014].

YAN ZHANG et al. WIRELESS MESH NETWORKING: Architectures, Protocols and Standards. 1a. ed. New York: Auerbach Publications, 2007, 592 p. ISBN 978-0-8493-7399-2.



## 8. Referencia de abreviaturas utilizadas

ADSL: *Asymmetric Digital Subscriber Line*

AP: *Access Point*

B.A.T.M.A.N.: *better approach to mobile ad-hoc networking*

BSSID: *Basic Service Set Identifier*

CPE: *Customer Premise grounded Equipment* - Equipo terminal de usuario

CWMP: *CPE WAN Management Protocol*

IETF: *Internet Engineering Task Force*

IVR: *Interactive Voice Response*

NAT: *Network Address Translator*

PoE: *Power over Ethernet*

PoTL: *Power over Telephone Line*

RDBMS: *Relational Database Management System*

S.O.: Sistema Operativo

SIP: *Session Initiation Protocol*

SNMP: *Simple Network Management Protocol*

SSID: *Service Set Identification*

UPS: *Uninterruptible Power Supply* - SAI: Sistema de Alimentación Ininterrumpible

VPS: *Virtual Private Server*

WLAN: *Wireless Local Area Network* - Red LAN inalámbrica.

## **Anexo A – Descripción de las pruebas de red mesh realizadas**

A principios de 2014 se realizó en la Cooperativa de Sarmiento, Chubut, una implementación y ensayo de una red mesh inalámbrica en un barrio de la ciudad de reciente urbanización, que aún no cuenta con redes cableadas para telefonía y datos.

### **A.1. Configuración y equipos utilizados en la red mesh**

La red estuvo compuesta por 11 nodos de tipo “Mesh Potato” dispuestos en ubicaciones fijas cumpliendo la función de nodos router. La distancia de cada nodo router con el nodo router más cercano fue de entre 50 y 150 m. Algunos nodos router se colocaron en postes de distribución de energía eléctrica de baja y media tensión, a una altura no mayor a 7 m, y otros se colocaron en puntos altos en el exterior de viviendas de la zona.

Uno de los nodos router también cumplió función de *Gateway*, estando conectado con un access point Ubiquiti Nanostation M5 funcionando a una frecuencia de 5GHz, enlazado con otro equipo Ubiquiti Nanostation M5 montado en el mástil de la antena de la Cooperativa ubicado a aproximadamente 300 m de distancia.

Por la disposición realizada todos los nodos router se encontraban a una distancia de hasta dos saltos del nodo gateway. Por ende los nodos cliente estarían siempre conectados a no más de tres saltos del gateway.

Para evaluar el comportamiento de la red mesh implementada se realizó un monitoreo de los distintos componentes de la red bajo diferentes condiciones de funcionamiento.

En ese monitoreo se utilizaron las herramientas que se describen a continuación.

### **A.2. Monitoreo realizado**

#### **A.2.1. Herramientas de monitoreo empleadas**

##### **A.2.1.1. SPUD (*Simple Unified Dashboard for mesh networks*)**

SPUD es una aplicación de visualización gráfica de datos de monitoreo de redes inalámbricas implementadas con el protocolo de ruteo B.A.T.M.A.N. (*better approach to mobile ad-hoc networking*). Presenta los datos que los equipos que utilizan este protocolo pueden registrar en el BATMAN Visualization Server, un módulo apéndice del proyecto B.A.T.M.A.N. que sirve para concentrar información de monitoreo de la red, que es provista

proactivamente por los módulos `batman-adv` instalados en los nodos inalámbricos.

