Diseñar e Implementar un Sistema Integrado de Puntos de Acceso Wi-Fi.

Antonio María Pérez

***Abstract*—Los puntos de acceso Wi-Fi permiten la conectividad en cualquier momento y lugar donde se tenga cobertura. Su sencillez y facilidad de uso hacen que rápidamente se estén desplegando en lugares que tienden a atraer a los usuarios nómadas. Sin embargo, su uso en muchos puntos tiende a una baja utilización y no son rentables. Con el propósito de mitigar las causas de esta baja utilización y poder aumentar el uso de puntos de acceso Wi-Fi para servicio público, es necesario diseñar e implementar en un único dispositivo, un sistema integrado de Access Point Wi-Fi que integre varios métodos de acceso a través de portales cautivos, con despliegue de publicidad controlada no invasiva, un servidor AAA y una interfaz de gestión web para toda la solución.**

***Index Terms*— Portal cautivo, RADIUS, redes de telecomunicaciones, redes inalámbricas, servidor AAA, Wi-Fi.**

I. Introducción

Los puntos de acceso Wi-Fi permiten la conectividad en cualquier momento y lugar donde se tenga cobertura[1]. Su sencillez y facilidad de uso hacen que rápidamente se estén desplegando en lugares que tienden a atraer a los usuarios nómadas. A pesar de que los puntos de acceso tienen un alcance limitado, un costo menor de instalación y sus anchos de banda de operación son más altos que otras alternativas como las redes inalámbricas 3G/4G, su uso en muchos puntos tiende a una baja utilización y no son rentables[2]. Esta baja utilización no se debe a la incompatibilidad tecnológica si no a otros factores, que inducen a un lento despliegue de nuevos puntos. Algunos factores que influyen son:

- La falta de rentabilidad para la sostenibilidad del servicio.
- El incremento de herramientas que vulneran la seguridad de las redes Wi-Fi. Por ejemplo: Aircrack, AirSnort, Kismet, Cain & Able, WireShark, Fern Wi-Fi Wireless Cracker, CoWPAtty, Airjack, WepAttack, NetStumbler, inSSIDer, Wifiphisher, entre otros[3].
- La vulnerabilidad al compartir datos privados y públicos en el mismo segmento de red.
- La baja oferta en el mercado de productos Wi-Fi con portales cautivos embebidos.
- La dependencia de gestión humana para la creación de cuentas de usuario Wi-Fi.
- La ausencia de diversos métodos de autenticación para acceder al servicio Wi-Fi por medio de una interfaz web.
- La falta de oportunidad de poder obtener información de los usuarios que acceden al servicio Wi-Fi.
- La carencia de poder emitir publicidad controlada a través de un portal cautivo.
- La necesidad de invertir en muchos dispositivos para brindar un servicio Wi-Fi a través de un portal cautivo.

Con el propósito de resolver las causas antes descritas y poder aumentar el uso de puntos de acceso Wi-Fi para servicio público, es necesario diseñar e implementar en único dispositivo, un sistema integrado de puntos de acceso Wi-Fi con gestión de portales cautivos[4] que permita las siguientes bondades técnicas:

- Conexión a redes Wi-Fi abiertas.
- Soporte de diversos métodos de autenticación para acceder al servicio Wi-Fi por medio de portales cautivos:
  1. Portal cautivo con capacidad de despliegue de un formulario personalizable con el objeto de poder capturar datos del usuario.
  2. Portal cautivo con capacidad de acreditar usuarios detrás de un sistema de autenticación digital descentralizado (OpenID)[5]. Servicio que permite identificarse ante una variedad de sitios web, que poseen las credenciales de un usuario y que son de uso reiterativo para él.
  3. Portal cautivo usuario/contraseña. Esquema que permite la conexión a bases de datos internas o externas donde se encuentran las credenciales del usuario, con posibilidad de recuperación de contraseñas a través de correo electrónico.
  4. Portal cautivo con acceso directo. Opción con la capacidad de conectar un determinado número de usuarios simultáneos sin necesidad de tener que escribir algún tipo de credencial. Muy funcional para escenarios que solo quieren desplegar información sin necesidad de capturar algún tipo de dato del usuario.
- Despliegue de publicidad o información corporativa controlada no invasiva.
- Seguridad a través de un servidor AAA[6] (autenticación, autorización y contabilización) interno o externo, bajo el estándar IEEE 802.1X[7].
- No requiere de un controlador central o máquinas virtuales para el despliegue del servicio de un portal cautivo.
- Capacidad de desplegar hasta dos portales cautivos simultáneos. Necesario cuando las credenciales de los usuarios vienen de dos fuentes diferentes de autenticación. Por ejemplo: Bases de datos locales y OpenID (API).

• Sistema operativo abierto. Permitirá la continuación y evolución en el desarrollo del producto.

Para el desarrollo de este trabajo se planteó como objetivo general diseñar e implementar un sistema integrado, flexible y seguro para portales cautivos Wi-Fi. Dentro del cual se desarrollaron los siguientes objetivos específicos:

• Diseñar un sistema de portal Wi-Fi que integre cuatro métodos de acceso con despliegue de publicidad controlada no invasiva, un servidor AAA y una interfaz de gestión web para toda la solución.

• Implementar un prototipo del sistema de portal Wi-Fi.

• Validar la funcionalidad y el desempeño del prototipo implementado.

• Validar las credenciales de un usuario a través de los diferentes métodos de portales cautivos, confirmando su acceso o denegación a la red restringida.

A continuación del presente artículo se encontrará organizado de la siguiente manera. Sección 2, trabajos similares al prototipo explicado en este artículo. Sección 3, se exponen las especificaciones del prototipo construido. Sección 4, se describe el desarrollo del proyecto. Sección 5, se presenta el protocolo de pruebas. Sección 6, análisis de resultados. Finalmente, en la sección 7, se presentan las conclusiones.

II. Trabajos Similares

Algunos fabricantes reconocidos de la industria como Cisco Systems, Hewlett-Packard, Aruba Networks y Ruckus poseen soluciones mixtas de Access Point con Wireless Lan Controller (dispositivo que se encarga de controlar todos los Access Point adscritos a él), que permiten algunas de las características técnicas anteriormente descritas y con la posibilidad de desplegar portales cautivos para requerimientos corporativos de gran escala. Estas soluciones no son funcionales para pequeños negocios o puntos de acceso con un área de cobertura limitada. No obstante, existe en la industria Access Point de tipo autónomo (dispositivo no dependiente de un tercero), que permiten ejecutar el servicio de un portal cautivo con autenticación de credenciales de manera local. Algunos de estos dispositivos son:

• LINKSYS LAPAC1750PRO Business Access Point Wireless: Permite establecer un portal cautivo personalizable, sin embargo, solo posibilita ciertos métodos de autenticación predefinidos. La personalización está limitada a texto e imágenes dentro de la misma página de gestión. La configuración se debe hacer por acceso directo al equipo o mediante una controladora de la misma marca. No dispone de un API (interfaz de programación de aplicaciones) para agregar más desarrollos propios a la funcionalidad del equipo.

• Aruba Networks IAP 105: Línea de Access Point autónomo, llamados IAP (punto de acceso instantáneo), con posibilidad de configurar un portal cautivo cuyo propósito es solamente la autenticación del usuario a través de credenciales soportadas sobre una base de datos local dentro del dispositivo.

• La compañía Cradlepoint y NetComm Wireless: Son Access Point autónomos con un portal cautivo embebido con el beneficio del 3G/4G, pero con sistema operacional cerrado. Solo permiten procesos de OEM para su comercialización.

III. Descripción

El proyecto realizado consiste en diseñar e implementar un prototipo de Access Point, que permite reducir en un solo dispositivo, la mayor cantidad de infraestructura posible, tanto en hardware como en software, en la implementación de servicios Wi-Fi. Este dispositivo permite proveer diferentes maneras de poder acceder a una red restringida por medio de portales cautivos que usan un servidor AAA para los servicios de autenticación como se muestra en ***Fig. 1***. El usuario envía su usuario y contraseña al portal cautivo, este portal envía la petición a RADIUS[8], quien realiza la autenticación, autorización y contabilización mediante un servidor AAA y si es un portal cautivo con proveedor externo, este utiliza un servidor LDAP, AD, ODBC o un API.

La integración y la seguridad ayudan a incrementar el despliegue de nuevos puntos de acceso Wi-Fi, principalmente en pequeños negocios que requieran sistemas sencillos y confiables que le permitan captar información de usuarios, distribuir contenidos, generar relaciones a largo plazo y aumentar así el grado de satisfacción.

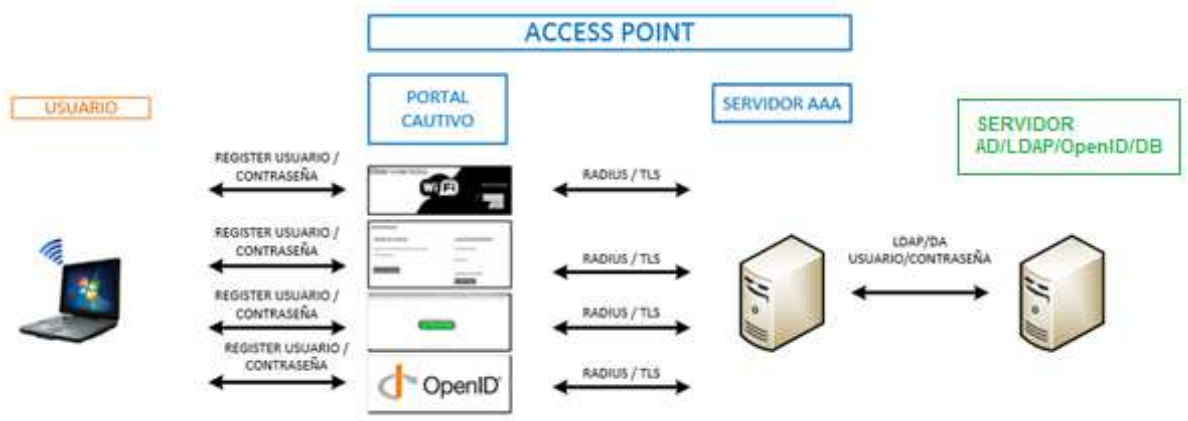

Fig. 1. Proceso de validación de credenciales.

El Access Point puede configurar y personalizar dos instancias, permitiendo una autenticación versátil desde diferentes fuentes donde se alojen las credenciales de los usuarios, para cualquiera de los siguientes portales cautivos:

- Generación de formularios para captura de datos.
- Acreditación de usuarios detrás de un sistema de autenticación digital descentralizado (OpenID).
- Conexión a través de una base de datos local por medio de usuario y contraseña.
- Acceso directo sin captura de credenciales.

