ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE META DISTRIBUCIÓN DEL SISTEMA OPERATIVO LINUX BAJO LICENCIA GPL ORIENTADA AL SERVICIO AAA (RADIUS) INTEGRANDO UN MÓDULO DE ADMINISTRACIÓN WEB

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN

CALDERÓN SUASNAVAS EVELYN ANDREA
eve29online@hotmail.com

LÓPEZ RAMÍREZ MARIO DAVID
md_nano@hotmail.es

DIRECTOR: ING. XAVIER CALDERÓN HINOJOSA, M.Sc
xavier.calderon@epn.edu.ec

Quito, Octubre 2011

DECLARACIÓN

Nosotros, Evelyn Andrea Calderón Suasnavas y Mario David López Ramírez, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

_____ _____

Evelyn Andrea Calderón Suasnavas Mario David López Ramírez

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Evelyn Andrea Calderón Suasnavas y Mario David López Ramírez, bajo mi supervisión.

_____

ING. XAVIER CALDERÓN, MSC.
DIRECTOR DE PROYECTO

DEDICATORIA

El momento en que el ser humano culmina una meta, es cuando se detiene a hacer un recuento de todas las ayudas recibidas, de las voces de aliento, de las expresiones de amor y comprensión; es por eso que dedico este triunfo a mis padres quienes con su infinita adhesión me han brindado todo su apoyo sin escatimar sacrificio alguno. A mis hermanos, porque han sido los amigos, los compañeros fieles en el camino hasta aquí recorrido.

A Dios por guiar mis pasos y ayudarme a superar los obstáculos que se me presentaron a lo largo del camino; y, por último quiero dedicar este logro a todos mis amigos testigos de mis triunfos y fracasos.

David L.

A mi familia, que son mi pilar y mi fuerza.

Evelyn C.

AGRADECIMIENTO

Hago llegar mi profundo agradecimiento primero a mi familia, a mi mami Hilda por enseñarme que en esta vida se debe luchar por lo que se quiere sin importar edad o género, a mi papa Mario por estar ahí siempre que lo he necesitado, a mis hermanos Mayra y Diego que me han apoyado en mis locuras y han sido participes de todas mis experiencias.

A mi compañera y amiga Evelyn que supo apoyarme en este proyecto y en muchas eventualidades de nuestra vida politécnica.

A mis amigos, a Jorge Luis un gran amigo del colegio que me ayudó mucho en esa etapa de mi vida, a Ramiro, Eduardo, David, Dianita y Oscar amigos de la universidad y camaradas de aventuras.

A mis amigos y compañeros del trabajo, a Andrés que al leer esto estará diciendo ¨¿Es muy larga tu historia?¨, a Verito, Luis y Santiago por brindarme su amistad y ayuda con este proyecto.

Al Master Xavier Calderón, Director de Proyecto de Titulación, quien nos orientó y guió en este proyecto que sella y da cuenta de un testimonio de trabajo, entrega y voluntad.

Y por último al Rojo (Gonzomovil) por ser un ícono en nuestra vida universitaria, por haberme llevado 5 años a la politécnica.

David L.

.

Primero quiero aclarar que la idea de este proyecto no fue mía, me la prestaron. Así que antes de nada, Deivid, alias Mario, (o viceversa) muchas gracias por compartir este proyecto conmigo, por invitarme a desarrollarlo y por tomar en cuenta mis ideas para mejorarlo.

Quiero agradecer a mi mami Olgui, por su terquedad y ternura al negarse a abandonarme en las tristes amanecidas del colegio y de la U. A mi papi Diego por sus sabios consejos y explicaciones precisas cuando más las necesitaba; a mis hermanas, Gaby y Vane, por su amistad y por motivarme a trabajar con más ahínco cuando de vez en cuando preguntaban "¿Y ya terminas?". Y claro, a mi Ferchito, por nunca haberme hecho esa misma pregunta cuando este proyecto consumió todo mi tiempo.

Gracias a las amigas del colegio, Natyta yo sé que todo cambia pero hemos trascendido y así será siempre. Gracias a los amigos de la U, por las horas de estudio compartidas, por los chistes privados (ejemplo: los ositos) y gracias Gatito, porque lo que se hace se paga.

Finalmente, quiero agradecer al Ingeniero Xavier Calderón por su guía durante el desarrollo de este proyecto y a mis compañeros (en realidad amigos) de trabajo, en especial a Luis por presentarnos en la China y a Verito por su invaluable amistad.

Evelyn C.

CONTENIDO

CONTENIDO I
ÍNDICE DE FIGURAS V
ÍNDICE DE TABLAS XI
ÍNDICE DE ESPACIO DE COMANDOS XII
RESUMEN XIV
PRESENTACIÓN XV
CAPÍTULO 1. Fundamentos Teóricos 1
1.1. Servicios AAA 1
1.1.1. Arquitectura AAA 2
1.1.1.1. Componentes Básicos 2
1.1.2. Funcionamiento de la Arquitectura 4
1.1.3. Autenticación 5
1.1.4. Autorización 7
1.1.4.1. Prácticas de Autorización en una Red AAA 7
1.1.5. Auditoría o Registro 8
1.2. Técnicas de Protección de la Información 9
1.2.1. Criptografía 9
1.2.1.1. Algoritmos de Cifrado Simétricos 9
1.2.1.2. Algoritmos de Cifrado Asimétricos 10
1.2.2. Certificados Digitales 11
1.2.2.1. Formato de los Certificados Digitales 13
1.3. Protocolos de Autenticación 14
1.3.1. PAP 14
1.3.2. CHAP 15
1.3.3. EAP 16
1.3.3.1. Formato del Paquete EAP 17
1.3.3.2. Proceso de Autenticación EAP 19
1.3.3.3. Métodos EAP 20
1.4. RADIUS 25
1.4.1. Generalidades 26
1.4.2. Formato del Paquete 26
1.4.2.1. Código 27
1.4.2.2. Identificador 28
1.4.2.3. Longitud 28
1.4.2.4. Autenticador 28
1.4.2.5. Atributos 28
1.4.3. Tipos de Paquete 30
1.4.3.1. Solicitud de Acceso (Access-Request) 30
1.4.3.2. Aprobación de Acceso (Access-Accept) 32
1.4.3.3. Denegación de Acceso (Access-Reject) 33
1.4.3.4. Desafío de Acceso (Access-Challenge) 33
1.4.3.5. Solicitud de Registro (Accounting-Request) 34

1.4.3.6. Respuesta de Registro (Accounting-Response) 36
1.4.4. Operación del Protocolo RADIUS 36
1.4.4.1. Autenticación y Autorización 36
1.4.4.2. Auditoría o Registro 38
1.5. Otros protocolos AAA 39
1.5.1. TACACS+ 39
1.5.2. DIAMETER 40
1.6. El Estándar 802.1x 41
1.6.1. Elementos de 802.1x 42
1.7. Meta Distribuciones Linux 43
1.7.1. Remastersys y el Concepto de Meta Distribución 43
1.7.2. Herramientas para la Generación de una Meta Distribución 44
1.7.2.1. InitramFS 44
1.7.2.2. Isolinux 46
1.7.2.3. UnionFS y AuFS 46
1.7.2.4. SquashFS 46
CAPÍTULO 2. Diseño e Implementación del Módulo de Administración Web 48
2.1. Generalidades 48
2.1.1. Lenguajes de Programación y Técnologías Utilizadas 49
2.1.1.1. HTTP 49
2.1.1.2. DOM 51
2.1.1.3. HTML 52
2.1.1.4. CSS 52
2.1.1.5. JavaScript 52
2.1.1.6. AJAX 52
2.1.1.7. PHP 54
2.2. Descripción de la Solución 54
2.3. Requerimientos de la Solución 56
2.3.1. Selección de la Distribución Base 56
2.3.2. Debian 58
2.3.3. FreeRADIUS 59
2.3.4. OpenSSL 59
2.3.4.1. La Licencia OpenSSL y GPL 60
2.3.5. MySqL 60
2.3.6. Samba 61
2.3.6.1. Winbind 61
2.3.6.2. Ntlm_auth 61
2.3.7. Kerberos 61
2.3.8. Cron 62
2.3.9. Apache 62
2.3.10. PhP5 62
2.3.11. CURL 63
2.4. Consideraciones del Diseño 63
2.4.1. Esquema Base de la Aplicación Web 63
2.4.1.1. Página HTML 64

2.4.1.2. Código JavaScript 64
2.4.1.3. Código PHP 65
2.4.1.4. Base de Datos 66
2.4.2. Autenticación Web y Perfiles de Usuario de la Aplicación 71
2.5. Secciones de la Aplicación Web 72
2.5.1. Configuración General 72
2.5.1.1. Información Básica 72
2.5.1.2. Recursos de Red 75
2.5.2. Administración de Certificados 87
2.5.2.1. Configuración Básica 87
2.5.2.2. Certificados 89
2.5.2.3. Protocolos de Autenticación 94
2.5.3. Configuración del Servidor 97
2.5.3.1. Clientes Radius (NAS) 97
2.5.3.2. Servicios 105
2.5.3.3. Usuarios Web 108
2.5.4. Credenciales 111
2.5.4.1. Usuarios Internos 111
2.5.4.2. Equipos Internos 119
2.5.4.3. Active Directory 128
2.5.4.4. LDAP 133
2.5.5. Políticas de Acceso 139
2.5.5.1. Reglas de Autenticación 139
2.5.5.2. Perfiles de Red 147
2.5.5.3. Reglas de Autorización 153
2.5.6. Visor de Eventos 161
2.5.6.1. Monitoreo 161
2.5.6.2. Resumen 162
2.5.6.3. Registro 165
2.5.6.4. Recuperación y Respaldos 168
CAPÍTULO 3. Implementación del Prototipo de Meta Distribución, Pruebas y Resultados……. 172
3.1. El prototipo de Meta Distribución “SAAAS!” 172
3.1.1. Descripción 172
3.1.1.1. Proyectos Similares 172
3.1.2. Requisitos de Hardware 175
3.1.3. Creación del Prototipo de Meta Distribución “SAAAS!” 175
3.1.3.1. Personalización de la Distribución Base 175
3.1.3.2. Creación del LiveCD 185
3.2. Pruebas y Resultados 194
3.2.1. Entorno de Prueba 194
3.2.1.1. Topología utilizada 194
3.2.1.2. Configuración de los Dispositivos Involucrados 196
3.2.2. Escenario I: Esquema AAA utilizando la base de datos local de “SAAAS!” como repositorio de credenciales 207

3.2.2.1. Objetivo 207
3.2.2.2. Alcance 207
3.2.2.3. Análisis de Resultados 208
3.2.3. Escenario II: Esquema AAA utllizando un repositorio de credenciales externo 216
3.2.3.1. Objetivo 216
3.2.3.2. Alcance 216
3.2.3.3. Análisis de Resultados 216
CAPÍTULO 4. Conclusiones y Recomendaciones 227
4.1. Conclusiones 227
4.2. Recomendaciones 229
BIBLIOGRAFÍA 232

ÍNDICE DE FIGURAS

Figura 1.1 Arquitectura AAA 5
Figura 1.2 Proceso General de Autorización 6
Figura 1.3 Cifrado Simétrico 10
Figura 1.4 Cifrado Asimétrico 11
Figura 1.5 Certificado Digital 12
Figura 1.6 Formato de un Certificado Digital 14
Figura 1.7 Autenticación PAP 15
Figura 1.8 Autenticación CHAP 16
Figura 1.9 Formato del Paquete EAP (Success/Failure) 18
Figura 1.10 Formato del Paquete EAP (Request/Response) 18
Figura 1.11 Proceso de Autenticación EAP 20
Figura 1.12 Intercambio de Paquetes EAP-TTLS en ambiente AAA 23
Figura 1.13 Formato del Paquete EAP-TTLS 24
Figura 1.14 Formato del Paquete RADIUS 27
Figura 1.15 Formato del Campo Atributos del Paquete RADIUS 30
Figura 1.16 Paquete Access-Request 31
Figura 1.17 Formato del Paquete Access-Accept 32
Figura 1.18 Formato del Paquete Access-Reject 33
Figura 1.19 Formato del Paquete Access-Challenge 34
Figura 1.20 Formato del Paquete Accounting-Request 35
Figura 1.21 Formato del Paquete Accounting-Response 36
Figura 1.22 Operación de RADIUS con Autenticación Fallida 38
Figura 1.23 Operación de RADIUS con Autenticación Exitosa 39
Figura 1.24 Protocolos mencionados en el estándar 802.1x 42
Figura 1.25 Proceso de arranque de un Sistema Operativo. 45
Figura 2.1 Ejemplo de Intercambio HTTP Request-Response 50
Figura 2.2 Ejemplo de Modelo DOM aplicado en NetBeans IDE 6.8 51
Figura 2.3 Comparación del modelo tradicional de aplicación Web y el propuesto por AJAX 53
Figura 2.4 Logotipo de "SAAAS!" 55
Figura 2.5 Lógica de la Aplicación 64
Figura 2.6 Esquema estándar de Base de Datos de FreeRADIUS 67
Figura 2.7 Esquema de Base de Datos de "SAAAS!" 68
Figura 2.8 Tabla para Administración de Perfiles 71
Figura 2.9 Página Información Básica del servidor "SAAAS!" 73
Figura 2.10 Información del Sistema mostrada por PHPSysInfo 73
Figura 2.11 Lógica de la Sección Configuración General 74
Figura 2.12 Página Recursos de Red/Pestaña: General de "SAAAS!" 75
Figura 2.13 Página Recursos de Red/Pestaña: Interfaces de Red de "SAAAS!" 76
Figura 2.14 Página "Recursos de Red"/Pestaña: Rutas del servidor "SAAAS!" 76
Figura 2.15 Página Recursos de Red/Pestaña: Nueva Interfaz Virtual del servidor "SAAAS!" 77

Figura 2.16 Clase: manejodeIP utilizada en la página Recursos de Red 77
Figura 2.17 Clase: interfazAdmin utilizada en la página Recursos de Red 78
Figura 2.18 Clase: sistema utilizada en la página Recursos de Red..... 79
Figura 2.19 Lógica de la acción: Mostrar pestaña General 80
Figura 2.20 Lógica de la acción Cambiar nombre de host 80
Figura 2.21 Lógica de la acción: Cambiar Dominio 81
Figura 2.22 Lógica de la acción: Cambiar DNS Principal 81
Figura 2.23 Lógica de la acción: Cambiar DNS Secundario..... 81
Figura 2.24 Lógica de la acción: Mostrar información de Interfaces de Red 82
Figura 2.25 Lógica de la función: Configurar Direccionamiento Estático en una Interfaz de Red 83
Figura 2.26 Lógica de la función: Configurar direccionamiento dinámico en una Interfaz de Red 83
Figura 2.27 Lógica de las funciones: Activar/Desactivar una interfaz de red 84
Figura 2.28 Lógica de la acción: Mostrar Tabla de Rutas 85
Figura 2.29 Lógica de la función: Mostrar Formulario para Nueva Interfaz Virtual 86
Figura 2.30 Lógica de la sesión: "Crear nueva Interfaz Virtual" 86
Figura 2.31 Página Configuración Básica de la sección Administración de Certificados del servidor "SAAAS!" 87
Figura 2.32 Lógica de la acción: Mostrar Formulario para crear Autoridad Certificadora 88
Figura 2.33 Lógica de la acción: Crear Autoridad Certificadora en el Servidor "SAAAS!" 88
Figura 2.34 Página Certificados de la sección Administración de Certificados del servidor "SAAAS!" 89
Figura 2.35 Lógica de la acción por defecto 90
Figura 2.36 Lógica de la acción Nuevo 91
Figura 2.37 Lógica de la acción Enviar 91
Figura 2.38 Lógica de la acción Ver 92
Figura 2.39 Lógica de la acción Descargar 93
Figura 2.40 Lógica de la acción Revocar 93
Figura 2.41 Lógica de la acción Revocar 94
Figura 2.42 Página Protocolos de Autenticación de la Sección Administración de Certificados del servidor "SAAAS!" 95
Figura 2.43 Lógica de la acción por defecto 96
Figura 2.44 Lógica de la acción TLS 96
Figura 2.45 Lógica de la acción Aplicar 97
Figura 2.46 Página Clientes Radius del servidor "SAAAS!", pestaña Ingreso de Clientes 98
Figura 2.47 Página Clientes Radius del servidor "SAAAS!" pestaña Grupos de Clientes 98
Figura 2.48 Esquema general de la página Clientes Radius 99
Figura 2.49 Lógica de la acción por defecto 100
Figura 2.50 Lógica de la acción: Buscar 101
Figura 2.51 Lógica de la acción: Nuevo 102

Figura 2.52 Lógica de la acción: Aceptar 103
Figura 2.53 Lógica de la acción: Actualizar 104
Figura 2.54 Lógica de la acción: Eliminar 105
Figura 2.55 Página Servicios de la sección Configuración del Servidor 106
Figura 2.56 Lógica de la página "Servicios" del servidor "SAAAS!" 106
Figura 2.57 Lógica de las acciones Iniciar/Detener/Reiniciar servicios de la página Servicios del servidor "SAAAS!" 107
Figura 2.58 Página Usuarios Web de la sección Configuración del Servidor del Servidor "SAAAS!" 108
Figura 2.59 Página Usuarios Web, formulario Modificar Usuario del Servidor "SAAAS!" 108
Figura 2.60 Página Usuarios Web, formulario Nuevo Usuario del servidor "SAAAS!" 109
Figura 2.61 Lógica de la Página Usuarios Web en la sección Configuración del Servidor 109
Figura 2.62 Lógica de la acción: Nuevo usuario en la página Usuarios Web 110
Figura 2.63 Lógica de la acción: Modificar usuario en la página Usuarios Web 110
Figura 2.64 Lógica de la acción Eliminar usuario de la página Usuarios Web 111
Figura 2.65 Página Usuarios Internos del servidor "SAAAS!" pestaña Usuarios Internos 112
Figura 2.66 Página Usuarios Internos del servidor "SAAAS!" pestaña Grupos de Usuarios 112
Figura 2.67 Esquema general de la página Usuarios Internos 113
Figura 2.68 Lógica de la acción por defecto 114
Figura 2.69 Lógica de la acción: Buscar Usuario Interno. 115
Figura 2.70 Lógica de la acción: Nuevo 116
Figura 2.71 Lógica de la acción: Aceptar 117
Figura 2.72 Lógica de la acción: Actualizar 118
Figura 2.73 Lógica de la acción: Eliminar 119
Figura 2.74 Página Equipos Internos del servidor "SAAAS!", pestaña Ingreso de Equipos 120
Figura 2.75 Página Equipos Internos del servidor "SAAAS!" pestaña Grupos de Equipos 120
Figura 2.76 Esquema general de la página Equipos Internos 121
Figura 2.77 Lógica de la acción por defecto 122
Figura 2.78 Lógica de la acción: Buscar 123
Figura 2.79 Lógica de la acción: Nuevo 124
Figura 2.80 Lógica de la acción: Aceptar 125
Figura 2.81 Lógica de la acción: Actualizar 126
Figura 2.82 Lógica de la acción: Eliminar 127
Figura 2.83 Página Active Directory del servidor "SAAAS!" pestaña Configuración 128
Figura 2.84 Página Active Directory del servidor "SAAAS!" pestaña Grupos de Dominio 129
Figura 2.85 Página Active Directory del servidor "SAAAS!" pestaña Test Usuarios 129
Figura 2.86 Esquema general de la página Active Directory 130

Figura 2.87 Lógica de la acción por defecto 130
Figura 2.88 Lógica de la acción: Verificar 131
Figura 2.89 Lógica de la acción: Guardar 132
Figura 2.90 Lógica de la acción: Buscar 132
Figura 2.91 Lógica de la acción: Verificar Usuario 133
Figura 2.92 Página LDAP del servidor "SAAAS!", pestaña Configuración 134
Figura 2.93 Página LDAP del servidor "SAAAS!" pestaña Grupos de Dominio 134
Figura 2.94 Página LDAP del servidor "SAAAS!" pestaña Test Usuarios 135
Figura 2.95 Esquema general de la página LDAP 135
Figura 2.96 Lógica de la acción por defecto 136
Figura 2.97 Lógica de la acción: Verificar 137
Figura 2.98 Lógica de la acción: Guardar 138
Figura 2.99 Lógica de la acción: "Buscar" 138
Figura 2.100 Lógica de la acción: Verificar Usuario 139
Figura 2.101 Página Reglas de Autenticación del servidor "SAAAS!" pestaña Repositorios 140
Figura 2.102 Página Reglas de Autenticación del servidor "SAAAS!" pestaña Edición de Reglas. 140
Figura 2.103 Esquema general de la página Reglas de Autenticación. 141
Figura 2.104 Lógica de la acción por defecto 142
Figura 2.105 Lógica de la acción Seleccionar Regla 143
Figura 2.106 Lógica de la acción Aplicar 144
Figura 2.107 Lógica de la acción Nueva Regla 144
Figura 2.108 Lógica de la acción Agregar Condición 145
Figura 2.109 Lógica de la acción Eliminar 146
Figura 2.110 Lógica de la acción Modificar 146
Figura 2.111 Página Perfiles de Red del servidor "SAAAS!" pestaña Perfiles 147
Figura 2.112 Página Perfiles de Red del servidor "SAAAS!" 148
Figura 2.113 Esquema general de la página Perfiles de Red. 148
Figura 2.114 Lógica de la acción por defecto 149
Figura 2.115 Lógica de la acción Seleccionar Perfil 149
Figura 2.116 Lógica de la acción Aplicar 150
Figura 2.117 Lógica de la acción Nuevo Perfil 151
Figura 2.118 Lógica de la acción Agregar Atributo 151
Figura 2.119 Lógica de la acción Eliminar 152
Figura 2.120 Lógica de la acción Modificar 153
Figura 2.121 Lógica de la acción Modificar 153
Figura 2.122 Página Reglas de Autorización del servidor "SAAAS!" pestaña Perfiles 154
Figura 2.123 Página Reglas de Autorización del servidor "SAAAS!" pestaña Edición de Reglas. 154
Figura 2.124 Esquema general de la página Reglas de Autorización. 155
Figura 2.125 Lógica de la acción por defecto 156
Figura 2.126 Lógica de la acción Seleccionar Regla 157
Figura 2.127 Lógica de la acción Aplicar 158

Figura 2.128 Lógica de la acción Nueva Regla 158
Figura 2.129 Lógica de la acción Agregar Condición 159
Figura 2.130 Lógica de la acción Eliminar 160
Figura 2.131 Lógica de la acción Modificar 160
Figura 2.132 Página Eventos del Sistema de la Sección Visor de Eventos de "SAAAS" 161
Figura 2.133 Lógica de la Sección Eventos del Sistema 162
Figura 2.134 Página Resumen del Registro AAA de la sección Visor de Eventos de "SAAAS" 163
Figura 2.135 Lógica para generar gráfico Solicitudes de Autenticación vs. NAS de la página Resumen del Registro AAA de "SAAAS!" 164
Figura 2.136 Lógica para generar gráfico Sesiones Activas vs. NAS de la página Resumen del Registro AAA de "SAAAS!" 164
Figura 2.137 Lógica para generar gráfico Solicitudes de Autenticación vs. Usuarios de la página Resumen del Registro AAA de "SAAAS!" 165
Figura 2.138 Página Registro AAA de la sección Visor de Eventos de "SAAAS" ... 166
Figura 2.139 Lógica para la generación de reportes en la página Registro AAA de "SAAAS!" 167
Figura 2.140 Lógica de la acción Descargar Reporte de la página Registro AAA de "SAAAS!" 167
Figura 2.141 Lógica de la función Cerrar Sesión del Reporte Sesiones Abiertas de la página "Registro AAA" 168
Figura 2.142 Página Recuperación y Respaldos de la Sección Visor de Eventos de "SAAAS" 169
Figura 2.143 Lógica de la página Recuperación y Respaldos de "SAAAS!" 169
Figura 2.144 Lógica de las funciones Crear/Eliminar Respaldos de la página Recuperación y Respaldos de "SAAAS!" 170
Figura 2.145 Lógica de la función Restaurar Sistema de la página Recuperación y Respaldos de "SAAAS!" 171
Figura 3.1 Interfaz gráfica de PhpRADmin 173
Figura 3.2 Interfaz gráfica de Cisco Secure ACS 5.x 173
Figura 3.3 GRUB2 Modificado para la Meta Distribución "SAAAS!" 183
Figura 3.4 Gestor de Arranque del LiveCD "SAAAS!" con ISOLINUX 187
Figura 3.5 Diálogo de Instalación de la Meta Distribución "SAAAS!" 189
Figura 3.6 Topología de red utilizada durante las pruebas del servidor "SAAAS!".. 194
Figura 3.7 Configuración de Clientes en el Servidor "SAAAS!" para las pruebas de Laboratorio 196
Figura 3.8 Usuarios y Equipos del dominio *saaas.ad* utilizado para las pruebas de laboratorio 197
Figura 3.9 Detalle del usuario *consultor1* miembro del grupo Consultores del dominio *saaas.ad* utilizado para las pruebas de laboratorio 198
Figura 3.10 Entrada DNS de la Cisco WLC 199
Figura 3.11 Ejemplo de ámbito DHCP creado para las pruebas de laboratorio 200
Figura 3.12 Cuentas de Correo creadas en el servidor Zimbra 201
Figura 3.13 Listas de Correo creadas en el servidor Zimbra 201

Figura 3.14 Detalle de usuarios miembros de una lista de correos en el servidor Zimbra 201
Figura 3.15 Configuración de la WLC como cliente RADIUS 202
Figura 3.16 Configuración de las interfaces en la WLC del esquema de laboratorio 203
Figura 3.17 Detalle de la configuración de una interfaz en la WLC 203
Figura 3.18 Esquema Fase 1 208
Figura 3.19 Paquete EAPoL (Start) 208
Figura 3.20 Paquete EAPoL (Request Identity) 209
Figura 3.21 Esquema Fase 2 209
Figura 3.22 Paquete EAPoL (Response) 209
Figura 3.23 Paquete Radius Access-Request 210
Figura 3.24 Paquete Radius Access - Challenge 211
Figura 3.25 Paquete EAPoL Request PEAP 211
Figura 3.26 Esquema Fase 3 211
Figura 3.27 Paquete TLSv1 Client Hello 212
Figura 3.28 Paquete TLSv1 Encapsulado sobre Radius 212
Figura 3.29 Paquete TLSv1 Encapsulado sobre Radius 213
Figura 3.30 Paquete EAPoL TLSv1 Server Hello 213
Figura 3.31 Esquema Fase 4 214
Figura 3.32 Paquete TLSv1 Túnel Encriptado 214
Figura 3.33 Paquete Radius Access-Accept 215
Figura 3.34 Paquete EAP-Success 215
Figura 3.35 Esquema Fase 1 216
Figura 3.36 Paquete EAPoL (Start) 217
Figura 3.37 Paquete EAPoL (Request Identity) 217
Figura 3.38 Esquema Fase 2 218
Figura 3.39 Paquete EAPoL (Response) 218
Figura 3.40 Paquete Radius Access-Request 219
Figura 3.41 Paquete LDAP Search 220
Figura 3.42 Paquete Radius Access - Challenge 220
Figura 3.43 Paquete EAPoL Request PEAP 221
Figura 3.44 Esquema Fase 3 221
Figura 3.45 Paquete TLSv1 Client Hello 221
Figura 3.46 Paquetes TLSv1 Encapsulados sobre RADIUS 222
Figura 3.47 Paquete EAPoL TLSv1 Server Hello 223
Figura 3.48 Esquema Fase 4 223
Figura 3.49 Paquete TLSv1 Túnel Encriptado 224
Figura 3.50 Paquete Samba NTLM 225
Figura 3.51 Paquete Radius Access-Accept 225
Figura 3.52 Paquete EAP-Success 226

ÍNDICE DE TABLAS

Tabla 1.1 Tipo de Paquetes EAP 18
Tabla 1.2 Campo Código en Paquetes RADIUS 27
Tabla 1.3 Atributos RADIUS más comunes..... 29
Tabla 1.4 Comparación de Protocolos AAA 41
Tabla 2.1 Características Generales de las Distribuciones Linux más populares. 56
Tabla 2.2 Arquitecturas de Hardware soportadas por algunas Distribuciones Linux. 56
Tabla 2.3 Características Técnicas de las Distribuciones Linux más populares. 57
Tabla 2.4 Matriz de Decisión de la Distribución Base de "SAAAS!" 57
Tabla 2.5 Versiones de Debian GNU/Linux..... 58
Tabla 2.6 Perfiles de Usuario de la Aplicación Web..... 72
Tabla 3.1 Comparación del prototipo de "Meta Distribución SAAAS!" con soluciones similares 174
Tabla 3.2 Requerimientos de Hardware de Debian GNU/Linux 175
Tabla 3.3 Tabla de direccionamiento IP utilizado en el esquema de laboratorio..... 195

ÍNDICE DE ESPACIO DE COMANDOS

Espacio de Comandos 2.1 Cambios del archivo *dialup.conf* (nas_query)..... 69
Espacio de Comandos 2.2 Cambios del archivo *dialup.conf* (authorize_check_query) 69
Espacio de Comandos 2.3 Cambios del archivo *dialup.conf* (authorize_reply_query) 70
Espacio de Comandos 2.4 Cambios del archivo *dialup.conf* (group_membership_query) 70
Espacio de Comandos 2.5 Cambios del archivo *dialup.conf* (authorize_group_check_query)..... 71
Espacio de Comandos 2.6 Cambios del archivo *class.WebpageXML.inc.php*..... 74
Espacio de Comandos 3.1 Instalación de paquetes necesarios para el funcionamiento de "SAAAS!"..... 176
Espacio de Comandos 3.2 Cambios al archivo */etc/apache2/sites-enabled/ssl*..... 177
Espacio de Comandos 3.3 Cambios al archivo */etc/apache2/mods-enabled/ssl.conf* 178
Espacio de Comandos 3.4 Cambios al archivo */etc/apache2/sites-enabled/000-default* 178
Espacio de Comandos 3.5 Descarga del paquete FreeRADIUS 2.1.8..... 179
Espacio de Comandos 3.6 Descarga de paquetes necesarios para la creación del paquete .deb de FreeRADIUS..... 179
Espacio de Comandos 3.7 Modificaciones al archivo *freeradius-server-2.1.8/debian/rules*..... 180
Espacio de Comandos 3.8 Creación de paquete .deb para la instalación de FreeRADIUS 180
Espacio de Comandos 3.9 Instalación de paquetes necesarios para usar el comando "dpkg-buildpackage"..... 181
Espacio de Comandos 3.10 Modificaciones en el archivo *sudoers*..... 182
Espacio de Comandos 3.11 Configuración del archivo *crontab* 183
Espacio de Comandos 3.12 Fragmento modificado en el archivo */boot/grub/grub.cfg* 184
Espacio de Comandos 3.13 Contenido del archivo *theme.txt* 184
Espacio de Comandos 3.14 Configuración de ISOLINUX..... 185
Espacio de Comandos 3.15 Configuración de ISOLINUX..... 186
Espacio de Comandos 3.16 Configuración de ISOLINUX..... 187
Espacio de Comandos 3.17 Comando para la creación de la Meta Distribución utilizando REMASTERSYS. 187
Espacio de Comandos 3.18 Modificación del archivo */usr/bin/remastersys* 188
Espacio de Comandos 3.19 Comando para la instalación en disco de "SAAAS!" .. 189
Espacio de Comandos 3.20 Fragmento del script generador del diálogo de instalación "SAAAS!"..... 190

Espacio de Comandos 3.21 Fragmento del script para el particionamiento automático del disco durante la instalación de "SAAAS!" 191
Espacio de Comandos 3.22 Fragmento del script para el formateo del disco durante la instalación de "SAAAS!" 192
Espacio de Comandos 3.23 Fragmento del script para el montaje del sistema de archivos durante la instalación de "SAAAS!" 193
Espacio de Comandos 3.24 Fragmento del script para copiar el sistema de archivos durante la instalación de "SAAAS!" 193
Espacio de Comandos 3.25 Fragmento del script para la instalación del GRUB durante la instalación de "SAAAS!" 194
Espacio de Comandos 3.26 Configuración de un usuario local en el switch de pruebas de laboratorio 204
Espacio de Comandos 3.27 Configuración de VLANs en el switch de pruebas de laboratorio 205
Espacio de Comandos 3.28 Configuración del esquema AAA en el switch para pruebas de laboratorio 205
Espacio de Comandos 3.29 Configuración como cliente RADIUS en el switch de pruebas de laboratorio 206
Espacio de Comandos 3.30 Configuración de 802.1x en los puertos de acceso del switch para pruebas de laboratorio. 206
Espacio de Comandos 3.31 Configuración de los puertos para servidores y enlace troncal en el switch para pruebas de laboratorio 207

RESUMEN

El presente documento describe el diseño y creación de un prototipo de Meta Distribución Linux que facilita la implementación de servicios de autenticación, autorización y registro en una red de datos de pequeña y mediana escala, a través de una herramienta de administración Web para FreeRADIUS desarrollada en PHP y AJAX.

En el primer capítulo se estudia la arquitectura de los servicios AAA (Autenticación, Autorización y Auditoría), los protocolos de autenticación más utilizados, las técnicas de protección de información a través del cifrado y el uso de certificados digitales. Se incluye además un estudio del protocolo RADIUS y la descripción del proyecto Metadistros Linux.

El segundo capítulo detalla las herramientas y lenguajes de programación utilizados para la implementación del módulo de administración Web “SAAAS!”, los paquetes requeridos para su funcionamiento adecuado y las consideraciones del diseño de la aplicación. También se presenta un resumen del funcionamiento de cada una de las secciones de la aplicación Web a través de diagramas lógicos.

El tercer capítulo muestra el proceso de creación del prototipo de Meta Distribución “SAAAS!” y las pruebas y resultados obtenidos al utilizar el prototipo en un ambiente de laboratorio.

Finalmente, en el cuarto capítulo se presentan las principales conclusiones y recomendaciones obtenidas durante la elaboración del proyecto.

PRESENTACIÓN

En la actualidad, el uso de esquemas de seguridad 802.1X en redes de área local es de vital importancia. Por esta razón los administradores de red se ven en la necesidad de buscar herramientas que permitan una implementación sencilla y eficaz de este estándar. El protocolo RADIUS es comúnmente utilizado en esquemas de seguridad para redes de área local basados en procesos de autenticación, autorización y registro, por lo que se ha decidido crear una solución de software libre basada en Linux que simplifique la implementación de este protocolo.

El Servidor de Autenticación, Autorización y Auditoría Simplificado "SAAAS!", es una herramienta para redes 802.1X que facilita la administración y configuración de un servidor de autenticación que maneja el protocolo RADIUS. El prototipo presentado se encuentra enfocado al personal encargado de redes de pequeña y mediana escala con poca experiencia en administración de servidores Linux pero familiarizado con el estándar 802.1x y el funcionamiento del protocolo RADIUS.

El servidor "SAAAS!" admite la integración con varios repositorios de credenciales como Active Directory o LDAP, facilitando la integración con entornos de red especializados. El servidor "SAAAS!" implementa políticas de acceso aplicadas a grupos de usuarios como asignación dinámica de VLANs, lo que permite una gran escalabilidad a nivel de políticas de acceso a la red.

El servidor "SAAAS!" permite una administración segura del servidor AAA mediante la autenticación de usuarios con privilegios sobre la aplicación web, los cuales pueden pertenecer a un perfil para administración que permite modificar todos los parámetros del servidor, un perfil de operador con acceso limitado y un perfil de solo lectura, adicionalmente incluye un visor de eventos para monitoreo del servidor y resolución de problemas de falla de autenticación.

CAPÍTULO 1. FUNDAMENTOS TEÓRICOS

1.1. SERVICIOS AAA

El acrónimo AAA proviene de los términos en inglés Authentication (Autenticación), Authorization (Autorización), y Accounting (Registro) y se usa para referirse a un conjunto de herramientas, procedimientos y protocolos que garantizan un tratamiento coherente de las tareas ejecutadas para proporcionar acceso seguro a un sistema de información.

El proceso de Autenticación en un servidor AAA provee un método para garantizar que una identidad sea válida y corresponda a su propietario; es decir, durante este proceso el dispositivo o usuario que solicita acceso al sistema de información debe responder a la pregunta: "¿Quién o qué es?".

Cuando se ha completado el proceso de Autenticación, se da inicio al proceso de Autorización. La Autorización se usa para restringir, en función de un perfil preestablecido, las acciones que un usuario o dispositivo autenticado puede realizar dentro del sistema de información. Es decir, durante esta etapa se responde a la pregunta "¿Qué se le permite hacer?".

Finalmente, se lleva a cabo el proceso de Registro o Auditoría. Durante esta etapa el servidor almacena los eventos de autenticación y autorización de los usuarios o dispositivos que hayan intentado acceder a la red, para llevar a cabo un seguimiento que más tarde puede utilizarse en la administración de los recursos de red. En resumen, se responde a la pregunta "¿Qué hizo?".