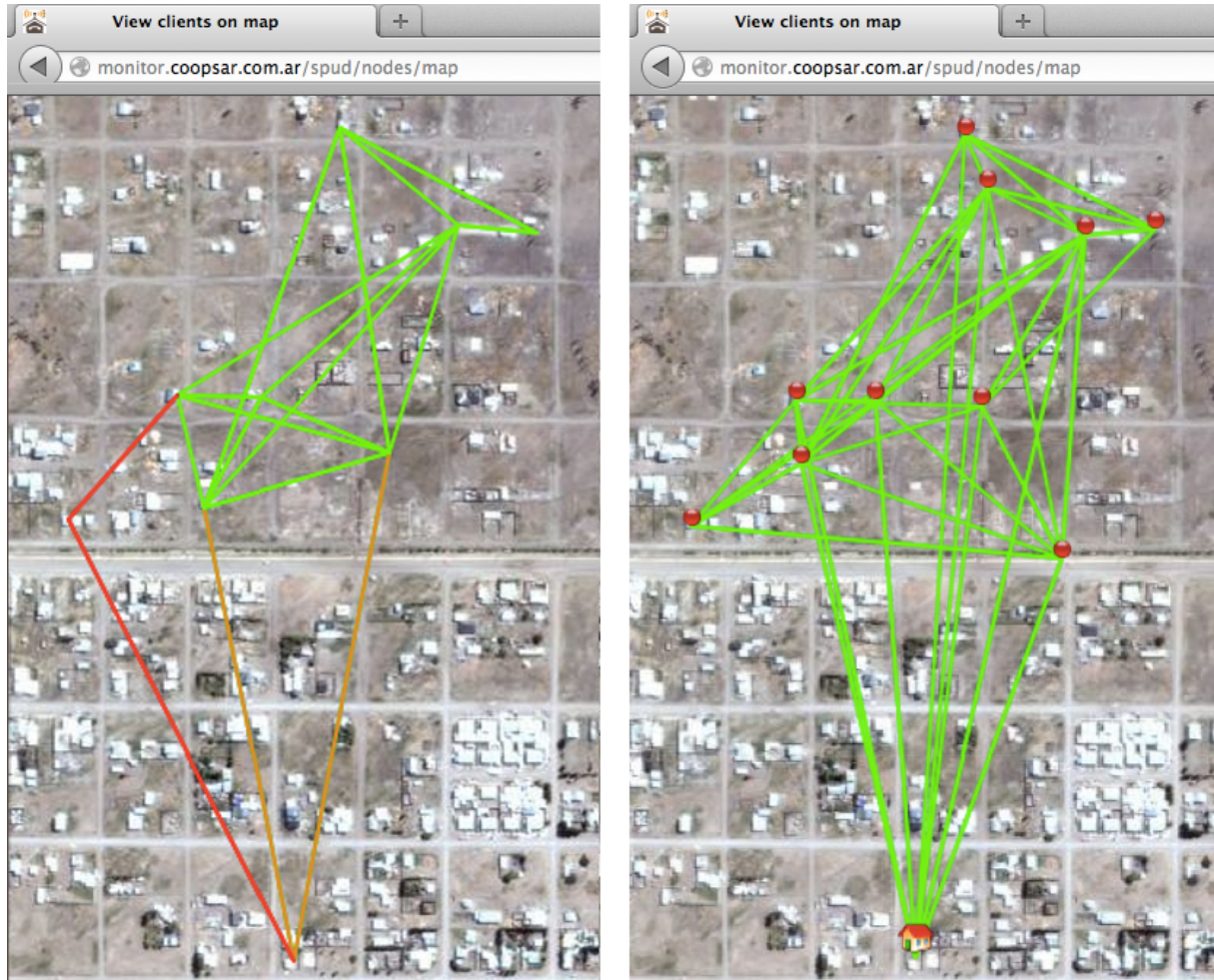
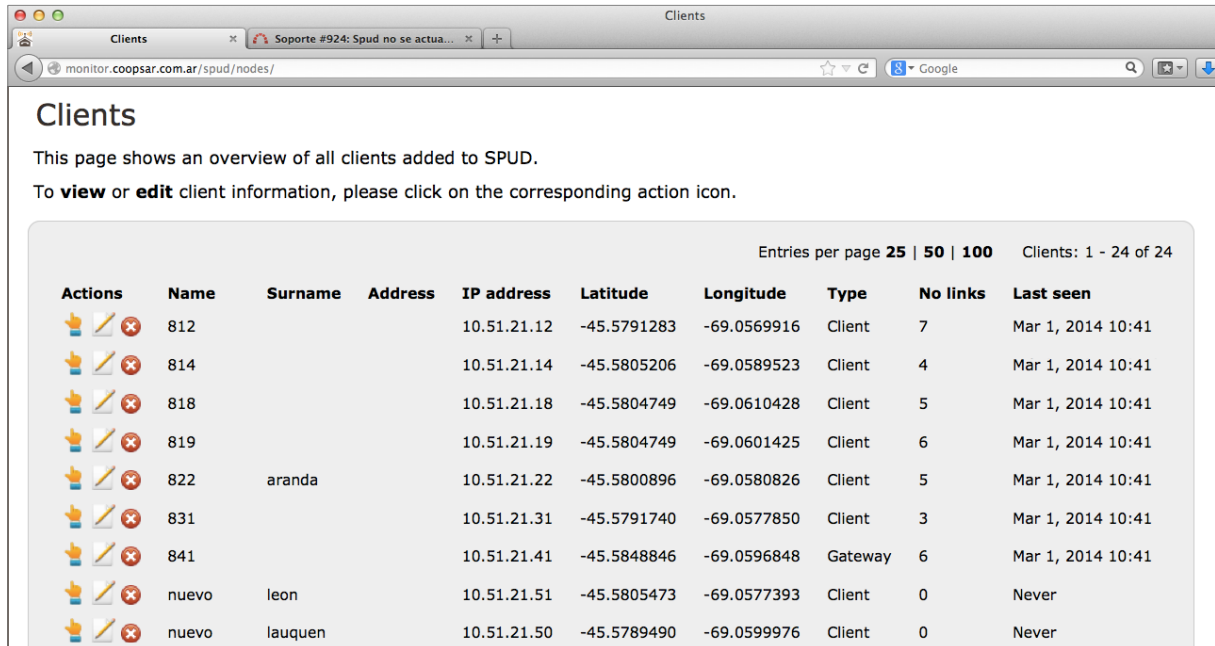


Figura 13: Captura de pantallas de SPUD tomadas en diferentes momentos. La de la izquierda se tomó con la disposición inicial de nodos, y la de la derecha se capturó luego de realizar varias mejoras en la disposición y configuración de los nodos de la red.