El dispositivo es autónomo, a menos que el proceso de autenticación de usuarios requerido se ejecute desde un servidor AAA externo. Tiene los suficientes recursos (memoria, capacidad de procesamiento y almacenamiento) para soportar la integración de las siguientes características técnicas:

- Sistema operativo abierto.
- Servidor AAA (FreeRADIUS).
- Portales cautivos (CoovaChilli).
- Aplicaciones complementarias de software libre (HostADP, SSH, Haserl, Mysql, Postfix, CherryPy y Samba).
- Personalización de los portales cautivos por medio de hojas de estilo en cascada (CSS).
- Despliegue de publicidad o información corporativa controlada no invasiva.
- Direccionamiento del usuario a un sitio web personalizado después de ser validado.
- Posibilidad de desconectar usuarios autenticados.
- Interfaz de gestión web para todo el sistema.

IV. Desarrollo del Proyecto

A. *Hardware*

Los principales componentes de hardware utilizados son los siguiente:

- Placa base: Ofrece el soporte electrónico principal de la solución, proveyendo tanto las capacidades de procesamiento, como el almacenamiento volátil requerido para desplegar todos los componentes lógicos necesarios para llevar a cabo las labores objetivo del sistema.
- Tarjeta de red inalámbrica: Provee el soporte electrónico para difundir redes inalámbricas, cumpliendo con ello el aspecto más relevante de la solución.

B. *Software*

El software utilizado es el siguiente:

- HostAPD: Ofrece el soporte lógico necesario para crear y gestionar redes de datos inalámbricas, logrando que la solución actué como un punto de acceso WiFi independiente.
- Portal Cautivo: Provee el componente lógico principal con que interactúa el usuario, permitiendo definir y controlar las diferentes modalidades de autenticación que ofrece el sistema; así como, proveyendo el control del flujo de datos que se traduce en el acceso final del usuario a los recursos de la red que se encuentra tras el punto de acceso.
- Servidor AAA: Provee los servicios esenciales de seguridad de la solución, integrándose con el portal cautivo para dar respuesta a las peticiones del usuario final; así mismo, constituye el motor general de procesos que se usa como núcleo para el despliegue y operación de todos los componentes adicionales, que se traducen en las características de valor agregado de la solución.

C. *Protocolos*

El funcionamiento de la solución planteada se soporta sobre los siguientes protocolos:

- RADIUS (UDP 1812 y 1813): Permite la comunicación y operatividad entre la entidad AAA (Servidor RADIUS) y el NAS Virtual (Portal Cautivo).
- HTTP y HTTPS (TCP 80 y 443): Permite la navegación web de los usuarios, así como la comunicación de agentes externos con el API WEB.
- Protocolo de acceso a la base de datos. Depende del motor de base de datos puntual que se implemente; ofrece el servicio de almacenamiento de datos. tanto para la entidad AAA, así como para los servicios de consulta y gestión del API WEB.

V. Protocolo de Pruebas

A continuación, se detallan las fases de pruebas a realizar, los objetivos a alcanzar en cada una de ellas y los componentes que se evaluarán.

- Pruebas Unitarias: Se tiene como objetivo evaluar los componentes propios de la solución, a saber: Workers, Webmin, WebAPI. A fin de revisar su correcto funcionamiento en diversos escenarios.
- Pruebas de integración: Pretenden revisar la integración directa entre los componentes verificados en las pruebas unitarias con los componentes mayores, a saber: Servidor AAA, Portal Cautivo y HostAPD. Así como la correcta ejecución de los procesos integrados entre los mismos.
- Pruebas del sistema: Tienen como objetivo verificar la correcta prestación de los servicios objetivos del sistema, bajo un entorno controlado. Se evaluarán todos los componentes, propios y de terceros con especial énfasis en su comportamiento bajo escenarios adversos.
- Pruebas de despliegue: Presentan el mismo objetivo y características que las pruebas del sistema, salvo que estas se realizarán en un ambiente de producción, bajo un flujo de operación dictaminado por las actividades de los usuarios, en lugar de en un escenario controlado de laboratorio.

A. *Pruebas unitarias*

PRUEBA GENERAL DEL WEBMIN	
OBJETIVO	Evaluar la eficacia general del Webmin
FUNCIONES A PROBAR	Configurar servicios, CRUD (Create, Read, Update and Delete) de usuarios administradores, CRUD de usuarios generales, herramientas de testeo.
CASO DE PRUEBA	Según la función a evaluar se crearán una serie de casos, en los cuales se pruebe la respuesta esperada en virtud del proceso realizado. Se crearán como mínimo tres casos de pruebas, uno perfecto según el formato de la función, uno con fallas leves de las cuales el componente debe sobreponerse y uno inaceptable que el componente debe rechazar.

METRICA	Se usará una escala relativa entre 0 y 1, la cual representa la relación entre el número de casos exitosos y el total de casos evaluados.

PRUEBA GENERAL DEL WEB API	
OBJETIVO	Evaluar la eficacia general del Web API
FUNCIONES A PROBAR	Recepción de las peticiones externas, verificación de seguridad, generación de peticiones para servicios externos, procesamiento de resultados.
CASO DE PRUEBA	Según la función a evaluar se realizarán 10 casos concretos, con diferentes niveles de complejidad. Para funciones que dependan de servicios externos, se crearan 10 casos para cada proveedor.
METRICA	Se usará una escala relativa entre 0 y 1, la cual representa la relación entre el número de casos exitosos y el total de casos evaluados.

PRUEBA GENERAL DE LOS WORKERS IT	
OBJETIVO	Evaluar la eficacia general de los Workers
FUNCIONES A PROBAR	Ejecución de scripts de configuración, ejecución de tareas almacenadas, ejecución de tareas programadas en calendario, ejecución de herramientas de apoyo, ejecución de subrutinas, ejecución de procesos de sincronización.
CASO DE PRUEBA	Según la función a evaluar se realizarán 3 casos de prueba, el primero donde la invocación al worker es perfectamente correcta, uno donde el worker debe sobreponerse a un error y uno donde el worker debe rechazar la ejecución. Se realizará este esquema con cada posible configuración.
METRICA	Se usará una escala relativa entre 0 y 1, la cual representa la relación entre el número de casos exitosos y el total de casos evaluados.

B. Pruebas de integración

PRUEBA AAA-PC	
OBJETIVO	Evaluar la correcta integración entre el servidor AAA y el portal cautivo.
FUNCIONES A PROBAR	Autenticación de un usuario, autorización de un usuario, registro de actividades del usuario, aplicación de reglas por atributo directo, aplicación de reglas por grupo.
CASO DE PRUEBA	Se crearán 100 credenciales de usuario, asignadas a 5 grupos de control, cada credencial de usuario tendrá diferentes configuraciones de atributos, así como cada grupo de control. 20 credenciales tendrán una configuración correcta, 20 tendrán atributos con errores menores, 20 tendrán atributos individuales que contradigan los atributos de su grupo, 20 tendrán atributos con errores críticos y 20 tendrán atributos incompatibles críticamente con su grupo. Para cada credencial se realizarán 3 intentos de conexión y navegación.
METRICA	Se usará una escala relativa entre 0 y 1, la cual representa la relación entre el número de casos exitosos y el total de casos evaluados.

PRUEBA AAA-DB	
OBJETIVO	Evaluar la correcta integración entre el servidor AAA y la base de datos
FUNCIONES A PROBAR	Búsqueda y recuperación de credenciales, carga de configuraciones.
CASO DE PRUEBA	Se ejecutará el servidor AAA en modo de pruebas forzándolo a ejecutar cada una de las consultas SQL necesarias para cooperar con la base de datos. Se realizarán 3 pruebas en cada caso, con sentencias correctas, datos ambiguos y sentencias incorrectas.
METRICA	Se usará una escala relativa entre 0 y 1, la cual representa la relación entre el número de casos exitosos y el total de casos evaluados.

PRUEBA AAA-AD	
OBJETIVO	Evaluar la correcta integración entre el servidor AAA y el servidor ADC
FUNCIONES A PROBAR	Recuperación de listas de usuario, recuperación de metadatos, autenticación y autorización de usuarios.
CASO DE PRUEBA	Mediante la herramienta RADTEST y los scripts de apoyo de samba se realizarán 50 pruebas de verificación de credenciales, 30 pruebas de recuperación de listas de usuario y 20 pruebas de recuperación de metadatos. Durante las pruebas se alterarán las condiciones de la red que comunica el AAA con el AD, así mismo se añadirán insistencias en los registros

	DNS y la sincronización de Fecha/Hora por NTP.
METRICA	Se usará una escala relativa entre 0 y 1, la cual representa la relación entre el número de casos exitosos y el total de casos evaluados.

PRUEBA AAA-API	
OBJETIVO	Evaluar la correcta integración entre el servidor AAA y el Web API
FUNCIONES A PROBAR	Ejecución de tareas solicitadas vía API, comprobación de credenciales de proveedores externos.
CASO DE PRUEBA	Se ejecutará el servidor AAA en modo de pruebas, para luego mediante la herramienta PyRequests enviarle peticiones de control mediante el API. Así mismo, mediante la herramienta RADTEST se realizarán pruebas de verificación de credenciales de proveedores externos. Se realizarán 10 pruebas por cada función del API y se verificarán 10 distintas credenciales de los proveedores externos Google, Facebook y Twitter.
METRICA	Se usará una escala relativa entre 0 y 1, la cual representa la relación entre el número de casos exitosos y el total de casos evaluados.

PRUEBA AAA-CONECTORES	
OBJETIVO	Evaluar la correcta integración entre el servidor AAA y los conectores de apoyo.
FUNCIONES A PROBAR	Verificación de credenciales vía ODB, verificación de credenciales vía OpenLDAP.
CASO DE PRUEBA	Se ejecutará el servidor AAA en modo de pruebas, para luego mediante la herramienta RADTEST realizar pruebas de verificación de credenciales de proveedores accesibles vía ODBC y OpenLDAP. Se realizarán 50 pruebas con credenciales vía ODBC, de motores Oracle, Informix y SQL Server; además de otras 50 pruebas a servidores LDAP.
METRICA	Se usará una escala relativa entre 0 y 1, la cual representa la relación entre el número de casos exitosos y el total de casos evaluados.