En la actualidad, AAA juega un papel importante en casi todas las formas en las que se accede a una red. Su implementación se considera una buena práctica de seguridad ya que permite limitar el acceso de los usuarios remotos a recursos de la red basándose en su identidad.

No está claro cuando se comenzó a utilizar el término AAA, pero se puede decir que coincidió con el desarrollo de RADIUS (Servicio de Autenticación Remota para Usuario de Acceso Telefónico) que en la actualidad es el protocolo AAA más utilizado por lo que se lo describe ampliamente en este primer capítulo.

1.1.1. ARQUITECTURA AAA

La arquitectura AAA ofrece un marco de gestión común para autenticar, autorizar y medir el consumo de recursos durante la conexión de un usuario, mediante la monitorización y generación de informes.

1.1.1.1. Componentes Básicos

En todo entorno AAA aparece un conjunto de entidades y componentes funcionales que permiten la toma de decisiones, y su aplicación en base a atributos y políticas.

Todos estos componentes son procedimientos de operación y no es obligatorio que cada uno esté en un equipo dedicado, un solo equipo puede realizar varias acciones a la vez.

Las características de estos componentes se detallan a continuación.

1.1.1.1.1. *Usuario*

El usuario es el dispositivo que requiere ingresar a la red. Un usuario puede ser un computador personal o dispositivos móviles como una laptop o un PDA, un cliente VPN, etc.

1.1.1.1.2. *Punto de Ejecución de Políticas (PEP)*

El Punto de Ejecución de Políticas PEP (RFC 2904[1]), también conocido como AEF *Access Control Enforcement Function*/*Función de Ejecución de Control de Acceso* (ISO-10181-3[2]); es la entidad lógica responsable de aplicar los términos del acceso de un usuario al sistema de información. El dispositivo que realiza esta tarea dentro de una red de datos es el NAS, *Network Access Server*/*Servidor de Acceso a la Red*, que puede ser un Access Point, un conmutador, un firewall, un gateway o un concentrador VPN.

El Punto de Ejecución de Políticas no contiene información acerca de qué usuarios pueden conectarse o qué credenciales son válidas. Las credenciales suministradas por el cliente se envían a un recurso que sabrá cómo procesarlas.

1.1.1.1.3. *Punto de Información de Políticas (PIP)*

El Punto de Información de Políticas PIP (RFC 2904) es el repositorio de la información que puede ser usado durante la evaluación de una solicitud de acceso a la red. Este repositorio puede guardar atributos relacionados a la identidad del usuario y los recursos de red.

El PIP podría ser una base de datos de identificadores de usuarios, un directorio activo (Active Directory de Windows) o un servidor LDAP, *Lightweight Directory Access Protocol/Protocolo Liviano de Active Directory*.

1.1.1.1.4. *Punto de Decisión de Políticas (PDP)*

El Punto de Decisión de Políticas PDP (RFC 2904), también conocido como ADF *Access Control Decision Function/Función para la Decisión de Control de Acceso*,

---

[1] AAA Authorization Framework, J. Vollbrecht y Otros, Agosto del 2000
[2] Security frameworks for open systems: Access control framework, 1996

(ISO-10181-3), es la entidad lógica encargada de la toma de decisiones dentro de la arquitectura AAA.

El PDP recoge las solicitudes de acceso del usuario a través del PEP y la información requerida para tomar la decisión de acceso del PIP y es la entidad que toma la decisión final de la admisión o denegación del acceso a la red y los recursos de la misma. El PDP es el núcleo del servicio AAA.

1.1.1.1.5. *Sistema de Registro y Auditoría*

El Sistema de Registro y Auditoría almacena la información referente a la actividad que tienen los usuarios durante los procesos de autenticación y autorización y el uso de recursos dentro de la red.

Este sistema puede ser parte del PDP o ser un sistema dedicado, generalmente está conformado por una base de datos.

1.1.2. FUNCIONAMIENTO DE LA ARQUITECTURA

En este punto se detalla la interacción de los componentes básicos de la arquitectura AAA durante los procesos de autenticación, autorización y registro.

Cuando un usuario intenta conectarse a una red protegida mediante el servicio AAA, el dispositivo PEP le solicita información que lo identifique. El PEP envía la información de la identidad del usuario al PDP, el cual realiza una consulta a la PIP configurada para determinar si la información que envió el cliente es válida. La PIP responde con un mensaje de éxito o fracaso de la validación de credenciales y envía información adicional del cliente al PDP.

El PDP evalúa la información sobre el usuario obtenida a través del PEP y el PIP y toma una decisión de autenticación para crear un perfil de autorización para el

usuario. Este resultado se envía al Sistema de Registro y Auditoría y al PEP.

En base a la información recibida, el PEP envía un mensaje de autenticación exitosa o fallida al usuario y pone en práctica el perfil de autorización del mismo. Todos estos procesos se detallan en la Figura 1.1.

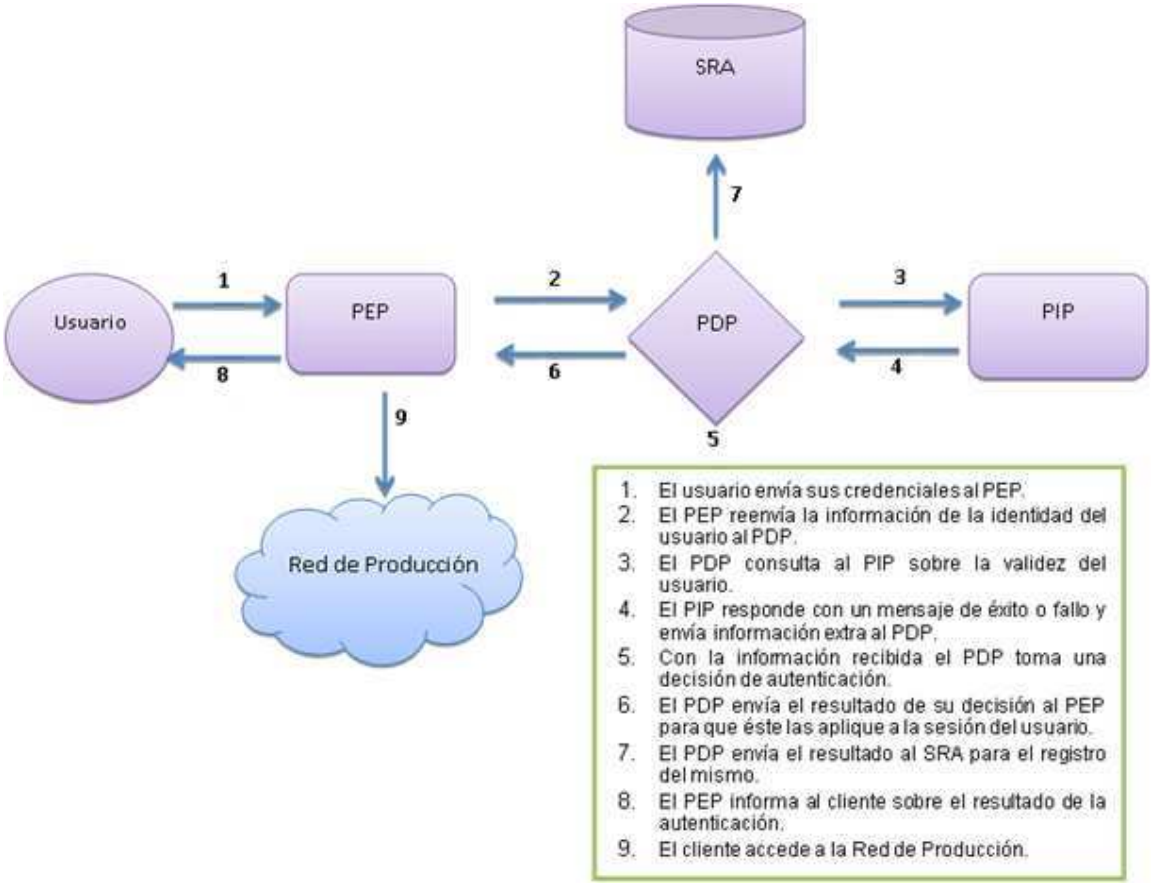


Figura 1.1 Arquitectura AAA[3]

1.1.3. AUTENTICACIÓN

En redes de información, se llama autenticación a la confirmación de que un usuario, que solicita los servicios de un sistema, es válido. La autenticación se realiza a través de la presentación de una identidad y credenciales.

---

[3] Imagen basada en la Figura 1: "A Client Connects to a AAA-Protected Network, tomada de: http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_10-1/101_aaa-part1.html

Los sistemas informáticos mantienen una relación de identidades personales (usuarios) asociadas normalmente con un perfil de seguridad. La autenticación permite a estos sistemas asumir con seguridad que el usuario o dispositivo que se está conectando a la red de información es quien dice ser. Una vez terminado el proceso de autenticación, las acciones que el usuario ejecuta en el sistema pueden ser referidas a la identidad que le corresponde y así, aplicar los mecanismos de autorización y auditoría oportunos.

El proceso general de autenticación se ilustra en la Figura 1.2 y consta de los siguientes pasos:

1) El usuario solicita acceso a un servicio.
2) El servidor solicita al usuario que se autentique.
3) El usuario aporta las credenciales que le identifican y permiten verificar la autenticidad de la identificación.
4) El servidor verifica según sus reglas si las credenciales aportadas son suficientes para dar acceso al usuario o no.

Figura 1.2 Proceso General de Autorización

La identidad del usuario se puede comprobar por diferentes tipos de credenciales entre estos están: contraseñas o secretos compartidos, Tokens, Certificados Digitales, Biométricas.

1.1.4. AUTORIZACIÓN

La autorización se refiere a la concesión de determinados servicios a un usuario basándose en su identidad, los servicios solicitados y el estado actual del sistema.

La autorización puede también estar basada en restricciones tales como el horario, localización física o el número de sesiones establecidas por el usuario. La autorización determina la naturaleza del servicio que se otorga.

1.1.4.1. Prácticas de Autorización en una Red AAA

La restricción de los servicios de red a un usuario se realiza mediante varias técnicas, entre ellas se nombra: filtrado de direcciones IP, asignación de direcciones, asignación de rutas, calidad de servicio, control de ancho de banda, gestión de tráfico y asignación de VLANs tanto en redes inalámbricas como cableadas.

A continuación se describen las prácticas más comunes para implementar autorización en redes AAA.

1.1.4.1.1. *Autorización Nula*

La autorización nula se caracteriza porque después de la autenticación se concede al usuario acceso total a la red. En la actualidad es la práctica más común en ambientes AAA, sin embargo cada vez más, se ve la necesidad de restringir los derechos de los usuarios según su rol en la organización.

1.1.4.1.2. *Segmentación de Capa Enlace*

La segmentación de capa enlace es la forma más común de autorización para puntos de acceso inalámbricos y conmutadores Ethernet. Esta segmentación se consigue dividiendo la red en múltiples segmentos lógicos, separando cierta clase de clientes de otros a través de VLANs[4].

1.1.4.1.3. *Filtrado en Capa Red*

La autorización mediante filtrado de capa red se logra aplicando listas de acceso ACLs. Las listas de acceso generalmente funcionan a nivel de capa red y capa transporte permitiendo autorizar o denegar servicios a equipos en particular basándose en su dirección IP.

1.1.5. AUDITORÍA O REGISTRO

La Auditoría se refiere al seguimiento del consumo de los recursos de red. Durante esta etapa se almacena la información referente a que usuarios accedieron a la red, que servicios utilizaron y cuando se desconectaron de la misma. Esta información puede usarse posteriormente para la administración, planificación, facturación, u otros propósitos.

La Auditoría en tiempo real es la información generada al mismo tiempo que se produce el consumo de los recursos.

La Auditoría por lotes es la información que se guarda hasta que se entrega en un momento posterior.

---

[4] "Virtual Local Area Network", Red de Área Local Virtual.

La información típica que un proceso de auditoría registra es la identidad del usuario, el tipo de servicio que se le proporciona, cuando comenzó a usarlo, y cuando terminó.

1.2. TÉCNICAS DE PROTECCIÓN DE LA INFORMACIÓN

1.2.1. CRIPTOGRAFÍA

Se entiende como criptografía a la técnica que permite transformar un mensaje legible a uno que pueda ser entendido tan solo por personas autorizadas. En sistemas de información, el método o sistema empleado para cifrar el texto claro se denomina *algoritmo de cifrado.*

El emisor y el receptor de un mensaje protegido por medios criptográficos, deben utilizar el mismo *algoritmo de cifrado* y poseer la *clave* adecuada para cifrar o descifrar el mensaje.

1.2.1.1. Algoritmos de Cifrado Simétricos

Los algoritmos de cifrado simétrico son aquellos que permiten cifrar y descifrar un texto con la misma clave por lo que su seguridad se basa en el secreto de la misma. La Figura 1.3 ilustra el funcionamiento básico de este tipo de algoritmos.

Los algoritmos de cifrado simétrico más usados son DES (Estándar de Cifrado de Datos), 3-DES (Triple DES) y AES (Estándar Avanzado de Cifrado).

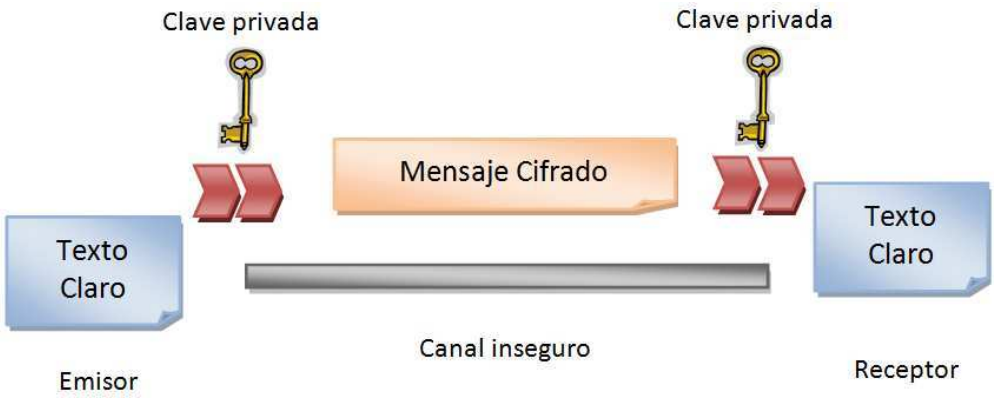


Figura 1.3 Cifrado Simétrico

1.2.1.2. Algoritmos de Cifrado Asimétricos

Los algoritmos de cifrado asimétrico utilizan dos claves diferentes para cifrar y descifrar un mensaje. El principio básico de este tipo de algoritmos es que si se cifra un mensaje con una de las claves solo puede descifrarse con la otra. La Figura 1.4 ilustra el funcionamiento básico de este tipo de algoritmos.

El par de claves utilizadas en el cifrado asimétrico están relacionadas matemáticamente y no se puede obtener una a través de la otra; una de ellas es conocida como clave pública y otra como clave privada.

Los algoritmos de cifrado asimétrico más difundidos son el RSA (Rivest, Shamir, Adelman) y DSA (Digital Signature Algorithm).

Figura 1.4 Cifrado Asimétrico

1.2.2. CERTIFICADOS DIGITALES

En el cifrado asimétrico se suelen utilizar certificados digitales emitidos por entidades de confianza para la transmisión de la clave pública.

"Un certificado digital es un documento electrónico que contiene datos que identifican a una persona o entidad y la clave pública de la misma, haciéndose responsable de la autenticidad de los datos que figuran en el certificado otra persona o entidad de confianza denominada Autoridad Certificadora"[5].

Una Autoridad Certificadora vincula a una clave pública con una entidad mediante el uso de firmas digitales. Se dice que un usuario firma un mensaje de datos cuando aplica una función hash[6] al mismo y cifra el resultado con su clave privada garantizando de esta manera la autenticación e integridad del mensaje.

---

[5] "Estudio Científico de las Redes de Ordenadores", Ángel Cobo Yera, pág 172.
[6] Una función hash es un algoritmo matemático utilizado para verificar la integridad de un mensaje. Mediante una función hash se obtiene un bloque de longitud fija del cual es imposible recuperar el mensaje original.

Un certificado digital provee un mecanismo que permite tener una mayor resistencia a la suplantación, pues a través de él se asegura que la clave pública que envía una entidad sea legítima.

Las recomendaciones para la generación de certificados digitales se encuentran registradas en la norma UIT-T X.509[7]. La Figura 1.5 muestra un ejemplo de certificado digital emitido bajo esta norma.

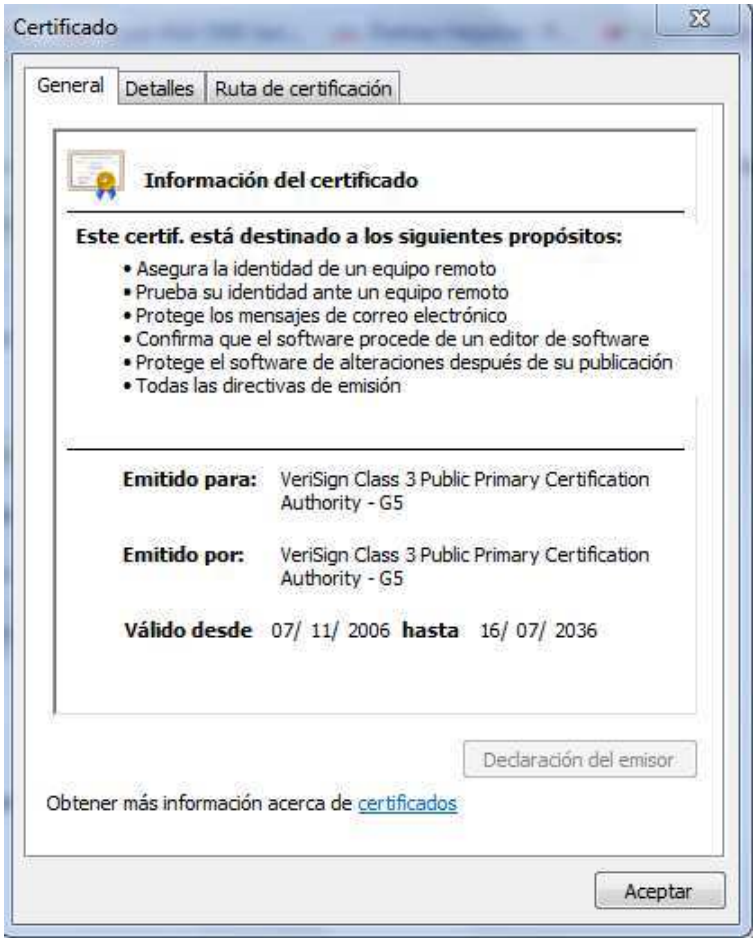


Figura 1.5 Certificado Digital

[7] Unión Internacional de Telecomunicaciones: “The Directory: Public-key and attribute certificate frameworks”.

1.2.2.1. Formato de los Certificados Digitales

Según el estándar X.509, un certificado digital debe incluir la siguiente información:

- *Versión:* Indica la versión de la norma X.509 bajo la cual se creó el certificado.
- *Número de Serie:* Es un identificador único que asigna una Autoridad Certificadora a cada certificado que emite.
- *Algoritmo de Firma*: El algoritmo utilizado para crear la firma.
- *Emisor*: El nombre de la entidad que verificó la información y emitió el certificado.
- *Válido desde y Válido hasta*: Periodo de validez del certificado.
- *Nombre* del Sujeto: Al que pertenece la clave pública certificada.
- *Clave Pública*: La clave pública del usuario junto a un identificador del algoritmo de cifrado utilizado.
- *Extensión:* Para agregar información adicional manteniendo el formato del certificado.
- *Firma:* Hash de los campos anteriores cifrado con la clave privada de la Autoridad Certificadora.

En la Figura 1.6 se puede observar parte de la información que incluye un certificado emitido bajo el estándar X.509.

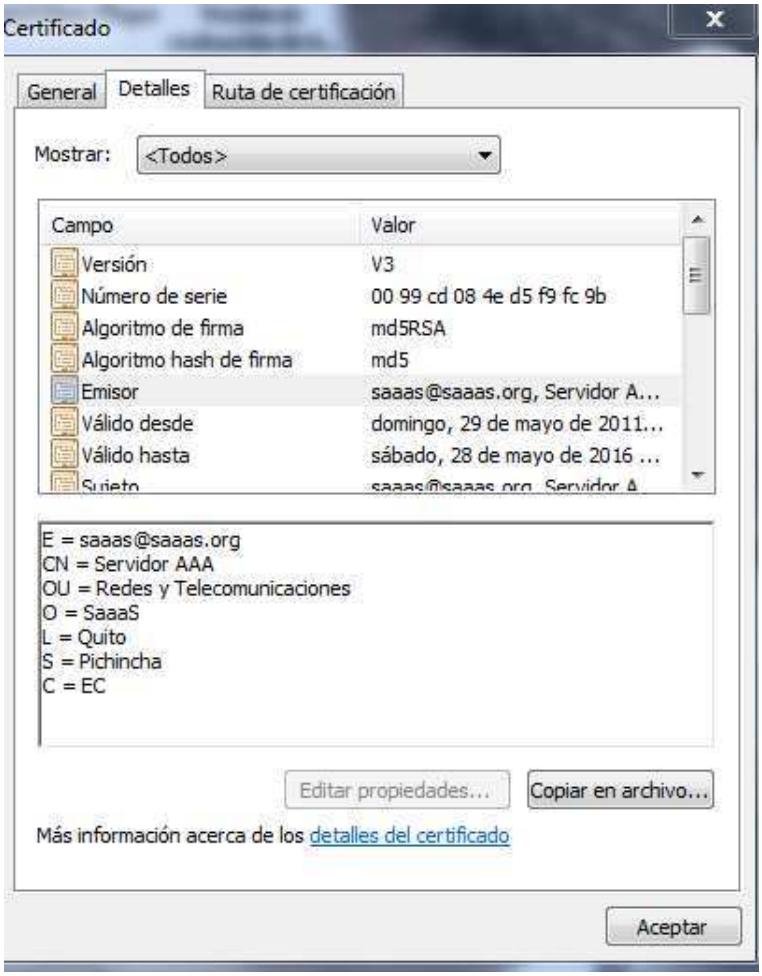


Figura 1.6 Formato de un Certificado Digital

1.3. PROTOCOLOS DE AUTENTICACIÓN

1.3.1. PAP

El Protocolo de Autenticación por Contraseña PAP (Password Authentication Protocol); especificado en el RFC 1334[8], provee un método simple de autenticación entre pares utilizando un intercambio de mensajes de dos vías durante el cual el usuario transmite su nombre y una palabra clave.

[8] "Protocolos de Autenticación PPP", B. Lloyd y W. Simpson, Octubre de 1992

Una vez terminado el establecimiento del enlace, configurado con autenticación PAP, el usuario envía sus credenciales en texto claro al dispositivo autentificador hasta que se realiza la autenticación o se termina la conexión. Este procedimiento se detalla en la Figura 1.7.

PAP es el método de autenticación más inseguro que existe por lo que su uso no es muy común, a pesar de esto se lo sigue utilizando en ciertos dispositivos como portales cautivos y en la autenticación para la administración de los equipos de red.

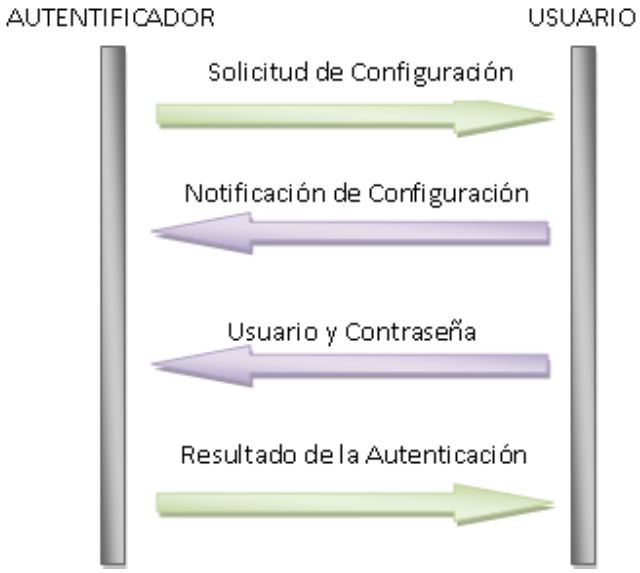


Figura 1.7 Autenticación PAP

1.3.2. CHAP

El Protocolo de Autenticación por Desafío CHAP (Challenge Handshake Authentication Protocol), definido en el RFC 1994[9], provee un método de autenticación entre pares utilizando un intercambio de mensajes de tres vías.

Una vez completada la fase de establecimiento de la conexión, el autentificador

[9] PPP Challenge Handshake Authentication Protocol (CHAP), W. Simpson, Agosto de 1996

envía un mensaje de desafío *challenge* al usuario, el cual consiste en un número aleatorio. El usuario responde al desafío recibido con el resultado de una operación HASH MD5 entre el número aleatorio recibido y el secreto compartido entre el autentificador y el usuario. Localmente, el autentificador realiza la misma operación y compara este resultado con el mensaje recibido, si ambos son iguales se permite el acceso; de lo contrario se rechaza.

El proceso de autenticación ocurre periódicamente en el transcurso de la conexión y se puede observar en la Figura 1.8.

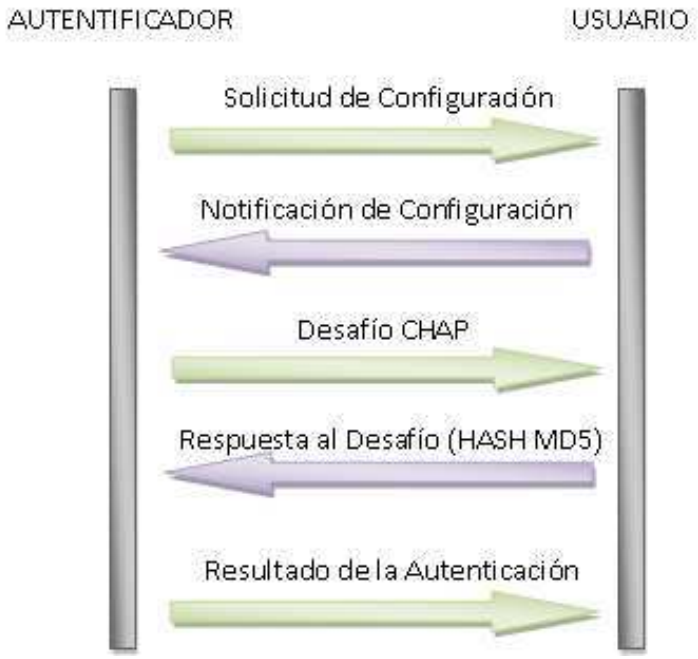


Figura 1.8 Autenticación CHAP

1.3.3. EAP

El Protocolo de Autenticación Extensible EAP definido en el RFC 3748[10] es una plataforma de autenticación frecuentemente utilizada en ambientes LAN inalámbricos que requieren procesos AAA, aunque su uso es cada vez más común en redes de área local cableadas.

---

[10]"Extensible Authentication Protocol (EAP)",B. Aboba y Otros, Junio 2004

EAP es una arquitectura de soporte que solo provee transporte confiable para diferentes métodos de autenticación y no un método de autenticación específico; sus principales características son:

- Funciona directamente sobre protocolos de capa dos como PPP o Ethernet sin requerir IP.
- Soporta un sólo paquete en transmisión por lo que no puede transportar eficientemente datos robustos, a diferencia de protocolos de capas superiores como TCP.
- Provee soporte para retransmisión pero no soporta el control de orden de recepción de tramas, esta tarea es delegada a las capas inferiores.
- No soporta fragmentación y re-ensamblaje, por lo que los métodos de autenticación con carga útil mayor al MTU[11] mínimo de EAP deben proveer soporte para la fragmentación. Por ejemplo, cuando se utiliza autentificación en base de certificados, si el certificado es más grande que el MTU de EAP el número de round-trips (viajes redondos de paquetes) entre cliente y servidor puede aumentar debido a la necesidad de fragmentar dicho certificado. Si se experimenta una significante pérdida de paquetes, se requerirán muchos round-trips y se reflejará en dificultades de conexión.
- En redes inalámbricas, los métodos modernos de EAP pueden proporcionar un mecanismo de autenticación seguro y negociar una PMK (Pair-wise Master Key) entre el usuario y el NAS. La llave primaria se utiliza para la sesión del cifrado inalámbrico con TKIP o AES.

1.3.3.1. Formato del Paquete EAP

El paquete EAP está dividido en cuatro campos. El primero es el campo Código que es un octeto que identifica el tipo de paquete EAP. Existen cuatro tipos de paquetes

[11] "Maximum Transmission Unit" es el tamaño máximo en bytes de la unidad de datos que puede transmitirse.

EAP, detallados en la Tabla 1.1.

Código	Valor Binario	Tipo de Paquete
1	0000001	Solicitud (Request)
2	0000010	Respuesta (Response)
3	0000011	Éxito (Success)
4	0000100	Falla (Failure)

Tabla 1.1 Tipo de Paquetes EAP

Al campo código le sigue el campo Identificador, que es un octeto que permite emparejar los paquetes de Respuesta con los de Solicitud durante el intercambio de mensajes EAP. Al Identificador le siguen dos octetos que indican la longitud del paquete y finalmente el campo Datos.

Para paquetes del tipo Request o Response el campo Datos se subdivide en Tipo y Datos del Tipo. En el octeto que conforma el campo Tipo se define el método de autenticación que se utilizará en el intercambio EAP.

El formato de los paquetes EAP de éxito y falla se ilustra en la Figura 1.9 y el formato de los paquetes de solicitud y respuesta en la Figura 1.10.

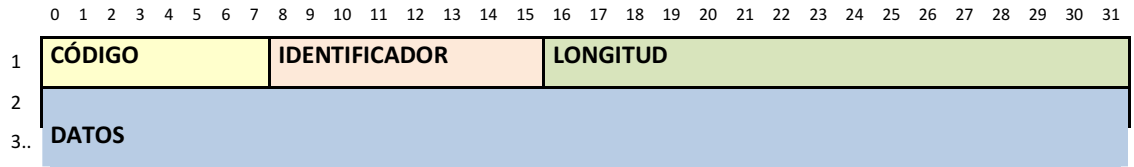


Figura 1.9 Formato del Paquete EAP (Success/Failure)

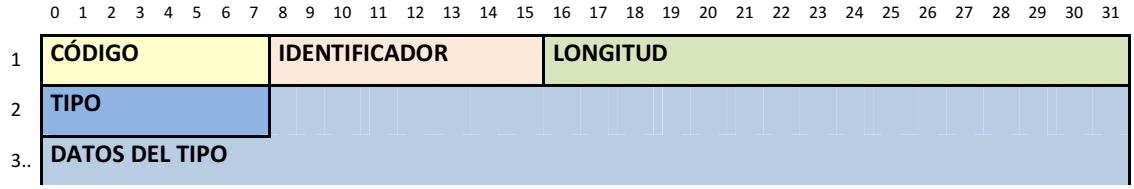


Figura 1.10 Formato del Paquete EAP (Request/Response)

1.3.3.2. Proceso de Autenticación EAP

El proceso de Autenticación EAP se resume en la Figura 1.11 y es el siguiente:

1) El servidor envía un paquete de EAP-Request hacia el cliente. No es obligatorio que el servidor envié el EAP-Request y este proceso puede pasarse por alto cuando se obtienen las contraseñas de otras formas; por ejemplo, mediante MAB[12] *Mac Address Bypass/Redirección de Direcciones MAC*.
2) El cliente envía un paquete EAP-Response en contestación al EAP-Request.
3) El servidor envía un paquete adicional EAP-Request y el cliente responde con otro paquete EAP-Response, este proceso se repite cuantas veces sea necesario, EAP es un protocolo de tipo bloqueante por lo que una vez enviado el paquete EAP-Request, se espera recibir un paquete EAP-Response, si éste no es enviado se procede a la retransmisión del paquete EAP-Request, después de un numero de retransmisiones sin respuesta el servidor da por terminada la sesión EAP. El servidor no puede enviar paquetes EAP-Success o EAP- Failure cuando está retransmitiendo.
4) El intercambio de mensajes continua hasta que se obtiene un resultado de conformidad o de rechazo en la autenticación entonces se enviarán los paquetes EAP-Success o EAP- Failure según corresponda.

[12] Mac Address Bypass utiliza para la autenticación la dirección MAC del equipo como nombre de usuario y contraseña.

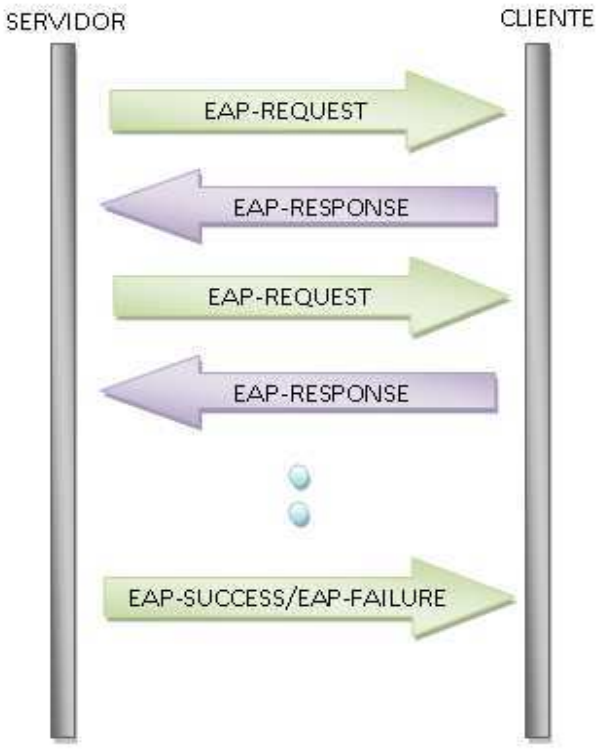


Figura 1.11 Proceso de Autenticación EAP

1.3.3.3. Métodos EAP

EAP no es un *mecanismo* de autenticación sino una *plataforma* de autenticación ya que define funciones comunes y procedimientos de negociación para diferentes técnicas de autenticación. Estas técnicas de autenticación son llamadas Métodos EAP y se han definido 40 diferentes; entre los principales se encuentran: EAP-MD5, EAP-OTP, EAP-GTC, EAP-TLS, EAP-IKEv2, EAP-SIM, EAP-AKA, PEAP, LEAP y EAP-TTLS.

1.3.3.3.1. *EAP-MD5*

EAP-MD5, *Message-Digest Algorithm 5/Algoritmo de Resumen del Mensaje 5,* está definido en el RFC 3748. Es un estándar abierto y ofrece una seguridad mínima. MD5 es vulnerable a ataques de diccionario y no soporta la autenticación mutua o de

generación de claves, lo cual lo hace inadecuado para su uso con WEP y WPA/WPA2.

1.3.3.3.2. *EAP-TLS*

EAP-TLS, *Transport Layer Security/Seguridad en Capa Transporte*[13], es un estándar abierto definido en el RFC 2716[14], que ofrece un método de autenticación mutua.

Utiliza una infraestructura de clave pública (PKI[15]) para asegurar la comunicación con el servidor de autenticación RADIUS. Proporciona una excelente seguridad, sin embargo, la generación e instalación de los certificados de cliente puede hacer que parezca desalentador de implementar.

Es considerado uno de los estándares EAP más seguro disponible y es apoyado por todos los fabricantes de hardware LAN inalámbrica y software, incluyendo Microsoft. El nivel de seguridad de EAP-TLS se basa en el requerimiento de un certificado de cliente. Cuando el certificado de cliente se encuentra en una tarjeta inteligente[16], se ofrece la mayor seguridad disponible porque no hay manera de robar el certificado de una tarjeta inteligente sin robar la misma.

1.3.3.3.3. *EAP-TTLS*

EAP-TTLS, *Tunneled Transport Layer Security/Seguridad en Capa Transporte sobre Túnel,* es un método EAP que proporciona funcionalidades adicionales a EAP-TLS.

---

[13]Transport Layer Security (TLS) es un protocolo de autenticación que permite tener una autenticación mutua entre cliente y servidor para así establecer una conexión segura para el intercambio de claves entre las partes. (RFC 4346)

[14] PPP EAP TLS Authentication Protocol, B. Aboba y D. Simon, Octubre 1999

[15]La tecnología PKI permite a los usuarios autenticarse frente a otros usuarios y usar la información de certificados de identidad (claves públicas de otros usuarios) para garantizar el envío seguro de la información.

[16] Una tarjeta inteligente (smart card) es cualquier tarjeta portable con circuitos integrados que pemiten la ejecución de cierta lógica programada.

En EAP-TLS realiza un *handshake/Intercambio de Mensajes* TLS para autenticar mutuamente al cliente y al servidor; mientras que EAP-TTLS extiende esta negociación de autenticación mediante el uso de un túnel establecido por el *handshake* TLS para el intercambio de información entre el cliente y el servidor.