SPUD utiliza la API de Goggle Maps para presentar sobre el mapa los nodos y representar mediante líneas los enlaces inalámbricos que existen entre ellos, pudiendo ser configurado para presentar de diferente forma los enlaces dependiendo su calidad de señal.



The screenshot shows a web browser window with the URL `monitor.coopsar.com.ar/spud/nodes/`. The page title is "Clients". Below the title, there is a brief description: "This page shows an overview of all clients added to SPUD. To view or edit client information, please click on the corresponding action icon." The main content is a table with 10 columns: Actions, Name, Surname, Address, IP address, Latitude, Longitude, Type, No links, and Last seen. The table contains 10 rows of client data. At the top right of the table area, it says "Entries per page 25 | 50 | 100" and "Clients: 1 - 24 of 24".




























Actions	Name	Surname	Address	IP address	Latitude	Longitude	Type	No links	Last seen
  	812			10.51.21.12	-45.5791283	-69.0569916	Client	7	Mar 1, 2014 10:41
  	814			10.51.21.14	-45.5805206	-69.0589523	Client	4	Mar 1, 2014 10:41
  	818			10.51.21.18	-45.5804749	-69.0610428	Client	5	Mar 1, 2014 10:41
  	819			10.51.21.19	-45.5804749	-69.0601425	Client	6	Mar 1, 2014 10:41
  	822	aranda		10.51.21.22	-45.5800896	-69.0580826	Client	5	Mar 1, 2014 10:41
  	831			10.51.21.31	-45.5791740	-69.0577850	Client	3	Mar 1, 2014 10:41
  	841			10.51.21.41	-45.5848846	-69.0596848	Gateway	6	Mar 1, 2014 10:41
  	nuevo	leon		10.51.21.51	-45.5805473	-69.0577393	Client	0	Never
  	nuevo	lauquen		10.51.21.50	-45.5789490	-69.0599976	Client	0	Never

Figura 14: Vista de nodos monitoreados en SPUD.

```
mysql> show tables;
+-----+
| Tables_in_spud_db |
+-----+
| limits             |
| links              |
| missing_nodes     |
| nodes              |
+-----+

mysql> select id, ip_addr, name, surname, type, lat, long, last_seen from nodes;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | ip_addr      | name      | surname | type   | lat          | long          | last_seen |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 25 | 10.51.21.17 | 4896817   |         | client | -45.5809822 | -69.0609894 | 1392396267 |
| 26 | 10.51.21.18 | 4896818   |         | client | -45.5804749 | -69.0610428 | 1392396267 |
| 27 | 10.51.21.19 | 4896819   |         | client | -45.5804749 | -69.0601425 | 1392396267 |
| 24 | 10.51.21.13 | 4896813   |         | client | -45.5814705 | -69.0622177 | 1392396267 |
| 20 | 10.51.21.16 | 816       | 4896816 | client | -45.5817223 | -69.0580368 | 1392396267 |
| 22 | 10.51.21.21 | 821       |         | client | -45.5788040 | -69.0588760 | 1392396267 |
| 23 | 10.51.21.14 | 814       |         | client | -45.5805206 | -69.0589523 | 1392396267 |
| 16 | 10.51.21.31 | 831       | 4896811 | client | -45.5791740 | -69.0577850 | 1392396267 |
| 17 | 10.51.21.12 | 4896812   |         | client | -45.5791283 | -69.0569916 | 1392396267 |
| 18 | 10.51.21.41 | 841       | 4896810 | gateway | -45.5848846 | -69.0596848 | 1392396267 |
| 21 | 10.51.21.15 | 4896815   |         | client | -45.5783844 | -69.0591278 | 1392396267 |
+-----+-----+-----+-----+-----+-----+-----+-----+

mysql> select * from missing_nodes;
+-----+-----+-----+-----+-----+
| id | ip_addr      | created          | modified          |
+-----+-----+-----+-----+-----+
| 12 | 10.51.21.10 | 2014-01-23 18:58:16 | 2014-01-23 18:58:16 |
+-----+-----+-----+-----+-----+

mysql> select node_id, neighbour, label, label_reverse, created, neighbour_id from links;
+-----+-----+-----+-----+-----+-----+-----+
| node_id | neighbour      | label | label_reverse | created          | neighbour_id |
+-----+-----+-----+-----+-----+-----+-----+
| 16 | 10.51.21.14 | 1 | 1.058 | 2014-02-14 13:44:27 | 23 |
| 16 | 10.51.21.12 | 1.02 | 1.007 | 2014-02-14 13:44:27 | 17 |
| 16 | 10.51.21.17 | 1.036 | 1.015 | 2014-02-14 13:44:27 | 25 |
| 16 | 10.51.21.19 | 1.011 | 1 | 2014-02-14 13:44:27 | 27 |
...
| 20 | 10.51.21.19 | 1.024 | 1 | 2014-02-14 13:44:27 | 27 |
| 20 | 10.51.21.13 | 1.071 | 1.058 | 2014-02-14 13:44:27 | 24 |
| 22 | 10.51.21.31 | 1.045 | 1.011 | 2014-02-14 13:44:27 | 16 |
| 22 | 10.51.21.12 | 1 | 1.007 | 2014-02-14 13:44:27 | 17 |
| 22 | 10.51.21.17 | 1.028 | 1.015 | 2014-02-14 13:44:27 | 25 |
| 22 | 10.51.21.16 | 1.007 | 1.071 | 2014-02-14 13:44:27 | 20 |
...
| 24 | 10.51.21.17 | 1.032 | 1.015 | 2014-02-14 13:44:27 | 25 |
| 24 | 10.51.21.18 | 1.032 | 1.015 | 2014-02-14 13:44:27 | 26 |
| 24 | 10.51.21.16 | 1.071 | 1.071 | 2014-02-14 13:44:27 | 20 |
| 24 | 10.51.21.19 | 1.058 | 1 | 2014-02-14 13:44:27 | 27 |
+-----+-----+-----+-----+-----+-----+-----+