PRUEBA WORKERS-ALL	
OBJETIVO	Evaluar la correcta integración entre los Workers y los componentes mayores.
FUNCIONES A PROBAR	Ejecución de las funciones maestras de los Workers, ejecución de tareas programadas.
CASO DE PRUEBA	Se ejecutarán los componentes mayores en modo de pruebas con salida de Log a texto; para luego forzar la ejecución de los Workers maestros. Para cada worker se realizarán 5 pruebas con diferentes niveles de complejidad.
METRICA	Se usará una escala relativa entre 0 y 1, la cual representa la relación entre el número de casos exitosos y el total de casos evaluados.

C. Pruebas del sistema

PRUEBA DE CONEXIÓN AL AP	
OBJETIVO	Evaluar la estabilidad en el proceso de conexión de diferentes clientes al AP.
FUNCIONES A PROBAR	Conexión, desconexión y reconexión de clientes.
CASO DE PRUEBA	Se escogerán diferentes equipos clientes (PC, LAPTOP, Tablet, etc.) de diferentes fabricantes y en diferentes escenarios de trabajo. Se evaluará la capacidad de los clientes de conectarse de forma estable a la solución, la correcta asignación de direcciones IP y el despliegue automático del portal cautivo. Se realizarán 10 pruebas por equipo cliente representativo.
METRICA	Se usará una escala relativa entre 0 y 1, la cual representa la relación entre el número de casos exitosos y el total de casos evaluados.

PRUEBA DE EFICACIA CON CREDENCIALES LOCALES	
OBJETIVO	Evaluar la eficacia del sistema para atender peticiones.
FUNCIONES A PROBAR	Autenticación, autorización, seguimiento, navegabilidad y gestión de atributos.
CASO DE PRUEBA	Se realizarán pruebas de servicio en base a 100 usuarios registrados localmente, para cada usuario se realizarán 3 intentos, uno correcto, uno con errores manejables y uno inaceptable. Se revisará la correcta asignación de atributos directos y por grupo, así como la capacidad de la solución para interpretar y aplicar los efectos de tales atributos.
METRICA	Se usará una escala relativa entre 0 y 1, la cual representa la relación entre

	el número de casos exitosos y el total de casos evaluados.

PRUEBA DE EFICACIA CON CREDENCIALES DE ADC	
OBJETIVO	Evaluar la eficacia del sistema para atender peticiones.
FUNCIONES A PROBAR	Autenticación, autorización, seguimiento, navegabilidad y gestión de atributos.
CASO DE PRUEBA	Se realizarán pruebas de servicio con base en 100 usuarios registrados en un controlador de directorio activo, para cada usuario se realizarán 4 intentos, uno correcto, uno con errores manejables, uno inaceptable y uno correcto, pero no registrado en ADC. Se revisará la correcta asignación de atributos directos y por grupo, así como la capacidad de la solución para interpretar y aplicar los efectos de tales atributos.
METRICA	Se usará una escala relativa entre 0 y 1, la cual representa la relación entre el número de casos exitosos y el total de casos evaluados.

PRUEBA DE EFICACIA CON CREDENCIALES DE PROVEEDOR PÚBLICO	
OBJETIVO	Evaluar la eficacia del sistema para atender peticiones.
FUNCIONES A PROBAR	Autenticación, autorización, seguimiento, navegabilidad y gestión de atributos.
CASO DE PRUEBA	Se realizarán pruebas de servicio con base en 120 usuarios registrados en los diferentes proveedores de credenciales públicas (Google, Facebook, y Twitter), para cada usuario se realizarán 4 intentos, uno correcto, uno con errores manejables, uno inaceptable y uno correcto, pero no registrado en el proveedor objetivo. Se revisará la correcta asignación de atributos directos y por grupo, así como la capacidad de la solución para interpretar y aplicar los efectos de tales atributos.
METRICA	Se usará una escala relativa entre 0 y 1, la cual representa la relación entre el número de casos exitosos y el total de casos evaluados.

PRUEBA DE EFICACIA CON CREDENCIALES DE DISPOSITIVO	
OBJETIVO	Evaluar la eficacia del sistema para atender peticiones.
FUNCIONES A PROBAR	Autenticación, autorización, seguimiento, navegabilidad y gestión de atributos.
CASO DE PRUEBA	Se realizarán pruebas de servicio con base en 100 usuarios registrados, que posean dispositivos asociados a sus credenciales, para cada usuario se realizarán 2 intentos, uno con el dispositivo asociado y otro con un dispositivo diferente suplantando al correcto. Se revisará la correcta asignación de atributos directos y por grupo, así como la capacidad de la solución para interpretar y aplicar los efectos de tales atributos.
METRICA	Se usará una escala relativa entre 0 y 1, la cual representa la relación entre el número de casos exitosos y el total de casos evaluados.

PRUEBA GENERAL DE WEBMIN	
OBJETIVO	Evaluar la eficacia de la herramienta unificada de gestión.
FUNCIONES A PROBAR	Gestión de usuarios, gestión de tareas, configuración general, herramientas de apoyo, herramientas de auditoria.
CASO DE PRUEBA	Se realizarán múltiples pruebas manuales a cada una de las prestaciones ofrecidas por el WebMin, con el fin de verificar que estas se ejecuten apropiadamente, aún en situaciones donde la solución se encuentra bajo altos niveles de carga.
METRICA	Se usará una escala relativa entre 0 y 1, la cual representa la relación entre el número de casos exitosos y el total de casos evaluados.

D. Pruebas de despliegue

Con el fin de validar las capacidades operativas de la solución, se recurrió a un escenario de pruebas masivo donde el sistema se viese sometido a todas las posibles variaciones de trabajo que puedan presentarse, tanto en términos de ejecución, como de integración con la infraestructura corporativa y la manipulación de la solución por diferentes tipos de usuario.

1. Se desplegó una unidad virtual de la solución sobre el servidor de virtualización central de la Universidad Católica de Pereira. Se asignaron recursos de hardware a la máquina virtual de tal forma que emularan idénticamente los recursos presentes en la solución física.
2. La solución se desplegó sobre el entorno de red corporativa mediante la creación de dos redes virtuales; tal que una se

conectó al directorio activo de la institución, mientras la otra ofrecía acceso directo a internet.

3. Se habilitaron credenciales para los estudiantes de la institución de tal forma que estos pudiesen acceder mediante credenciales almacenadas dentro de la solución. Para docentes y administrativos se habilitó la autenticación vía directorio activo, mientras que para visitantes se habilitaron los métodos de autenticación por registro y proveedor público.

4. Mediante el sistema de seguimiento del servidor RADIUS, y con la ayuda del personal de informática de la institución, se realizó un seguimiento minucioso de la actividad de los usuarios.

5. Se registró y detalló minuciosamente todos los cambios de configuración que se realizaron sobre la solución y su entorno de red; reportando todos los inconvenientes.

E. Pruebas específicas

1) Prueba para el portal cautivo con proveedor público

El esquema de pruebas para este modelo, se representa en ***Fig. 2***

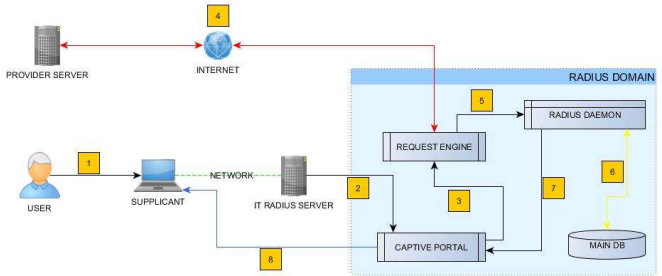

Fig. 2. Prueba para el portal cautivo con proveedor público

Donde se realizan los siguientes procesos:

1. El usuario de prueba accede a la dirección portal cautivo mediante un suplicante (PC, teléfono, etc.); se verifica el acceso de red al portal, así como el correcto despliegue del mismo. El usuario ingresa sus credenciales de acceso de proveedor público disparando el siguiente proceso.

2. La petición llega al servidor RADIUS, ingresando en el dominio del servicio; el portal cautivo recibe las credenciales y la identifica como de tipo público, invocando el siguiente proceso. Se verifica que el portal cautivo reconozca el dominio externo mediante el token [@domain].

3. El motor de peticiones recibe la solicitud del portal cautivo y genera una petición de autenticación pública según el API del proveedor correspondiente. Esta petición es enviada vía internet y se espera la respuesta

4. Se valida que se reciba una respuesta del proveedor correcto y que esta sea positiva (STATUS 0), ver Fig. 3, en tal caso se inicia el siguiente proceso; si se recibe una respuesta negativa, se ejecuta la siguiente tarea, pero marcando un REJECT como resultado.

```
root@radius:~# getPublicProvider -d google.com -u jherran07
PWD:

===============RESPONSE===============
{"status": 0, "id": "104561500744669318081", "user": "jherran07", "provider": "g
oogle"}
```

Fig. 3. Ejemplo de solicitud de autenticación sobre un usuario de google.

5. Se ejecuta un proceso de autenticación sobre el demonio RADIUS, de tal manera que este identifique al usuario externo y proceda con las fases de autorización y seguimiento.

6. El demonio RADIUS accede a la base de datos local, extrae los atributos correspondientes al usuario por medio de su grupo(s) asociado(s). Se valida que el demonio RADIUS identifique correctamente el esquema de grupos

7. El demonio RADIUS emite una respuesta en formato de protocolo RADIUS que es enviada al punto de acceso; este formato es estándar y se aplica indistintamente a todos los modelos de portal cautivo, salvo por algunos atributos que varían según el modelo. Se valida que la respuesta contenga todos los atributos esperados y que sea positiva (ACCEPT) o negativa (REJECT) según sea el caso de la prueba. Ver ***Fig. 4***.

```
root@radius:~# radtest usertest usertest localhost 0 radius
Sending Access-Request of id 32 to 127.0.0.1 port 1812
        User-Name = "usertest"
        User-Password = "usertest"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 0
        Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=32, length=37
        Tunnel-Type:0 = VLAN
        Tunnel-Medium-Type:0 = IEEE-802
        Tunnel-Private-Group-Id:0 = "417"
```

Fig. 4. Ejemplo de respuesta en formato de protocolo RADIUS

8. El portal cautivo reconoce la respuesta emitida por el demonio RADIUS y procede a conceder acceso al usuario en caso recibir un indicador positivo; en caso contrario emite los indicadores de error y procede a solicitar nuevamente la autenticación.

2) Prueba para el portal cautivo con proveedor externo

El esquema de pruebas para este modelo, se representa en ***Fig. 5***.