En EAP-TTLS, el *handshake* TLS puede ser mutuo, o puede ser de sentido único (el servidor es autenticado sólo por el cliente). La conexión segura establecida por el *handshake* puede ser utilizada para permitir que el servidor autentique al cliente utilizando los mecanismos de autenticación interna.

Los mecanismos de autenticación interna pueden ser EAP, PAP, CHAP, MS-CHAP o MS-CHAP-V2. Por lo tanto, EAP-TTLS permite delegar el control a protocolos de autenticación por contraseña que se pueden validar contra un repositorio de credenciales. EAP-TTLS también permite que cliente y servidor establezcan las llaves para la conexión de datos entre el cliente e intermediador. Las llaves se establecen implícitamente entre el cliente y el servidor basado en el *handshake* TLS.

En la Figura 1.12 se puede observar un ejemplo del intercambio de mensajes que tiene lugar durante la autenticación utilizando EAP-TTLS en un ambiente AAA, mientras que la Figura 1.13 ilustra el formato de un paquete EAP-TTLS.

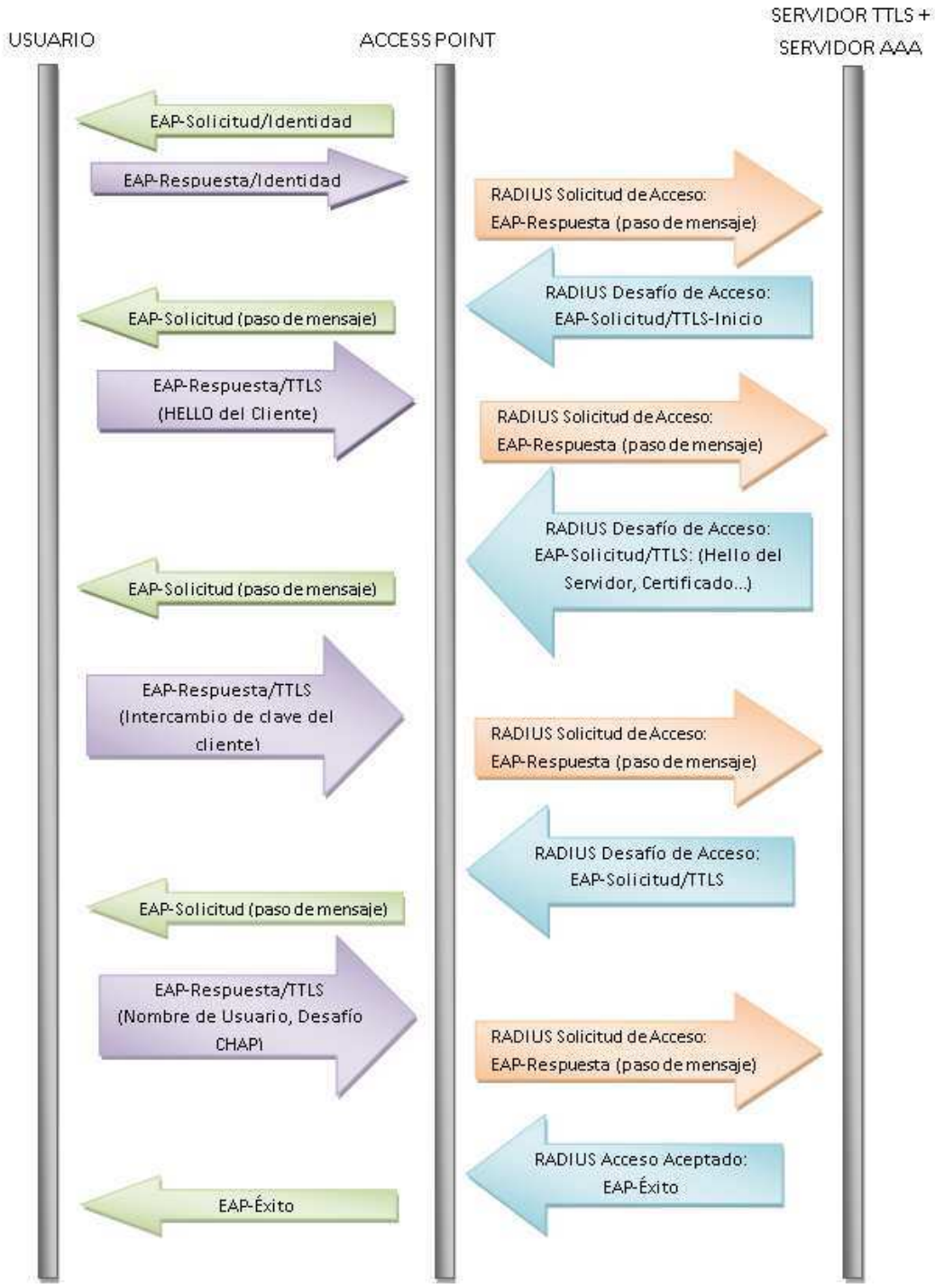


Figura 1.12 Intercambio de Paquetes EAP-TTLS en ambiente AAA

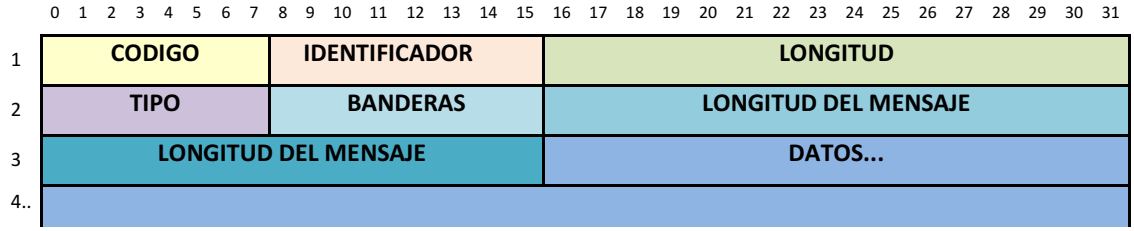


Figura 1.13 Formato del Paquete EAP-TTLS

1.3.3.3.4. *PEAP*

PEAP, *Protected EAP/EAP Protegido,* es una propuesta conjunta de Cisco Systems, Microsoft y RSA Security como un estándar abierto. Ofrece muy buena seguridad y soporte para diversos dispositivos. Es similar a EAP-TTLS y requiere una PKI, *Public Key Infraestructure/Infraestructura de Claves Públicas,* del lado del servidor para crear un túnel TLS para proteger la autenticación del usuario, de igual manera que EAP-TTLS delega la seguridad a protocolos de autenticación por contraseña, los mismos que pueden ser EAP-MSCHAPv2 y EAP-GTC.

Existen 2 subtipos:

- PEAPv0/EAP-MSCHAPv2 es el segundo estándar EAP con más soporte después de EAP-TLS.
- PEAPv1/EAP-GTC fue creado por Cisco como una alternativa a PEAPv0/EAP-MSCHAPv2. Permite el uso de un protocolo de autenticación interna diferente a MSCHAPv2 de Microsoft. EAP-GTC, *EAP Generic Token Card*/*EAP con Tarjeta Token Genérica,* se define en el RFC 3748. Lleva en texto plano el reto desde el servidor de autenticación y una respuesta que se supone que es generada por un token de seguridad. EAP-GTC no protege los datos de autenticación de ninguna manera. PEAPv1/EAP-GTC no tiene soporte nativo de SO Windows.

PEAPv0 y PEAPv1 se refieren al método de autenticación externo y son un

mecanismo que crea el túnel TLS para proteger las transacciones posteriores, mientras que la autenticación EAP-MSCHAPv2 y EAP-GTC se refieren al método de autenticación interna que facilita la autenticación de cliente o dispositivo.

1.3.3.3.5. *LEAP*

LEAP, *Lightweight EAP/EAP Ligero*, es un método EAP desarrollado por Cisco Systems. No existe soporte nativo para LEAP en cualquier sistema operativo Windows. El protocolo es vulnerable a ataques de diccionario; sin embargo si las contraseñas que se utilizan son lo suficientemente complejas LEAP puede ser seguro.

1.3.3.3.6. *EAP-FAST*

EAP-FAST, *Flexible Authentication via Secure Tunneling/Autenticación Flexible a través de Túnel Seguro*, se define en el RFC 4851[17]. EAP-FAST es un método diseñado por Cisco Systems para corregir las debilidades de LEAP. El uso del certificado del servidor es opcional.

EAP-FAST utiliza una Credencial de Acceso Protegido (PAC). EAP-FAST tiene tres fases. La Fase 0 es opcional, en esta fase se proporciona el PAC de forma manual o dinámica. En la Fase 1, el cliente y el servidor AAA utilizan el PAC para establecer el túnel TLS. En la Fase 2, el cliente envía la información del usuario a través del túnel.

1.4. RADIUS

RADIUS, *Remote Authentication Dial-In User Server/Servicio de Autenticación Remota para Usuario de Acceso Telefónico*, es un protocolo estándar que se utiliza para proporcionar servicios de autenticación, autorización y registro para el acceso y

[17]The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST), N. Cam-Winget y otros, Mayo del 2007.

uso de recursos de red.

RADIUS fue creado por Livingston Enterprises Inc. en 1991, y en 1997 pasó a formar parte de los estándares de la IETF al ser definido en el RFC 2058 y el RFC 2059, los cuales se encuentran actualmente descontinuados.

Actualmente, el funcionamiento del protocolo para los procesos de autenticación y autorización se describe en el RFC 2865[18], y los procesos para registro o auditoría se encuentran descritos en el RFC 2866[19].

1.4.1. GENERALIDADES

Entre las características más importantes del protocolo RADIUS se mencionan:

- Maneja un modelo Cliente/Servidor en el que el cliente es un NAS (Servidor de Acceso a la Red).
- Los puertos UDP asignados oficialmente para el protocolo RADIUS son el 1812 para Autenticación y Autorización y el 1813 para Auditoria o Registro.
- Establece un nivel de seguridad en la conexión Cliente/ Servidor mediante la autenticación entre estos dos elementos con secreto compartido.
- Soporta PAP, CHAP, EAP y otros mecanismos de autenticación.

1.4.2. FORMATO DEL PAQUETE

El paquete RADIUS es de longitud variable y se encapsula dentro de UDP. En la Figura 1.14 se detallan los campos del mismo.

---

[18]Remote Authentication Dial In User Service (RADIUS), C. Rigney y Otros, Junio del 2000
[19]RADIUS Accounting, C. Rigney, Junio 2000

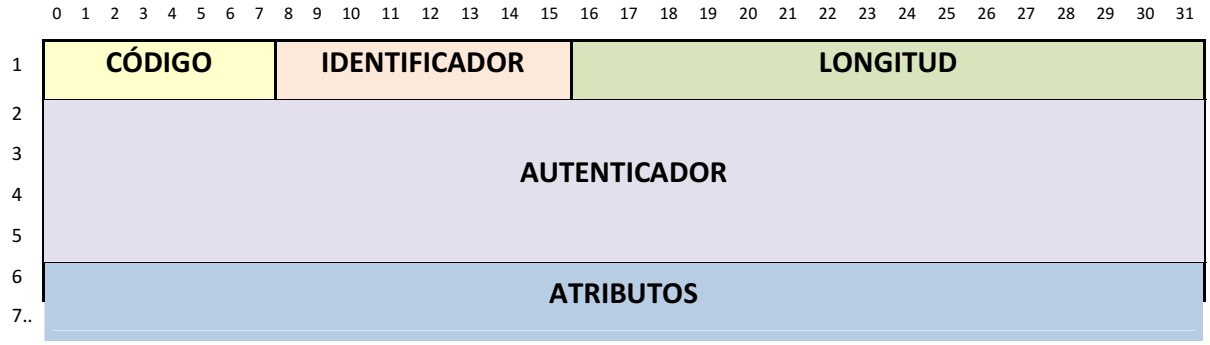


Figura 1.14 Formato del Paquete RADIUS

1.4.2.1. Código

El campo código es un octeto que identifica el tipo de paquete RADIUS que se transmite. La Tabla 1.2 detalla el valor binario y el código que le corresponde a cada tipo de paquete RADIUS.

Código	Valor Binario	Tipo de Paquete
1	0000001	Solicitud de Acceso (Access-Request)
2	0000010	Aprobación de Acceso (Access-Accept)
3	0000011	Denegación de Acceso (Access-Reject)
4	0000100	Solicitud de Registro (Accounting-Request)
5	0000101	Respuesta de Registro (Accounting-Response)
11	0001011	Desafío de Acceso (Access-Challenge)
12	0001100	Estado del Servidor (Experimental)
13	0001101	Estado del Cliente (Experimental)
255	1111111	Reservado

Tabla 1.2 Campo Código en Paquetes RADIUS

1.4.2.2. Identificador

El campo Identificador es un octeto que numera al paquete RADIUS para ayudar al servidor a emparejar peticiones con respuestas.

1.4.2.3. Longitud

El campo longitud son dos octetos que indican la longitud total del paquete, incluyendo el campo Atributos. La longitud mínima de un paquete RADIUS es 20 octetos y la máxima es 4096 octetos.

1.4.2.4. Autenticador

El campo Autenticador son 16 octetos que contienen la información que el cliente y el servidor RADIUS utilizan para autenticarse mutuamente.

1.4.2.5. Atributos

El campo Atributos de un paquete RADIUS contiene la información necesaria para los procesos de autenticación, autorización y ciertos detalles de configuración. En la Tabla 1.3 se detallan los atributos RADIUS más comunes.

Tipo	Nombre	Longitud	Descripción
1	User-Name	L≥3	Nombre de Usuario a ser autenticado.
2	User-Password	130≥L≥18	Contraseña del usuario a ser autenticado.
3	CHAP-Password	L=19	Respuesta del Usuario CHAP al desafío.
4	NAS-IP-Address	L=6	La dirección IP del NAS.
5	NAS-Port	L=6	Número de puerto FÍSICO del NAS que está autenticando al usuario.
6	Service-Type	L=6	Tipo de Servicio solicitado o provisto.
7	Framed-Protocol	L=6	Protocolo utilizado para el acceso (PPP, SLIP, etc.)
8	Framed-IP-Address	L=6	Dirección IP a asignarse al cliente.
9	Framed-IP-Mask	L=6	Máscara de Red de la IP que el NAS asigna al cliente antes de la autenticación.

Tipo	Nombre	Longitud	Descripción
11	Filter-Id	L≥3	Id o el nombre del filtro que se aplicará al usuario.
18	Reply-Message	L≥3	Texto que se muestra al usuario, su uso varía de acuerdo al tipo de paquete que se envía.
24	State	L≥3	Estado de la Conexión.
25	Class	L≥3	Atributo usado para el proceso de Registro.
26	Vendor-Specific	L≥7	Campo Reservado para implementaciones propietarias del proveedor.
27	Session-Timeout	L=6	Máximo número de segundos de duración de la sesión.
30	Called-Station-Id	L≥3	Número de teléfono en el que se recibió la llamada. En las conexiones de red privada virtual (VPN), se trata de la dirección IP del servidor VPN.
31	Calling-Station-Id	L≥3	Número de teléfono desde el que se realizó la llamada. En las conexiones de red privada virtual (VPN), se trata de la dirección IP del cliente VPN.
32	NAS-Identifier	L≥3	Nombre completo de dominio del servidor de acceso remoto.
40	Acct-Status-Type	L=6	Indica el estado de la solicitud de registro
41	Acct-Delay-Time	L=6	Indica el tiempo en segundos que el cliente ha intentado enviar el registro.
42	Acct-Input-Octets	L=6	Indica el número de octetos que se han recibido del puerto durante el transcurso del servicio.
43	Acct-Output-Octets	L=6	Indica el número de octetos que se han recibido del puerto durante el transcurso del servicio.
44	Acct-Session-Id	L≥3	Es un ID de Registro único.
45	Acct-Authentic	L=6	Indica la forma en la que el usuario fue autenticado.
46	Acct-Session-Time	L=6	Indica la duración en segundos de la sesión.
47	Acct-Input-Packets	L=6	Indica el número de paquetes recibidos en el puerto por el que se conecta el usuario durante la sesión
48	Acct-Output-Packets	L=6	Indica el número de paquetes enviados desde el puerto por el que se conecta el usuario.
49	Acct-Terminate-Cause	L=6	Indica la forma en la que se terminó la sesión.
60	CHAP-Challenge	L≥7	Desafío que envía el servidor de acceso remoto durante la autenticación CHAP.
61	NAS-Port-Type	L=6	Indica el tipo de puerto físico del NAS que autentica al usuario.
64	Tunnel-Type	L=6	Este atributo indica los protocolos de túnel a usarse. (PPTP, L2TP, VTP, etc.).
65	Tunnel-Medium-Type	L=6	Este atributo indica el medio de transporte a usarse al crear un túnel.
81	Tunnel-Private-Group-ID	L≥3	Indica el ID de grupo para una sesión de túnel en particular.

Tabla 1.3 Atributos RADIUS más comunes

El campo Atributos es de longitud variable y maneja un formato TLV (Tipo, Longitud,

Valor). Esto se puede observar en la Figura 1.15.

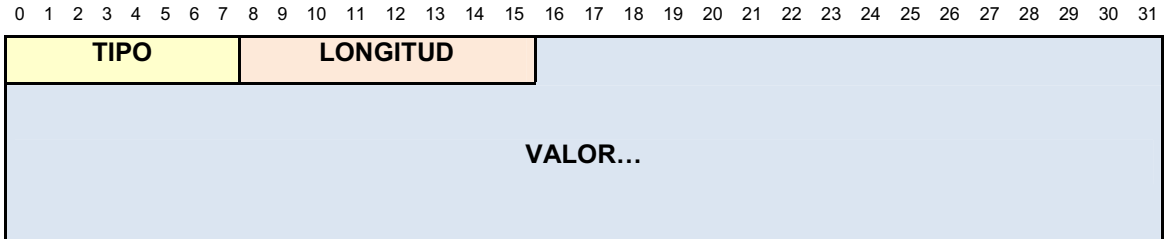


Figura 1.15 Formato del Campo Atributos del Paquete RADIUS

1.4.2.5.1. *Tipo*

El campo Tipo es un octeto cuyo valor identifica el tipo de atributo. Los valores del 192 al 223 están reservados para uso experimental, los valores del 224 al 240 están reservados para uso específico de la implementación y los valores del 241 al 255 están reservados y no deben usarse.

1.4.2.5.2. *Longitud*

El campo Longitud es un octeto que especifica la longitud total del campo Atributos.

1.4.2.5.3. *Valor*

El campo valor puede contener cero o más octetos y contiene la información específica del atributo.

1.4.3. TIPOS DE PAQUETE

1.4.3.1. Solicitud de Acceso (Access-Request)

Los paquetes de Solicitud de Acceso se envían al servidor RADIUS para entregar la información necesaria para el proceso de autenticación y autorización del usuario.

Un paquete de Solicitud de Acceso debe incluir:

- Un atributo que identifique al usuario que requiere acceso a la red, (User-Name).
- Al menos un atributo que identifique de forma única al NAS, (NAS-IP-Address, NAS-Identifier, o ambos).
- Un atributo que contenga la contraseña con la que el Usuario va a autenticarse. (User-Password, CHAP-Password o State).
- Al menos un atributo que identifique el puerto físico del NAS a través del cual el usuario va a acceder a la red, a menos que el tipo de acceso solicitado no involucre un puerto o el NAS no los distinga. (NAS-Port o NAS-Port-Type).

En un paquete de Solicitud de Acceso (Access-Request), el campo Código tiene un valor de 1 y los 16 octetos del campo Autenticador son un número aleatorio e impredecible llamado Autenticador de Solicitud (Request Authenticator). Este número aleatorio se utiliza en el proceso de cifrado del atributo User-Password.

La Figura 1.16 ilustra el formato del Paquete Access-Request.

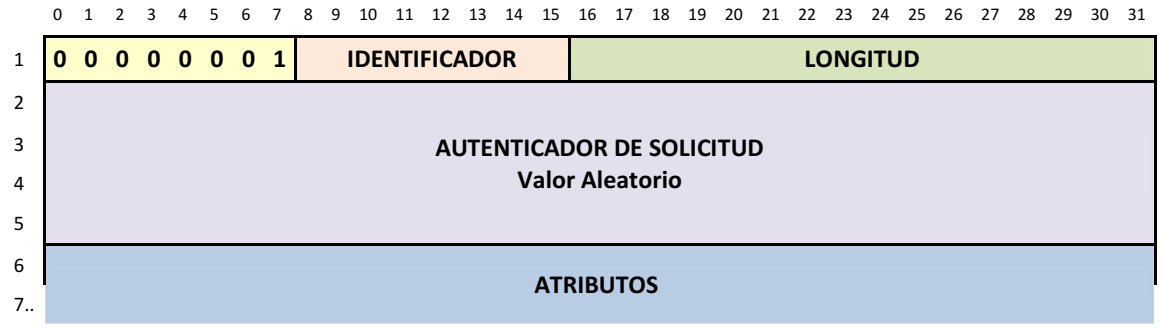


Figura 1.16 Paquete Access-Request

1.4.3.2. Aprobación de Acceso (Access-Accept)

Los paquetes de Aprobación de Acceso son enviados por el servidor RADIUS para entregar la información necesaria para empezar la entrega del servicio al usuario una vez que todos los atributos recibidos en el paquete Access-Request han sido aceptados.

En un paquete de Aprobación de Acceso (Access-Accept):

- El campo Código tiene un valor de 2.
- El campo Identificador es igual al del paquete Access-Request que generó la respuesta.
- Los 16 octetos del campo Autenticador son el resultado del cálculo hash MD5[20] del contenido del paquete RADIUS concatenado al Secreto compartido del NAS y el Servidor RADIUS[21]. Este campo toma el nombre de Autenticador de Respuesta (Response Authenticator).

La Figura 1.17 ilustra el formato del Paquete Access-Accept.

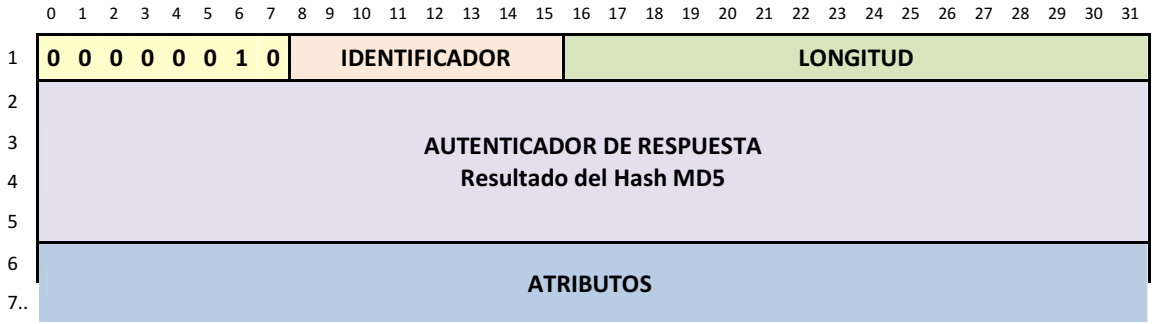


Figura 1.17 Formato del Paquete Access-Accept

[20] Message-Digest Algorithm 5 (Algoritmo de Resumen del Mensaje 5) es un algoritmo criptográfico de reducción.
[21] ResponseAuth = MD5(Codigo+ID+Longitud+RequestAuth+Atributes+Secret).

1.4.3.3. Denegación de Acceso (Access-Reject)

Un paquete de Denegación de Acceso es enviado por el servidor RADIUS, como respuesta a un paquete Access-Request que contiene atributos que el servidor no aceptó. El paquete puede contener uno o varios atributos del tipo Reply-Message que el NAS puede presentar al usuario.

En un paquete de Denegación de Acceso (Access-Reject):

- El campo Código tiene un valor de 3.
- El campo Identificador es igual al del paquete Access-Request que generó la respuesta.
- Los 16 octetos del campo Autenticador son del tipo Autenticador de Respuesta (Response Authenticator).

La Figura 1.18 ilustra el formato del Paquete Access-Reject.

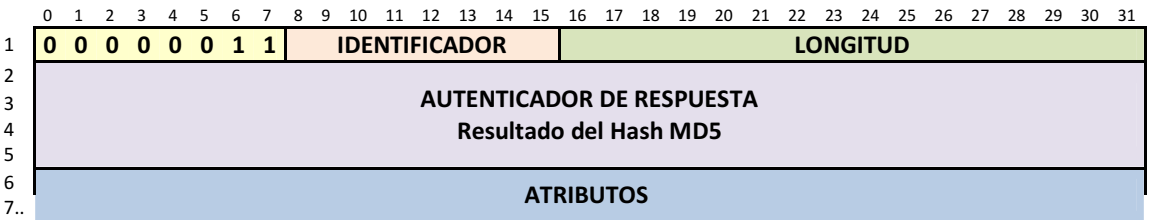


Figura 1.18 Formato del Paquete Access-Reject

1.4.3.4. Desafío de Acceso (Access-Challenge)

Un paquete de Desafío de Acceso es enviado por el servidor RADIUS al NAS, cuando se requiere información extra para la autenticación del usuario.

En un paquete de Desafío de Acceso (Access-Challenge):

- El campo Código tiene un valor de 11.

- El campo Identificador es igual al del paquete Access-Accept que generó la respuesta.
- Los 16 octetos del campo Autenticador son del tipo *Autenticador de Respuesta (Response Authenticator).*

Un paquete de Desafío de Acceso puede incluir:

- Uno o más atributos Reply-Message.
- Uno o ningún atributo State.
- Los atributos Vendor-Specific, Session-Timeout o Proxy-State.

Cuando el NAS recibe un paquete Access-Challenge debe enviar una nueva Solicitud de Acceso (Access-Request) con la respuesta del usuario al desafío del servidor. Si el NAS no soporta, la operación actúa como si hubiera recibido un paquete Access-Reject.

La Figura 1.19 ilustra el formato del Paquete Access-Challenge.

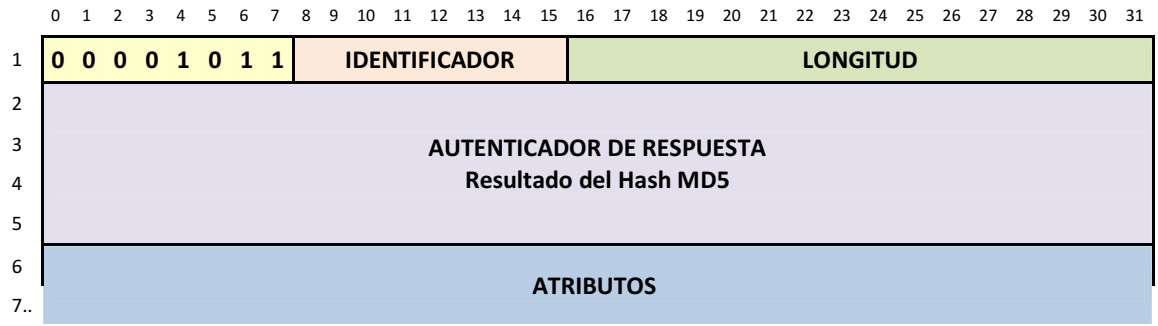


Figura 1.19 Formato del Paquete Access-Challenge

1.4.3.5. Solicitud de Registro (Accounting-Request)

La solicitud de Registro es enviada por el Cliente (NAS) al Servidor RADIUS para entregar la información necesaria para realizar el Registro del Servicio que se le

entregó al usuario.

En un paquete de Solicitud de Registro (Accounting-Request):

- El campo Código tiene un valor de 4.
- Los 16 octetos del campo Autenticador son el resultado del hash MD5 del paquete RADIUS concatenado al Secreto Compartido entre el NAS y el Servidor.[22]

Un paquete de Solicitud de Registro NO puede incluir:

- El atributo User-Password.
- El atributo CHAP-Password.
- El atributo Reply-Message.
- El atributo State.

Un paquete de Solicitud de Registro DEBE incluir:

- El atributo NAS-IP-Address o NAS-Identifier.
- El atributo NAS-Port o NAS-Port-Type o ambos a menos que el servicio no involucre los puertos o el NAS no los distinga.

La Figura 1.20 ilustra el formato del Paquete Accounting-Request.

Figura 1.20 Formato del Paquete Accounting-Request

[22] RequestAuth = MD5(Código+ID+Longitud+16 octetos de valor 0+Atributos+Secreto Compartido)

1.4.3.6. Respuesta de Registro (Accounting-Response)

Los mensajes de Respuesta de Registro son enviados por el servidor RADIUS al NAS para informar que la Solicitud de Registro fue recibida y procesada exitosamente.

En un paquete de Respuesta de Registro (Accounting-Response):

- El campo Código tiene un valor de 5.
- Los 16 octetos del campo Autenticador son el resultado del hash MD5 del paquete RADIUS concatenado al Secreto Compartido entre el NAS y el Servidor.[23]

La Figura 1.21 ilustra el formato del Paquete Accounting-Response.

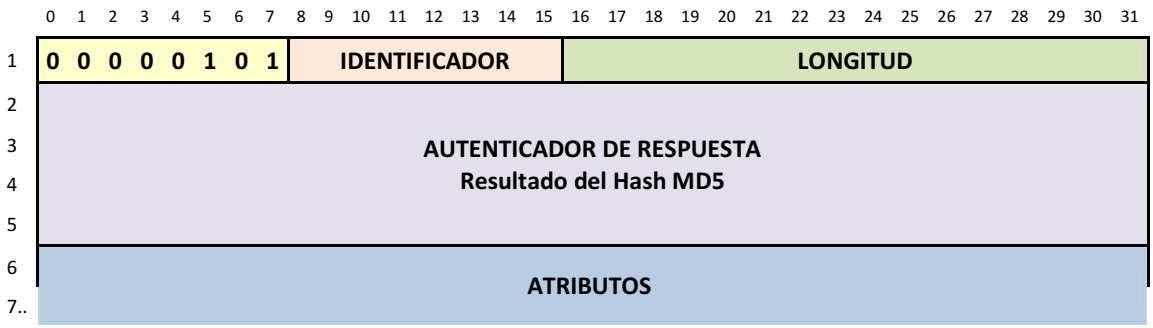


Figura 1.21 Formato del Paquete Accounting-Response

1.4.4. OPERACIÓN DEL PROTOCOLO RADIUS

1.4.4.1. Autenticación y Autorización

Cuando un usuario intenta ingresar a una red en producción a través de un dispositivo NAS configurado como cliente RADIUS, debe presentar la información

[23] ResponseAuth = MD5(Código+ID+Longitud+RequestAuth del Mensaje al que se Responde+Atributos+Secreto Compartido)

necesaria para el proceso de autenticación al NAS al inicio de la conexión.

Tras recibir las credenciales del usuario, el Cliente NAS envía un paquete *Access-Request/Solicitud de Acceso,* al Servidor RADIUS con información como el nombre de usuario, la contraseña y el puerto del dispositivo a través del cual el usuario accede a la red.

Una vez que el servidor RADIUS recibe la solicitud, valida la identidad del Cliente NAS. Si el servidor RADIUS y el Cliente NAS no tienen un secreto compartido la solicitud de acceso se descarta; si por el contrario el cliente es válido, el Servidor RADIUS consulta una base de datos que contiene las condiciones que el usuario debe cumplir para obtener acceso a la red. Se pueden obtener tres resultados diferentes de esta operación:

- Si alguna condición no se cumple, el Servidor RADIUS envía un mensaje *Access-Reject/Denegación de Acceso*.
- Si se cumplen todas las condiciones y el Servidor RADIUS desea consignar un desafío al usuario, se envía un mensaje *Access-Challenge/Desafío de Acceso.* Si el cliente soporta Challenge/Response solicita una respuesta al usuario y retransmite un mensaje *Access-Request/Solicitud de Acceso* al servidor, con la respuesta al desafío en el atributo User-Password.
- Si se cumplen todas las condiciones, el Servidor RADIUS envía un mensaje *Access-Accept/Aprobación de Acceso*, en el que se incluyen todos los valores necesarios para configurar la sesión del usuario.

La Figura 1.22 ilustra un ejemplo del intercambio de mensajes que se lleva a cabo durante un proceso de autenticación fallida utilizando RADIUS.

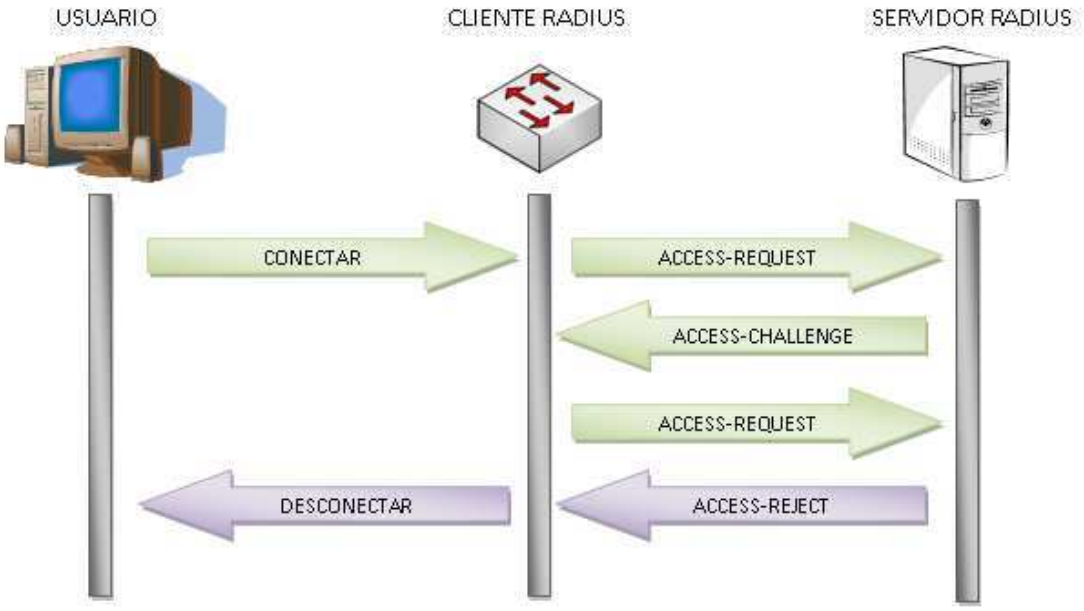


Figura 1.22 Operación de RADIUS con Autenticación Fallida

1.4.4.2. Auditoría o Registro

Cuando se configura a un dispositivo de acceso a la red como un cliente RADIUS, éste genera un paquete *Accounting-Request/Solicitud de Registro* en el instante en el que se empieza a entregar el servicio al usuario y lo envía al servidor RADIUS.

El paquete *Accounting-Request/Solicitud de Registro* al inicio del proceso de registro, lleva información del usuario, detalles de su conexión a la red y permisos asignados. Cuando el servidor RADIUS recibe y procesa un paquete *Accounting-Request/Solicitud de Registro* envía un mensaje de confirmación al cliente.

Cuando la sesión del usuario termina, el cliente RADIUS envía un paquete *Accounting-Request/Solicitud de Registro* solicitando se finalice el registro de la sesión. Este mensaje describe el tipo de servicio otorgado y puede llevar información extra como el tiempo de duración de la sesión o el número de paquetes intercambiados.

La Figura 1.23 ilustra el intercambio de mensajes que tiene lugar durante un proceso

de autenticación exitosa usando RADIUS.

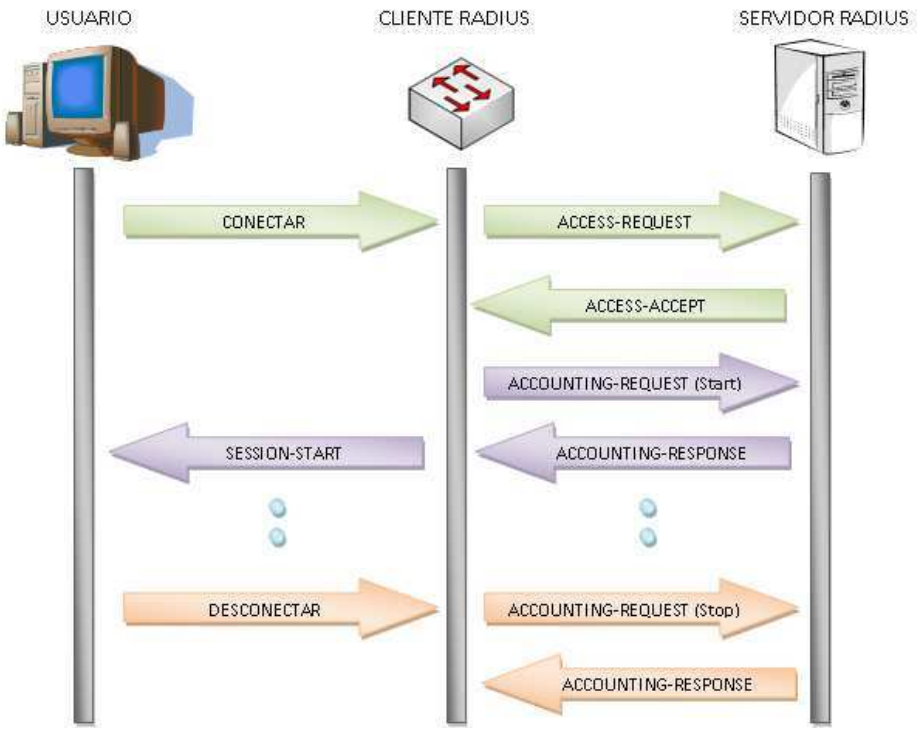


Figura 1.23 Operación de RADIUS con Autenticación Exitosa

1.5. OTROS PROTOCOLOS AAA

1.5.1. TACACS+

TACACS+ (Terminal Access Controller Access-Control System Plus) es un protocolo propietario de Cisco que provee servicios AAA y utiliza el puerto 49 de TCP para transporte.