```

Figura 15: Ejemplo consultas a la base de datos de SPUD.

Además de las herramientas de consulta gráficas, SPUD mantiene un esquema con un diseño de alto nivel en una base de datos relacional, sobre la que el usuario puede realizar consultas más específicas o concretas cuando la información disponible en los gráficos no sea suficiente.

### A.2.1.2. SmokePing

SmokePing es una herramienta de software libre y código abierto que sirve para monitorear la latencia en redes IP.

Para efectuar la medición de latencia admite diferentes mecanismos que se pueden incorporar en forma de *plug-ins*. También dispone de una arquitectura *master/slave* para poder realizar mediciones en entornos distribuidos.

Tiene la capacidad de presentar la información obtenida en forma de gráficos.

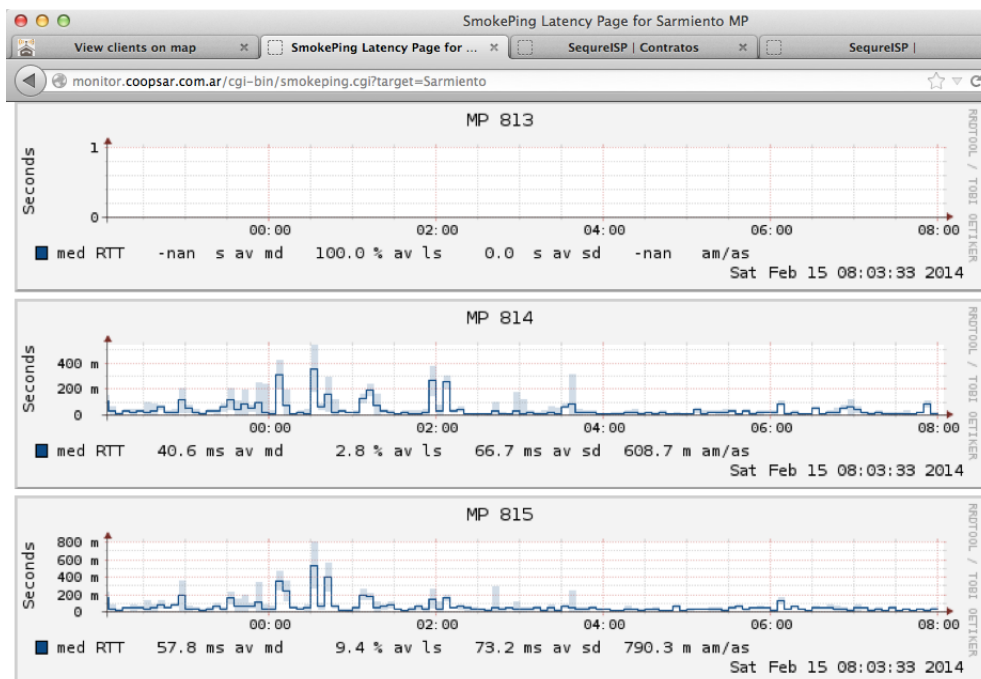


Figura 16: Vista de gráficos de SmokePing que muestra los valores de latencia medidos para los nodos monitoreados en los últimos 8 minutos.

### A.2.1.3. Cacti

Cacti es una herramienta de código abierto, desarrollada en tecnologías web que sirve para monitorear y representar gráficamente redes de datos. Es una aplicación que sirve como interfaz de usuario para RRDtool, una herramienta de colección de datos estándar de la industria.

Cacti permite al usuario consultar periódicamente los datos en intervalos predefinidos, y presentarlos en forma gráfica. Habitualmente se lo utiliza para mostrar la evolución en el tiempo de parámetros tales como nivel de utilización de CPU y utilización de ancho de banda



en redes de datos.

#### **A.2.1.4. Iperf**

Iperf es una herramienta prevista para realizar mediciones de performance de redes IP.