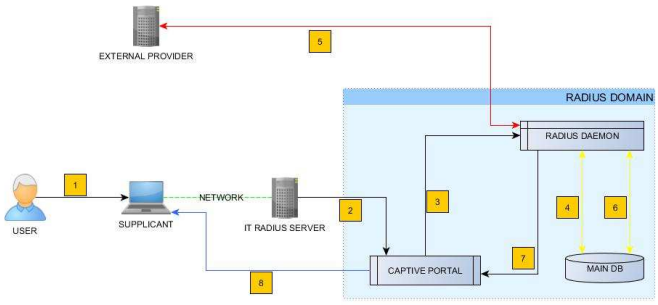

Fig. 5. Prueba para el portal cautivo con proveedor externo

Donde se realizan los siguientes procesos:

1. El usuario de prueba accede a la dirección portal cautivo mediante un suplicante (PC, teléfono, etc.); se verifica el acceso de red al portal, así como el correcto despliegue del mismo. El usuario ingresa sus credenciales de acceso de proveedor externo ejecutando el siguiente proceso.

2. La petición llega al servidor RADIUS, ingresando en el dominio del servicio; el portal cautivo recibe las credenciales y la identifica como de tipo común (para el portal cautivo es indiferente un usuario de proveedor externo a uno local), invocando la siguiente tarea. Se verifica que el portal cautivo determine correctamente el tipo de usuario y genere la petición correspondiente.

3. El demonio de RADIUS recibe la petición del portal cautivo, identifica el tipo de usuario en su fase PRE-AUTH y procede al siguiente proceso. Se verifica que el demonio RADIUS identifique el tipo de usuario como EXTERNO y ejecute el proceso correcto.

4. El demonio RADIUS busca en la base de datos principal, para verificar si posee datos de cache sobre la contraseña del usuario en cuestión. Si es así omite el proceso 5 y salta al 6; en caso contrario realiza el proceso 5. Se valida que el demonio RADIUS envíe la consulta a la base de datos local y reciba la respuesta adecuada.

5. El demonio RADIUS emite una petición al proveedor externo que corresponda al usuario en cuestión, eso lo hace a través de sus módulos auxiliares (LDAP, NTHL, ODBC, PY, etc.); una vez recibida la respuesta del proveedor, si es positiva marca el proceso para devolver un ACCEPT y almacenar la contraseña recibida; en caso contrario marca el proceso para devolver un REJECT. Se valida que el demonio RADIUS identifique correctamente al proveedor externo y que pueda comunicarse satisfactoriamente con él.

6. El demonio de RADIUS accede a la base de datos local, extrae los atributos correspondientes al usuario por medio de su grupo(s) asociado(s). Se valida que el demonio RADIUS identifique correctamente el esquema de grupos.

7. El demonio RADIUS emite una respuesta en formato de protocolo RADIUS que es enviada al portal cautivo; este formato es estándar y se aplica indistintamente a todos los modelos de portal, salvo por algunos atributos que varían según el modelo. Se valida que la respuesta contenga todos los atributos esperados y que sea positiva (ACCEPT) o negativa (REJECT) según sea el caso de la prueba. Ver ***Fig. 6***

```
root@radius:~# radtest -t mschap usertest usertest localhost 0 radius
Sending Access-Request of id 176 to 127.0.0.1 port 1812
        User-Name = "usertest"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 0
        Message-Authenticator = 0x00000000000000000000000000000000
        MS-CHAP-Challenge = 0x9c0b65c413134ca4
        MS-CHAP-Response = 0x000100000000000000000000000000000000000000000000000000000000
094bd91d4aa3905484eb27630ffd6f21c49d8b3fba42d3063
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=176, length=101
        Tunnel-Type:0 = VLAN
        Tunnel-Medium-Type:0 = IEEE-802
        Tunnel-Private-Group-Id:0 = "417"
        MS-CHAP-MPPE-Keys = 0xb5e6fa9da3fcf1e8804db8a0d67f5835bf4affe9493e369700
00000000000000
        MS-MPPE-Encryption-Policy = 0x00000001
        MS-MPPE-Encryption-Types = 0x00000006
```

Fig. 6. Ejemplo de respuesta en formato de protocolo RADIUS invocada sobre un proveedor de directorio activo

8. El portal cautivo reconoce la respuesta emitida por el demonio RADIUS y procede a conceder acceso al usuario en caso de recibir un indicador positivo; en caso contrario emite los indicadores de error y procede a solicitar nuevamente la autenticación.

3) Prueba para el portal cautivo con acceso directo sin captura de credenciales de autenticación

El esquema de pruebas para este modelo, se representa en ***Fig. 7***.

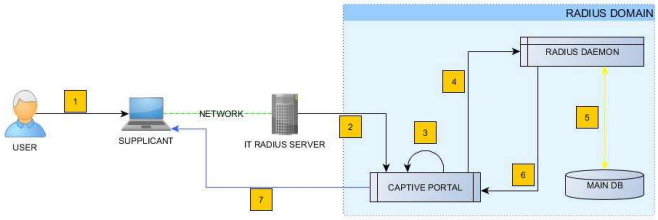

Fig. 7. Prueba para el portal cautivo con acceso directo sin captura de credenciales de autenticación

Donde se realizan los siguientes procesos:

1. El usuario de prueba accede a la dirección portal cautivo mediante un suplicante (PC, teléfono, etc.); se verifica el acceso de red al portal, así como el correcto despliegue del mismo. El usuario no ingresa ningún dato, solo procede a solicitar el acceso al medio, ejecutando el siguiente proceso.

2. La petición llega al servidor RADIUS, ingresando en el dominio del servicio; el portal cautivo recibe las credenciales vacías y la identifica como de tipo genérico, invocando la siguiente tarea. Se verifica que el portal cautivo determine correctamente el tipo de usuario y proceda con el proceso correspondiente.

3. El portal cautivo extrae de su configuración las credenciales del usuario genérico y establece con ellas una petición estándar para el demonio RADIUS. Se valida que el portal cautivo reconozca las credenciales vacías y sea capaz de transformarlas en una petición genérica.

4. El demonio RADIUS recibe la petición del portal cautivo, identifica el tipo de usuario en su fase PRE-AUTH y ejecuta el siguiente proceso. Se verifica que el demonio RADIUS identifique el tipo de usuario como LOCAL y realice el proceso correcto.

5. El demonio RADIUS accede a la base de datos local, extrae los atributos correspondientes al usuario por medio de su grupo(s) asociado(s). Se valida que el demonio RADIUS identifique correctamente el esquema de grupos.

6. El demonio RADIUS emite una respuesta en formato de protocolo RADIUS que es enviada al portal cautivo; este formato es estándar y se aplica indistintamente a todos los modelos de portal, salvo por algunos atributos que varían según el modelo. Se valida que la respuesta contenga todos los atributos esperados y que sea positiva (ACCEPT) o negativa (REJECT) según sea el caso de la prueba.

7. El portal cautivo reconoce la respuesta emitida por el demonio RADIUS y procede a conceder acceso al usuario en caso de recibir un indicador positivo, de lo contrario emite los indicadores de error y procede a solicitar nuevamente la autenticación.

4) Prueba para el portal cautivo con conexión por medio de credenciales almacenadas

El esquema de pruebas para este modelo, se representa en ***Fig. 8***

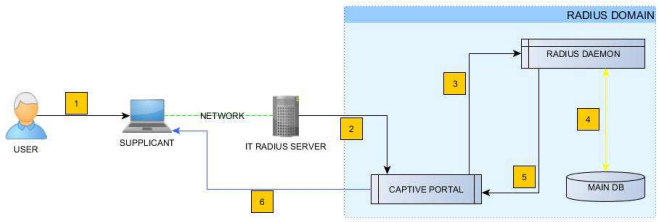

Fig. 8. Prueba para el portal cautivo con conexión por medio de credenciales almacenadas

Donde se realizan los siguientes procesos:

1. El usuario de prueba accede a la dirección portal cautivo mediante un suplicante (PC, teléfono, etc.); se verifica el acceso de red al portal, así como el correcto despliegue del mismo. El usuario ingresa sus credenciales de acceso local, realizando el siguiente proceso.
2. La petición llega al servidor RADIUS, ingresando en el dominio del servicio; el portal cautivo recibe las credenciales vacías y la identifica como de tipo genérico, invocando la siguiente tarea. Se verifica que el portal cautivo determine correctamente el tipo de usuario y desarrolle el proceso correspondiente.
3. El demonio RADIUS recibe la petición del portal cautivo, identifica el tipo de usuario en su fase PRE-AUTH y elabore la siguiente tarea. Se verifica que el demonio RADIUS identifique el tipo de usuario como LOCAL y proceda con el proceso correcto.
4. El demonio RADIUS accede a la base de datos local, extrae los atributos correspondientes al usuario por medio de su grupo(s) asociado(s). Se valida que el demonio RADIUS identifique correctamente el esquema de grupos.
5. El demonio RADIUS emite una respuesta en formato de protocolo RADIUS que es enviada al portal cautivo; este formato es estándar y se aplica indistintamente a todos los modelos de portal, salvo por algunos atributos que varían según el modelo. Se valida que la respuesta contenga todos los atributos esperados y que sea positiva (ACCEPT) o negativa (REJECT) según sea el caso de la prueba.
6. El portal cautivo reconoce la respuesta emitida por el demonio RADIUS y procede a conceder acceso al usuario en caso recibir un indicador positivo de lo contrario emite los indicadores de error y procede a solicitar nuevamente la autenticación.

5) Prueba para el portal cautivo para captura de datos por medio de formularios

El esquema de pruebas para este modelo, se representa en ***Fig. 9***.