El protocolo TACACS+ es la última generación del protocolo de autenticación TACACS[24], se caracteriza porque las funciones de autenticación, autorización y registro son independientes entre sí; por esta razón, en una implementación de

[24]RFC 1492, "An Access Control Protocol, Sometimes Called TACACS"

TACACS+ no es obligatoria la configuración de estos tres servicios aunque los tres son soportados.

Un beneficio importante de la separación de servicios AAA en el protocolo TACACS+ es que la autorización puede ser un proceso dinámico y el servicio de registro se puede convertir en servicio de facturación y auditoría.

1.5.2. DIAMETER

DIAMETER es un protocolo desarrollado para proveer servicios de autenticación, autorización y registro que superen las debilidades de RADIUS en redes que requieren de movilidad IP y roaming. El protocolo base de DIAMETER se encuentra definido en el RFC 3588[25].

DIAMETER es escalable a través de la definición de aplicaciones. En este contexto, una aplicación es un protocolo basado en el protocolo base de DIAMETER que funciona como una extensión del mismo. Entre las aplicaciones DIAMETER más importantes se pueden nombrar:

- *"Diameter Mobile IPv4 Application"* (MobileIP, RFC 4004)
  Que permite la implementación de DIAMETER en ambientes que requieren movilidad IP.
- *"Diameter Network Access Server Application"* (NASREQ, RFC 4005)
  Que provee un método de actualización hacia el protocolo DIAMETER para los NAS que utilicen RADIUS.
- "*Diameter Extensible Authentication Protocol Application"* (RFC 4072)
  Que permite la implementación de EAP como método de autenticación para ambientes en los que se utilice un servidor DIAMETER.

---

[25] "Diameter Base Protocol", P. Calhoun y Otros, Septiembre 2003

- *"Diameter Session Initiation Protocol Application"* (RFC 4740)
  Que permite la interacción de un Servidor DIAMETER con un Servidor SIP para los procesos de autenticación y autorización.

La Tabla 1.4 muestra una breve comparación entre los tres protocolos AAA mencionados en este capítulo.

CARACTERÍSTICA	RADIUS	TACACS+	DIAMETER
Estándar	RFC 2865 RFC 2866	Propietario de Cisco	RFC 3588
Arquitectura	Cliente/Servidor	Cliente/Servidor	Peer to Peer
Protocolo Capa Transporte	UDP	TCP	TCP ó SCTP
Puerto	1812 y 1813	49	3868
Seguridad del Enlace	Sólo provee encriptación de la contraseña.	Provee encriptación para el cuerpo entero del paquete.	Utiliza seguridad a nivel de capa Transporte IPSec o TLS
Descubrimiento Dinámico de Pares	NO	NO	SI
Procesos de Autenticación, Autorización y Registro independientes	No, combina la autenticación y autorización.	Sí, lo cual lo hace más escalable.	Sí, se definen en aplicaciones independientes.
Control de comandos para administración de Routers	NO	SI	NO

Tabla 1.4 Comparación de Protocolos AAA

1.6. EL ESTÁNDAR 802.1X

El estándar IEEE 802.1x, *Port-Based Network Access Control/Control de Acceso en Redes basadas en Puerto*, describe la arquitectura, los elementos funcionales y los protocolos que proveen mecanismos de autenticación, autorización y cifrado para lograr una comunicación segura entre dispositivos conectados en una red de área

local. Fue publicado por primera vez en el 2001 y la última revisión del mismo fue lanzada oficialmente el 5 de Febrero del 2010.

Este estándar menciona a varios protocolos, entre ellos EAP y RADIUS. EAP, *Extensible Authentication Protocol/ Protocolo de Autenticación Extensible*, es un componente importante de 802.1x. En este estándar se define la encapsulación de EAP en paquetes Ethernet para uso en la LAN, la cual es conocida como EAPOL, *EAP over LAN/EAP sobre Redes de Área Local.*

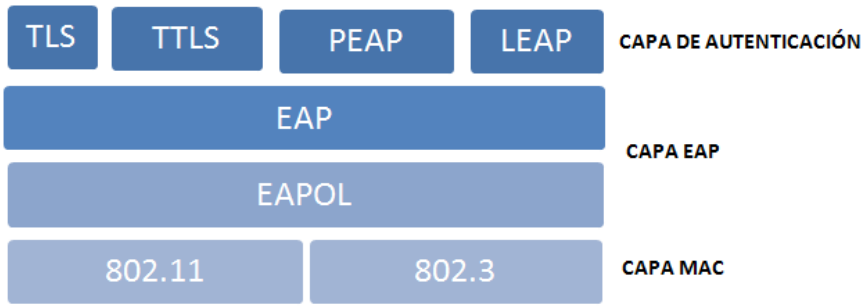


Figura 1.24 Protocolos mencionados en el estándar 802.1x

1.6.1. ELEMENTOS DE 802.1X

El estándar define tres elementos para proporcionar autenticación mutua extremo a extremo en una red LAN:

- Usuario, también llamado suplicante, debe soportar los métodos de autenticación que menciona el estándar.
- Autenticador, que generalmente es un punto de acceso inalámbrico, análogo al PEP en una arquitectura AAA.
- Servidor de autenticación, con funciones análogas al PDP en una arquitectura AAA.

1.7. META DISTRIBUCIONES LINUX

1.7.1. REMASTERSYS Y EL CONCEPTO DE META DISTRIBUCIÓN

Meta Distribución es un sistema autónomo completo, auto-arrancable e instalable, basado en una distribución GNU/Linux personalizada para un entorno de trabajo concreto.

Por su forma de instalación, se considera a una Meta Distribución la base para otras distribuciones. A diferencia de una Distribución Linux común o una LIVE Distro, los paquetes de la Meta Distribución no se instalan, solo son copiados al disco duro, por lo que se encuentran con las modificaciones y configuraciones personalizadas por cada usuario desde el momento que inicia el LiveCD.

Las distribuciones Linux y Live distros instalan los paquetes, por lo que estos se inician con las configuraciones por defecto teniendo así siempre el mismo resultado final.

El proceso de arranque de una Meta Distribución es del tipo LIVE; es decir tiene lugar desde un CD-ROM de tal forma que el sistema completo arranca y se ejecuta sin necesidad de utilizar el disco duro. El arranque en modo LIVE permite al usuario probar el sistema operativo sin necesidad de instalarlo en disco duro.

El término Meta distribución, proviene del desaparecido proyecto METADISTROS de HispaLinux[26]. El objetivo de este proyecto fue ser un espacio de colaboración entre la gente que desarrollaba distribuciones Linux enfocadas a un público determinado. Actualmente, la página oficial del proyecto[27] cambió de dominio y se encuentra abandonada. Sin embargo, se puede encontrar aún valiosa información respecto al proceso de creación de meta distribuciones y también se distribuyen sistemas

[26] Asociación de Usuarios Españoles de Linux: http://www.hispalinux.es/
[27] https://forja.rediris.es/projects/metadistros

operativos que utilizaron este concepto para su desarrollo. Uno de los ejemplos más claros es “Guadalinex”, que es un sistema operativo basado en Debian impulsado por la Junta de Andalucía para brindar al usuario común las herramientas básicas para el uso en el hogar y así brindar un instrumento de acceso a la Sociedad del Conocimiento.[28]

La desaparición del proyecto METADISTROS se debe en parte a la falta de organización y a la aparición de programas que facilitaron la creación de éstos sistemas operativos basados en otras distribuciones como REMASTERSYS[29].

REMASTERSYS es una herramienta que puede utilizarse para crear un LIVE CD de un respaldo completo de una distribución Debian que pueda instalarse en otro equipo; o se puede crear una copia distribuible y personalizada de Debian siguiendo el concepto de Meta Distribuciones.

1.7.2. HERRAMIENTAS PARA LA GENERACIÓN DE UNA META DISTRIBUCIÓN

En este punto se detallarán las herramientas que REMASTERSYS utiliza para lograr que una distribución Linux arranque en modo LIVE y pueda después instalarse en el disco duro.

1.7.2.1. InitramFS

El initramFS es un sistema de archivos raíz mínimo embebido en el kernel de Linux desde la Versión 2.7, que se utiliza para montar un pequeño sistema Linux temporal sobre la memoria RAM para hacer detecciones de hardware básicas antes del arranque del proceso init[30].

---

[28] Citado de: http://www.guadalinex.org, página oficial del proyecto.
[29] http://geekconnection.org/remastersys/index.html, página oficial del paquete.
[30] Es el primer proceso que se ejecuta en un Sistema Linux y el padre de todos los demás procesos.

REMASTERSYS utiliza initramFS para el arranque LIVE de la distribución que inicia desde el CD.

La Figura 1.25 detalla el orden del proceso de arranque de un sistema operativo cualquiera y el proceso de arranque de una distribución Linux.

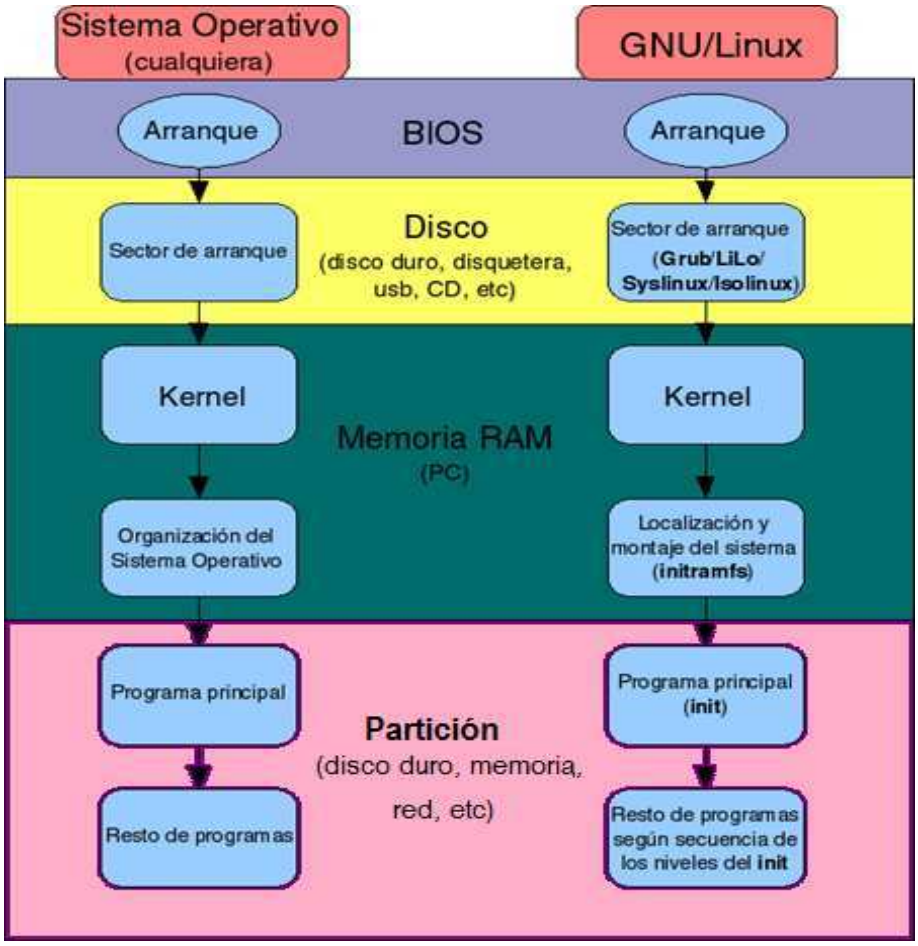


Figura 1.25 Proceso de arranque de un Sistema Operativo.[31]

[31]Todos los PC comienzan el proceso de arranque de un sistema operativo ejecutando un código en la BIOS para encontrar el sector 0 en el disco de arranque, el cual contiene un gestor de arranque que localiza el kernel, lo carga y lo ejecuta para comenzar los procesos de arranque de Linux o el kernel del sistema. Cuando el kernel del sistema se ha cargado completamente, el initramFS es cargado en la memoria RAM a través del cual se inicializará el proceso init. Imagen obtenida de: http://forja.guadalinex.org/webs/guadalinexv5/doku.php?id=initramfs:que_es.

1.7.2.2. Isolinux

Isolinux es el nombre de un gestor de arranque para Linux/i386 que funciona sobre ISO 9660/El Torito CD-ROM[32] permitiendo crear CDs autoejecutables.

1.7.2.3. UnionFS y AuFS

UnionFS[33] y AuFS[34], *Another UnionFS/Otro UnionFS,* son sistemas de archivos de unión apilable de Linux que permiten la unión del contenido de varios directorios (sistemas de archivos) aunque el contenido esté separado físicamente (sistemas de archivos diferentes).

Los paquetes *unionfs-modules* y *aufs-modules* son requisito para el funcionamiento de REMASTERSYS y la creación del LIVECD. Permiten superponer un sistema de archivos modificable sobre un CD que es un medio de solo lectura de tal forma que el usuario lo vea como un sistema de archivos en el que es posible modificar, eliminar y crear archivos.

1.7.2.4. SquashFS

Una imagen SquashFS[35] es un sistema de archivos de sólo lectura para Linux que comprime directorios o sistemas de archivos completos y es distribuido como un parche del kernel de Linux a través de las herramientas *mksquashfs* y *unsquashfs.*

Mksquashfs es la herramienta que permite crear nuevos sistemas de archivos squash e incluir nueva información en sistemas de archivos squash existentes.

---

[32] Extensión de la especificación ISO 9660 CD ROM (estándar del sistema de archivos para discos compactos) diseñada para permitir que un computador arranque desde un CD-ROM.
[33] http://unionfs.filesystems.org/, página oficial del proyecto.
[34] http://aufs.sourceforge.net/, página oficial del proyecto.
[35] http://squashfs.sourceforge.net/, página oficial del proyecto.

Unsquashfs es la herramienta para extraer información de un sistema de archivos squash.

REMASTERSYS usa la imagen SquashFS para que la copia de una distribución Linux en producción quepa en un CD.

CAPÍTULO 2. DISEÑO E IMPLEMENTACIÓN DEL MÓDULO DE ADMINISTRACIÓN WEB

2.1. GENERALIDADES

En vista de la creciente necesidad de implementar servicios basados en identidad en redes empresariales y la escasez de herramientas de software que lo faciliten; se ha desarrollado una Aplicación Web que facilita las tareas de configuración y administración de un servidor AAA sin costos de licenciamiento.

Se llama Aplicación Web a un programa anidado en un servidor con el que los usuarios interactúan a través de un navegador Web. La principal ventaja de este tipo de aplicaciones es la capacidad de actualizarlas y mantenerlas sin necesidad de distribuir e instalar software adicional a los clientes debido a la amplia difusión de los navegadores.

Tradicionalmente las páginas Web eran estáticas y no permitían la interacción del usuario con el contenido que presentaban pero hoy en día existe una nueva tendencia llamada Web 2.0 que se inclina a crear páginas Web que proveen servicios capaces de reemplazar a las aplicaciones de escritorio. Existe otra tendencia llamada Web 3.0 dirigida a crear Webs inteligentes que puedan interpretar las consultas de los usuarios basándose en la semántica y el contexto de la solicitud. Esta tendencia está orientada al mundo del Internet y la World Wide Web, por lo que no es aplicable al desarrollo de herramientas de administración Web como la descrita en este documento.

A continuación se realiza una breve descripción de las tecnologías y lenguajes de programación utilizados para el desarrollo de la Aplicación Web.

2.1.1. LENGUAJES DE PROGRAMACIÓN Y TÉCNOLOGÍAS UTILIZADAS

2.1.1.1. HTTP

HTTP[36], *Hypertext Transfer Protocol/Protocolo de Transferencia de Hipertexto*, es el protocolo estándar para intercambio de contenido (archivos HTML, imágenes, etc.) en la World Wide Web. HTTP es un protocolo orientado a transacciones y sigue un esquema petición-respuesta entre el cliente y el servidor sin guardar ninguna información sobre conexiones anteriores, por lo que se dice que HTTP es un protocolo *stateless/sin estado*.

Las aplicaciones Web actuales, frecuentemente necesitan mantener el estado de la conexión para establecer la noción de sesiones. Entre los métodos para conseguir este objetivo se encuentra el uso de *cookies* que son fragmentos de información que el explorador web almacena en el disco duro del usuario a petición del servidor HTTP.

2.1.1.1.1. Métodos HTTP

Un método HTTP es la operación que el cliente solicita al servidor Web se realice sobre un recurso identificado. Se definen ocho métodos HTTP en el estándar pero los más utilizados son los métodos GET y POST.

El método GET se utiliza para obtener la información de un recurso específico sin que esta operación tenga efectos secundarios sobre el mismo por lo que puede repetirse refrescando la página en el explorador.

Las solicitudes del método GET, solo se pueden enviar a través de parámetros

[36] RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1, Junio 1999.

codificados en el URI[37] o como cookies; por lo que este método no puede ser utilizado para operaciones que requieren que se envíe gran cantidad de información al servidor, además la información enviada es visible a través del explorador lo cual implica una brecha en la seguridad. La Figura 2.1 ilustra un ejemplo del intercambio de mensajes HTTP cuando se utiliza el método GET.

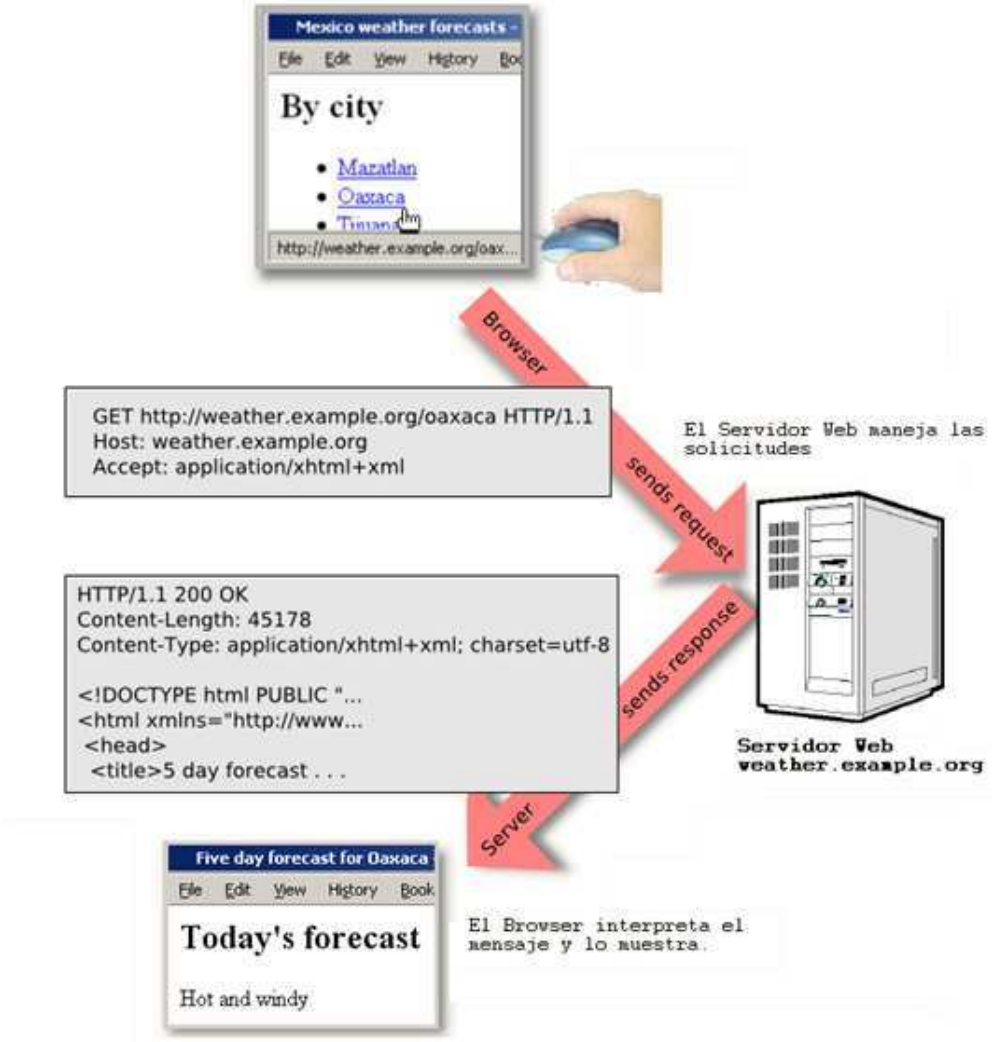


Figura 2.1 Ejemplo de Intercambio HTTP Request-Response[38]

[37] Uniform Resource Identifier, una cadena corta de caracteres que identifica inequívocamente a un recurso accesible a través de la red.

[38] Imagen obtenida de: "What's a URI and why does it matter?" , Henry S. Thompson, Agosto 2010, http://www.ltg.ed.ac.uk/~ht/WhatAreURIs/

El método POST se utiliza para operaciones que tienen efectos secundarios sobre los recursos solicitados por lo que no se pueden repetir de manera segura. El método envía datos en el cuerpo de la solicitud HTTP para ser procesados en el servidor, lo cual puede resultar en la actualización o creación de nuevos recursos.

2.1.1.2. DOM

DOM, *Document Object Model/Modelo de Objetos para Documentos*, es una especificación del W3C[39] que define un modelo lógico para API, *Application Programming Interface/Interfaz de Programación de Aplicaciones,* que permite que un script de código acceda y actualice el contenido, estructura y estilo de un documento Web.

El DOM reconoce cada parte de un documento Web como un nodo que puede ser texto, un elemento o un atributo; y ofrece métodos para modificarlos o crearlos separando el diseño Web en tres capas: Estructura (HTML), Presentación (CSS) y Comportamiento (JavaScript). La Figura 2.2 presenta gráficamente el DOM de una página HTML.

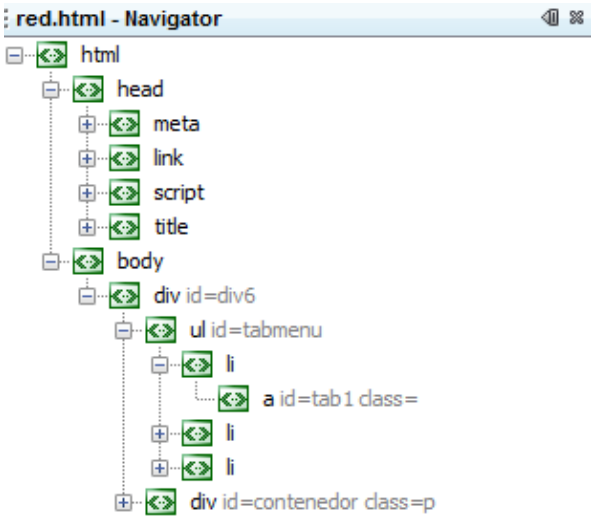


Figura 2.2 Ejemplo de Modelo DOM aplicado en NetBeans IDE 6.8

[39] World Wide Web Consortium

2.1.1.3. HTML

HTML, *HyperText Markup Language/Lenguaje de Marcación de Hipertexto*, es un sistema de etiquetas utilizado para crear documentos de hipertexto portables entre plataformas, lo que lo ha convertido en el lenguaje de publicación utilizado por la World Wide Web.

HTML permite describir la estructura de una página Web con cabeceras, texto, tablas, listas, fotos, enlaces y formularios para realizar transacciones con servidores remotos, entre otros.

2.1.1.4. CSS

Mientras que HTML describe la estructura de una página Web; CSS, *Cascading Style Sheets/Hoja de Estilo en Cascada*, provee el formato de la presentación de la misma.

CSS es el lenguaje para describir el diseño de páginas Web, incluyendo colores, forma y fuentes de texto. CSS es independiente de HTML y puede usarse con cualquier lenguaje basado en etiquetas.

2.1.1.5. JavaScript

JavaScript es un lenguaje de programación orientado objetos que no necesita compilarse y se ejecuta en el explorador Web. JavaScript utiliza DOM para modificar el documento o controlar el explorador Web que lo muestra. Es decir, JavaScript añade comportamiento a los documentos HTML que de otra forma son estáticos.

2.1.1.6. AJAX

AJAX, *Asynchronous JavaScript And XML/JavaScript Asíncrono y XML*, es la fusión de diferentes tecnologías de desarrollo Web empleadas para permitir que una página

Web solicite y reciba información del servidor en segundo plano y de manera asíncrona sin interferir con la presentación de la página; y así crear aplicaciones interactivas siguiendo la tendencia de la Web 2.0. La Figura 2.3 muestra una comparación entre la forma en que funcionan las aplicaciones Web tradicionales y una aplicación que utiliza AJAX.

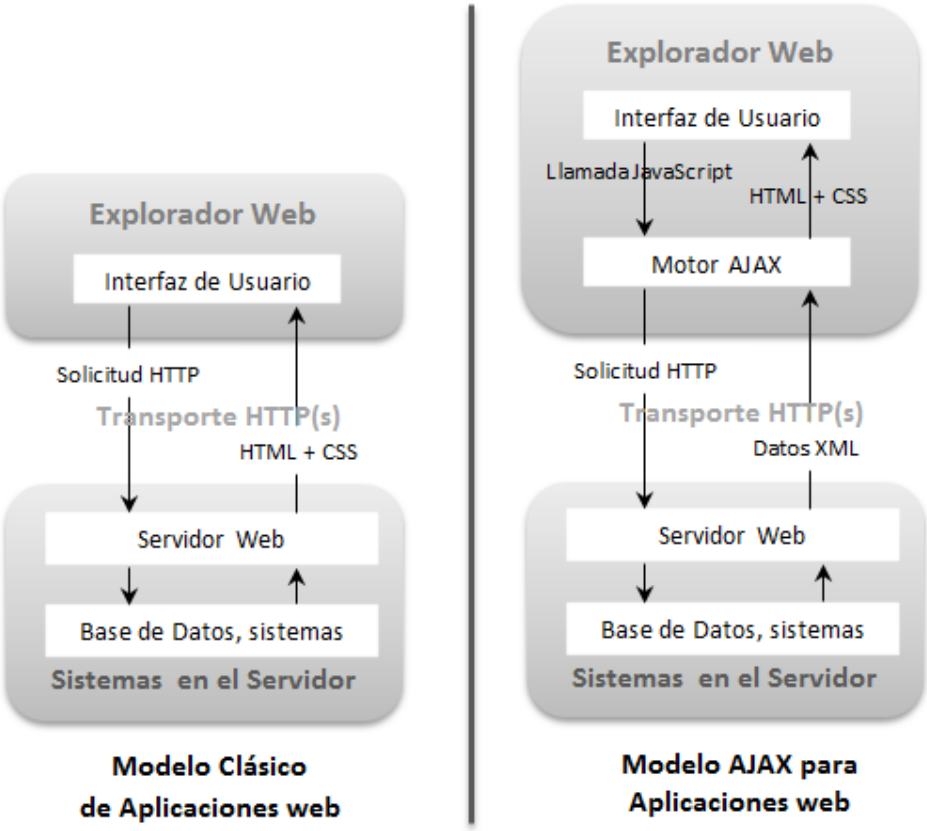


Figura 2.3 Comparación del modelo tradicional de aplicación Web y el propuesto por AJAX [40]

AJAX incorpora:

- HTML y CSS para la presentación de la página Web.
- DOM para proveer interacción dinámica.
- XML o texto plano para intercambio y manipulación de datos.

[40] Imagen obtenida de: "Ajax: A New Approach to Web Applications", Jesse James Garrett, Febrero 2005, http://www.adaptivepath.com/ideas/essays/archives/000385.php

- El objeto XMLHttpRequest[41] para recuperación asincrónica de datos.
- Y JavaScript para unir todas las tecnologías.

2.1.1.7. PHP

PHP, *Hypertext Preprocessor/Preprocesador de HyperTexto*, es un lenguaje de código abierto orientado al desarrollo de aplicaciones Web dinámicas con acceso a información almacenada en Base de Datos y que puede ser desplegado en la mayoría de los servidores Web y sistemas operativos sin costo alguno.

Las páginas Web que usan PHP se tratan igual que páginas HTML comunes y corrientes y se pueden crear o editar de la misma manera.

2.2. DESCRIPCIÓN DE LA SOLUCIÓN

El Servidor de Autenticación, Autorización y Auditoría Simplificado "SAAAS!", es una consola Web de administración y configuración para un servidor FreeRADIUS anidado en Debian Linux que aplica en su diseño los conceptos de la Web 2.0.

"SAAAS!" es una herramienta enfocada al personal encargado de redes de pequeña y mediana escala con poca experiencia en administración de servidores Linux pero familiarizado con el estándar 802.1x y el funcionamiento del protocolo RADIUS. La Figura 2.4 ilustra el Logotipo de la aplicación.

[41] Inicialmente implementado por Microsoft como un objeto ActiveX, permite que JavaScript realice peticiones HTTP a un servidor remoto sin necesidad de recargar la página web.

Figura 2.4 Logotipo de "SAAAS!"

Esta solución de software admite la integración del servidor RADIUS con Active Directory para el manejo de credenciales, garantiza la compatibilidad del servidor con el protocolo EAP, implementa políticas de acceso aplicadas a grupos de usuarios como asignación dinámica de VLANs e incluye un visor de eventos para monitoreo del servidor que permite la recuperación a fallas mediante una restauración de configuraciones anteriores.

“SAAAS!” permite una administración segura del servidor AAA mediante la autenticación de los usuarios de la aplicación Web, los cuales pueden pertenecer a un perfil para administración que permite modificar todos los parámetros del servidor, un perfil de operador con acceso limitado y un perfil de solo lectura.

El Servidor de AAA Simplificado está dividido en 6 secciones:

- Configuración básica del servidor.
- Administración de Certificados.
- Configuración del Servidor.
- Credenciales.
- Políticas de Acceso.
- Visor de Eventos.

Se describirá el diseño y funcionalidades de cada sección en el punto 2.5 de éste documento.

2.3. REQUERIMIENTOS DE LA SOLUCIÓN

2.3.1. SELECCIÓN DE LA DISTRIBUCIÓN BASE

Para la selección se tomó en cuenta las características de las distribuciones Linux más populares descritas en las Tablas 2.1, 2.2 y 2.3.

Características Generales						
Distribución	Creador	Año de Lanzamiento	Distribución base	Propósito	Soporte REMASTERSYS	Coste
CentOS	CentOS Project	2003	RHEL	Servidor	No	Gratuito
Debian	Ian Murdock	1993	No	Propósito general	Si	Gratuito
Fedora	Fedora Project	2003	Red Hat Linux	Propósito general	No	Gratuito
Mandriva Linux	Mandrakesoft S.A.	1998	Red Hat Linux	Propósito general	No	Gratuito
Red Hat Linux	Red Hat	1995	No	Servidor	No	Licenciado
Slackware	Patrick Volkerding	1993	SLS	Escritorio	No	Gratuito
Ubuntu	Canonical Ltd.	2004	Debian	Escritorio	Si	Gratuito

Tabla 2.1 Características Generales de las Distribuciones Linux más populares.

Arquitecturas soportadas										
Distribución	x86	x86-64	IA-64	sparc32	sparc64	hppa	mips	s390	s390x	Alpha
CentOS	Sí	Sí	Sí	No	No	No	No	Sí	Sí	Sí
Debian	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Fedora	Sí	Sí	No	No	No	No	No	No	No	No
Mandriva Linux	Sí	Sí	No	No	No	No	No	No	No	No
Red Hat Linux	Sí	Sí	Sí	No	No	No	No	No	Sí	No
Slackware	Sí	Sí	No	No	No	No	No	Sí	No	No
Ubuntu	Sí	Sí	Sí	No	No	No	No	No	No	No

Tabla 2.2 Arquitecturas de Hardware soportadas por algunas Distribuciones Linux.

Características Técnicas					
Distribución	Número de paquetes	Instalación gráfica	Gestor de ventanas	Gestor de paquetes	Formato de paquetes
CentOS	1660	Sí	GNOME	RPM, yum/up2date	RPM
Debian	23000	Sí	GNOME, KDE	APT	Deb
Fedora	8000	Sí	GNOME, KDE, Xfce	Yum	RPM
Mandriva Linux	23560	Sí	KDE	urpmi, RPMDrake23	RPM
Red Hat Linux	3000	Sí	GNOME	RPM,yum	RPM
Slackware	544	No	KDE, Xfce	installpkg, upgradepkg	Tgz
Ubuntu	23000	Sí	GNOME, Unity	APT	Deb

Tabla 2.3 Características Técnicas de las Distribuciones Linux más populares.

En base a las características descritas se realizó la matriz de decisión mostrada en la Tabla 2.4, en donde se escoge a la distribución Debian.

Factores	**Propósito**		**Coste**		**Múltiples Arquitecturas**		**Compatibilidad con Aplicaciones**		**Soporte Remastersys Meta-Distros**		**Total**
Pesos	3		2		4		5		4		
CentOS	3	9	3	6	2	8	1	5	1	4	32
Debian	3	9	3	6	3	12	3	15	3	12	54
Fedora	3	9	3	6	1	4	2	10	1	4	33
Mandriva Linux	3	9	3	6	1	4	3	15	1	4	38
Red Hat Linux	3	9	1	2	2	8	2	10	1	4	33
Slackware	1	3	3	6	1	4	1	5	1	4	22
Ubuntu	1	3	3	6	1	4	3	15	3	12	40

Tabla 2.4 Matriz de Decisión de la Distribución Base de "SAAAS!"

2.3.2. DEBIAN

"SAAAS!" ha sido diseñado para correr sobre un servidor DEBIAN GNU/Linux, por lo que no funcionará de manera correcta si se intenta anidarlo en otro sistema operativo.

Debian es una distribución Linux desarrollada en 1993 que se caracteriza por su estabilidad y compatibilidad con miles de paquetes de software pre compilado que facilitan la implementación de servicios de red y el desarrollo de software, razón por la cual fue el sistema operativo elegido como plataforma de "SAAAS!".

La versión 6.0 de Debian fue lanzada en marzo del 2011 con el nombre clave "Squeeze". La aplicación Web descrita en este documento se desarrolló sobre la actualización 6.0.1 de Debian Squeeze. La Tabla 2.4 muestra el listado de las versiones estables de Debian publicadas hasta el 2011.

Versión	Nombre Clave	Fecha de Lanzamiento	Paquetes Soportados
1.1	Buzz	Junio, 1996	474
1.2	Rex	Diciembre, 1996	848
1.3	Bo	Junio, 1997	974
2.0	Hamm	Julio, 1998	1500
2.1	Slink	Marzo, 1999	2250
2.2	Potato	Agosto, 2000	3900
3.0	Woody	Julio, 2002	8500
3.1	Sarge	Junio, 2005	15400
4.0	Etch	Abril, 2007	18000
5.0	Lenny	Febrero, 2009	25000
6.0	Squeeze	Marzo, 2011	29000

Tabla 2.5 Versiones de Debian GNU/Linux

2.3.3. FREERADIUS

FreeRADIUS es un paquete de código abierto distribuido bajo licencia GNU GPL[42] usado para la implementación del protocolo RADIUS. El paquete incluye servidor, clientes y librerías de desarrollo. Soporta todos los atributos especificados en los RFC 2865 y RFC 2866 e incluye módulos para soportar EAP, EAP-MD5, EAP-SIM, EAP-TLS, EAP-TTLS, EAP-PEAP y LEAP EAP de CISCO.

"SAAAS!" se desarrolló para trabajar con la versión 2.1.8 de FreeRadius lanzada el 30 de Diciembre del 2009. Para habilitar todas las funciones de "SAAAS!" se debe compilar de manera especial los paquetes de FreeRADIUS disponibles para Debian, y se debe agregar scripts en el servidor para el uso de grupos del Active Directory para la asignación de políticas. Estos procedimientos se describirán a fondo en el capítulo IV de este documento.

2.3.4. OPENSSL

OpenSSL es una implementación de código abierto de los protocolos SSL V2/V3[43] y TLS V1 que permite la creación de certificados digitales y provee funciones criptográficas a otras aplicaciones.

"SAAAS!" utiliza la versión 0.9.8g de OpenSSL para la generación de los certificados usados en la implementación de FreeRadius con autenticación EAP-TLS y EAP-TTLS.