Permite inyectar tráfico de datos de protocolo TCP y UDP, y medir la capacidad de transmisión de la red que los transporta.

Está desarrollada en lenguaje C, y se distribuye como software libre, con licencia BSD. Está disponible para diferentes plataformas, incluyendo Linux, Unix y Windows (tanto en forma nativa como sobre Cygwin).

Cuando se usa para medir la capacidad de transmisión de datos de protocolo UDP, Iperf permite al usuario especificar el tamaño de los datagramas, y devuelve como resultado la cantidad de datagramas transmitidos y el nivel de pérdida de paquetes.

Cuando se usa para medir la capacidad de transmisión de datos de protocolo TCP, Iperf mide la capacidad de transmisión neta de la red (el *payload* de los paquetes).

Las mediciones pueden expresarse tanto en MiB (1024 x 1024 bytes) como en Mb (10<sup>6</sup> bits).

La utilidad de Iperf consiste en que es una aplicación multi-plataforma, por lo que puede usarse sobre cualquier red para extraer mediciones estándar de performance. Gracias a eso, puede ser usada para comparar equipamiento tanto para redes inalámbricas como cableadas.

Además, el hecho de ser una herramienta de código abierto permite que el usuario observe cual es la metodología de medición utilizada.

#### **A.2.1.5. SequireISP**

SequireISP es una suite de productos prevista para proveedores de servicio de Internet (ISPs). Tiene un modo de licenciamiento mixto, en el que el core del producto es libre (licencia AGPL), mientras que algunos módulos verticales se proveen con cargo.

Incluye módulos para las siguientes funciones:

- facturación
- portal de auto-gestión para el usuario
- caché de datos, tanto para contenido web como para video
- medición de tráfico de datos y establecimiento de límites

- mesa de ayuda
- DHCP
- restricción de horarios de acceso
- monitoreo de equipos AP Ubiquiti

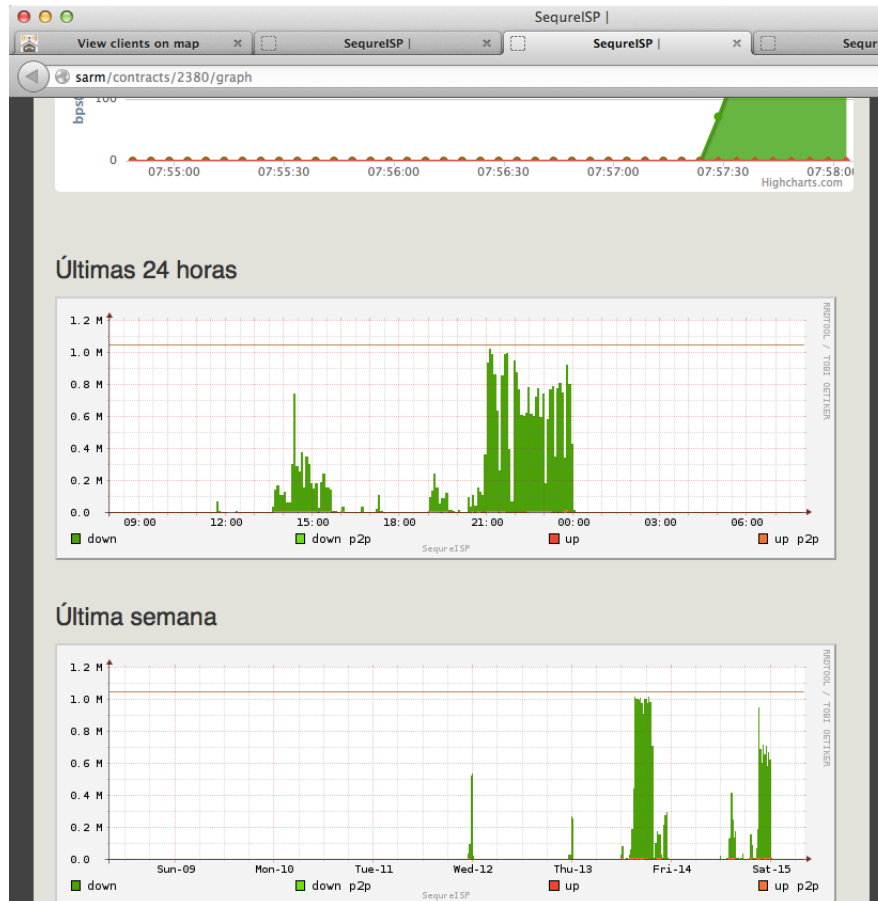


Figura 17: ejemplo de gráficos obtenidos de Iperf que muestran el nivel de tráfico de datos de un cliente particular, en las últimas 24 horas o en la última semana.