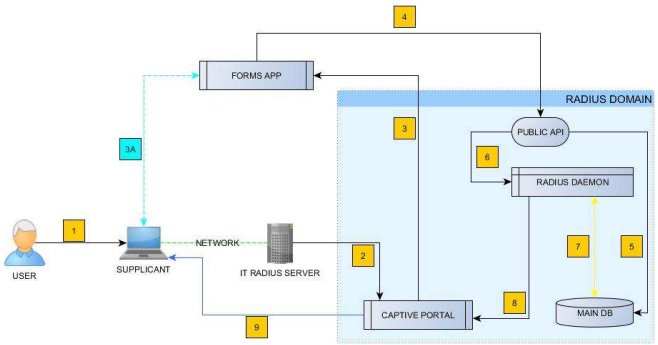

Fig. 9. Prueba para el portal cautivo para captura de datos por medio de formularios

Donde se realizan los siguientes procesos:

1. El usuario de prueba accede a la dirección portal cautivo mediante un suplicante (PC, teléfono, etc.); se verifica el acceso de red al portal, así como el correcto despliegue del mismo. El usuario al no poseer credenciales de acceso invoca el proceso de REGISTRO/CAPTURA DE DATOS.
2. La petición llega al servidor RADIUS, ingresando en el dominio del servicio; el portal cautivo recibe la solicitud de registro y redirección al usuario a la aplicación de formularios; esta aplicación puede ser local o externa en un servidor de red. Se valida que el portal cautivo identifique la solicitud de registro y que sea capaz de re direccionar el tráfico hacia ella, se verifica la configuración del GARDEN de direcciones y dominios necesarios para que funcione el APP de formularios.
3. El portal cautivo sede el control a la aplicación de formularios, el usuario interactúa directamente con la aplicación de formularios hasta completar su proceso de registro. Se valida la interacción del usuario con el APP de formularios.
4. El APP de formularios se comunica con el dominio RADIUS a través del API publica (ver anexo A), registrando los datos y atributos del nuevo usuario. El APP de formularios re direcciona nuevamente al cliente a la zona de espera del portal cautivo. Se valida que el APP de formularios se comunique correctamente con el API pública y que sea capaz de re direccionar al usuario al portal cautivo.
5. Los autómatas del API publica modifican la base de datos local para crear/actualizar los datos y atributos del usuario. Se valida que los autómatas del API publica se ejecuten correctamente y fijen los datos en la base de datos local.
6. Los autómatas del API publica invocan al demonio RADIUS para que ejecute un proceso común, pasándole los datos de usuario que han sido fijados en la base de datos. Se valida que el demonio RADIUS reciba la petición y las credenciales correctamente.
7. El demonio de RADIUS recibe la petición del portal cautivo, identifica el tipo de usuario en su fase PRE-AUTH y efectúa la siguiente tarea. Se verifica que el demonio RADIUS identifique el tipo de usuario como LOCAL y desarrolle el proceso correcto.

Posteriormente el demonio de RADIUS accede a la base de datos local, extrae los atributos correspondientes al usuario por medio de su grupo(s) asociado(s). Se valida que el demonio RADIUS identifique correctamente el esquema de grupos.

8. El demonio RADIUS emite una respuesta en formato de protocolo RADIUS que es enviada al portal cautivo; este formato es estándar y se aplica indistintamente a todos los modelos de portal, salvo por algunos atributos que varían según el modelo. Se valida que la respuesta contenga todos los atributos esperados y que sea positiva (ACCEPT) o negativa (REJECT) según sea el caso de la prueba.

9. El portal cautivo reconoce la respuesta emitida por el demonio RADIUS y procede a conceder acceso al usuario en caso de recibir un indicador positivo; en caso contrario emite los indicadores de error y procede a solicitar nuevamente la autenticación.

6) Proceso de verificación de ejecución del servidor FreeRADIUS 3.X

A continuación, se describe el proceso de verificación de ejecución del servidor FreeRADIUS.

1. Verificar el correcto despliegue del servicio, mediante la ejecución en modo activo y con DEBUG en primer plano. Como súper usuario se ejecuta el comando:

freeradius –s –X

Se evalúa la trama de salida del depurador, esperando encontrar el mensaje **"Ready to process request".** Ver ***Fig. 10***.

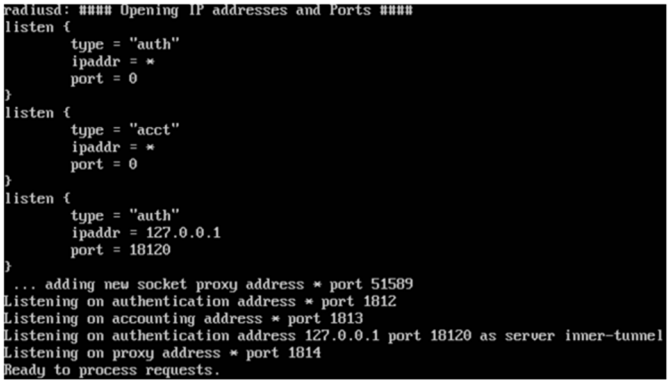

Fig. 10. Segmento final del servidor RADIUS ejecutado en modo de depuración

2. Se corre el servicio FreeRADIUS como demonio maestro del sistema. Como súper usuario se ejecuta el comando: **service freeradius start**

Se evalúa la respuesta, esperando encontrar el mensaje "OK". Ver ***Fig. 11***

Fig. 11. Salida del gestor de servicios al iniciar correctamente el servidor RADIUS.

3. Se verifica que el proceso de FreeRADIUS este activo en el sistema. Como súper usuario se ejecuta el comando:

ps –A | grep Radius

Se espera encontrar al menos un proceso de RADIUS activo.

Fig. 12. Salida del monitor de procesos.

4. Se verifica la carga y prioridad del servicio RADIUS. Como súper usuario se ejecuta el comando: **top**

Mediante el teclado se navega por la interfaz de texto hasta encontrar los datos pertenecientes al demonio de RADIUS.

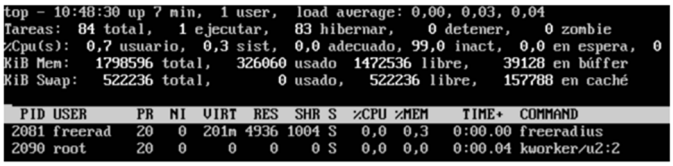

Fig. 13. Salida de la utilidad TOP fijada sobre el demonio del servicio RADIUS.

5. Se verifica el estado de los puertos UDP asociados al demonio RADIUS. Como súper usuario se ejecuta el comando:

lsof –i udp | grep rad

Se espera encontrar entradas relacionadas con los puertos configurados en el servidor FreeRADIUS.

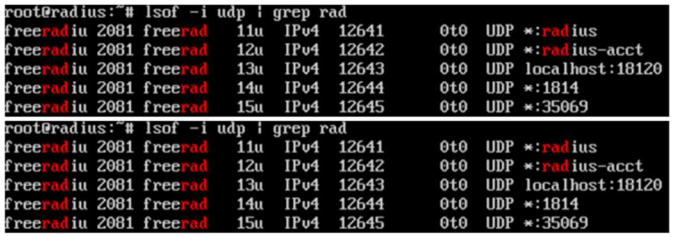

Fig. 14. Puertos configurados en el servidor FreeRADIUS.

VI. Análisis de Resultados

A continuación, se presenta una revisión estadística descriptiva de los resultados obtenidos en las pruebas realizadas.

A. Resultados de pruebas unitarias

RESULTADO DE PRUEBA GENERAL DEL WEBMIN			
CASO DE PRUEBA	**RESULTADO**		**RELEVANCIA O IMPACTO**
Crear perfiles o configuraciones	Caso ideal	1	30%
	Caso estándar	0,75	
	Caso crítico	0,5	
Consultar o recuperar perfiles o configuraciones	Caso ideal	1	30%
	Caso estándar	1	
	Caso crítico	0,8	

Actualizar perfiles o configuraciones	Caso ideal	1	10%
	Caso estándar	0,75	
	Caso crítico	0,5	
Anular perfiles o configuraciones	Caso ideal	0,9	20%
	Caso estándar	0,7	
	Caso crítico	0,8	
Generar reportes o estadísticas	Caso ideal	1	10%
	Caso estándar	1	
	Caso crítico	0,8	
RESULTADO GENERAL	**0,83**		**100%**

El resultado general se obtiene como la media ponderada entre el resultado obtenido en cada caso multiplicado por el factor apreciativo de relevancia o impacto. El factor de relevancia o impacto representa qué tan importante es el éxito de ese caso para que se considere que el módulo es eficiente y que puede cumplir con sus funciones.

En la prueba general del Webmin se tuvieron en cuenta las funciones primordiales que realiza el Webmin (crear, consultar o recuperar, actualizar y anular perfiles o configuraciones, y generar reportes o estadísticas) en tres diferentes escenarios: un caso ideal, en el cual se cumplen todas las condiciones esperadas para que los procesos se lleven a cabo correctamente; un caso estándar, en el cual se pueden admitir ligeros errores, tales como equivocaciones de entrada, fallas del usuario, requerimientos no satisfechos, entre otros; y un caso crítico inaceptable en el cual la suma de los problemas o errores debe desencadenar en que el sistema rechace o bloquee el proceso para evitar inconsistencias en el sistema.

Un caso crítico se alcanza esencialmente cuando la suma de las acciones actuales o anteriores del usuario llevan a punto sin retorno en los cuales o bien el sistema es incapaz de operar o uno de sus procesos entra en conflicto directo con otro; casos típicos de este escenario pueden ser: múltiples servicios corriendo en el mismo puerto, exceso de carga en la pila de servicios, imposibilidad de acceso a un recurso, conflictos de seguridad, entre otros.

Análisis. De los resultados presentados por esta prueba, es posible deducir que en los casos críticos el Webmin tiende a fallar en una proporción relativamente alta, esto debido a que sus labores desencadenan muchos procesos en paralelo, los cuales son difíciles de rastrear y verificar en su totalidad. En el caso relacionado con la anulación de elementos, la proporción de errores se incrementa incluso en casos ideales, ya que al anular un elemento se corre el riesgo de provocar fallos indirectos por integridad referencial o resolución de dependencias. Aun así, es posible estimar la prueba como exitosa.

RESULTADO DE PRUEBA GENERAL DEL WEB API			
CASO DE PRUEBA	**RESULTADO**		**RELEVANCIA O IMPACTO**
Consumir servicios externos	Complejidad baja	1	60%
	Complejidad media	0,9	
	Complejidad alta	0,7	
Proveer medios de cooperación con servicios externos	Complejidad baja	1	40%
	Complejidad media	0,8	
	Complejidad alta	0,5	
RESULTADO GENERAL	**0,82**		100%

En la prueba general del web API se tuvieron en cuenta las capacidades del sistema para acceder a recursos de proveedores públicos y consumir los servicios que éstos ofrecen. Así mismo, se evaluó la capacidad de la solución de proveer medios por los cuales sistemas externos puedan cooperar o hacer uso de las prestaciones del ecosistema RADIUS.

Análisis. Esta prueba evidencia como el componente web es propenso a errores, salvo en condiciones ideales, principalmente por la necesidad de contar tanto con un entorno de red estable como con un esquema de configuración que no presente limitaciones, además de la absoluta necesidad de que el lado externo se encuentre disponible, y correctamente configurado para para llevar a cabo el proceso. Es posible considerar la prueba como exitosa ya que se cumple de forma estable y en mayor medida con los objetivos de los casos planteados.