---

[42] General Public Licence, licencia orientada a proteger la libre distribución, modificación y uso de software.

[43] Secure Sockets Layer/Protocolo de Capa de Conexión Segura, es un protocolo criptográfico que proporciona cifrado de datos, autenticación de servidores e integridad de mensajes.

2.3.4.1. La Licencia OpenSSL y GPL

OpenSSL utiliza una licencia que permite su libre distribución y uso bajo ciertas condiciones que no son compatibles con los términos de la licencia GPL.

La licencia GPL incluye una "excepción especial" que permite que un programa bajo GNU-GPL pueda incluir librerías incompatibles con esta licencia cuando forman parte del sistema operativo sobre el que corre la aplicación.

Para cumplir los términos de la licencia GPL, "SAAAS!" incluye la siguiente excepción en su licencia, tal y como recomienda el Proyecto OpenSSL[44]:

"This program is released under the GPL with the additional exemption that compiling, linking, and/or using OpenSSL is allowed."

"Este programa es liberado bajo los términos de la licencia GPL con la excepción adicional de que compilarlo, enlazarlo, y/o usarlo con OpensSSL está permitido."

2.3.5. MYSQL

MySQL es un motor de base de datos relacional, multihilo y multiusuario de código abierto distribuido y soportado por "Oracle Corporation" con licencia GNU GPL.

"SAAAS!" utiliza una base de datos MySQL Versión 5.0.51a-24+Squeeze4 (Debian) para el manejo de credenciales en base de datos local, para almacenar la información de los clientes, para recopilar logs del servicio RADIUS y para el registro de autenticación y permanencia de usuarios que se lleva a cabo durante el proceso de "Accounting" (Registro) del servicio AAA.

---

[44] En la página oficial del Proyecto OpenSSL: http://www.openssl.org/support/faq.html#LEGAL2.

2.3.6. SAMBA

Samba es un conjunto de programas distribuidos con licencia GNU GPL que permiten la interoperabilidad de Linux y Unix con sistemas operativos Windows.

Para que FreeRADIUS pueda usar Active Directory como repositorio para el manejo de credenciales, el servidor Linux, en el que se anida, debe pertenecer a un dominio Windows; "SAAAS!" utiliza la versión 3.2.5 de Samba para este propósito.

2.3.6.1. Winbind

Winbind es el componente de Samba que permite que un equipo Unix se convierta en miembro de un dominio NT, de tal forma que Linux reconozca a los usuarios y grupos del dominio Windows como si fueran nativos del sistema operativo.

"SAAAS!" usa el componente Winbind de Samba para que el servidor en el que se anida FreeRADIUS acceda a la información de usuarios y grupos existentes en un dominio Windows NT.

2.3.6.2. Ntlm_auth

Ntlm_auth es una herramienta que usa Winbind para evaluar solicitudes NTLM[45]. Esta herramienta permite verificar las credenciales de los usuarios en el controlador de dominio y devuelve mensajes de éxito o de error.

2.3.7. KERBEROS

Kerberos es un servicio de seguridad desarrollado por el MIT, *Massachusetts Institute of Technology/Instituto de Tecnología de Massachusetts*, que provee

[45] NTLM (NT LAN Manager) es un protocolo de autenticación del tipo Desafío/Respuesta utilizado en ambientes Windows.

autenticación, integridad y privacidad de datos a través del uso de cifrado de clave privada.

Kerberos V5 es la tecnología de autenticación por defecto utilizada en Active Directory de Windows Server 2003/2008 para proveer autenticación mutua. En ambientes mixtos con Active Directory de Windows 2000 se utiliza junto al protocolo NT LAN Manager.

“SAAAS!” utiliza Kerberos V5 para permitir el uso de Active Directory como repositorio de credenciales.

2.3.8. CRON

Cron es un demonio de sistemas operativos basados en Unix que permite programar tareas. En el presente proyecto, este demonio se utiliza para el almacenamiento de logs del servidor RADIUS en una base de datos.

2.3.9. APACHE

Apache es el servidor Web de código abierto sobre el que corre la aplicación Web desarrollada en este proyecto de titulación. El servidor HTTP Apache es un proyecto de *The Apache Software Foundation/Fundación de Software Apache*, y se encuentra en continuo desarrollo.

2.3.10. PHP5

Para el correcto funcionamiento de la aplicación Web se instaló la herramienta php5 y la librería php5-mysql para permitir la manipulación de la base de datos a través del explorador Web y la librería php5-ldap para realizar consultas al directorio de credenciales.

2.3.11. CURL

CURL es una herramienta de software libre para transferir archivos con sintaxis URL. Se utiliza en la aplicación desarrollada para la ejecución programada de scripts PHP mediante Cron.

2.4. CONSIDERACIONES DEL DISEÑO

La aplicación Web sigue la tendencia de AJAX, involucrando las siguientes tecnologías:

- HTML se utiliza para la presentación al usuario final. A pesar de utilizar código HTML, el formato entregado al cliente Web es .php para facilitar la manipulación de cookies de las sesiones de usuarios y permitir un manejo más dinámico de la hoja de estilo a la que se hace referencia en las páginas.
- PHP ejecuta las funciones requeridas en el lado del servidor.
- JavaScript se encarga de los efectos visuales y la comunicación asincrónica con el servidor, el código se ejecuta en el lado del cliente.

2.4.1. ESQUEMA BASE DE LA APLICACIÓN WEB

Para explicar el funcionamiento del código fuente de la aplicación se decidió utilizar diagramas lógicos basados en la Figura 2.3, la cual describe el modelo AJAX para aplicaciones Web. Además los diagramas de secuencia de la aplicación se pueden encontrar en el Anexo A de este documento.

Todas las páginas de “SAAAS!” manejan el esquema básico detallado en la Figura 2.5. El funcionamiento y la lógica con la que se trabaja en cada sección se detalla en el punto 2.5 “Secciones de la Aplicación Web”.

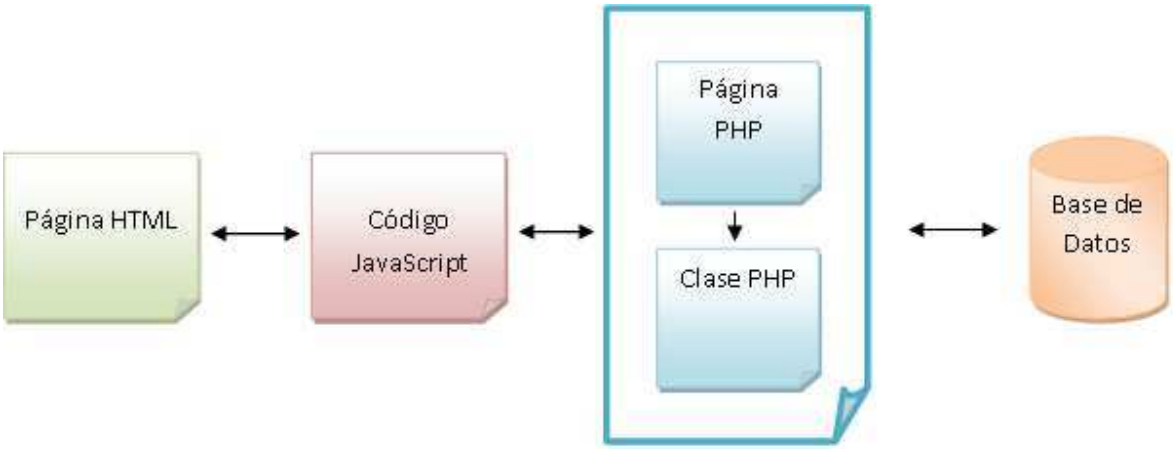


Figura 2.5 Lógica de la Aplicación

2.4.1.1. Página HTML

En las páginas HTML se encuentran objetos <*div*> que actuarán como contenedores de código HTML generado por el código JavaScript; adicionalmente contienen:

- Código para la comprobación de una cookie "Session" utilizada para la autenticación local con el servidor Web y la verificación del perfil al que pertenece el usuario autenticado.
- El enlace a la página de configuración general, que se encuentra en el directorio: *.../configuracion/config.php.* Este script provee información de la hoja de estilo escogido para toda la aplicación Web.

2.4.1.2. Código JavaScript

El código JavaScript ejecutará las funciones de llamada asincrónica al servidor (código PHP) utilizando el objeto XMLHttpRequest para leer y modificar el contenido de los objetos <*div*> presentes en la página HTML. Las funciones de llamada asincrónica son las siguientes:

- *crearXMLHttpRequest*

Crea el objeto ActiveXObject si el cliente Web es Internet Explorer o el objeto XMLHttpRequest para cualquier otro explorador.

- *enviarFormulario*
  Parámetros: datos, funcion, método, url.
  Llama a la función *crearXMLHttpRequest* para enviar los datos al servidor y define la función que se ejecutará una vez obtenida la respuesta.

Además el código JavaScript se encarga de manejar la presentación de los objetos <*div*> que para el usuario aparecen como pestañas en la interfaz Web. Las funciones utilizadas para este propósito son:

- Función activar
  Parametro: tab
  Cambia el atributo class del <*div*> (pestaña en la interfaz Web) a active. Además elimina las funciones sobre los atributos onmouseover y onmouseout.

- Función desactivar
  Parámetro: tab
  Cambia el atributo class del <*div*> que actuará como pestaña a al valor desactive, y activa la función *className='active'* sobre el atributo "onmouseover" de la pestaña,

2.4.1.3. Código PHP

En el código PHP se han escrito las funciones a ejecutarse en el servidor, como la conexión a la base de datos, la modificación de los archivos de configuración del servidor y los procesos de inicio de sesión de usuarios de la aplicación Web.

En todas las páginas se hace referencia a un conjunto de funciones definidas en el archivo *my_functions.php.*

2.4.1.4. Base de Datos

Se utiliza una base de datos MySQL para almacenar los logs del servidor, la información de inicio de sesión de los usuarios de la aplicación Web, las credenciales de los usuarios locales y demás información necesaria para el correcto funcionamiento de los procesos AAA.

FreeRADIUS provee un esquema estándar para la base de datos orientado a almacenar individualmente los atributos por usuario. Este esquema se ilustra en la Figura 2.6.

En la implementación de “SAAAS!” se modificó esta base de datos para ajustarla a una arquitectura orientada a perfiles y reglas de autenticación y autorización. Por otro lado, se incluyó tablas complementarias que almacenan la información de usuarios Web y logs del sistema. El esquema de base de la base de datos modificada se puede observar en la Figura 2.7.

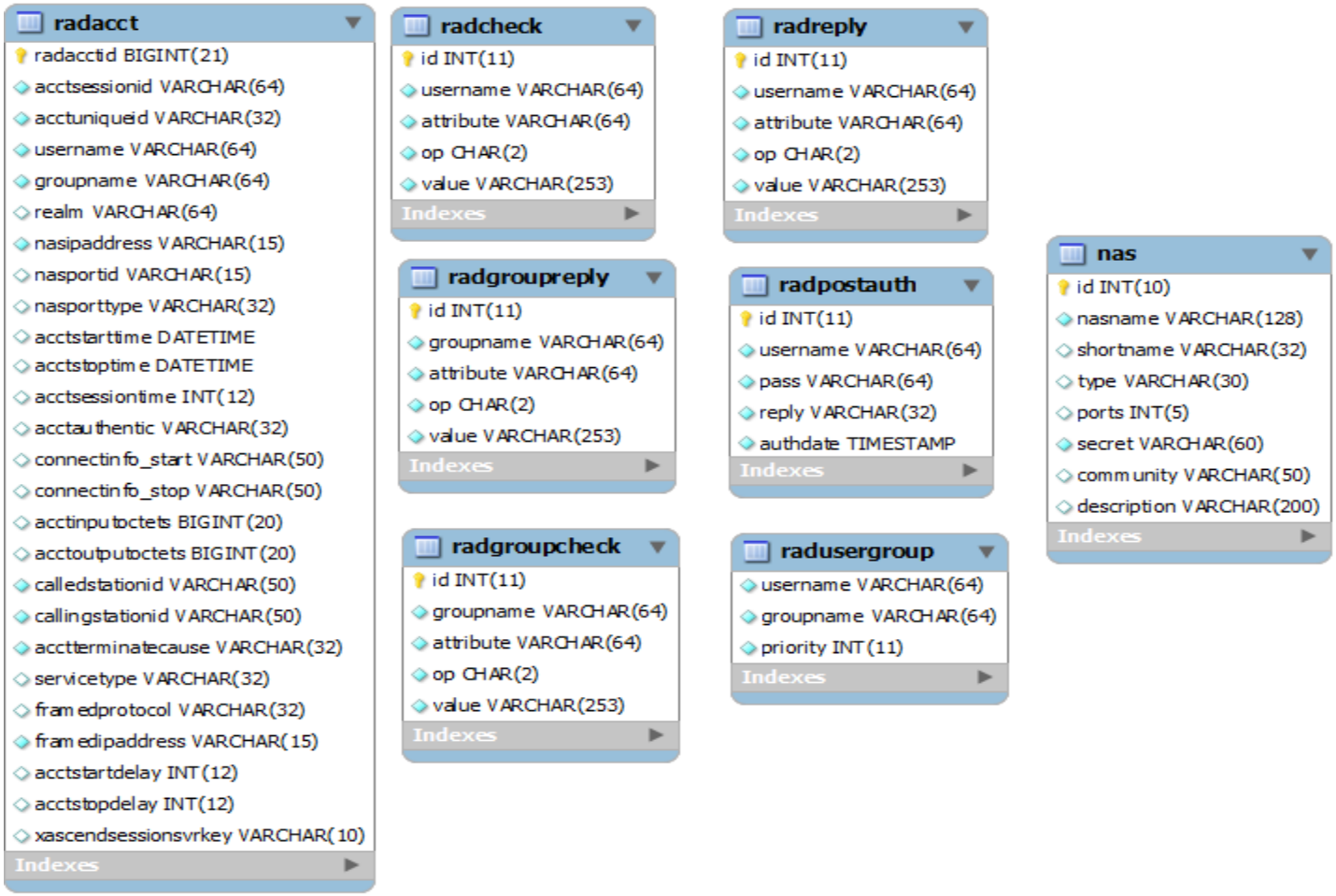


Figura 2.6 Esquema estándar de Base de Datos de FreeRADIUS

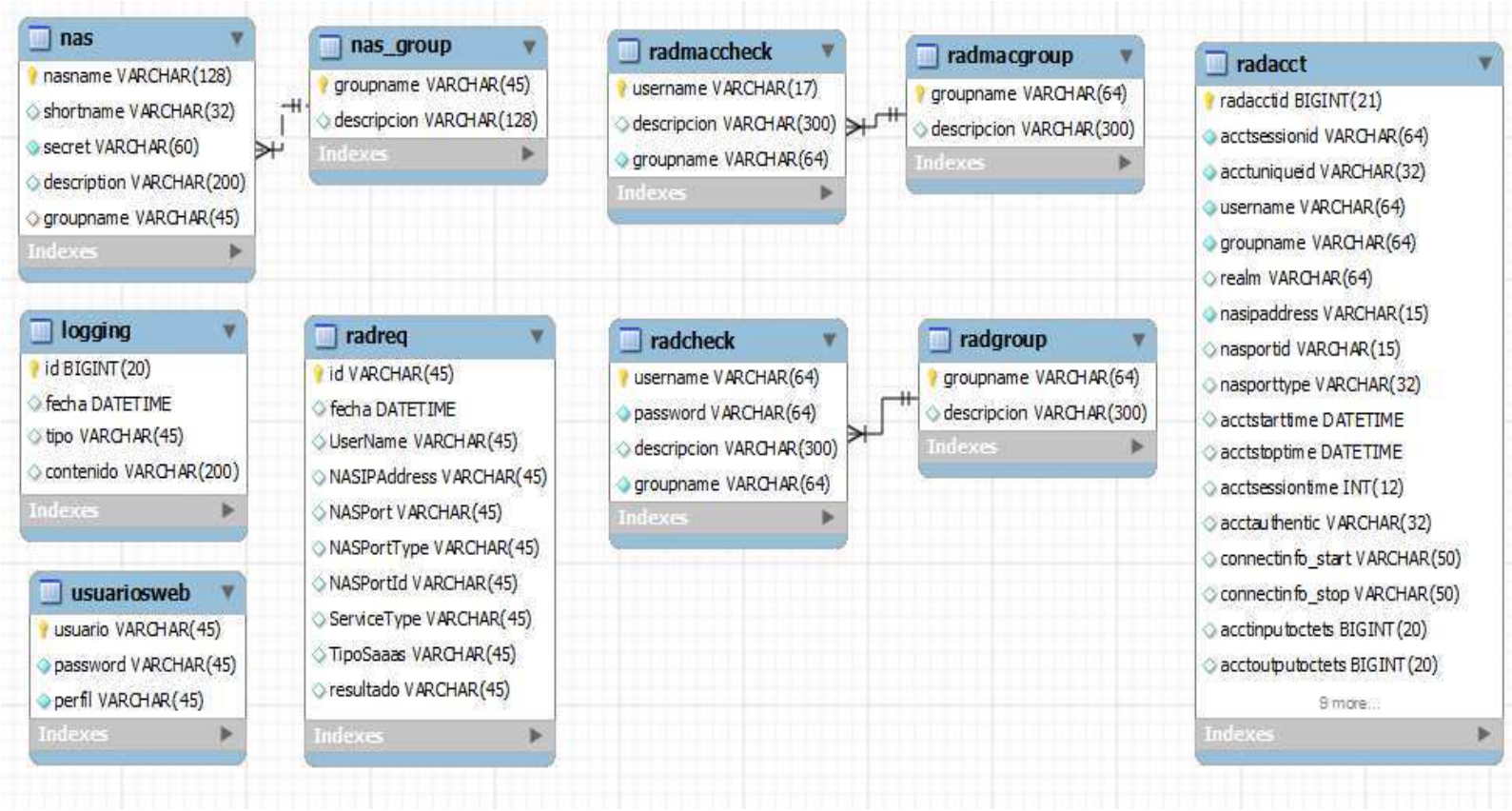


Figura 2.7 Esquema de Base de Datos de "SAAAS!"

Para la integración de FreeRADIUS con el nuevo esquema de la base de datos fue necesario modificar el archivo *dialup.conf* ubicado en el directorio */etc/freeradius/sql/mysql.* En este archivo se configuran las consultas que realizará el módulo SQL de FreeRADIUS para obtener la información de autenticación, autorización y auditoría.

Los cambios realizados en el archivo *dialup.conf* se detallan a continuación.

Original	**nas_query = "SELECT id, nasname, shortname, type, secret FROM ${nas_table}"**
Modificado	nas_query = "SELECT '1' as id, nasname, shortname,'other' as type, secret FROM ${nas_table}"

Espacio de Comandos 2.1 Cambios del archivo *dialup.conf* (nas_query)

FreeRADIUS lee las columnas secuencialmente, razón por la cual no se modificó el número ni el orden de los atributos consultados, tan solo se reemplazó las columnas innecesarias por valores por defecto, por ejemplo el campo id se puso por defecto el valor de "1". El parámetro por defecto escogido en la columna *"type"* es *"other"* debido a un error en el módulo de lectura de clientes de FreeRADIUS que provoca inestabilidad en el servicio si este valor es diferente. Estos cambios se observan en el Espacio de Comandos 2.1.

Original	**authorize_check_query = "SELECT id, username, attribute, value, op ** **FROM ${authcheck_table} ** **WHERE username = '%{SQL-User-Name}' ** **ORDER BY id"**
Modificado	authorize_check_query = "SELECT descripcion,username, 'Cleartext-Password' as attribute, password,':=' as op \ FROM ${authcheck_table} \ WHERE username = '%{SQL-User-Name}'"

Espacio de Comandos 2.2 Cambios del archivo *dialup.conf* (authorize_check_query)

En el Espacio de Comandos 2.2 se observa que los parámetros por defecto en las columnas *attribute* y *op* son *'Cleartext-Password* e *:=,* forzando a la tabla *"radcheck"*

a almacenar solo las contraseñas de usuario; los demás atributos serán adicionados a través de políticas de acceso, en la sección *Politicas de Acceso*.

Original	**authorize_reply_query = "SELECT id, username, attribute, value, op ** **FROM ${authreply_table} ** **WHERE username = '%{SQL-User-Name}' ** **ORDER BY id"**

Espacio de Comandos 2.3 Cambios del archivo *dialup.conf* (authorize_reply_query)

Se eliminó la consulta presentada en el Espacio de Comandos 2.3 debido a que la tabla a la que hace referencia no existe en el nuevo esquema.

Original	**group_membership_query = "SELECT groupname ** **FROM ${usergroup_table} ** **WHERE username = '%{SQL-User-Name}' ** **ORDER BY priority"**
Modificado	group_membership_query = "SELECT groupname \ FROM ${usergroup_table} \ WHERE username = '%{SQL-User-Name}'"

Espacio de Comandos 2.4 Cambios del archivo *dialup.conf* (group_membership_query)

En el Espacio de Comandos 2.4 se observa que se eliminó la sentencia de ordenamiento por prioridad en la *group_membership_query* ya que la columna *priority* no existe en el nuevo esquema. Adicionalmente, en el módulo de SQL no se hará referencia a la tabla *radusergroup* si no a la tabla *radcheck* a la cual se le añadió la columna *groupname*.

Original	**authorize_group_check_query = "SELECT id, groupname, attribute, ** **Value, op ** **FROM ${groupcheck_table} ** **WHERE groupname = '%{Sql-Group}' ** **ORDER BY id"**
Modificado	authorize_group_check_query = "SELECT descripcion, groupname, 'SqlGroup' as attribute,groupname, ':=' as op \

```
FROM ${groupcheck_table} \
WHERE groupname = '%{Sql-Group}'"
```

Espacio de Comandos 2.5 Cambios del archivo *dialup.conf* (authorize_group_check_query)

En el Espacio de Comandos 2.5 se observa la sentencia que lee los atributos del grupo al que el usuario pertenece razón por la cual se forzará al módulo SQL a leer la columna *groupname* y pasarlo al módulo de FreeRADIUS como el atributo *SqlGroup*, el cual se creó en el archivo */etc/freeradius/dictionary* para identificar los grupos de usuarios internos. La consulta se la realizará a la tabla *radgroup*.

2.4.2. AUTENTICACIÓN WEB Y PERFILES DE USUARIO DE LA APLICACIÓN

"SAAAS!" utiliza HTTPS para permitir un acceso seguro a la aplicación Web, de igual forma protege la administración del servidor mediante el uso de perfiles de autenticación definidos en la base de datos local. La Figura 2.8 ilustra la tabla de la base de datos en la que se almacena la información de autenticación de los usuarios Web.

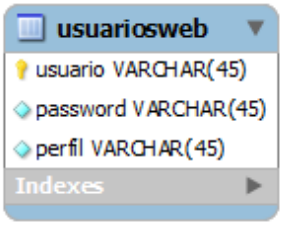


Figura 2.8 Tabla para Administración de Perfiles

Se definen tres perfiles con distintos privilegios de acceso, los cuales se detallan en la Tabla 2.6.

NOMBRE	DESCRIPCIÓN	PRIVILEGIOS
Administrador	Nivel de privilegios alto, permite configuración de todos los parámetros del servidor.	Todos
Operador	Nivel de privilegios orientado a las funciones de Soporte a Usuario. Tiene acceso a configuraciones básicas del servicio AAA.	Añadir, eliminar y modificar Clientes Radius (NAS). Iniciar, detener y reiniciar Servicios. Crear, eliminar y modificar usuarios internos (credenciales). Visor de Logs del sistema.
Monitoreo	Nivel de privilegios bajo, permite el control a nivel de auditoría para la generación de reportes.	Visor de Logs del sistema

Tabla 2.6 Perfiles de Usuario de la Aplicación Web

2.5. SECCIONES DE LA APLICACIÓN WEB

2.5.1. CONFIGURACIÓN GENERAL

En esta sección se detalla la información básica del servidor (nombre de host, dirección IP, estado de la memoria, etc) y también permite realizar la configuración de las interfaces de red (dirección IP, gateway), el nombre del equipo, el DNS, etc.

2.5.1.1. Información Básica

Como se ilustra en la Figura 2.9, esta página muestra la dirección IP del servidor “SAAAS!”, el kernel, el sistema operativo, el porcentaje de uso de la red, las características de hardware y el porcentaje de memoria utilizado.

Figura 2.9 Página Información Básica del servidor "SAAAS!"

Para esta página se adaptó la versión 3.0.7 del proyecto PHPSysinfo con el objetivo de mostrar información de hardware, memoria y uso de red del equipo sobre el que se ejecuta "SAAAS!". PHPSysInfo es un script PHP distribuido bajo los términos de la licencia GNU-GPL que muestra información del equipo sobre el que se lo ejecuta. La Figura 2.10 muestra una captura de la aplicación.

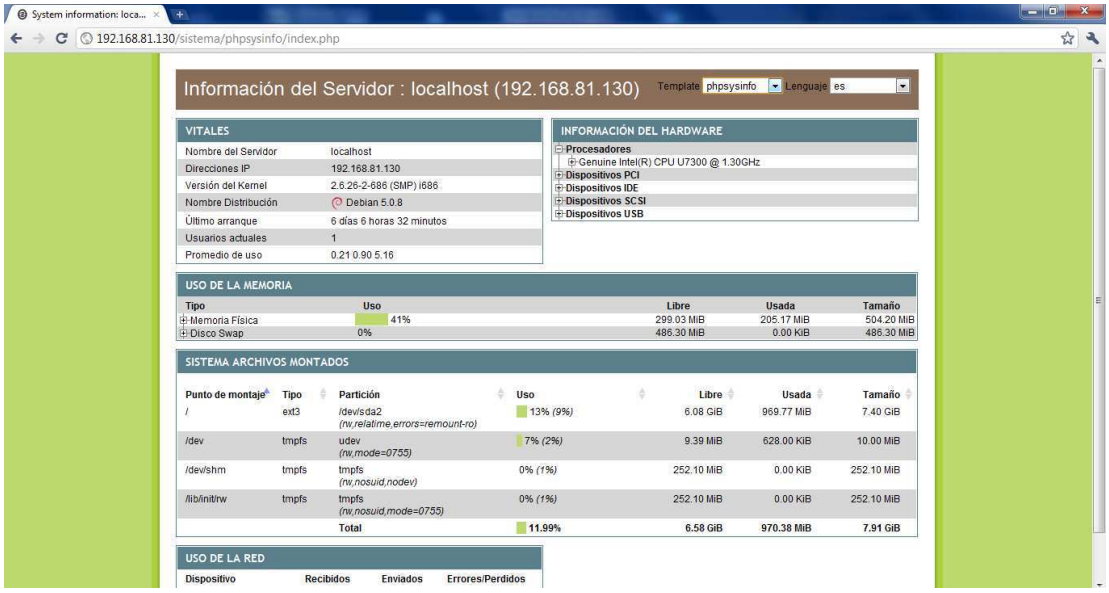


Figura 2.10 Información del Sistema mostrada por PHPSysInfo

La función *run()* de la clase WebpageXML, cuyo código se encuentra en el archivo *class.WebpageXML.inc.php* de PHPSysInfo, transforma un objeto *SimpleXMLElement* de PHP a formato XML y lo muestre en el explorador. Para evitar el tiempo de procesamiento que representaba esta acción, en el código de "SAAAS!" se añadió una nueva función a la clase llamada *run2()* la cual retorna el objeto *SimpleXMLElement.* El Espacio de Comandos 2.6 muestra la función que se añadió.

Original

```
public function run()
  {
    header("Cache-Control: no-cache, must-revalidate\n");
    header("Content-Type: text/xml\n\n");
    $xml = $this->_xml->getXml();
    echo $xml->asXML();
  }
```

Nuevo

```
public function run2(){
    return $this->_xml->getXml();
  }
```

Espacio de Comandos 2.6 Cambios del archivo *class.WebpageXML.inc.php*

2.5.1.1.1. Lógica de la Sección.

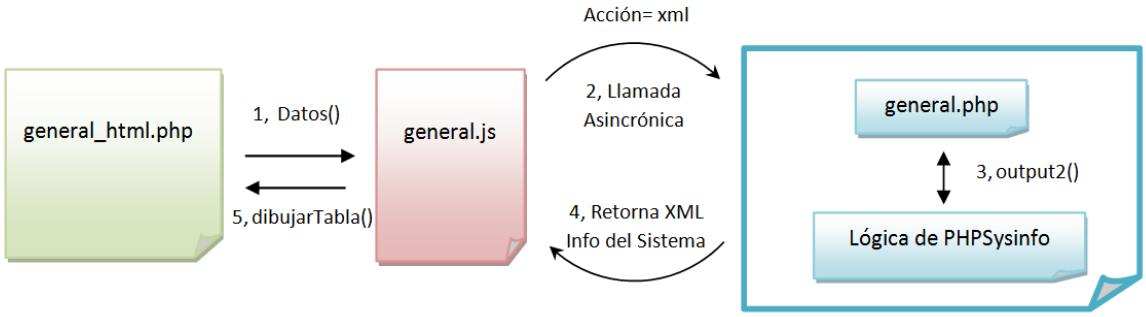


Figura 2.11 Lógica de la Sección Configuración General

Como se observa en la Figura 2.11, al cargarse por primera vez, la página *general_html.php* llama a la función *Datos()* alojada en el código JavaScript del

archivo *general.js.* Esta función realiza una llamada asincrónica al archivo *general.php,* enviando a través del método POST el valor de una variable llamada *acción*. El código del archivo *general.php* se ejecuta de acuerdo al tipo de acción solicitada.

Si la acción solicitada es: *XML* en el archivo *general.php* se llama a la clase WebpageXML de PHPSysInfo para entregar la información básica del servidor en formato XML al código JavaScript que inició la llamada asincrónica. Finalmente, al recibir la información solicitada el código JavaScript crea objetos HTML para mostrar esta información en la página *general_html.php*, el caso de uso para esta acción se lo presenta en el Anexo B1.

2.5.1.2. Recursos de Red

La página Recursos de Red del servidor “SAAAS!” permite visualizar y modificar los aspectos básicos de conectividad a la red del sistema. Como se observa en la Figura 2.12, en la pestaña *General* se puede cambiar el nombre del equipo, el dominio al que pertenece y se puede configurar los servidores DNS con los que trabajará.

Figura 2.12 Página Recursos de Red/Pestaña: General de "SAAAS!"

Como se observa en la Figura 2.13, en la pestaña *Interfaces de Red* se puede visualizar el estado de las interfaces de red físicas y virtuales del servidor. Además se puede modificar las direcciones IP asignadas a las mismas y eliminar interfaces virtuales.

Figura 2.13 Página Recursos de Red/Pestaña: Interfaces de Red de "SAAAS!"

Como se observa en la Figura 2.14 al activar la pestaña *Rutas* se muestra la tabla de enrutamiento que maneja el servidor. En vista de que “SAAAS!” provee servicios AAA y no actúa como proxy o enrutador no es necesario que el usuario pueda agregar rutas en esta tabla.

Figura 2.14 Página "Recursos de Red"/Pestaña: Rutas del servidor "SAAAS!"

La Figura 2.15 muestra el contenido que se despliega al activar la pestaña *Nueva Interfaz Virtual.* Esta pestaña permite crear interfaces virtuales asociadas a las interfaces físicas instaladas en el servidor.

Figura 2.15 Página Recursos de Red/Pestaña: Nueva Interfaz Virtual del servidor "SAAAS!"

2.5.1.2.1. Detalle de Clases

La página Recursos de Red utiliza tres clases para mostrar y modificar la configuración de red del servidor. La clase *manejodeIP* tiene como único atributo un arreglo denominado *$mascaras,* que contiene el detalle de las máscaras de red válidas para una dirección IP. Las funciones de esta clase proveen la lógica y el procesamiento para calcular un direccionamiento de red válido y aplicable a una interfaz.

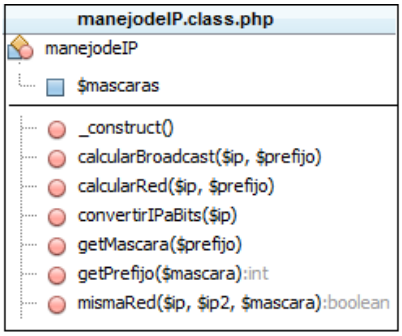


Figura 2.16 Clase: manejodeIP utilizada en la página Recursos de Red

La clase *interfazAdmin* tiene como atributos el texto *$archivoInterfaces,* el array *$interfaces,* y la constante *PATH_INTERFACES.* Esta clase permite la interacción del sistema con los servicios y ficheros de configuración de interfaces de red del servidor; en particular el archivo */etc/network/interfaces*.

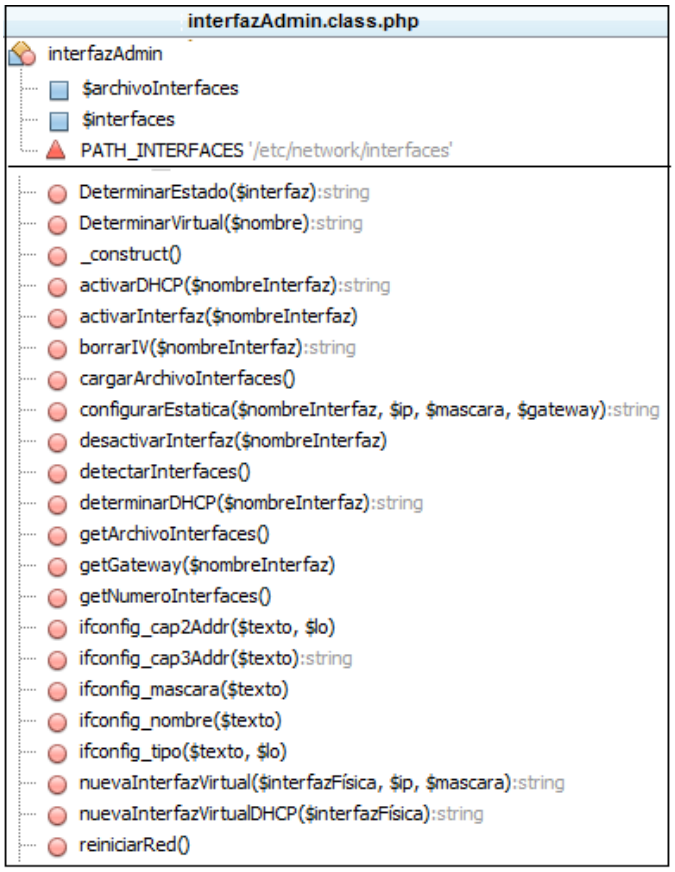


Figura 2.17 Clase: interfazAdmin utilizada en la página Recursos de Red

La clase *sistema* tiene como atributos las constantes PATH_DHCP, PATH_DNS, PATH_HOSTNAME y PATH_HOST, que representan la ubicación y el nombre de los archivos de Debian involucrados en la configuración del nombre del equipo, dominio, servidores DNS del equipo. Esta clase permite la configuración de los elementos visibles en la pestaña *General* de la página Recursos de Red.

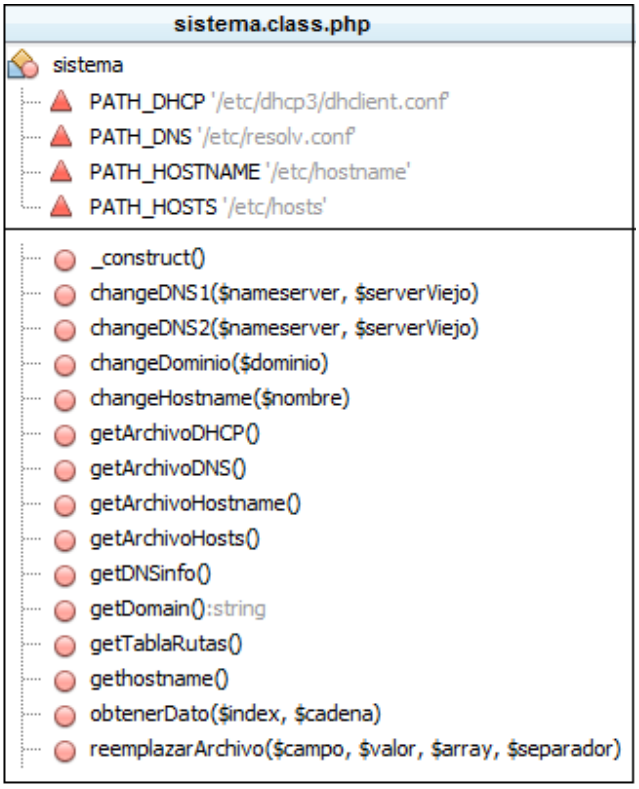


Figura 2.18 Clase: sistema utilizada en la página Recursos de Red

2.5.1.2.2. Lógica de la Sección

La Figura 2.19 muestra que la página *sistema_html.php,* al cargarse por primera vez llama a la función *activeTab(1)* alojada en el código JavaScript del archivo *sistema.js.* Esta función realiza una llamada asincrónica al archivo *sistema.php* enviando a través del método POST la variable accion con el valor “general”.