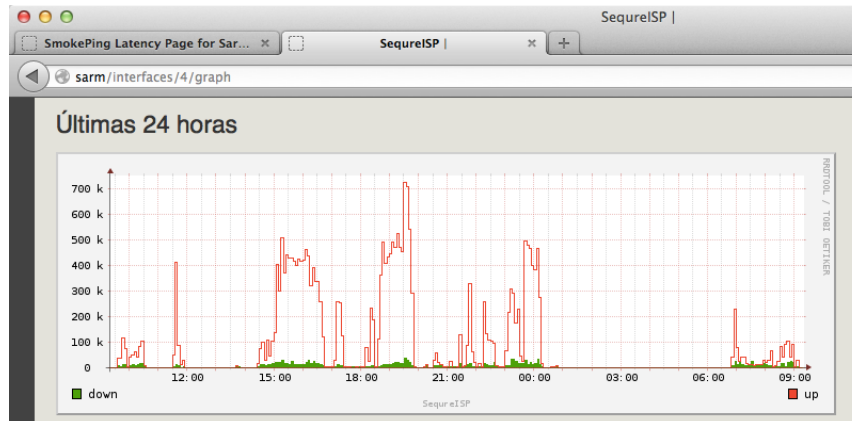


Figura 18: ejemplo de gráfico obtenidos de Iperf que muestra el nivel de tráfico de datos de un nodo de la red, en las últimas 24 horas.

#### A.4. Conclusiones de la prueba

Las mediciones realizadas desde nodos cliente alejados por hasta tres saltos del nodo gateway arrojaron los siguientes resultados:

- los parámetros de calidad de servicio para soportar servicio de telefonía IP se encuentran dentro de los márgenes adecuados;
- los parámetros de calidad de servicio como ISP logrados permiten ofrecer servicio de banda ancha utilizando la red mesh.

Con un despliegue de 11 nodos mesh se logró asegurar cobertura de voz y datos a una superficie de aproximadamente 9 hectáreas.

La implementación de una red mesh inalámbrica de estas características requiere, además de un cuidado planeamiento previo, de la verificación *in situ* de los resultados del despliegue realizado, para resolver problemas que puedan presentarse por obstáculos en la línea de visión entre los equipos (ej. árboles y tanques de agua), y asegurar la existencia de enlaces redundantes entre los nodos de la red.

Se observó una correcta integración de los equipos MP con la central telefónica Asterix de la cooperativa utilizando protocolo SIP.

Se observó la necesidad de cuidar las frecuencias (canales) con la que se configuran los nodos mesh y los *access point* que se utilicen en el área para evitar interferencia y solapamiento de señales con el fin de lograr la mayor capacidad de los equipos Mesh Potato.

## Anexo B – Solución desarrollada

### B.1. Aplicación web de registración

A continuación se presentan capturas de pantalla de ejemplo de la aplicación web desarrollada para mantener y consultar la base de datos de equipos configurados.



Figura 19: Diálogo de autenticación para ingreso a la aplicación

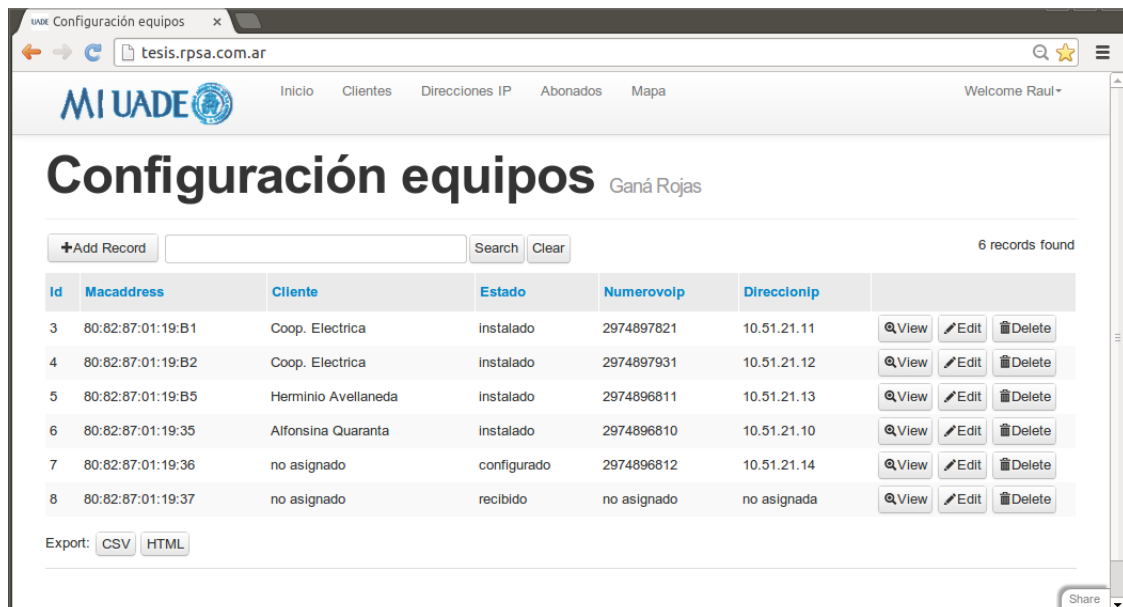


Figura 20: Función de altas, bajas, modificaciones y consulta de equipos.