RESULTADO DE PRUEBA GENERAL DE LOS WORKERS			
CASO DE PRUEBA	**RESULTADO**		**RELEVANCIA O IMPACTO**
Ejecución de scripts de configuración	Caso ideal	1	40%
	Caso estándar	0,8	
	Caso crítico	0,5	
Ejecución de tareas almacenadas	Caso ideal	1	15%
	Caso estándar	0,8	
	Caso crítico	0,7	
	Caso ideal	1	10%

Ejecución de tareas programadas en calendario	Caso estándar	0,8	
	Caso crítico	0,7	
Ejecución de herramientas de apoyo	Caso ideal	1	5%
	Caso estándar	1	
	Caso crítico	1	
Ejecución de subrutinas	Caso ideal	1	20%
	Caso estándar	0,7	
	Caso crítico	0,7	
Ejecución de procesos de sincronización	Caso ideal	0,9	10%
	Caso estándar	0,6	
	Caso crítico	1	
RESULTADO GENERAL	**0,80**		**100%**

En la prueba general de los workers se evaluaron individualmente todos los componentes intermedios que se encargan de mantener sincronizado el ecosistema RADIUS, logrando con ello proveer las funciones básicas y de valor agregado

Análisis. Los resultados de esta prueba evidencian que si bien existen fallas en la ejecución de los workers, producto de la inevitable complejidad de los mismos, éstos son en mayor medida estables, salvo por los escenarios de caso crítico. Dado el gran volumen de elementos que componen los workers y la eminente entropía degenerada por sus labores, es posible considerar a la prueba exitosa, más halla los fallos que se presentan.

Es igualmente importante mencionar que en casos donde los workers dependen de elementos externos, el escenario crítico no suele fallar nunca, ya que el comportamiento esperado en tales circunstancias es que el proceso se aborte para evitar consecuencias negativas, y dado que en el escenario crítico el recurso externo no está disponible, resulta natural que se dé el comportamiento esperado de forma sencilla.

B. Resultados de pruebas de integración

RESULTADO DE PRUEBA AAA-PC			
CASO DE PRUEBA	**RESULTADO**		**RELEVANCIA O IMPACTO**
Autenticación y autorización de un usuario	Caso ideal	1	50%
	Caso estándar	0,9	
	Caso crítico	1	
Registro de actividades del usuario	Caso ideal	1	20%
	Caso estándar	1	
	Caso crítico	0	
Aplicación de reglas por atributo	Caso ideal	1	30%
	Caso estándar	0,7	
	Caso crítico	0,5	
RESULTADO GENERAL	**0,83**		**100%**

En la prueba AAA-PC se evaluó la capacidad de cooperación entre el servidor de protocolo RADIUS y el servicio de portal cautivo, tanto en sus funciones básicas de autenticación, autorización y seguimiento como en su capacidad de ofrecer servicios enriquecidos a través del sistema de usuarios, grupos y atributos.

Análisis. Los resultados de esta prueba permiten extraer tres conclusiones primordiales, a saber:

1. El proceso de autenticación y autorización se comporta de forma perfecta en el escenario óptimo, ya que todos los elementos se encuentran disponibles alcanzando, por tanto, el resultado esperado. Así mismo, en el escenario crítico el comportamiento es perfecto, ya que el par externo no se encuentra disponible y se alcanza un estado de rechazo en la petición, lo cual es precisamente el resultado esperado en tales circunstancias.

2. En el caso de registro de actividades, bajo el escenario crítico se obtiene el peor resultado posible, esto debido a que la prueba está planteada de tal forma que en condiciones críticas el registro debería detenerse, sin embargo, dada la configuración base del servidor RADIUS el proceso de registro es universal y no se puede discriminar su ejecución por escenario o usuario.

3. La aplicación de reglas por atributo presentan un elevado índice de fallas en condiciones críticas, dado que el protocolo RADIUS no define ningún mecanismo para priorizar atributos cuando un perfil presenta redundancias en sus definiciones, por lo que la solución a esto se deja completamente al azar.

Bajo la perspectiva anterior es posible considerar a la prueba como exitosa.

RESULTADO DE PRUEBA AAA-DB			
CASO DE PRUEBA	**RESULTADO**		**RELEVANCIA O IMPACTO**
Búsqueda y recuperación de credenciales	Caso ideal	1	50%
	Caso estándar	0,9	
	Caso crítico	0,7	
Carga de configuraciones	Caso ideal	1	50%

	Caso estándar	0,8	
	Caso crítico	0,6	
RESULTADO GENERAL	**0,83**		**100%**

En la prueba AAA-DB se evalúa la capacidad del servidor de protocolo RADIUS de interactuar con un proveedor de base de datos relacionales para realizar las funciones de manejo de información vitales para la ejecución del ecosistema.

Análisis. Los resultados de esta prueba, evidencian claramente como en casos críticos, donde la configuración del motor RDBMS no satisface los requisitos del servidor RADIUS, se incrementan notablemente los errores de la solución, especialmente en las labores relacionadas con la carga dinámica de configuraciones, lo que a su vez puede degenerar en problemas catastróficos. Sin embargo, dado que los escenarios críticos son muy escasos salvo que se produzcan intencionadamente, es posible considerar esta prueba como exitosa.

RESULTADO DE PRUEBA AAA-AD			
CASO DE PRUEBA	**RESULTADO**		**RELEVANCIA O IMPACTO**
Recuperación de listas de usuario	Caso ideal	0,8	25%
	Caso estándar	0,7	
	Caso crítico	1	
Recuperación de metadatos	Caso ideal	0,8	15%
	Caso estándar	0,7	
	Caso crítico	1	
Autenticación y autorización de usuarios	Caso ideal	0,9	60%
	Caso estándar	0,75	
	Caso crítico	1	
RESULTADO GENERAL	**0,86**		**100%**

La prueba AAA-AD tiene como objetivo evaluar el rendimiento del componente RADIUS enlazado a un controlador de directorio activo como fuente de credenciales, explorando con ello las diferentes complicaciones que pueden originarse en tal escenario.

Análisis. En primer lugar, es importante recalcar que la solución opera perfectamente en condiciones críticas, ya que está diseñada para rechazar todo tipo de peticiones si el par remoto no se encuentra disponible, lo cual es el comportamiento deseado para tales circunstancias. De igual forma, es importante mencionar que en escenarios comunes el rendimiento presentado es claramente mediocre, esto principalmente debido a la inestabilidad inherente que existe al acoplar un controlador AD con un servidor base Linux. A pesar de que la prueba puede considerarse como exitosas, es necesario concluir que la conexión directa AAA-AD no es ideal y debe ser mejorada.

RESULTADO DE PRUEBA AAA-API			
CASO DE PRUEBA	**RESULTADO**		**RELEVANCIA O IMPACTO**
Ejecución de tareas solicitadas vía API	Caso ideal	1	40%
	Caso estándar	0,9	
	Caso crítico	0,7	
Comprobación de credenciales de proveedores externos.	Caso ideal	1	60%
	Caso estándar	0,8	
	Caso crítico	1	
RESULTADO GENERAL	**0,90**		**100%**

La prueba AAA-API permite evaluar la correcta sincronización entre el tráfico gestionado mediante el API y sus efectos sobre el servidor RADIUS base, tanto desde la perspectiva del consumidor de recursos remotos como desde el enfoque de servidor público desligado.

Análisis. Los resultados de esta prueba remarcan como la solución está preparada para proceder correctamente en escenarios críticos donde la colaboración con elementos externos es fundamental y estos no se encuentran disponibles. Así mismo, es apropiado estimar que el enfoque como prestador de recursos en condiciones ideales y normales es casi perfecto, decayendo en eficiencia solo en escenarios críticos donde el entorno de red es altamente desfavorable.

RESULTADO DE PRUEBA AAA-CONECTORES			
CASO DE PRUEBA	**RESULTADO**		**RELEVANCIA O IMPACTO**
Verificación de credenciales vía ODBC	Caso ideal	0,9	50%
	Caso estándar	0,8	
	Caso crítico	1	
Verificación de credenciales vía OpenLDAP.	Caso ideal	1	50%
	Caso estándar	0,9	
	Caso crítico	1	
RESULTADO GENERAL	**0,93**		**100%**

El objetivo de la prueba AAA-CONECTORES es evaluar el rendimiento del sistema RADIUS cuando trabaja con fuentes externas como proveedores de credenciales, siempre y cuando la conexión con estos se realice bajo protocolos y herramientas de libre acceso.

Análisis. Esta prueba puede considerarse altamente exitosa tanto por sus resultados directos como al compararla con la prueba AAA-AD, la cual es muy similar en concepto, pero radicalmente diferente en ejecución, dada la naturaleza cerrada del protocolo y las tecnologías que esta precisa. Teniendo esto en cuenta es posible concluir sintéticamente que el sistema RADIUS es más estable y eficiente cuando trabaja bajo esquemas de conexión abiertos que cuando debe operar bajo esquemas privativos.

RESULTADO DE PRUEBA WORKERS-ALL			
CASO DE PRUEBA	**RESULTADO**		**RELEVANCIA O IMPACTO**
Ejecución de las funciones maestras de los Workers	Caso ideal	0,9	80%
	Caso estándar	0,7	
	Caso crítico	0,5	
Ejecución de tareas programadas.	Caso ideal	1	20%
	Caso estándar	0,8	
	Caso crítico	0,8	
RESULTADO GENERAL	**0,73**		**100%**

El objetivo primordial de la prueba WORKERS-ALL es avaluar la sinergia que debe existir entre los Workers y los servicios fundamentales del ecosistema RADIUS, dado que de su correcta integración depende la estabilidad y eficiencia general de la solución.

Análisis. En términos generales es posible estimar esta prueba como exitosa, sin embargo, es preciso mencionar que en condiciones críticas y muchos escenarios normales, la complejidad y la entropía que desencadenan los Workers maestros tiende a provocar errores, la mayoría de esos aleatorios; por lo que es necesario concluir que el esquema Workers debe sr revisado y simplificado a fin de mejorar la estabilidad general del sistema.

C. Resultados de pruebas del sistema

RESULTADO DE PRUEBA DE CONEXIÓN AL AP			
CASO DE PRUEBA	**RESULTADO**		**RELEVANCIA O IMPACTO**
Conexión, desconexión y reconexión de clientes.	Caso ideal	1	100%
	Caso estándar	0,8	
	Caso crítico	0,5	
RESULTADO GENERAL	**0,76**		**100%**

La prueba de conexión al AP tiene como objetivo evaluar el rendimiento de la solución en escenarios donde las condiciones físicas del entorno afectan directamente a la experiencia de usuario.