En el archivo *sistema.php* se ejecutan las funciones necesarias para recuperar la información de red básica del servidor y se la entrega en formato XML al código JavaScript que inició la llamada asincrónica. Al recibir la información solicitada el código JavaScript crea objetos HTML para mostrar esta información en la primera pestaña de la página *sistema_html.php.*

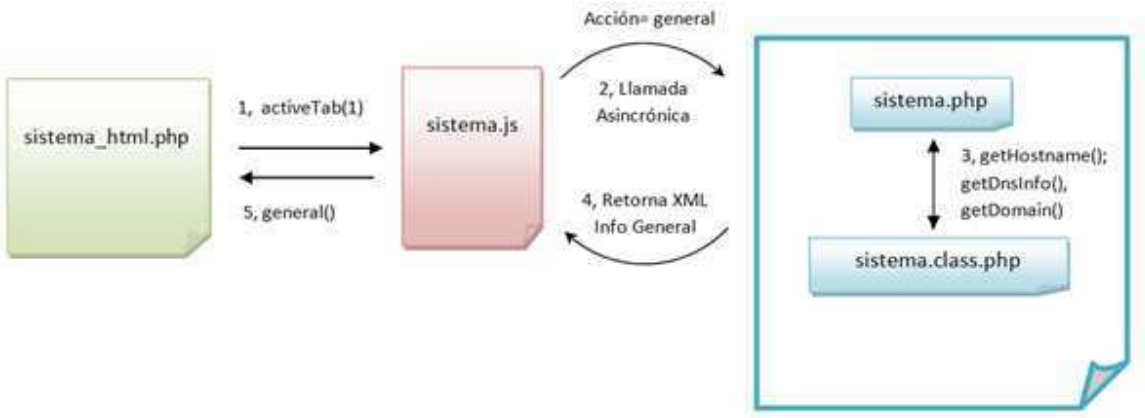


Figura 2.19 Lógica de la acción: Mostrar pestaña General

Si el usuario desea modificar las características del servidor, se ejecuta una serie de funciones de acuerdo al parámetro a modificarse. La Figura 2.20 detalla la lógica que se ejecuta en la aplicación cuando el usuario cambia el nombre de host del servidor.

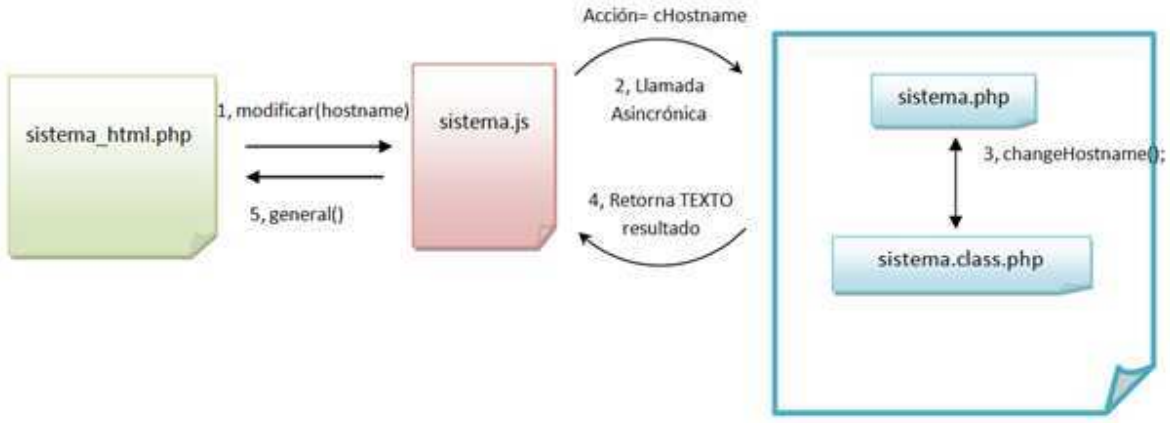


Figura 2.20 Lógica de la acción Cambiar nombre de host

La Figura 2.21 muestra la lógica que se ejecuta cuando el usuario cambia el nombre de dominio del servidor.

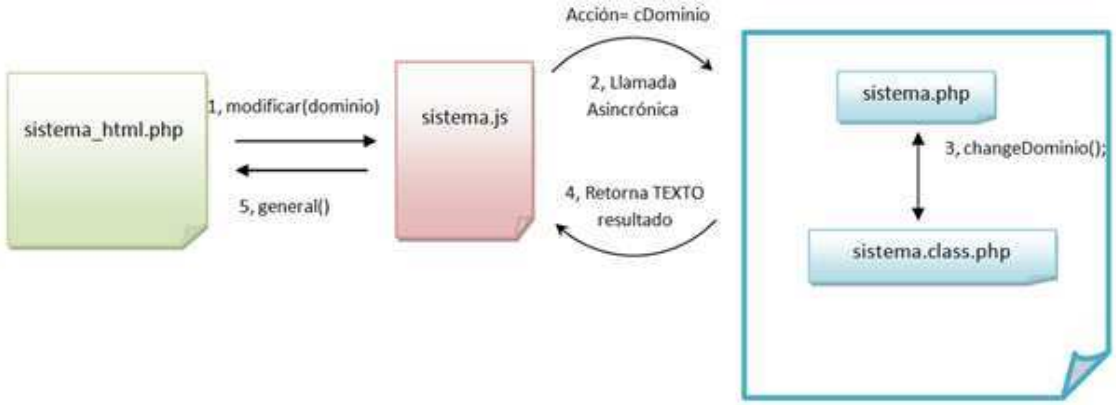


Figura 2.21 Lógica de la acción: Cambiar Dominio

Las Figuras 2.22 y 2.23 muestran la lógica que se ejecuta en la aplicación cuando el usuario modifica la información de DNS primario y secundario.

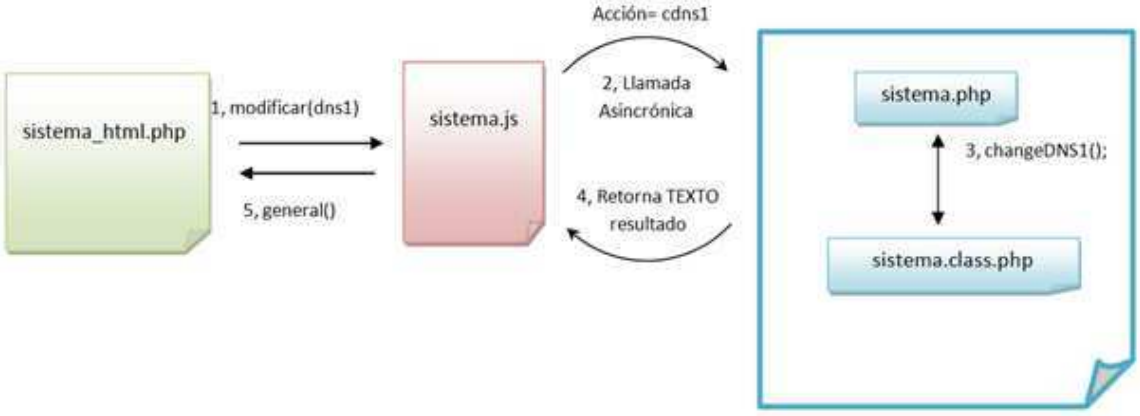


Figura 2.22 Lógica de la acción: Cambiar DNS Principal

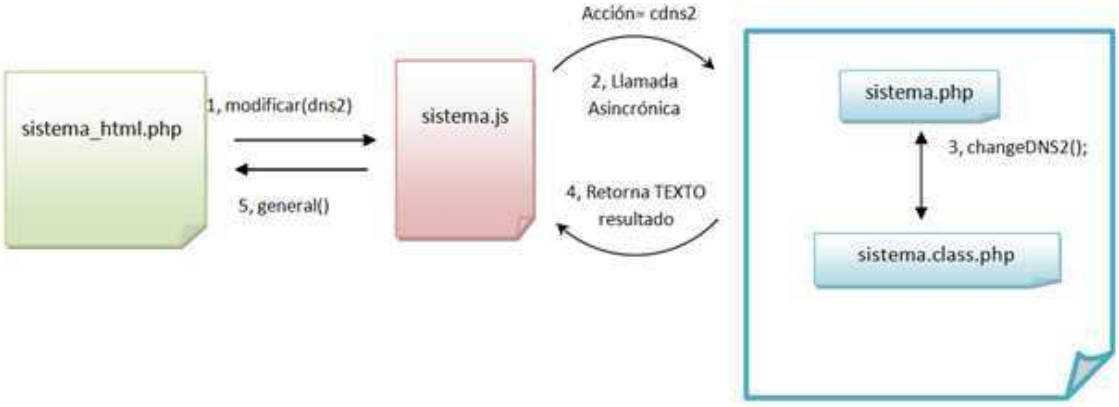


Figura 2.23 Lógica de la acción: Cambiar DNS Secundario

La Figura 2.24 ilustra la lógica que se ejecuta en la aplicación Web cuando un usuario hace clic sobre la pestaña *Interfaces de Red.* La página *sistema_html.php* llama a la función *activeTab(2)* alojada en el código JavaScript del archivo *sistema.js.* Esta función realiza una llamada asincrónica al archivo *sistema.php* enviando a través del método POST la variable accion con el valor "red".

En el archivo *sistema.php* se verifica la variable acción y al coincidir con "red" se ejecutan las funciones necesarias para recuperar información del estado y configuración de las interfaces de red físicas y virtuales del sistema. Esta información se entrega en formato XML al código JavaScript que inició la llamada asincrónica. Al recibir la información solicitada el código JavaScript crea objetos HTML para mostrar una tabla con los datos de las interfaces de red en la segunda pestaña de la página *sistema_html.php.*

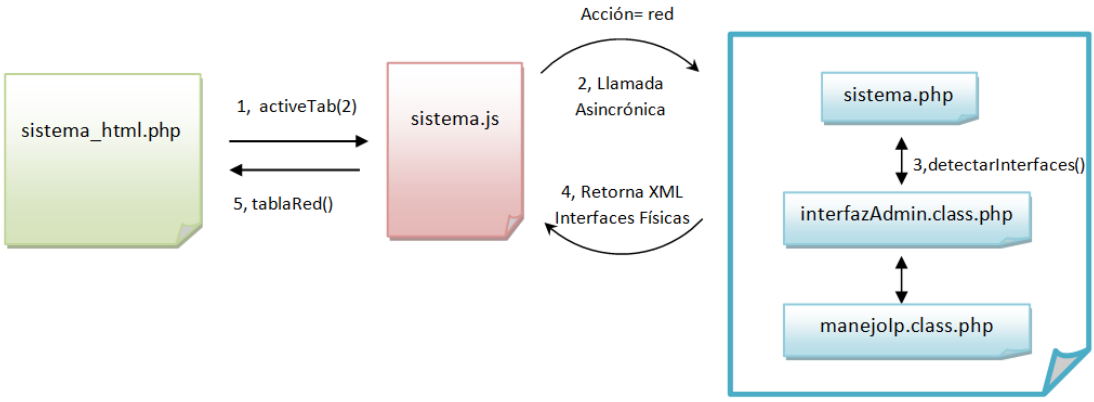


Figura 2.24 Lógica de la acción: Mostrar información de Interfaces de Red

La Figura 2.25 muestra las tareas que se llevan a cabo en la aplicación cuando un usuario configura una interfaz con direccionamiento estático. La página *sistema_html.php* llama a la función *modInterfaz(i)* del archivo *sistema.js.* Esta función realiza una llamada asincrónica al archivo *sistema.php* enviando a través del método POST la variable accion con el valor "cambiarEstatica" y las variables necesarias para ejecutar la función *configurarEstatica()* de la clase *manejoIP* que

retorna un mensaje de resultado en formato de texto al código para mostrarse como alerta al usuario, el caso de uso para esta acción se lo presenta en el Anexo B2.

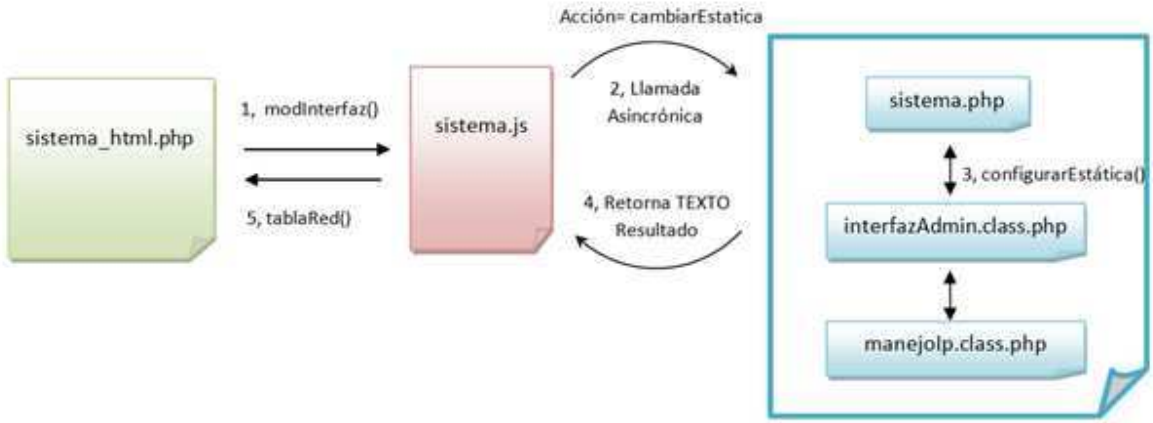


Figura 2.25 Lógica de la función: Configurar Direccionamiento Estático en una Interfaz de Red

Cuando el usuario desea activar direccionamiento dinámico en la interfaz, la función *modInterfaz(i)* del archivo *sistema.js envía* la variable accion con el valor "activarDHCP" para que la llamada asincrónica ejecute la función *activarDHCP()* de la clase *manejoIP.* Estos procedimientos se ilustran en la Figura 2.26.

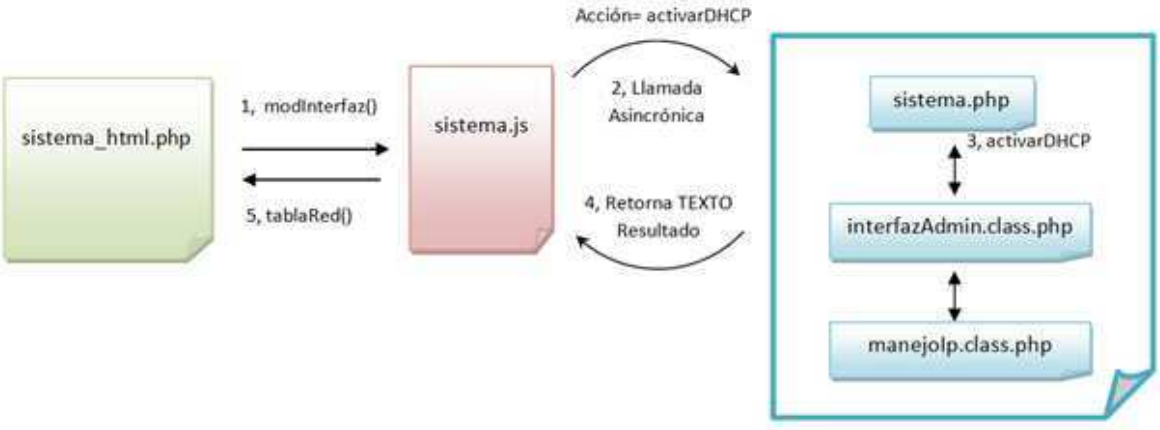


Figura 2.26 Lógica de la función: Configurar direccionamiento dinámico en una Interfaz de Red

La Figura 2.27 muestra la lógica que se ejecuta en la aplicación cuando un usuario desea Activar o Desactivar una interfaz de red[46]. Para cambiar el estado de una interfaz de activo a inactivo, o viceversa; la página *sistema_html.php* llama a la función *cambiarEstado(i)* del archivo *sistema.js.* Esta función crea un objeto XMLHttpRequest para realizar una llamada asincrónica al archivo *sistema.php* enviando a través del método POST la variable accion con el valor "activar" o "desactivar" dependiendo del estado de la interfaz y se ejecuta la función *activarInterfaz()* o *desactivarInterfaz()* de la clase *manejoIP*, el caso de uso para esta acción se lo presenta en el Anexo B4.

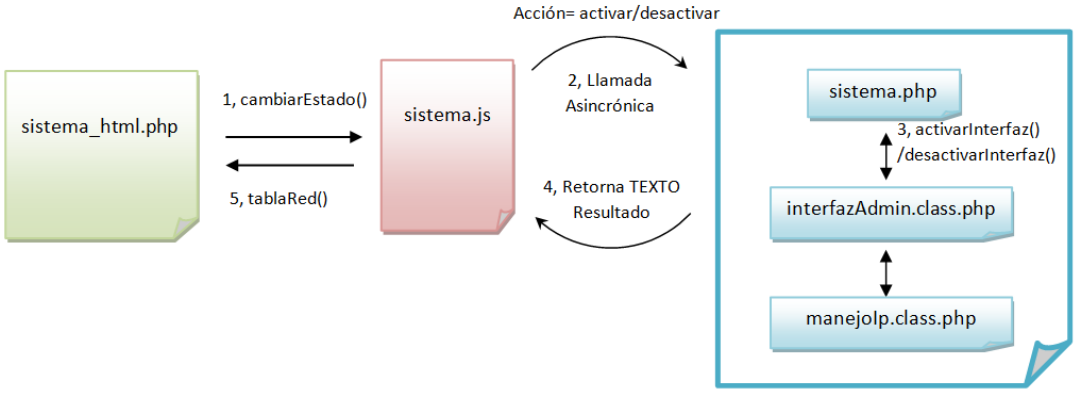


Figura 2.27 Lógica de las funciones: Activar/Desactivar una interfaz de red

Si el usuario hace clic sobre la pestaña *Rutas,* la página *sistema_html.php* llama a la función *activeTab(3)* del archivo *sistema.js.* Esta función realiza una llamada asincrónica al archivo *sistema.php* enviando a través del método POST la variable accion con el valor "rutas". En el archivo *sistema.php* se verifica la variable acción y al coincidir con "rutas" se ejecuta la función *getTablaRutas()* de la clase *sistema* que retorna un array con la información de la tabla de rutas del sistema. Esta información se entrega en formato XML al código JavaScript que inició la llamada asincrónica. Al recibir la información solicitada el código JavaScript crea objetos HTML para mostrar la tabla de rutas en la tercera pestaña de la página *sistema_html.php.* Estos

[46] Esta opción solo se activa en servidores que cuentan con más de una interfaz física.

procedimientos se ilustran en la Figura 2.28, el caso de uso para esta acción se lo presenta en el Anexo B5.

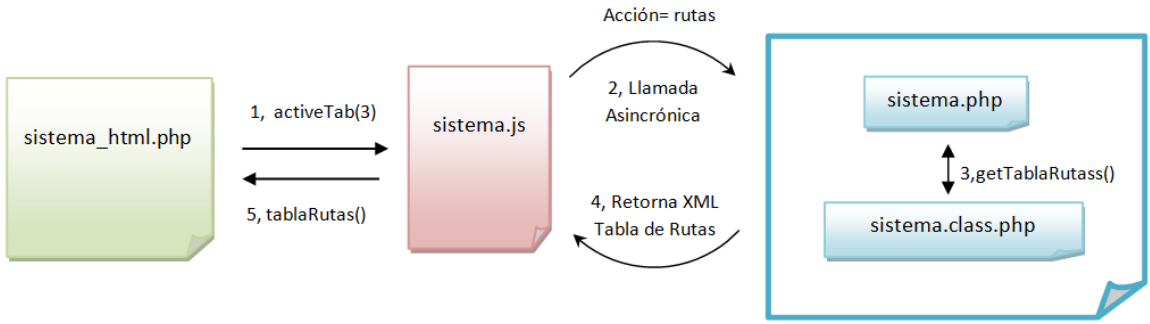


Figura 2.28 Lógica de la acción: Mostrar Tabla de Rutas

La Figura 2.29 detalla la lógica que se ejecuta en la aplicación cuando un usuario hace clic sobre la pestaña Nueva Interfaz Virtual. La página *sistema_html.php* llama a la función *activeTab(4)* del archivo *sistema.js.* Esta función realiza una llamada asincrónica al archivo *sistema.php* enviando a través del método POST la variable accion con el valor “IV”.

En el archivo *sistema.php* se verifica la variable acción y al coincidir con “IV” se ejecuta la función *detectarInterfaces()* de la clase *interfazAdmin* para recuperar la lista de interfaces de red físicas en el servidor. Esta información se entrega en formato XML al código JavaScript que inicio la llamada asincrónica. Al recibir la información solicitada el código JavaScript crea objetos HTML para mostrar un formulario para la creación de una nueva interfaz virtual en la cuarta pestaña de la página *sistema_html.php.*

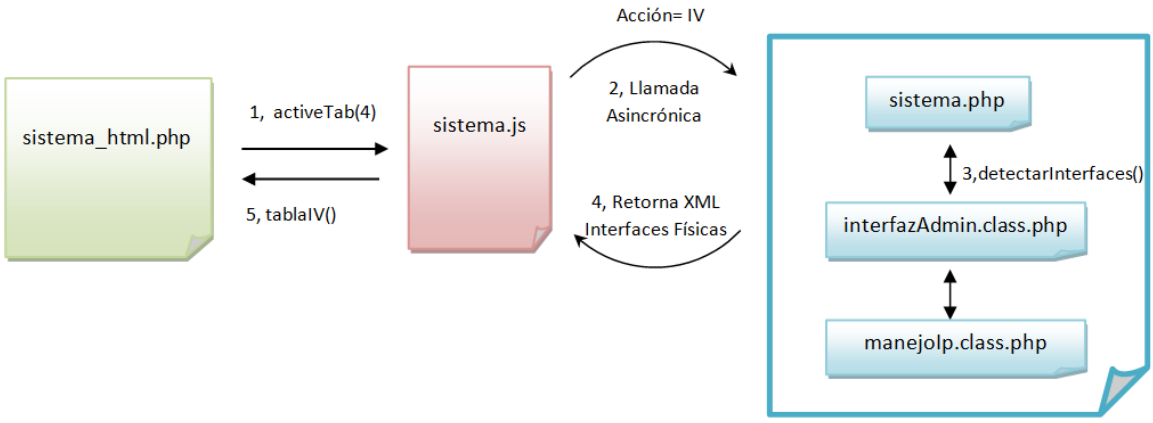


Figura 2.29 Lógica de la función: Mostrar Formulario para Nueva Interfaz Virtual

Para crear un Interfaz Virtual, el usuario pulsa sobre el botón guardar del formulario de la página *sistema_html.php*, esta acción realiza una llamada asincrónica al archivo *sistema.php* enviando a través del método POST la variable acción con el valor "crearIVEstatica" y las variables necesarias para ejecutar la función *nuevaInterfazVirtual()* de la clase *interfazAdmin* que retorna un mensaje de resultado en formato de texto al código para mostrarse como alerta al usuario. Estos procedimientos se ilustran en la Figura 2.30, el caso de uso para esta acción se lo presenta en el Anexo B6.

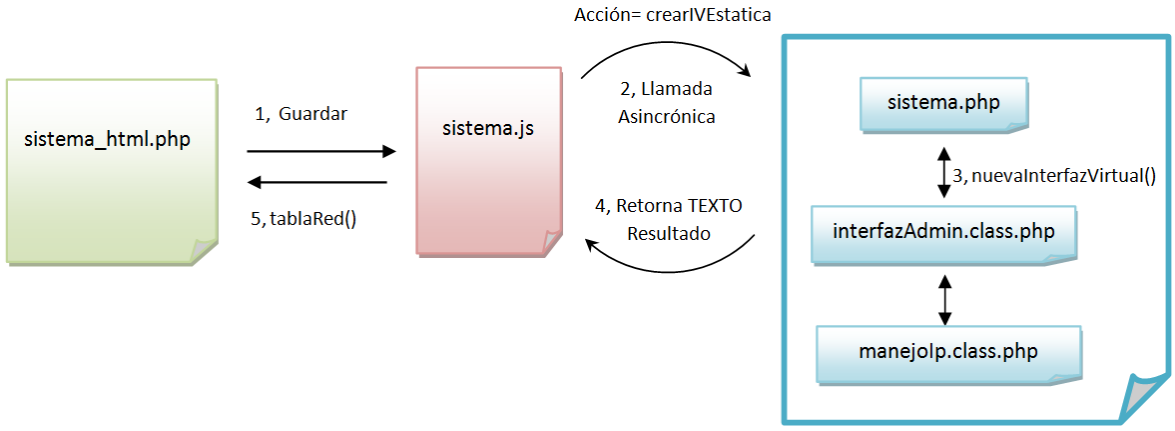


Figura 2.30 Lógica de la sesión: "Crear nueva Interfaz Virtual"

2.5.2. ADMINISTRACIÓN DE CERTIFICADOS

Ciertos protocolos de autenticación necesitan de la generación de certificados digitales para funcionar. La sección Administración de Certificados del servidor "SAAAS!" permite la creación de una estructura de certificados local para la creación de certificados auto firmados y la configuración de los protocolos utilizados en la etapa de autenticación del servicio AAA.

2.5.2.1. Configuración Básica

La página Configuración Básica de la sección Administración de Certificados se ilustra en la Figura 2.31. En ésta página se empieza a crear la estructura de certificados local, al llenar el formulario y presionar enviar se ejecuta en script que crea la Autoridad Certificadora.

Figura 2.31 Página Configuración Básica de la sección Administración de Certificados del servidor "SAAAS!"

2.5.2.1.1. Lógica de la Sección

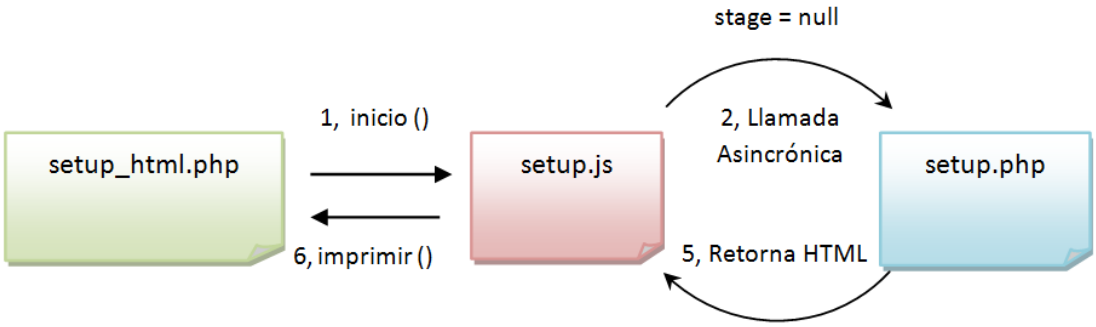


Figura 2.32 Lógica de la acción: Mostrar Formulario para crear Autoridad Certificadora

Cuando se carga por primera vez la página Configuración Básica de la sección Administración de Certificados se ejecuta la función *inicio()* del código JavaScript del archivo *setup.js* la cual realiza una llamada asincrónica a la página *setup.php* transmitiendo a través del método POST la variable stage con valor nulo. Al recibir esta información, la página *setup.php* retorna el código HTML que permitirá que se muestre el formulario de creación de la Autoridad Certificadora en el explorador después de que se ejecuta la función *imprimir()*.

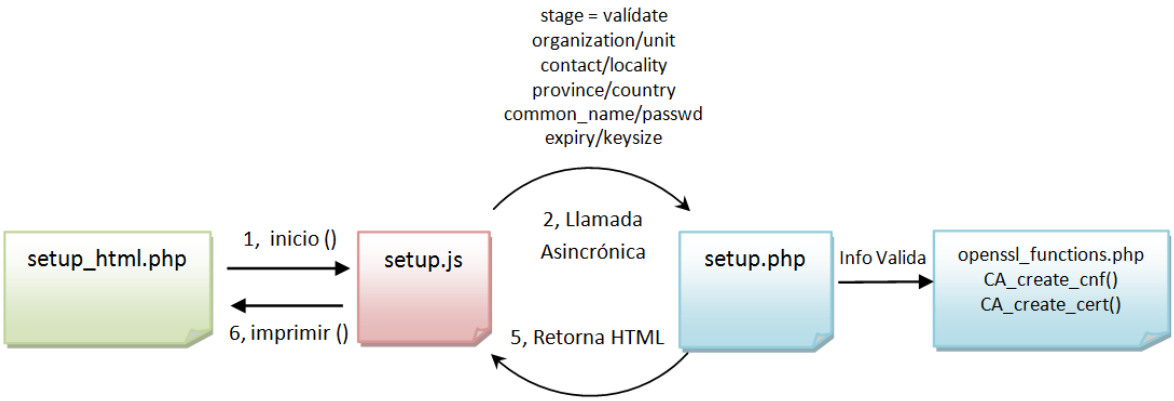


Figura 2.33 Lógica de la acción: Crear Autoridad Certificadora en el Servidor "SAAAS!"

Cuando un usuario envía el formulario de creación de la Autoridad Certificadora, se inicia una nueva llamada asincrónica hacia la página *setup.php,* esta vez con el valor "validate" en la variable stage y el resto de variables necesarias para crear la AC. Al inicio, el servidor verifica que la información recibida tenga el formato correcto y de ser así ejecuta las funciones *CA_create_cnf()* y *CA_create_cert()* que se encargan de guardar los datos de la Autoridad Certificadora, generar su clave privada, el certificado de la Autoridad Certificadora y el certificado autofirmado para distribución de su clave pública, tal como se ilustra en la Figura 2.33, el caso de uso para esta acción se lo presenta en el Anexo B7.

2.5.2.2. Certificados

La página Certificados de la sección: Administración de Certificados. Maneja la creación, visualización, revocación, descarga y eliminación de los certificados.

La Figura 2.34 ilustra la página Certificados Esta página muestra una lista de los certificados digitales creados en el servidor, además se pueden crear nuevos certificados y revocar, visualizar y descargar los certificados existentes.

Figura 2.34 Página Certificados de la sección Administración de Certificados del servidor "SAAAS!"

2.5.2.2.1. Lógica de la Sección

Al cargarse por primera vez, la página *certificados_html.php* llama a la función *inicio()* alojada en el código JavaScript del archivo *certificados.js.* Esta función realiza una llamada asincrónica al archivo *certificados.php* enviando a través del método POST el valor de una variable stage, el código del archivo *certificados.php* se ejecuta de acuerdo al tipo de acción solicitada.

Como se puede observar en la Figura 2.35, si el valor de la variable stage es “null” en el archivo *certificados.php* se ejecuta la función por defecto *mostrar()*, esta función genera el código HTML necesario para crear la tabla de visualización de la estructura de los certificados, previamente recupera el registro de los certificados del archivo */var/www/openssl/cert/CA/index*, y genera el menú inferior (Nuevo, Ver, Descargar, Revocar y Renovar) apuntando a las funciones JavaScript correspondientes para la manipulación de estos registros , el caso de uso para estas acciones se lo presenta en el Anexo B8.

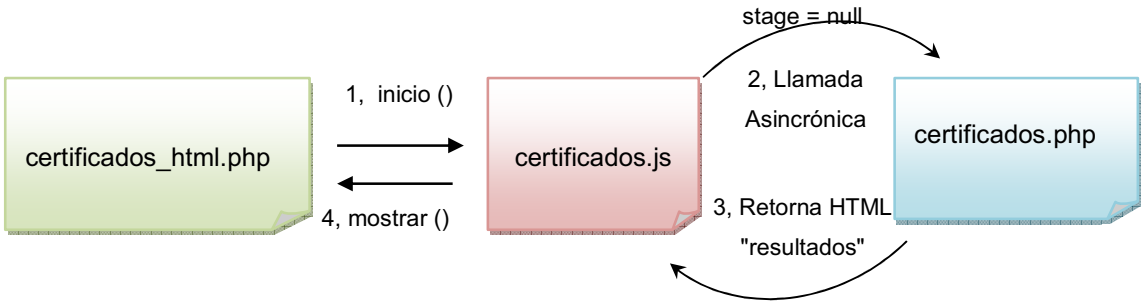


Figura 2.35 Lógica de la acción por defecto

En la Figura 2.36 se puede observar la lógica que se ejecuta en la aplicación cuando el usuario presiona el botón *Nuevo*. La página *certificados_html.php* llama a la función *enviar(),* alojada en el código JavaScript del archivo *certificados.js.* Esta función realiza una llamada asincrónica al archivo *certificados.php* enviando a través del método POST la variable stage con el valor “nuevo”.

Cuando la variable stage tiene el valor “nuevo”; el archivo *certificados.php* ejecuta la función por defecto *mostrar()*, y se genera el código HTML, que muestra al usuario campos en blanco que deben ser llenados con la información del certificado a crear.

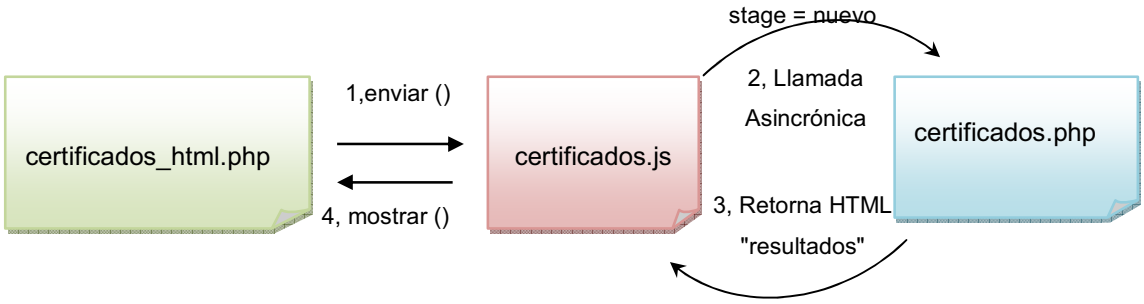


Figura 2.36 Lógica de la acción Nuevo

La Figura 2.37 ilustra las acciones que la aplicación lleva a cabo cuando el usuario presiona el botón *Enviar*. La página *certificados_html.php* llama a la función *enviar(),* alojada en el código JavaScript del archivo *certificados.js.* Esta función realiza una llamada asincrónica al archivo *certificados.php* enviando a través del método POST la variable stage con el valor “nuevocertificado”.

Cuando la variable stage tiene el valor “nuevocertificado”; en el archivo *certificados.php* se ejecuta la función *nuevo()*, que verifica si todos los campos fueron ingresados correctamente; de ser así ejecuta los comandos de openssl para crear un certificado nuevo y firmarlo por la Autoridad Certificadora (AC) luego se llama a la función *mostrar()* como en los casos anteriores.

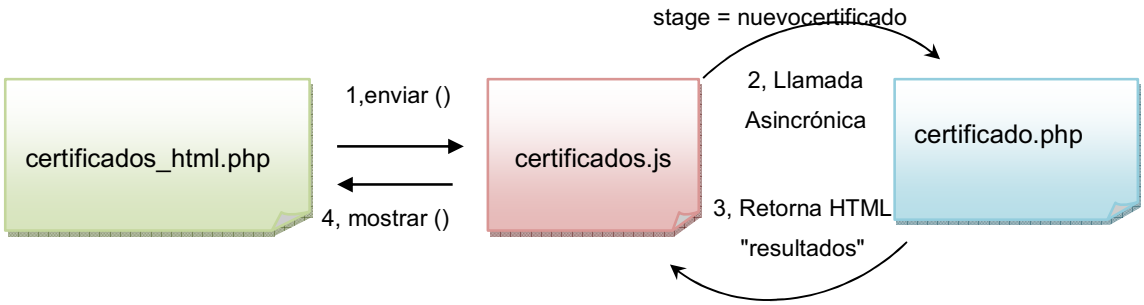


Figura 2.37 Lógica de la acción Enviar

En la Figura 2.38 se puede observar la lógica que se ejecuta en la aplicación al presionar el botón *Ver*. La página *certificados_html.php* llama a la función *enviar(),* alojada en el código JavaScript del archivo *certificados.js.* Esta función realiza una llamada asincrónica al archivo *certificados.php* enviando a través del método POST la variable stage con el valor “display”.

Si el valor de la variable stage es “display”*;* en el archivo *certificados.php* se ejecuta la función *ver(),* la misma que lee los atributos del certificado seleccionado. Usando comandos de openssl estos atributos se muestran en una tabla que incluye los valores tanto del certificado como de la AC que lo firmo.