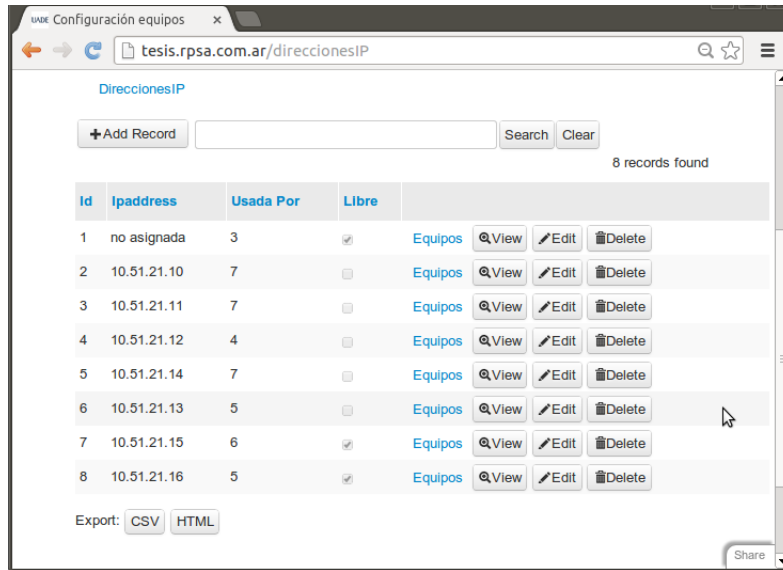


Figura 21: Función de mantenimiento del stock de direcciones IP para asignar.

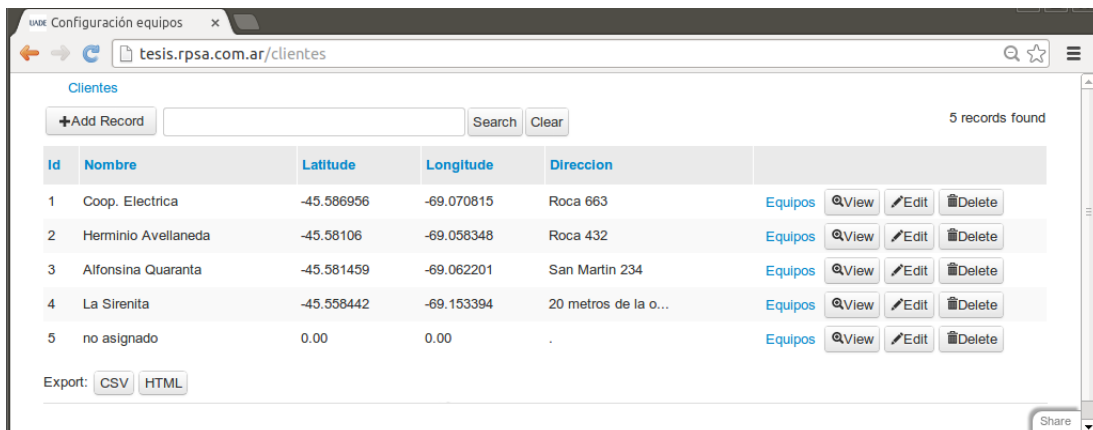


Figura 22: Función de mantenimiento de clientes con equipos asociados.

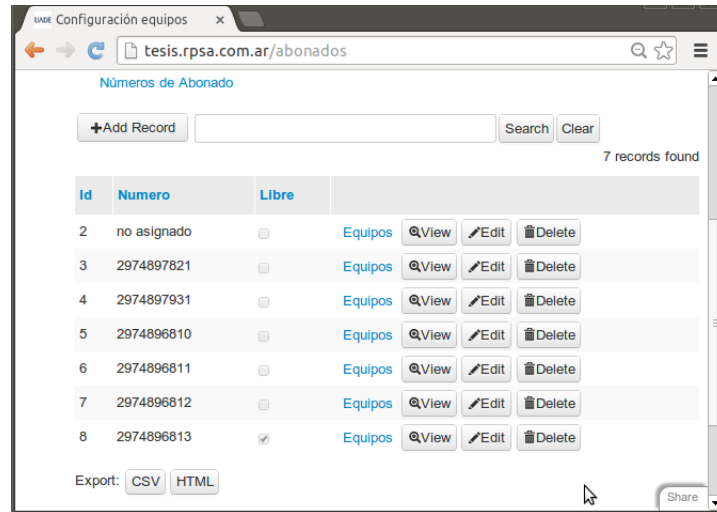


Figura 23: Función de mantenimiento del stock de números telefónicos para asignar.