Análisis. Es posible considerar esta prueba como exitosa, ya que, si bien los resultados en escenarios críticos son mediocres, esto se debe a imposibilidades físicas insorteables, salvo mediante modificaciones de hardware que están fuera de los alcances del proyecto.

RESULTADO DE PRUEBA DE EFICACIA CON CREDENCIALES LOCALES			
CASO DE PRUEBA	**RESULTADO**		**RELEVANCIA O IMPACTO**
Autenticación, autorización, seguimiento, navegabilidad y gestión de atributos.	Caso ideal	1	100%
	Caso estándar	0,9	
	Caso crítico	0,6	
RESULTADO GENERAL	**0,83**		**100%**

El objetivo de esta prueba es evaluar la estabilidad y rendimiento general de la solución, cuando se usan únicamente los recursos locales como proveedores de credenciales.

Análisis. Esta prueba puede considerarse altamente exitosa, dados los resultados tanto en escenarios ideales como generales, aun así, es preciso mencionar que bajo condiciones críticas queda un amplio margen de mejora para futuras revisiones de la solución.

RESULTADO DE PRUEBA DE EFICACIA CON CREDENCIALES DE ADC			
CASO DE PRUEBA	**RESULTADO**		**RELEVANCIA O IMPACTO**
Autenticación, autorización, seguimiento, navegabilidad y gestión de atributos.	Caso ideal	1	100%
	Caso estándar	0,8	
	Caso crítico	1	
RESULTADO GENERAL	**0,93**		**100%**

La prueba de eficacia con credenciales de ADC tiene como objetivo evaluar la estabilidad y el rendimiento del sistema RADIUS cuando se usa un controlador de directorio activo como único proveedor de credenciales.

Análisis. Los resultados de esta prueba pueden considerarse como contradictorios si se contrastan con los de la prueba AAA-AD, sin embargo, tales diferencias pueden explicarse si

se tiene en cuenta que en esta ocasión la prueba se basa en medir la eficacia de la solución, es decir su capacidad de realizar las labores que se esperan de ella, y que para tal fin se tiene en cuenta el resultado final que ofrece todo el ecosistema, en lugar de únicamente el arrojado por el núcleo RADIUS como era el caso en la prueba anterior. Una vez explicada la discrepancia en los resultados es posible considerar esta prueba como exitosa.

RESULTADO DE PRUEBA DE EFICACIA CON CREDENCIALES DE PROVEEDOR PÚBLICO			
CASO DE PRUEBA	**RESULTADO**		**RELEVANCIA O IMPACTO**
Autenticación, autorización, seguimiento, navegabilidad y gestión de atributos.	Caso ideal	1	100%
	Caso estándar	0,9	
	Caso crítico	1	
RESULTADO GENERAL	**0,96**		**100%**

La prueba de eficacia con credenciales de proveedor público como objetivo evaluar la estabilidad y el rendimiento del sistema RADIUS cuando se usan servicios públicos como única fuente de credenciales.

Análisis. Gracias a los componentes de apoyo que le permiten al ecosistema RADIUS colaborar de forma trasparente con los principales proveedores públicos de credenciales, es posible considerar esta prueba como altamente exitosa, ya que solo bajo escenarios muy puntuales donde el entorno de red es inestables, aleatorio y adverso, se presentan resultados no deseados.

RESULTADO DE PRUEBA DE EFICACIA CON CREDENCIALES DE DISPOSITIVO			
CASO DE PRUEBA	**RESULTADO**		**RELEVANCIA O IMPACTO**
Autenticación, autorización, seguimiento, navegabilidad y gestión de atributos.	Caso ideal	1	100%
	Caso estándar	0,8	
	Caso crítico	0,6	
RESULTADO GENERAL	**0,8**		**100%**

La prueba de eficacia con credenciales de dispositivo tiene como objetivo evaluar la estabilidad y el rendimiento del sistema RADIUS cuando se usan las direcciones físicas de los dispositivos clientes como credencial de autenticación.

Análisis. Si bien es cierto que los resultados de la prueba permiten catalogar a esta como satisfactoria, es igualmente cierto que se observa un margen de mejora considerable, esto se debe principalmente a que el proceso de autenticación por dispositivo no es trasparente para el usuario final, sino que implica un proceso de registro previo el cual puede fallar por desconocimiento del usuario sobre los conceptos mínimos necesarios para completar tal labor. También es preciso tener en cuenta que este método depende ampliamente de que los dispositivos clientes ofrezcan verídicamente su dirección física durante el proceso de autenticación y autorización, algo que, si bien suele darse por sentado, en la realidad está bastante alejado del ideal esperado, ya sea por negligencia de los fabricantes que comercializan dispositivos con direcciones clonadas, o por la presencia de software malicioso en el dispositivo el cual puede alterar la firma física del mismo.

RESULTADO DE PRUEBA GENERAL DE WEBMIN			
CASO DE PRUEBA	**RESULTADO**		**RELEVANCIA O IMPACTO**
Gestión de usuarios	Caso ideal	1	30%
	Caso estándar	0,9	
	Caso crítico	0,7	
Gestión de tareas	Caso ideal	1	20%
	Caso estándar	0,8	
	Caso crítico	0,8	
Configuración general	Caso ideal	1	30%
	Caso estándar	0,8	
	Caso crítico	0,6	
Herramientas de apoyo	Caso ideal	1	10%
	Caso estándar	1	
	Caso crítico	1	
Herramientas de auditoría.	Caso ideal	1	10%
	Caso estándar	0,8	
	Caso crítico	0,8	
RESULTADO GENERAL	**0,86**		**100%**

La prueba general de Webmin tiene como objetivo evaluar el Webmin como componente sistémico vital de la solución, ya no centrándose en sus funciones atómicas sino en las prestaciones y responsabilidades generales que recaen sobre él.

Análisis. En primer lugar, es posible estimar los resultados de la prueba como exitosas, no ideales, pero sí satisfactorios. Sin embargo, aunque en términos generales todos los aspectos evaluados se encuentran dentro de lo esperado, esta prueba evidencia claramente como la interacción del usuario humano afecta drásticamente los resultados de un proceso que puede considerar infalible. Tal efecto se debe principalmente al desconocimiento del usuario tanto de conocimientos esenciales sobre informática, como de detalles específicos propios del diseño de la solución o la interfaz del Webmin. Por tanto, es

necesario concluir que el Webmin debe ser revisado para hacerlo más “amable” con usuario final, y recomendar especialmente que las funciones complejas de la solución se encarguen solo a usuarios con un nivel de conocimiento técnico apropiado.

D. Relación general de errores por tipo

Tabla 1. Relación general de errores por tipo

TIPO ERROR	CANTIDAD	PORCENTAJE
LOGICO DE PROGRAMACIÓN	231	8,1
DE ENTORNO DE RED	417	14,5
DE CONFIGURACIÓN	129	4,5
DE INTEGRACIÓN CON AD	239	8,3
DE INTEGRACIÓN EXTERNA	69	2,4
DE MANIPULACIÓN DE USUARIO FINAL	1781	62,1

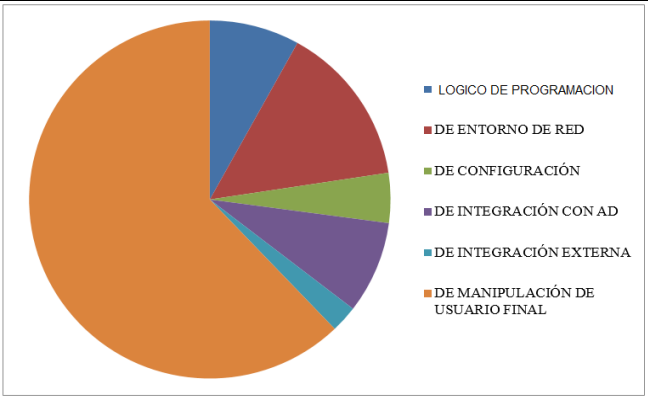

Fig. 15. Relación general de errores por tipo

E. Relación de errores lógicos de la solución

Tabla 2. Relación de errores lógicos de programación

CAUSA	POCENTAJE
PLANTILLA DE CONFIGURACIÓN INCORRECTA	15,3
CASO ESPECIAL OMITIDO	10,2
MANEJO INCORRECTO DE PROTOCOLO	25,4
CONFIGURACIÓN DE INTEGRACIÓN ERRONEA	18,9
CONFIGURACIÓN DE EJECUCIÓN ERRONEA	11,7
ERRORES GENERALES DE LOGICA	18,5

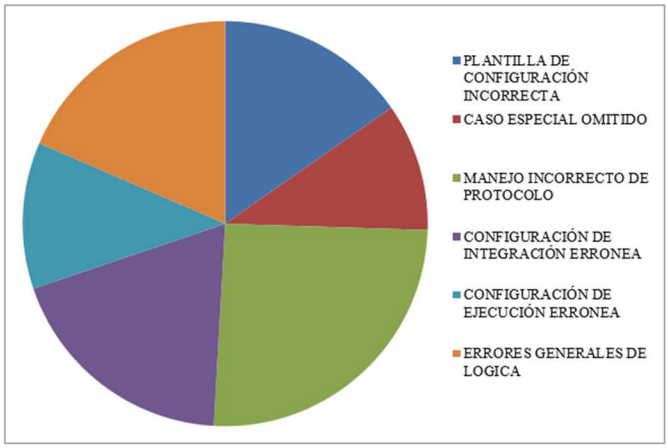

Fig. 16. Relación de errores lógicos de programación

F. Relación de errores ocasionados por el entorno de red

Tabla 3. Relación de errores ocasionados por el entorno de red

CAUSA	POCENTAJE
RUTAS ESTATICAS MAL CONFIGURADAS	33,8
POLITICAS DE SEGURIDAD DE RED INCOMPATIBLES	40,2
CONFIGURACIÓN IP/VLAN INCORRECTA	15,9
MEDIO DE TRASMISIÓN DEFECTUOSO	10,1

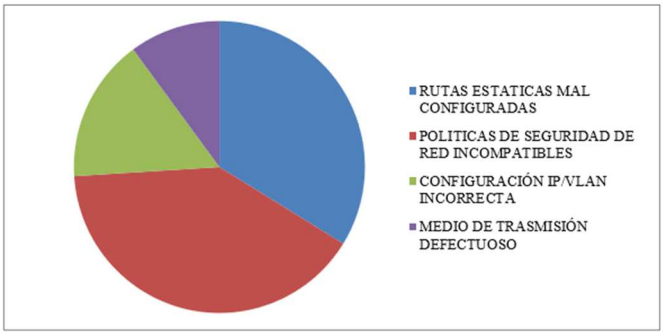

Fig. 17. Relación de errores ocasionados por el entorno de red

G. Relación de causas probables de los errores ocasionados por intervención humana

Tabla 4. Relación de causas probables de los errores ocasionados por intervención humana