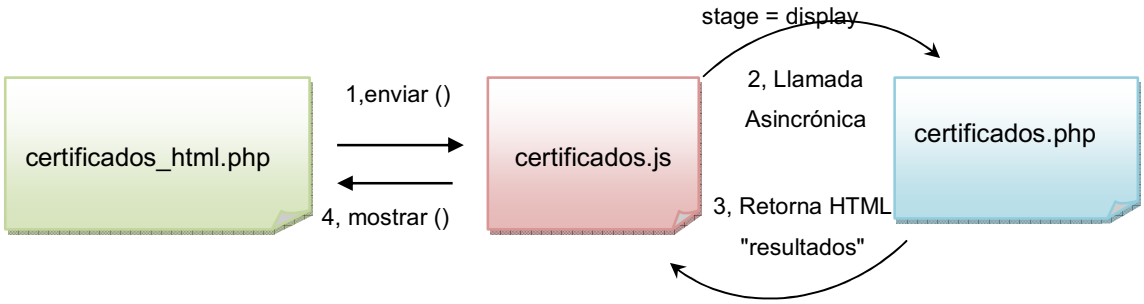


Figura 2.38 Lógica de la acción Ver

Al presionar el botón *Descargar*, la página *certificados_html.php* llama a la función *descargar(),* alojada en el código JavaScript del archivo *certificados.js.* Esta función realiza una llamada asincrónica al archivo *certificados.php* enviando a través del método POST la variable stage con el valor “download”. Si el valor de la variable stage es: “download”*;* el archivo *certificados.php* responde con la ubicación del certificado: */openssl/cert/CA/certs/$serial.der*, la cual es leída por el archivo *certificados.js* y se procede a recuperar el archivo desde el servidor. Este proceso lógico se ilustra en la Figura 2.39.

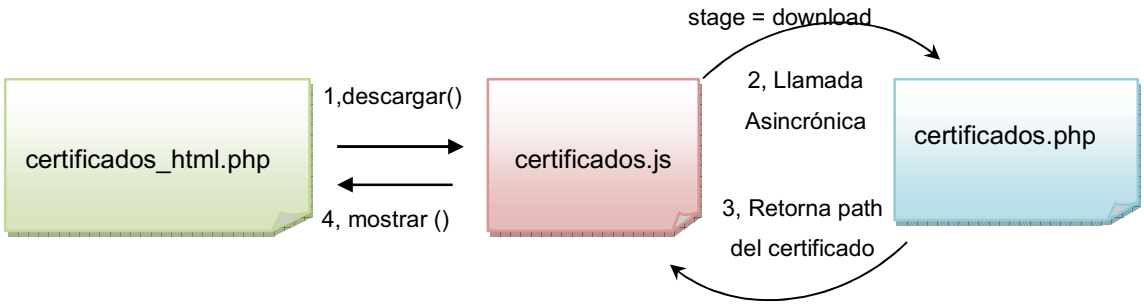


Figura 2.39 Lógica de la acción Descargar

La Figura 2.40 ilustra la lógica que se ejecuta en la aplicación Web cuando el usuario presiona el botón *Revocar*. La página *certificados_html.php* llama a la función *revocar(),* alojada en el código JavaScript del archivo *certificados.js.* Esta función muestra un mensaje de confirmación si se desea revocar el certificado si se responde afirmativamente se realiza una llamada asincrónica al archivo *certificados.php* enviando a través del método POST la variable stage con el valor "revoke". Cuando el valor de la variable stage es "revoke"*;* en el archivo *certificados.php* se ejecuta la función *revocar()* que ejecuta los comandos openssl para revocar el certificado.

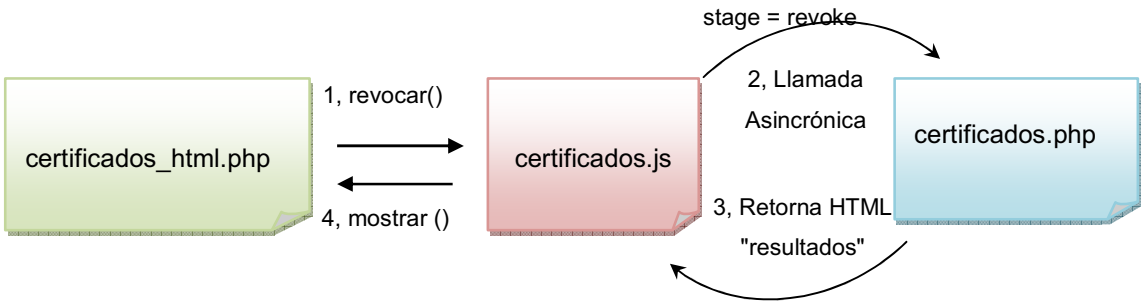


Figura 2.40 Lógica de la acción Revocar

Al presionar el botón *Renovar*, la página *certificados_html.php* llama a la función *renovar(),* alojada en el código JavaScript del archivo *certificados.js.* Esta función

muestra un mensaje en donde se debe indicar por cuantos años adicionales se va a renovar el certificado, luego se realiza una llamada asincrónica al archivo *certificados.php* enviando a través del método POST la variable stage con el valor "renew". Cuando esto sucede, el archivo *certificados.php* se ejecuta la función *renovar()* que ejecuta los comandos openssl para renovar el certificado, como se observa en la Figura 2.41.

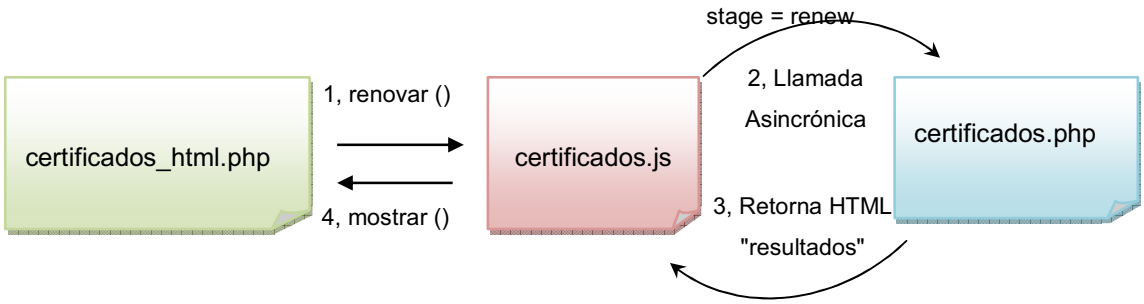


Figura 2.41 Lógica de la acción Revocar

2.5.2.3. Protocolos de Autenticación

En la página Protocolos de Autenticación de la sección Administración de Certificados se presenta una lista de los protocolos soportados por FreeRADIUS, esta pantalla permite seleccionar que protocolos serán habilitados a nivel de EAP. Es importante mencionar que los protocolos PAP y CHAP son habilitados por defecto. Adicionalmente, esta página permite configurar el certificado que es usado por el servidor para los métodos EAP-TLS, EAP-TTLS y EAP-PEAP. La Figura 2.42 muestra una captura de la página Protocolos de Autenticación.

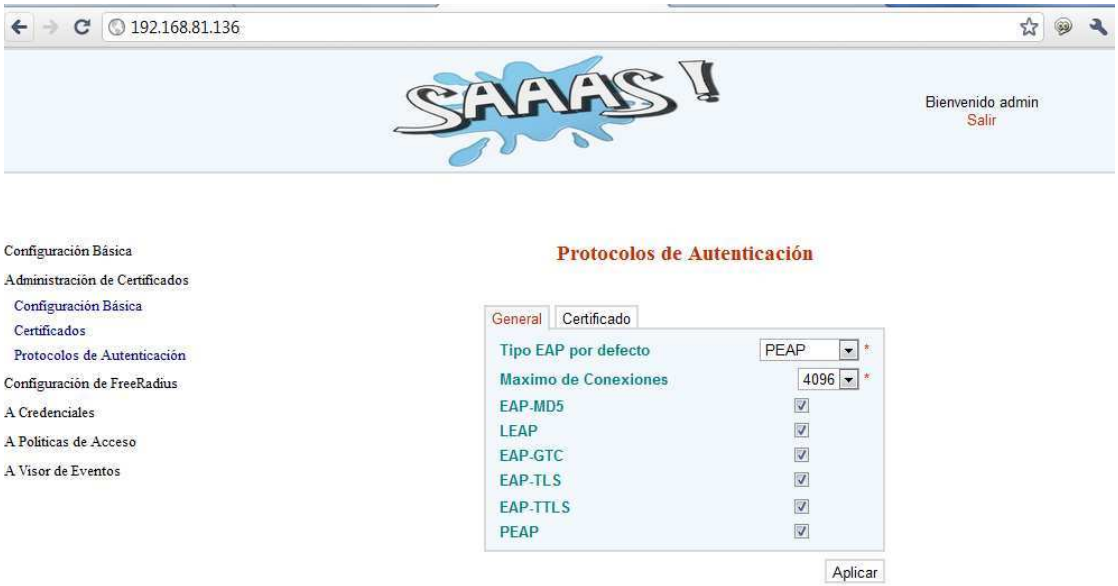


Figura 2.42 Página Protocolos de Autenticación de la Sección Administración de Certificados del servidor "SAAAS!"

2.5.2.3.1. Lógica de la Sección

La página Protocolos de Autenticación permite elegir que protocolos serán admitidos en el método de autenticación EAP y manipular el certificado digital con el que se realizará el proceso TLS.

Al cargarse por primera vez, la página *protocolos_html.php* llama a la función *inicio()* alojada en el código JavaScript del archivo *protocolos.js.* Esta función realiza una llamada asincrónica al archivo *protocolos.php* enviando a través del método POST el valor “null” dentro de la variable stage. Cuando esto sucede, en el archivo *protocolos.php* se ejecuta la función por defecto *mostrar()*, que recupera los parámetros del archivo */etc/freeradius/eap.conf* y genera el código HTML necesario para mostrar la configuración actual de los protocolos EAP. Si la opción EAP-TLS se encuentra activa, esta función muestra una segunda pestaña para la configuración de los certificados que se usan en este módulo. Esta lógica se ilustra en la Figura 2.43.

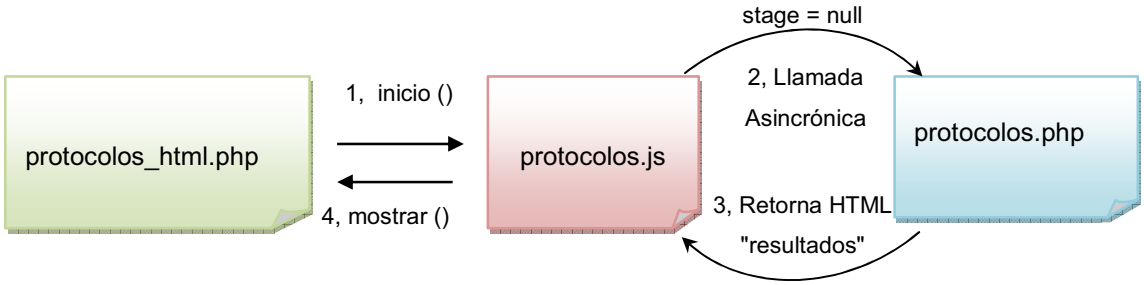


Figura 2.43 Lógica de la acción por defecto

En la Figura 2.44 se puede observar los procesos que se ejecutan en la aplicación Web cuando el usuario activa el control tipo checkbox de EAP-TLS en la página Protocolos de Autenticación. El archivo *protocolos_html.php* llama a la función *enviar()* alojada en el código JavaScript del archivo *protocolos.js.* Esta función realiza una llamada asincrónica al archivo *protocolos.php* enviando a través del método POST el valor "tls" en la variable stage; al recibir este valor el archivo *protocolos.php* ejecuta la función *certificados(),* la cual habilita los módulos EAP-TTLS y EAP-PEAP y genera el código HTML necesario para mostrar una pestaña adicional para la configuración de los certificados a usarse en EAP-TLS.

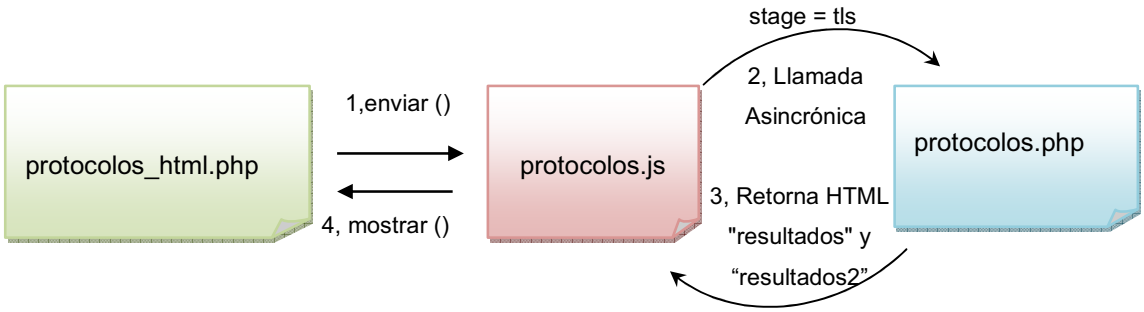


Figura 2.44 Lógica de la acción TLS

Al presionar el botón Aplicar de la pestaña Certificados, la página *protocolos_html.php* llama a la función *aplicar()* alojada en el código JavaScript del archivo *protocolos.js.* Esta función realiza una llamada asincrónica al archivo

protocolos.php enviando a través del método POST el valor “aplicar” en la variable stage. Cuando esto sucede, el archivo *protocolos.php* ejecuta la función *aplicar()*, la cual recopila toda la configuración para generar y remplazar el archivo eap.conf y luego llama a la función por defecto *mostrar();* como se muestra en la Figura 2.45, el caso de uso para esta acción se lo presenta en el Anexo B9.

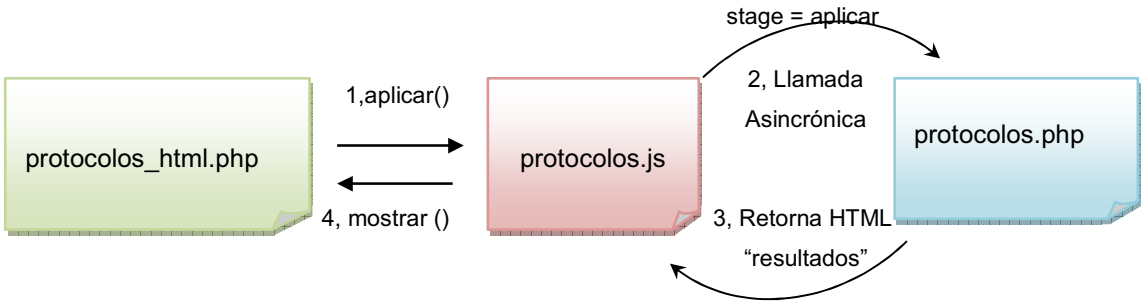


Figura 2.45 Lógica de la acción Aplicar

2.5.3. CONFIGURACIÓN DEL SERVIDOR

2.5.3.1. Clientes Radius (NAS)

Como se observa en las Figuras 2.46 y 2.47 esta página se encuentra dividida en dos pestañas: la primera recupera los datos de los clientes NAS, el nombre, el secreto compartido, el grupo al que pertenecen y una descripción del cliente de la base de datos; mientras que la segunda recupera los nombres de los grupos de clientes existentes y su descripción correspondiente. Estos clientes y grupos de clientes corresponden a los intermediadores en el esquema de un entorno AAA con Radius.

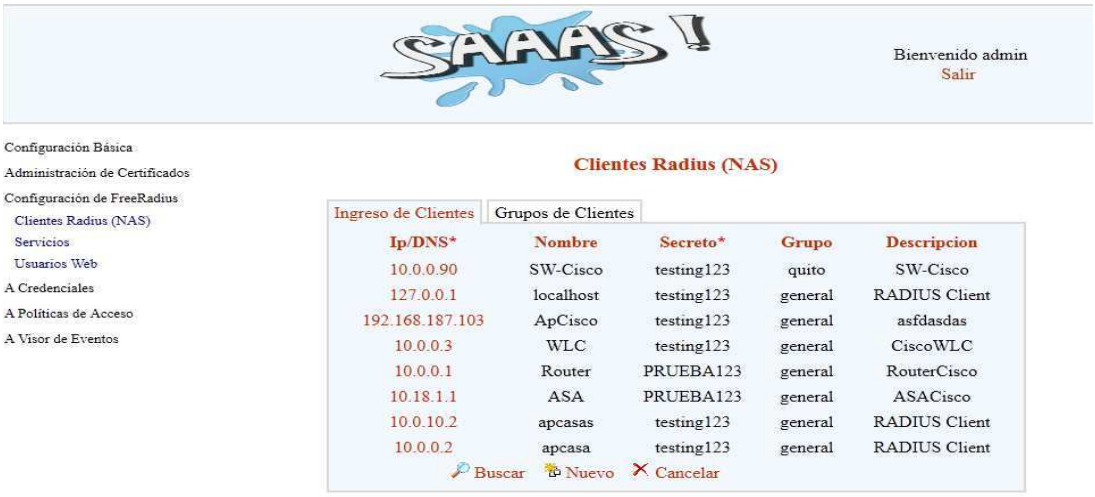


Figura 2.46 Página Clientes Radius del servidor "SAAAS!", pestaña Ingreso de Clientes

.

Figura 2.47 Página Clientes Radius del servidor "SAAAS!" pestaña Grupos de Clientes

2.5.3.1.1. Lógica de la Sección

La página Clientes Radius maneja dos páginas con código php *nas.php* e *nas_g.php* las cuales manejan de manera independiente las pestañas Ingreso de Clientes y

Grupos de Clientes respectivamente, de igual forma recuperan y manipulan los datos de las tablas correspondientes en la base de datos, estas son: *nas* para los clientes y *nas_group* para los grupos. Esto se ilustra en la Figura 2.48, el caso de uso para la manipulación de los clientes se lo presenta en el Anexo B10 y para la manipulación de grupo de se lo presenta en el Anexo B11.

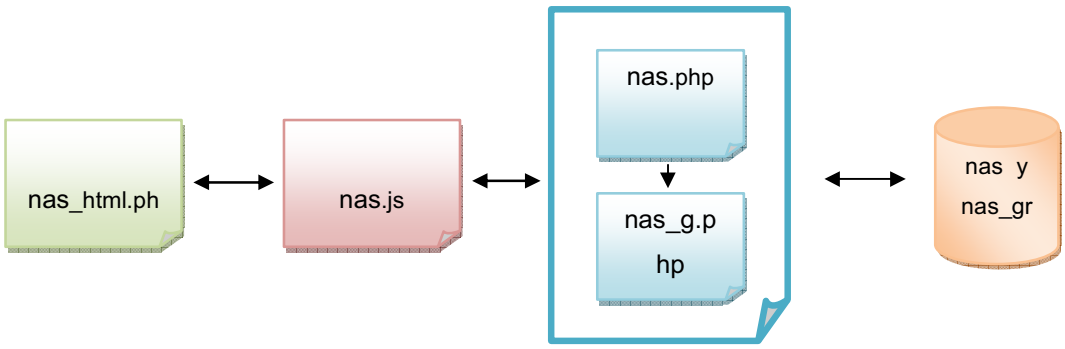


Figura 2.48 Esquema general de la página Clientes Radius

La Figura 2.49 ilustra la lógica que se ejecuta en la aplicación Web cuando se carga por primera vez la página Clientes RADIUS. El archivo *nas_html.php* llama a la función *inicio()* alojada en el código JavaScript del archivo *nas.js.* Esta función realiza una llamada asincrónica al archivo *nas.php y al archivo nas_g.php* enviando a través del método POST el valor "null" en la variable stage. Al recibir esta variable el archivo *nas.php* ejecuta la función por defecto *mostrar()*, esta función genera el código HTML necesario para crear la tabla de visualización de los clientes y de los grupos de clientes. Previamente realiza una consulta a la tabla *nas* de la base de datos para obtener los primeros diez registros, y genera el menú inferior apuntando a las funciones JavaScript correspondientes para la manipulación de estos registros.

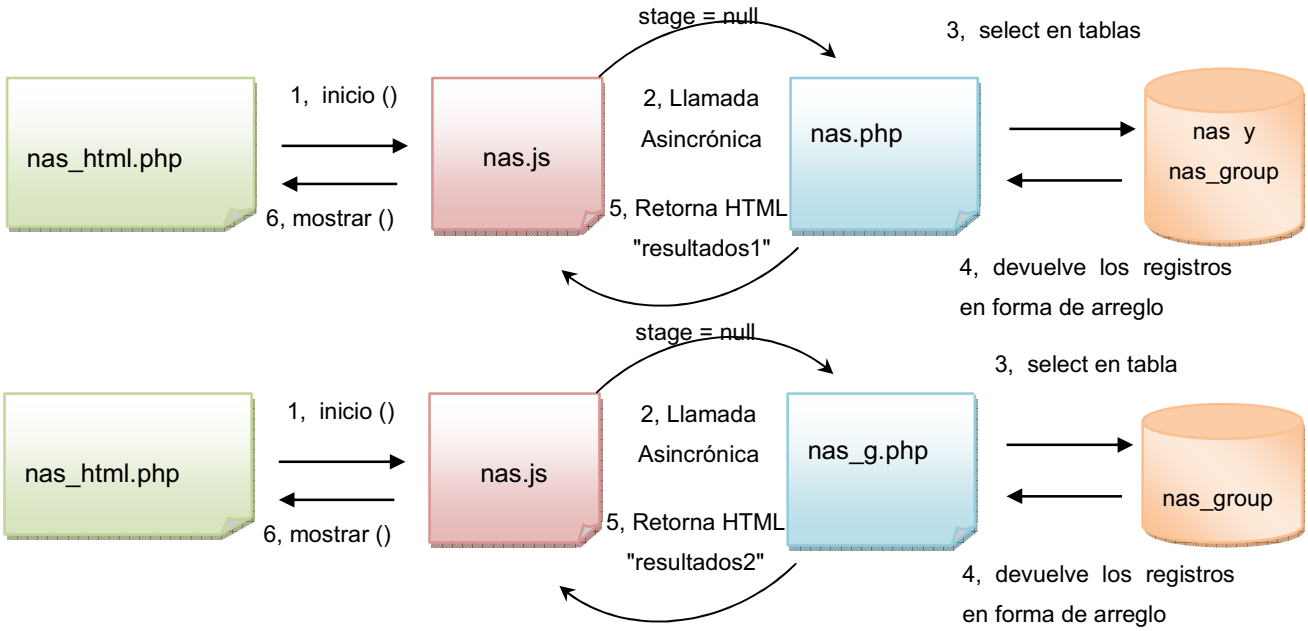


Figura 2.49 Lógica de la acción por defecto

Cuando el usuario presiona el botón *Buscar* en la página Clientes NAS, el archivo *nas_html.php* llama a la función *buscar(),* si se lo hizo desde la pestaña Ingreso de Clientes, o *buscar_g(),* si se lo hizo desde la pestaña Grupos de Clientes. Estas funciones muestran un mensaje al usuario con un campo vacío para ingresar el nombre parcial o completo del cliente o grupo a buscar. Una vez ingresado este campo se realiza una llamada asincrónica al archivo *nas.php* o *al archivo nas_g.php* enviando a través del método POST las variables stage y buscar con los valores "null" y "valor ingresado" respectivamente. Ya que la variable stage es "null"*;* el archivo *nas.php o nas_g.php* ejecuta la función por defecto *mostrar() como* se indicó en la sección anterior, con la diferencia de que la consulta a realizarse esta vez es condicionada al valor ingresado para mostrar todas las posibles coincidencias de este valor con los registros almacenados. Estos procedimientos se ilustran en la Figura 2.50.

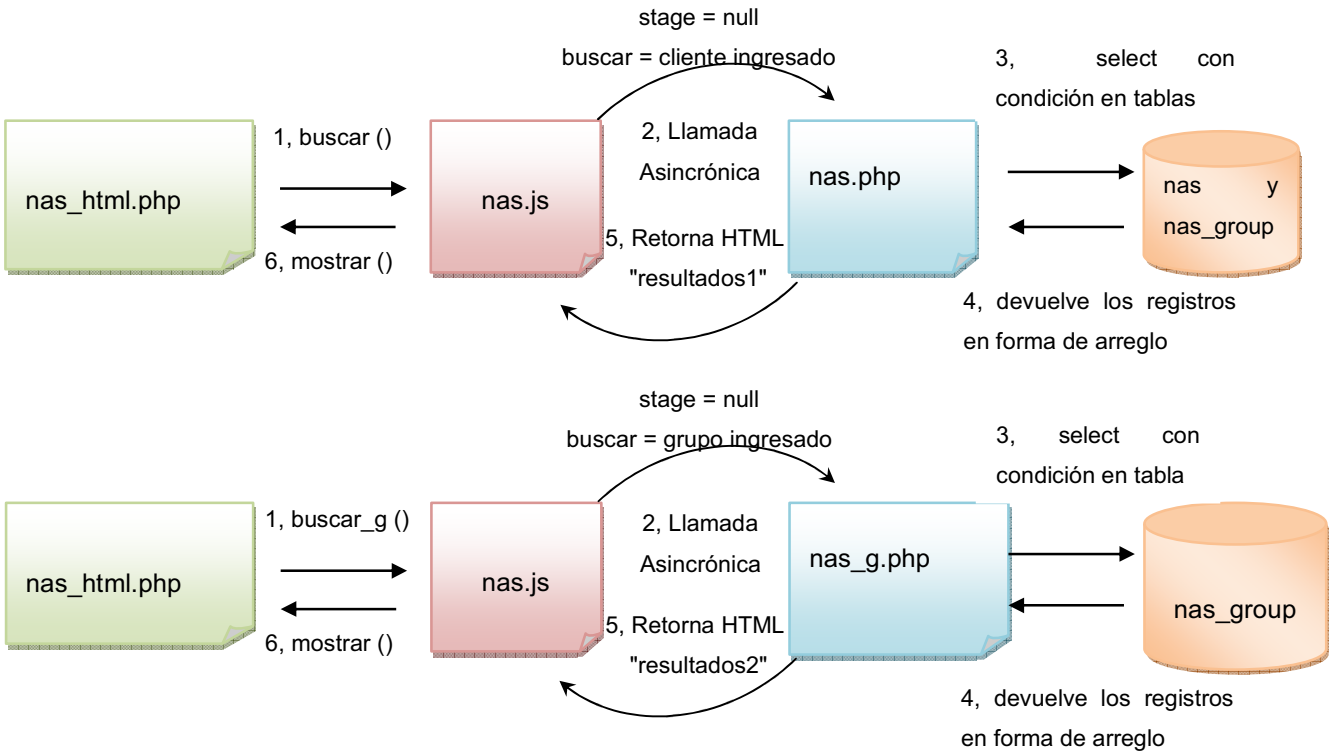


Figura 2.50 Lógica de la acción: Buscar

La Figura 2.51 ilustra la lógica que se ejecuta en la aplicación Web cuando el usuario hace clic sobre el botón *Nuevo,* para registrar un nuevo NAS en el servidor AAA o para crear un nuevo grupo de NAS. La página *nas_html.php* llama a la función enviar(), desde la pestaña Ingreso de Clientes, o *enviar_g(),* desde la pestaña Grupos de Clientes; estas funciones están alojadas en el código JavaScript del archivo *nas.js* y realizan una llamada asincrónica al archivo *nas.php o al archivo nas_g.php* enviando a través del método POST el valor “null” en la variable stage y el valor “nuevocliente” o “nuevogrupo” en la variable puntero.

Dependiendo de la pestaña en la cual se haya generado la acción. Ya que la variable stage es “null”, el archivo *nas.php o nas_g.php* ejecuta la función por defecto *mostrar()*, y se genera el código HTML necesario para crear la tabla de visualización

como se lo indicó en las secciones anteriores, con la diferencia que la consulta a realizarse esta vez es de los últimos registros almacenados y la última fila de la tabla estará conformada por campos vacíos para que el usuario ingrese los datos necesarios para la creación de un nuevo NAS.

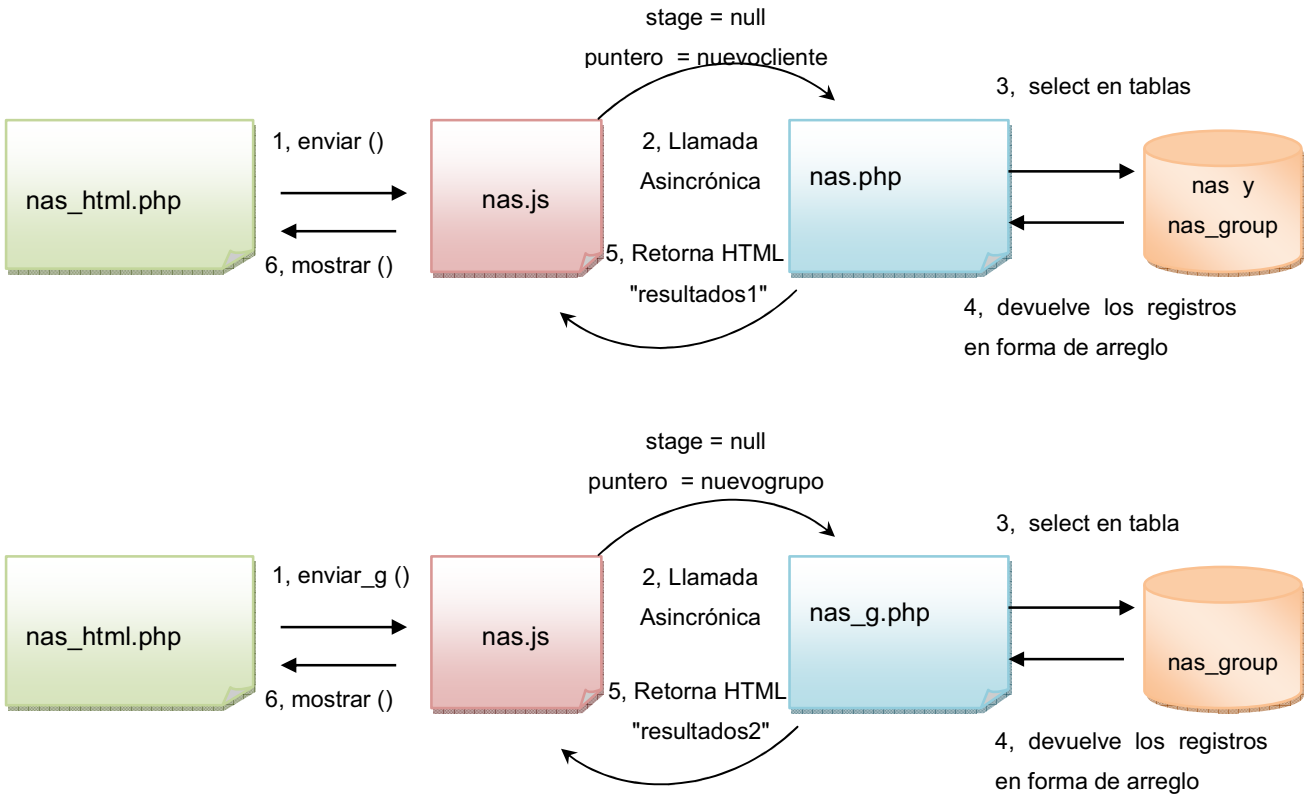


Figura 2.51 Lógica de la acción: Nuevo

Para guardar el cliente o grupo de cliente nuevo el usuario debe presionar el botón *Aceptar,* cuando esto sucede la página *nas_html.php* llama a la función *enviar()* o *enviar_g(),* dependiendo de la pestaña en la que se ejecutó la acción. Estas funciones realizan una llamada asincrónica al archivo *nas.php o al archivo nas_g.php* respectivamente, enviando a través del método POST el valor “nuevo” en la variable stage junto con los datos necesarios para crear el nuevo registro de NAS en el servidor. Al recibir esta variable, el archivo *nas.php o nas_g.php* ejecuta la función

nuevo(), que recupera y revisa si los campos vacíos fueron ingresados correctamente y procede a insertar el registro en la base de datos. Si los campos están vacíos o mal ingresados devuelve el error pertinente por medio de una ventana de alerta de JavaScript. Si no se produjo ningún error se llama a la función *mostrar()* como en las secciones anteriores (ver Figura 2.52).

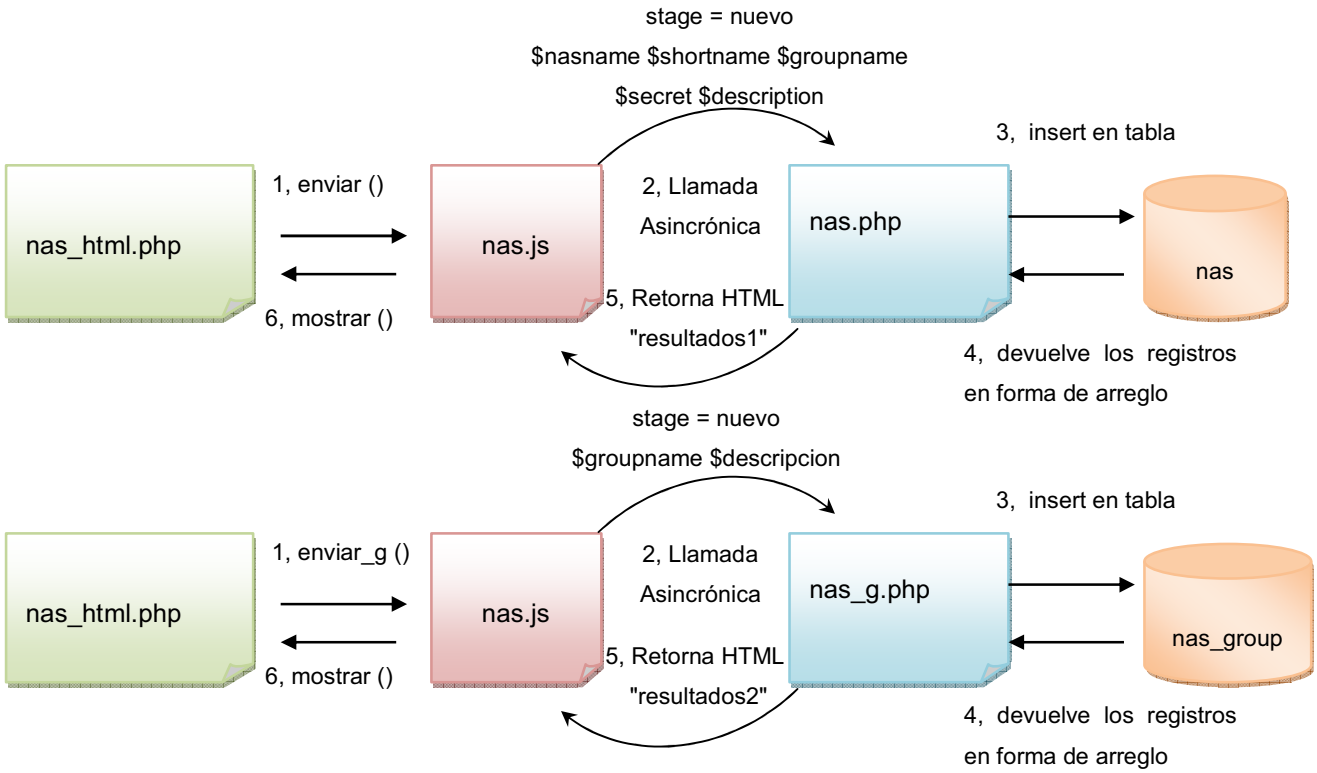


Figura 2.52 Lógica de la acción: Aceptar

Cuando el usuario presiona el botón *Actualizar* en la página Clientes NAS, el archivo *nas_html.php* llama a la función *enviar(),* si se lo hizo desde la pestaña Ingreso de Clientes, o *enviar_g(),* si se lo hizo desde la pestaña Grupos de Clientes. Estas funciones realizan una llamada asincrónica al archivo *nas.php* o *al archivo nas_g.php* enviando a través del método POST la variable stage con el valor “actualizar”.

Ya que la variable stage es “actualizar”*;* el archivo *nas.php o nas_g.php* ejecuta la

función *actualizar()*, esta función recupera y revisa si los campos fueron ingresados correctamente y procede a actualizar el registro en la base de datos. Si los campos están mal ingresados devuelve el error pertinente por medio de una ventana de alerta de JavaScript. Si no se producen errores se llama a la función *mostrar()* como en las secciones anteriores. Estos procedimientos se ilustran en la Figura 2.53.