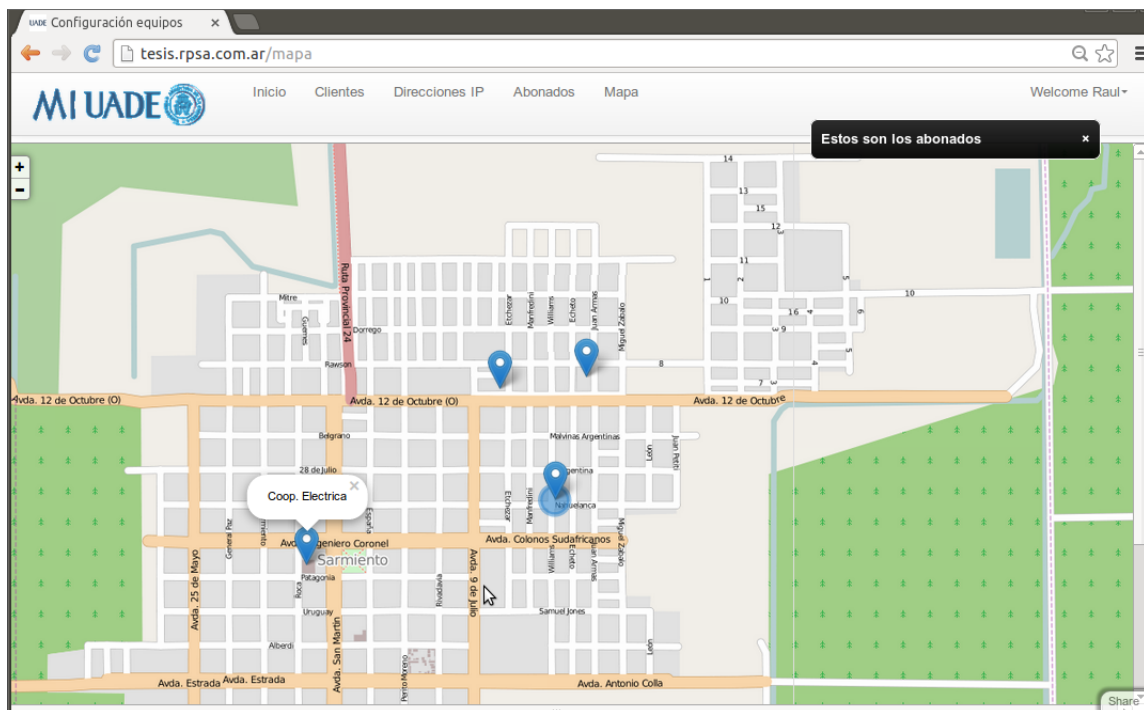


Figura 24: Visualización de nodos instalados sobre mapa de OpenStreetMap.org.

## B.2. Servicio web de configuración

Este servicio se encarga de aplicar en el equipo a configurar los cambios necesarios para lograr la configuración deseada. Concretamente en el caso de los equipos Mesh Potato genera y luego transfiere vía SCP los siguientes archivos de configuración:

```

/etc/config/secn
/etc/config/network
/etc/asterisk/modules.conf
/etc/asterisk/extensions.conf
/etc/init.d/config_secn

```

### B.3. Modelo de datos

En la base de datos de equipos configurados se modelaron las siguientes clases de objetos:

**numero-voip:** constituye el stock de números telefónicos disponibles para ser asignados a los equipos a configurar que incorporen telefonía por IP.

**ip-address:** constituye el stock de direcciones IP disponibles para ser asignadas a los equipos a configurar.

**cliente:** es una referencia que contiene los datos mínimos de los clientes a los que se asignaron los equipos configurados.

**equipo:** contiene el registro de los equipos configurados, los valores asignados y su estado.

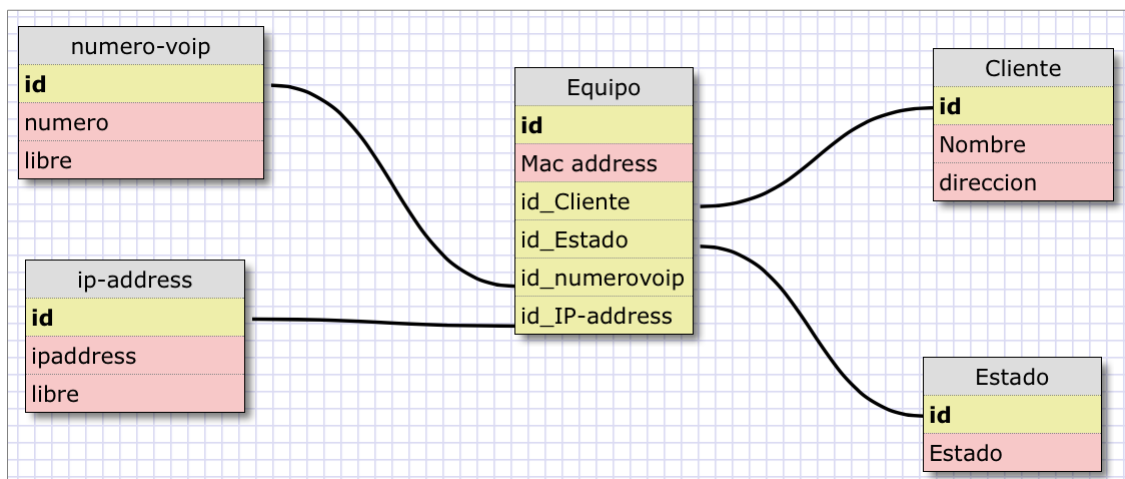


Figura 25: Modelo de datos de la aplicación de registro de equipos configurados