CAUSA	PORCENTAJE
FALTA DE DOCUMENTACIÓN PARA CONSULTA	48,0

FALTA DE CONOCIMIENTO TECNICO PARA MANIPULAR LA SOLUCIÓN	15,5
FALTA DE CAPACITACIÓN GENERAL DE LOS USUARIOS FINALES	32,8
INTERFAZ DE USUARIO FINAL POCO INTUITIVA	3,7

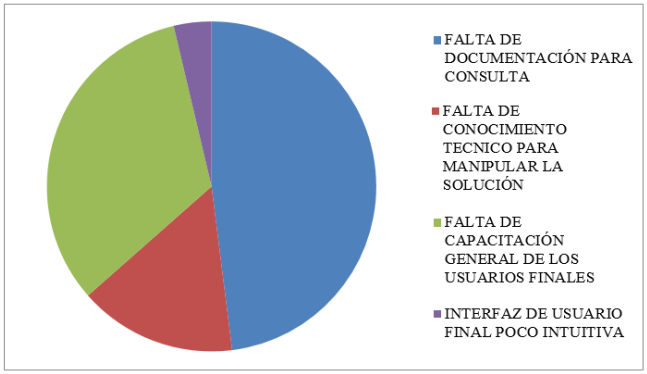

Fig. 18. Relación de causas probables de los errores ocasionados por intervención humana

H. Relación usuarios concurrentes – errores de usuario final

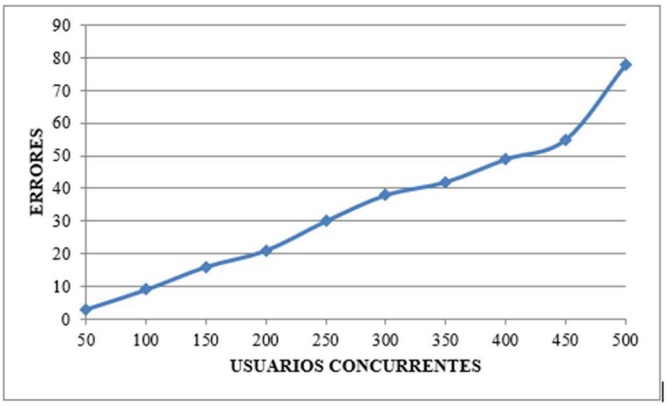

Fig. 19. Relación entre usuarios concurrentes y errores de usuario final.

En la ***Fig. 19*** se describe el crecimiento de la aparición de errores de usuario final, a medida que crece la carga de usuarios concurrentes en el sistema; los valores representados no son puntuales ni absolutos, sino una aproximación basada en la media total de los errores reportados durante el periodo de pruebas.

Errores de usuario final desencadenados exclusivamente por las acciones del usuario que consume los servicios de la solución, es decir, aquellos que se conectan al entorno de red protegido por RADIUS. Estos problemas pueden variar desde ingresos inadecuados de datos en formularios de registro y autenticación, fallas de hardware en los equipos clientes o hasta violaciones a los protocolos de seguridad establecidos.

Del gráfico se puede inferir que existe una relación proporcional entre el número de usuarios conectados simultáneamente al sistema con los que se reportan por eventos de usuario final, por tanto, es posible afirmar que a mayor carga concurrente, el sistema será más propenso a sufrir colapsos relacionados con fallas humanas.

I. Relación carga del sistema – errores

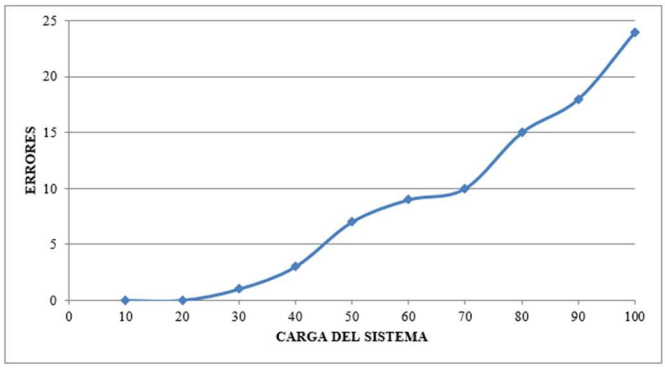

Fig. 20. Relación entre carga del sistema y errores.

En la ***Fig. 20*** se muestra el crecimiento de la aparición de errores, respecto al incremento de la carga del sistema respecto al uso del procesador y memoria. Los datos presentados son una ponderación de los diferentes factores de carga del sistema, contra una media aproximada de los casos de problemas reportados.

En esta gráfica se expresan los errores técnicos relacionados con la ejecución de procesos de software y rendimiento general de hardware. Tales errores pueden implicar desde colapsos por tareas incompletas hasta fallos críticos en el soporte físico de la solución.

Es posible identificar que, en cargas relativamente bajas, el sistema no sufre de problemas considerables. Sin embargo, durante picos de crecimiento en la carga, se presenta un crecimiento proporcional en el número de errores. De lo anterior, es posible deducir que el sistema alcanza su punto de equilibrio alrededor del 50% de la carga. Así mismo, se puede inferir, que aún en cargas extremas, no presenta un número significativo de errores.

VII. CONCLUSIONES

El logro de poder construir un prototipo de un Access Point Wi-Fi, con el soporte de la arquitectura ARM en cuanto a procesador se refiere, un protocolo de autenticación y autorización para aplicaciones de acceso a la red o movilidad IP (FreeRADIUS Server) y su integración con el desarrollo de cuatro (4) posibles métodos de puntos de acceso, permite reducir los costos de adquisición, implementación, sencillez, movilidad y gestión para proyectos y soluciones que requieran

similares características. Muy funcional para proveedores de Internet, sucursales empresariales y pequeños negocios con poca infraestructura tecnológica.

La diversidad en los diferentes métodos de acceso con características como la capacidad de despliegue de un formulario personalizable con el objeto de poder capturar datos del usuario, la capacidad de acreditar usuarios detrás de un sistema de autenticación digital descentralizado (OpenID), el acceso por medio de usuario y contraseña, y el no uso de credenciales, ayudan a reducir la inversión tecnológica, tanto en infraestructura como desarrollo de software de cualquier persona o compañía que requiera dos o más de estos servicios.

Usuarios potenciales del prototipo implementado son aquellos proveedores de servicios de Internet, cadena de restaurantes, centros comerciales, terminales de transporte aéreo y terrestre, clínicas/hospitales, centros educativos y cualquier compañía que tenga sucursales con poca infraestructura tecnológica. Siendo su principal uso el acceso al servicio de Internet y/o red corporativa, captura de datos, publicidad, seguridad en el proceso de autenticación y variedad en la fuente donde se alojan las credenciales de los usuarios (base de datos local, OpenID, LDAP, DA y ODBC).

El prototipo permite obtener diferentes beneficios que pueden ayudar a mejorar la seguridad en redes inalámbricas Wi-Fi y el buen uso de los recursos informáticos adquiridos a operadores de Internet. Esta característica se ve reflejada en el mismo momento que es reemplazado los procedimientos de protección como WEP, WPA-Personal o WPA2-Personal, por el uso de portales cautivos con métodos de autenticación soportados en 802.1X. La autenticación 802.1X usa un servidor de autenticación para validar las credenciales de los usuarios y proporcionar acceso a la red.

Otra de las bondades del proyecto realizado es la versatilidad de poder extender los servicios de una compañía que requiera capturar datos por medio de formularios web de forma fácil y sencilla. Para este caso, aplicarían clínicas (registro de salida de los pacientes), centros comerciales (captura de usuarios frecuentes para aplicación de servicios de Crossmedia), empresas (acceso invitados), terminales aéreas (ubicación de información de vuelos, servicios de acceso a la Internet, mapa del sitio, entre otros) y demás escenarios que requieran desplegar o capturar información de sus usuarios.

Gracias a los diferentes tipos de atributos soportados por FreeRADIUS, este servicio nos permite crear perfiles de clientes personalizados con características como: Control de ancho banda, VLAN (red de área local virtual), calidad de servicio, mensajes de bienvenida, entre otros. La diferenciación de servicio, ayuda a maximizar el buen uso de los recursos informáticos, la percepción del servicio por parte del usuario y el fortalecimiento de la relación cliente – empresa.

Para terminar, la pluralidad de operadores de Internet, la necesidad de usar los canales entregados por estos operadores de manera local, el requerimiento de despliegue de portales cautivos, la adaptación de los planes de datos a los usuarios del sistema (atributos FreeRADIUS), la difusión de contenidos publicitarios de manera personalizada por cada punto de acceso y la reducción en la cantidad de equipos electrónicos por sitio, obliga de alguna manera la creación de un dispositivo con las características del prototipo diseñado e implementado en este proyecto.

REFERENCIAS

[1] H. X. H. Xia and J. Brustoloni, “Virtual prepaid tokens for Wi-Fi hotspot access,” *29th Annu. IEEE Int. Conf. Local Comput. Networks*, 2004.

[2] A. Stone, “For-Fee Hot Spots Strive to Make Wi-Fi Pay,” *Pervasive Comput. IEEE*, 2003.

[3] “20 Popular Wireless Hacking Tools,” 2016. [Online]. Available: http://resources.infosecinstitute.com/20-popular-wireless-hacking-tools-updated-for-2016/. [Accessed: 03-Apr-2016].

[4] M. Rouse, “Captive Portal.” [Online]. Available: http://searchmobilecomputing.techtarget.com/definition/captive-portal. [Accessed: 14-Nov-2016].

[5] X. Ding and J. Wei, “A scheme for confidentiality protection of OpenID authentication mechanism,” in *Proceedings - 2010 International Conference on Computational Intelligence and Security, CIS 2010*, 2010, pp. 310–314.

[6] M. Rouse, “Authentication, Authorization, and Accounting (AAA).” [Online]. Available: http://searchsecurity.techtarget.com/definition/authentication-authorization-and-accounting. [Accessed: 14-Nov-2016].

[7] “Seguridad avanzada en redes Wireless 802.1X.” [Online]. Available: http://www.jacksecurity.com/files/publications/Jack42.pdf. [Accessed: 03-Apr-2016].

[8] Internet Engineering Task Force (IETF), “RFC2865 - Remote Authentication Dial In User Service (RADIUS),” 2000.