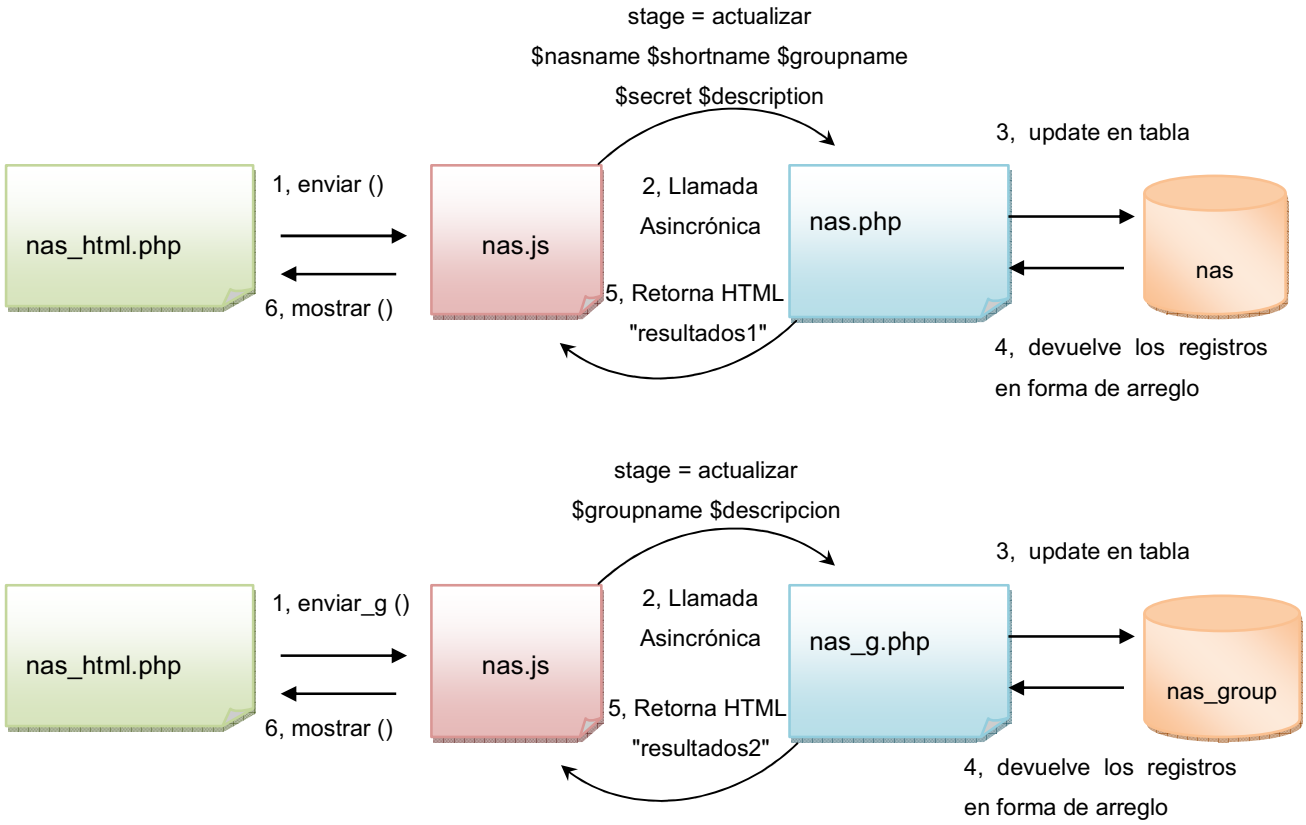


Figura 2.53 Lógica de la acción: Actualizar

Para eliminar el registro de un Cliente NAS o un Grupo de Clientes NAS el usuario debe presionar el botón *Eliminar* en las pestañas de la página Clientes Radius, cuando esto sucede la página *nas_html.php* llama a la función *borrar()* o *borrar_g(),* dependiendo de la pestaña en la que se ejecutó la acción. Estas funciones realizan una llamada asincrónica al archivo *nas.php o al archivo nas_g.php* respectivamente, enviando a través del método POST el valor “eliminar” en la variable stage junto con una variable que identifica al registro que se eliminará.

Al recibir estas variables, el archivo *nas.php o nas_g.php* ejecuta la función *borrar()*; si lo que se debe borrar es un grupo la función revisa si el mismo contiene clientes, de ser así envía una ventana de alerta, caso contrario procede a eliminar el registro. Si no se produce ningún error se llama a la función *mostrar()* como en las secciones anteriores (ver Figura 2.54).

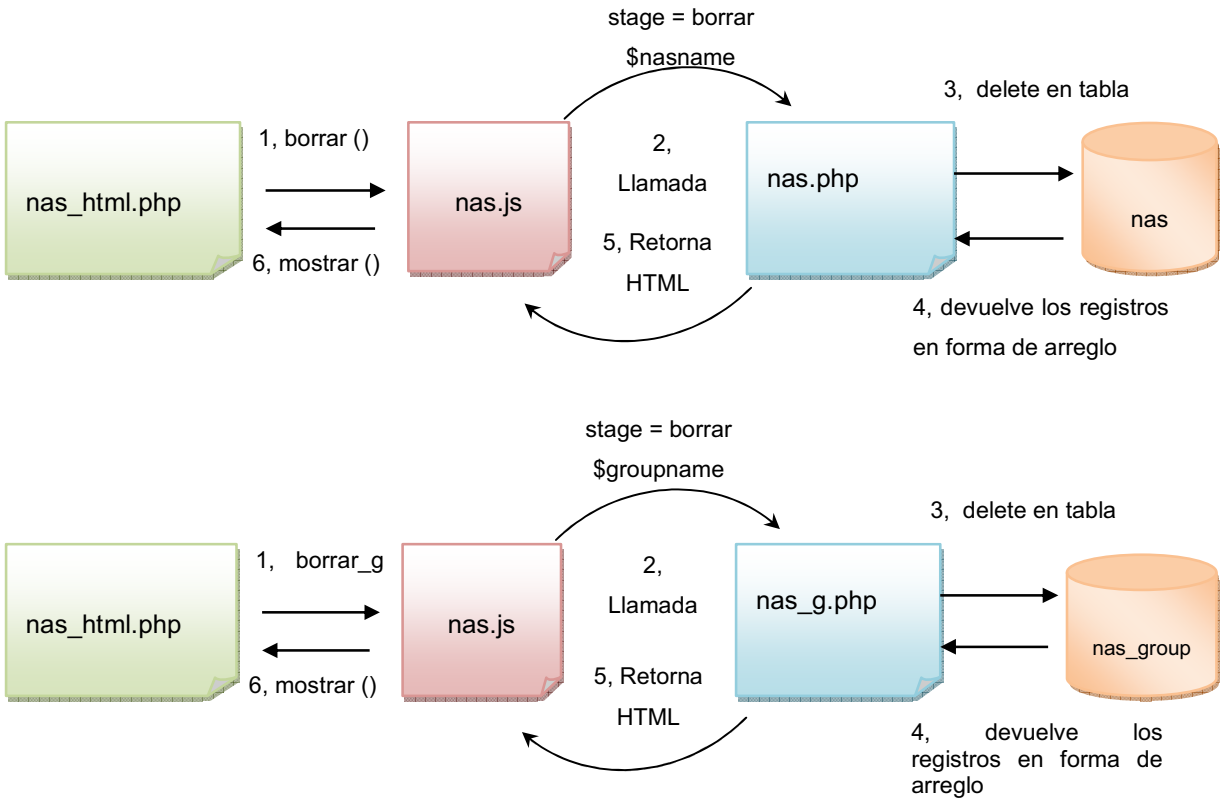


Figura 2.54 Lógica de la acción: Eliminar

2.5.3.2. Servicios

Esta página permite al usuario detener o iniciar los servicios más importantes que corren sobre "SAAAS!". En la Figura 2.55 se puede observar la interfaz gráfica de esta sección de la aplicación Web.

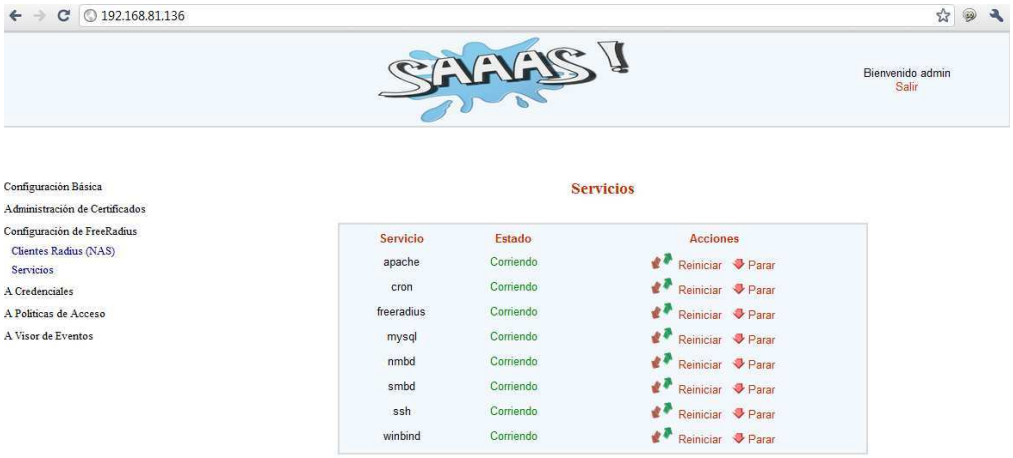


Figura 2.55 Página Servicios de la sección Configuración del Servidor

2.5.3.2.1. Lógica de la Sección

Como se observa en la Figura 2.56, al cargarse por primera vez la página *servicios_html.php* llama a la función *inicializar()* alojada en el código JavaScript de *servicios.js.* Esta función recupera la información del estado de los servicios Apache, Cron, FreeRADIUS, MySQL, nmbd, smbd, ssh y winbind que corren sobre "SAAAS!" a través de una llamada asincrónica a la página *servicios.php* y la muestra al usuario a través de la función *imprimir()*, los casos de uso para estas acciones se los presentan en el Anexo B12, B13 y B14.

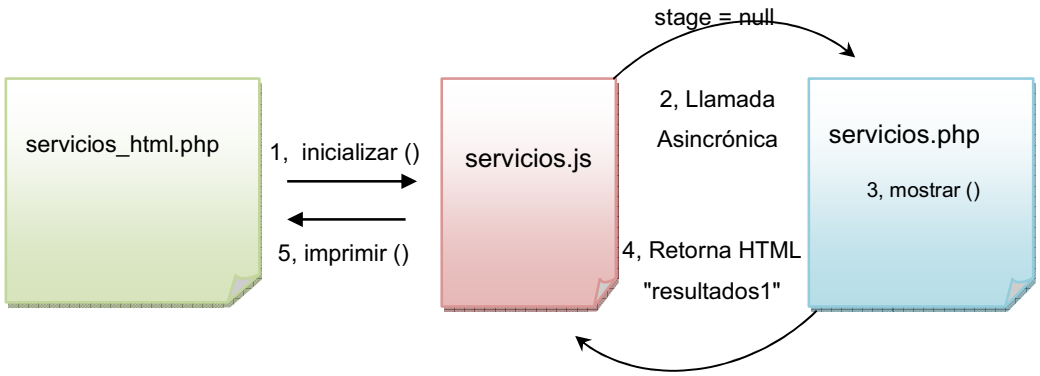


Figura 2.56 Lógica de la página "Servicios" del servidor "SAAAS!"

Cuando el usuario presiona los botones *Iniciar*, *Detener* o *Reiniciar*, se ejecuta la

función *"enviar(stage)",* en la que *stage* representa la acción a realizarse. Esta función crea una llamada asincrónica hacia la página *servicios.php* enviando a través del método POST las variables *stage* y *servname* que identifica el nombre del servicio sobre el que se realizará la acción. En la página *servicios.php* se ejecuta el código para iniciar, detener o reiniciar el servicio que el usuario requiere, después de hacerlo retorna el código HTML a mostrarse en la página *servicios_html.php* (Ver Figura 2.57).

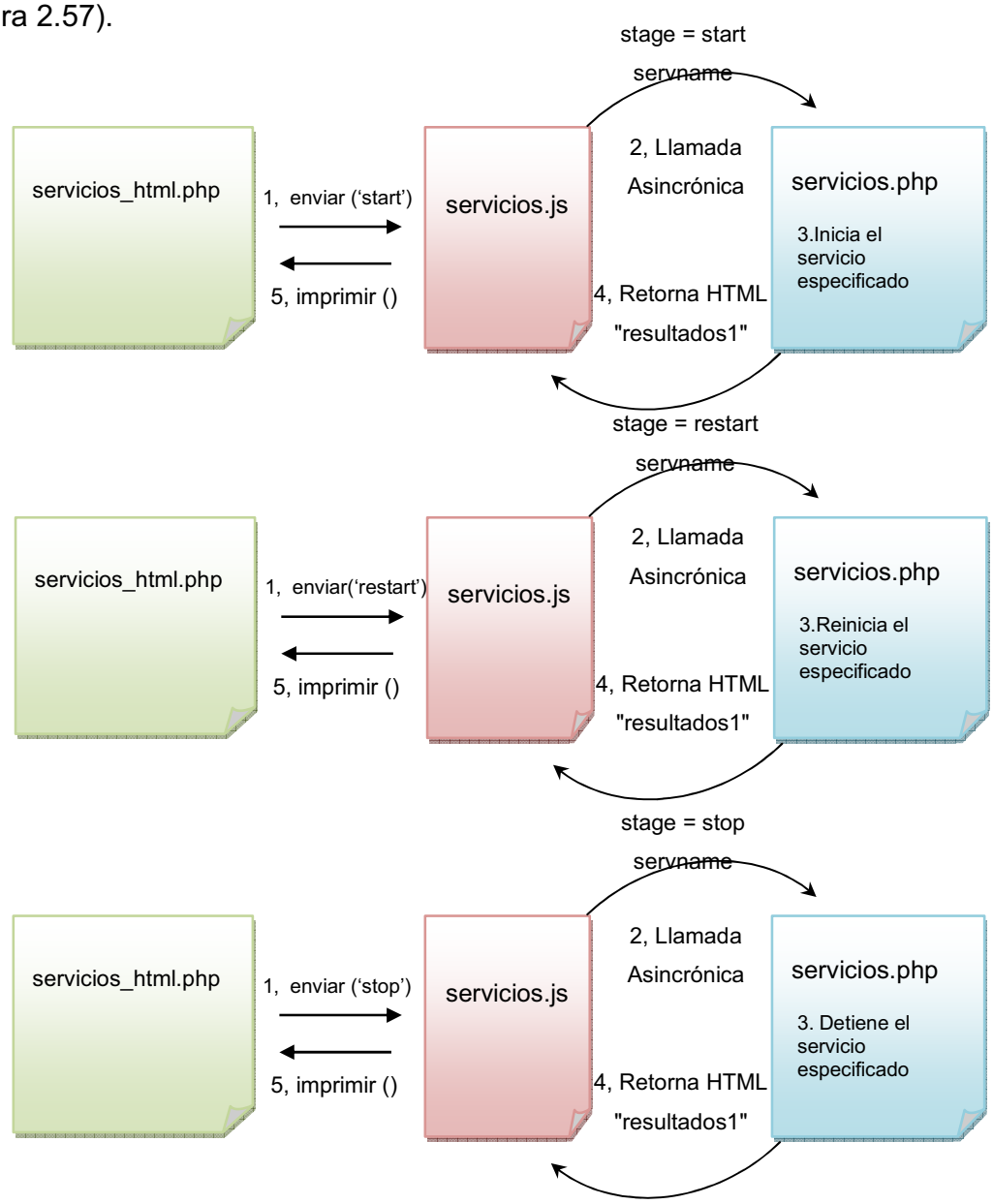


Figura 2.57 Lógica de las acciones Iniciar/Detener/Reiniciar servicios de la página

2.5.3.3. Usuarios Web

En esta página se pueden crear, eliminar y modificar los usuarios Web del servidor “SAAAS!”. Las figuras 2.58, 2.59 y 2.60 muestran la interfaz de esta página para cada una de estas acciones.

Figura 2.58 Página Usuarios Web de la sección Configuración del Servidor del Servidor "SAAAS!"

Figura 2.59 Página Usuarios Web, formulario Modificar Usuario del Servidor "SAAAS!"

Figura 2.60 Página Usuarios Web, formulario Nuevo Usuario del servidor "SAAAS!"

2.5.3.3.1. Lógica de la Sección

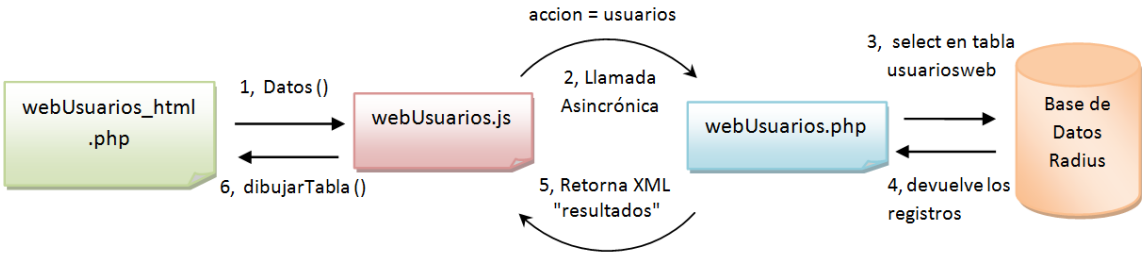


Figura 2.61 Lógica de la Página Usuarios Web en la sección Configuración del Servidor

Como se observa en la Figura 2.61, al cargarse por primera vez la página *webUsuarios_html.php* se ejecuta la función *Datos()* del código JavaScript del archivo *webUsuarios.js*, esta función realiza una llamada asincrónica a la página *webUsuarios.php* enviando la variable accion con el valor “usuarios” a través del método POST. En la página *webUsuarios.php* se ejecuta la conexión a la base de datos y la consulta para generar el listado de Usuarios Web que se finalmente se muestra al usuario.

De manera similar, se ejecutan llamadas asincrónicas para las acciones *crear nuevo usuario* (ver Figura 2.62), *modificar usuario* (ver Figura 2.63) y *eliminar usuario* (ver Figura 2.63). Las variables enviadas son diferentes lo que provoca una interacción diferente con la base de datos (consulta, inserción, actualización o eliminación de campos), los casos de uso para estas acciones se los presentan en el Anexo B15, B16 y B17.

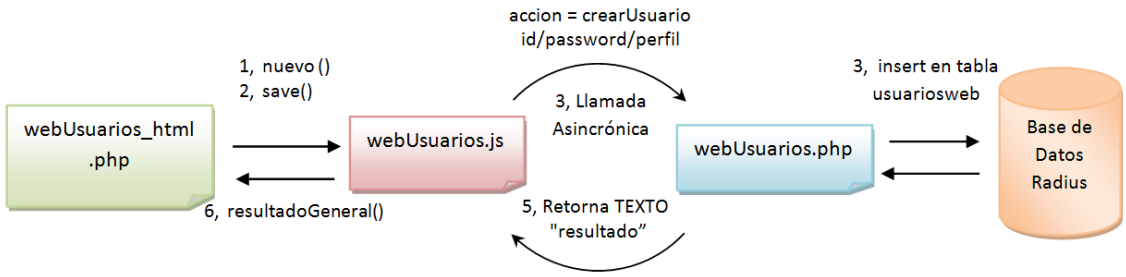


Figura 2.62 Lógica de la acción: Nuevo usuario en la página Usuarios Web

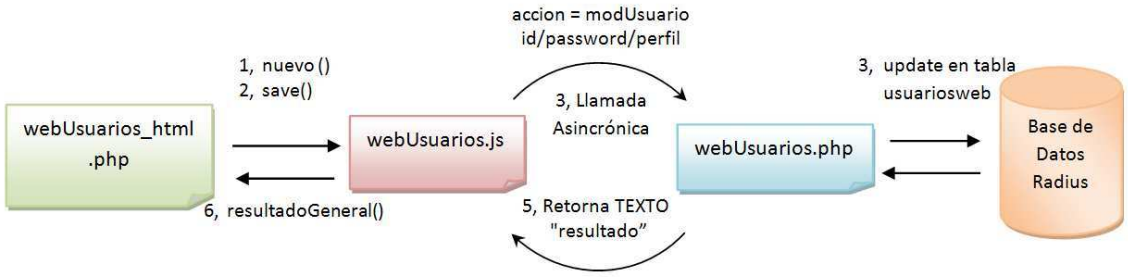


Figura 2.63 Lógica de la acción: Modificar usuario en la página Usuarios Web

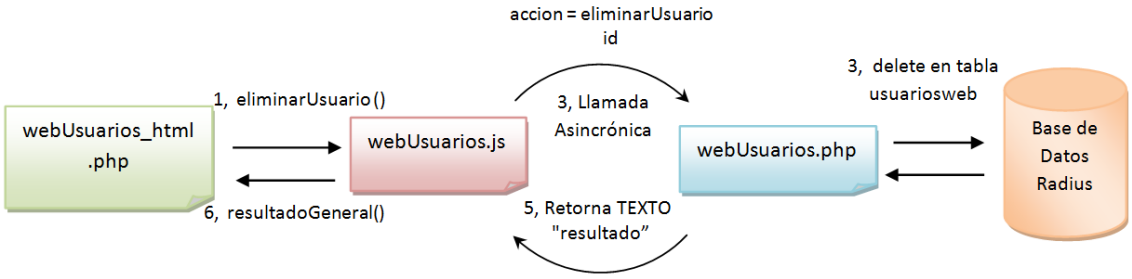


Figura 2.64 Lógica de la acción Eliminar usuario de la página Usuarios Web

2.5.4. CREDENCIALES

En esta sección se visualiza y se manipula la configuración de los repositorios de credenciales tanto internos como externos.

2.5.4.1. Usuarios Internos

Como se observa en la Figura 2.65, esta página se encuentra dividida en dos pestañas, la una recupera los datos de los usuarios internos, contraseña, el grupo al que pertenecen y una descripción del usuario de la base de datos, mientras que la otra pestaña recupera los nombres de los grupos existentes y su descripción correspondiente. Estos usuarios y grupos pertenecen al repositorio interno dedicado únicamente a la autenticación mediante RADIUS.

Figura 2.65 Página Usuarios Internos del servidor "SAAAS!" pestaña Usuarios Internos

La Figura 2.66 es una captura de la pestaña de configuración de Grupos de Usuarios de la aplicación Web. El uso de grupos en la base de datos local permite una implementación más sencilla de las políticas a utilizarse en la autorización del esquema AAA.

Figura 2.66 Página Usuarios Internos del servidor "SAAAS!" pestaña Grupos de Usuarios

2.5.4.1.1. Lógica de la sección

La página Usuarios Internos maneja dos páginas con código php *internos.php* e *internos_g.php* las cuales manejan de manera independiente las pestañas Usuarios Internos y Grupos de Usuarios, respectivamente, de igual forma recuperan y manipulan los datos de las tablas correspondientes en la base de datos, éstas son: *radcheck* para los usuarios y *radgroup* para los grupos (Ver Figura 2.67), el caso de uso para la manipulación de usuarios internos se lo presenta en el Anexo B18 y el de manipulación de grupos de usuarios internos se lo presenta en el Anexo B19.

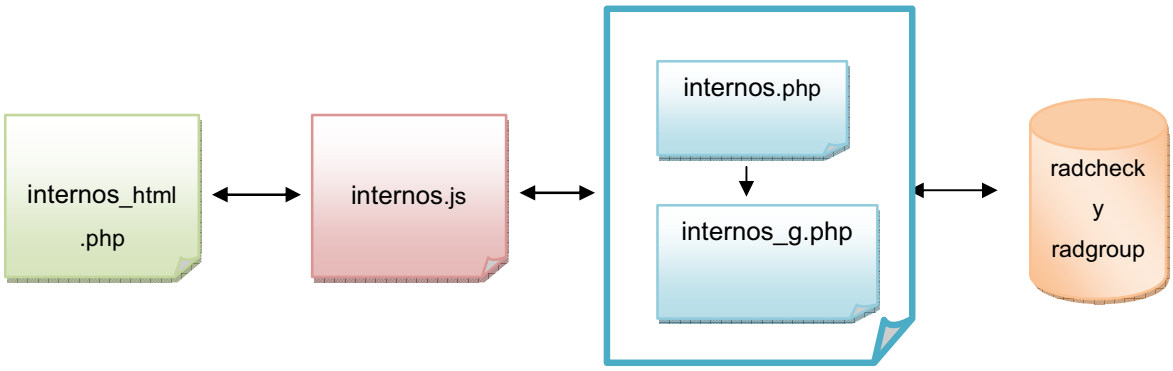


Figura 2.67 Esquema general de la página Usuarios Internos

Al cargarse por primera vez, la página *internos_html.php* llama a la función *inicio()* alojada en el código JavaScript del archivo *internos.js.* Esta función realiza una llamada asincrónica al archivo *internos.php y al archivo internos_g.php* enviando a través del método POST el valor "null" en una variable llamada stage; el código del archivo *internos.php* y *internos_g.php* se ejecuta de acuerdo al tipo de acción solicitada.

Si la variable stage es "null"*;* en el archivo *internos.php* se ejecuta la función por defecto *mostrar()*, esta función genera el código HTML necesario para crear la tabla de visualización de los usuarios, previamente realiza una consulta a la tabla *radcheck*

de la base de datos obteniendo así los primeros diez registros, y genera el menú inferior apuntando a las funciones JavaScript correspondientes para la manipulación de estos registros, lo mismo sucede en el archivo *internos_g.php* si la llamada a la función se realiza desde la pestaña de grupos (ver Figura 2.68)

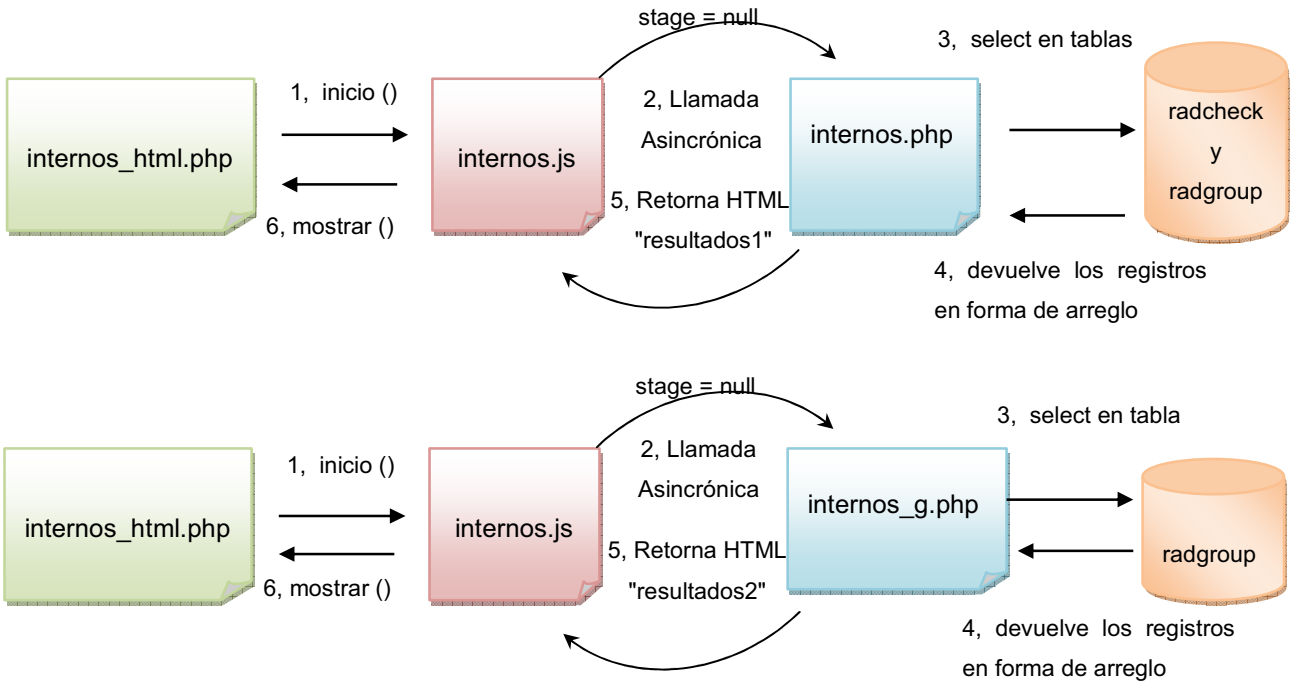


Figura 2.68 Lógica de la acción por defecto

La Figura 2.69 ilustra la lógica que se ejecuta en la aplicación cuando el usuario presiona el botón *Buscar* en las pestañas de la sección Usuarios Internos. La página *internos_html.php* llama a la función *buscar(),* si se lo hizo desde la pestaña Ingreso de Usuarios, o *buscar_g(),* si se lo hizo desde la pestaña Grupos de Usuarios, alojadas en el código JavaScript del archivo *internos.js.* Esta función despliega una ventana de JavaScript mostrando un campo vacío para ingresar el nombre parcial o completo del usuario o grupo a buscar.

Una vez ingresado este campo se realiza una llamada asincrónica al archivo *internos.php* o al archivo *internos_g.php* enviando a través del método POST el valor

"null" en la variable stage y el valor indicado por el usuario en la variable buscar. Cuando la variable stage es "null"; en el archivo *internos.php o internos_g.php* se ejecuta la función por defecto *mostrar()*, y se genera el código HTML necesario para crear la tabla de visualización como se lo indicó en la sección anterior, con la diferencia que la consulta a realizarse esta vez es condicionada al valor ingresado, lo que mostraría todas las posibles coincidencias del valor ingresado con los registros almacenados.

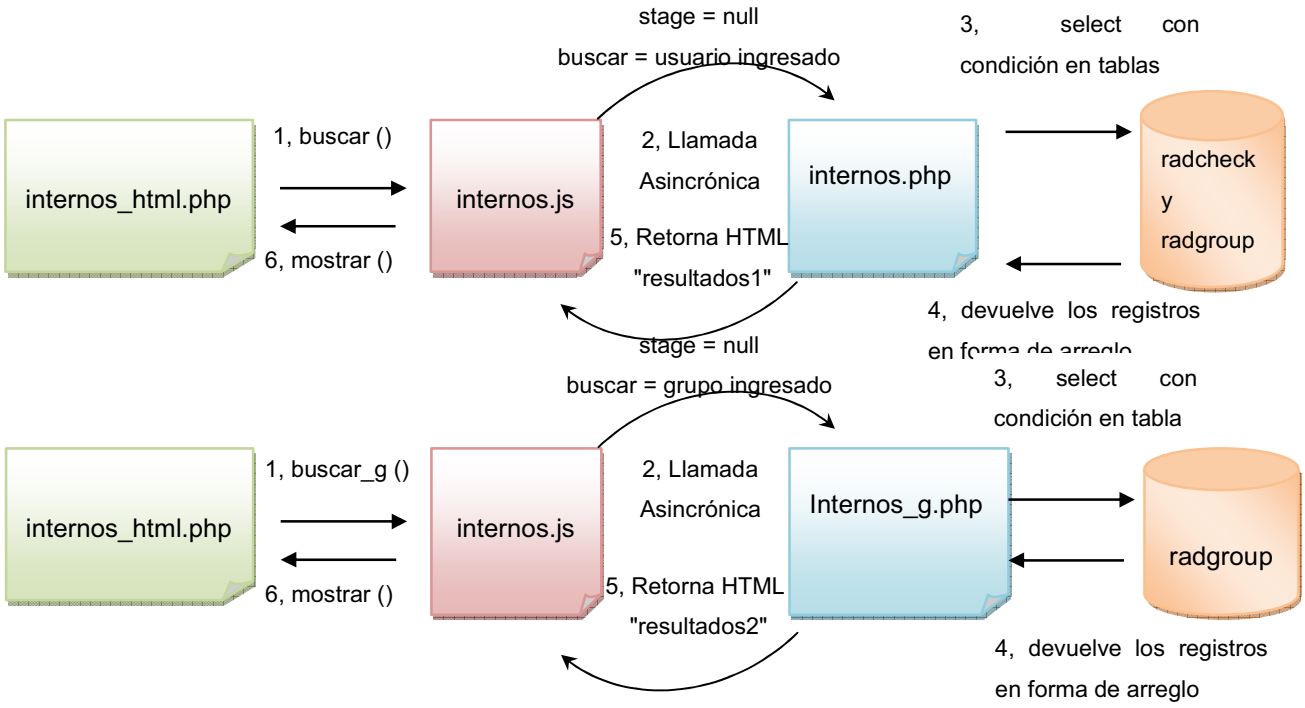


Figura 2.69 Lógica de la acción: Buscar Usuario Interno.

Si el usuario desea crear un nuevo registro de usuario o de grupo en el repositorio interno presiona el botón *Nuevo,* esto provoca que la página *internos_html.php* llame a la función *enviar(),* si se lo hizo desde la pestaña Ingreso de Usuarios, o *enviar_g(),* si se lo hizo desde la pestaña Grupos de Usuarios. Esta función realiza una llamada asincrónica al archivo *internos.php o al archivo internos_g.php* enviando a través del método POST el valor "null" en la variable stage y el valor "nuevousuario" o "nuevogrupo" en la variable puntero.

Ya que la variable stage es: “null”*;* en el archivo *internos.php o internos_g.php* se ejecuta la función por defecto *mostrar()*, y se genera el código HTML necesario para crear la tabla de visualización como se lo indicó en las secciones anteriores, con la diferencia que la consulta a realizarse esta vez es de los últimos registros almacenados y al final de estos, se muestra una fila de campos vacíos que serán llenados de acuerdo a lo que corresponda cada uno (ver Figura 2.70).

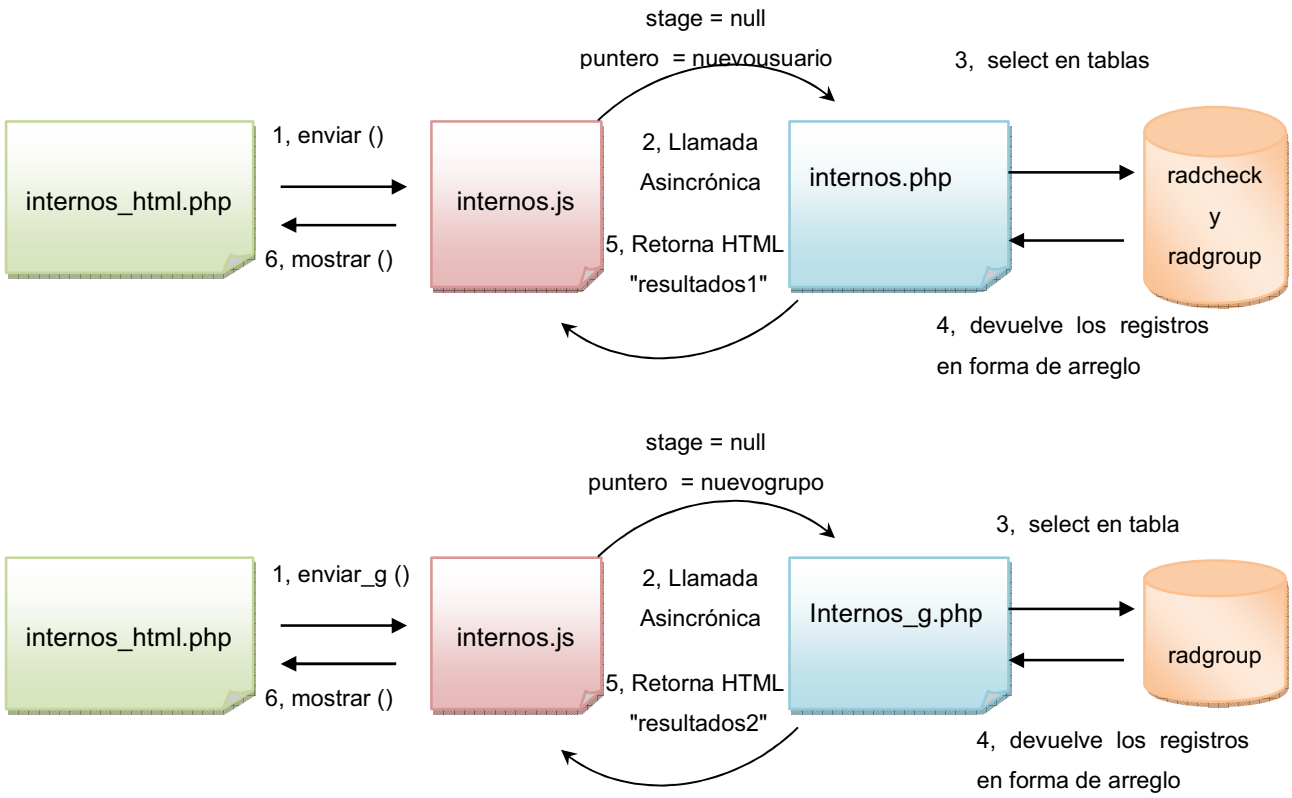


Figura 2.70 Lógica de la acción: Nuevo

Después de llenar los campos para la creación del nuevo usuario o grupo en la base de datos local, el usuario de la aplicación Web debe presionar el botón *Aceptar* para que los cambios se guarden.

La lógica que se ejecuta en la aplicación al realizar esta acción se detalla a continuación y en la Figura 2.71. La página *internos_html.php* llama a la función *enviar(),* si se lo hizo desde la pestaña Ingreso de Usuarios, o *enviar_g(),* si se lo hizo desde la pestaña Grupos de Usuarios, alojadas en el código JavaScript del archivo *internos.js.* Esta función realiza una llamada asincrónica al archivo *internos.php o* al archivo *internos_g.php* enviando a través del método POST el valor "nuevo" en la variable stage, junto con los datos ingresados por el usuario (nombre de usuario, contraseña, grupo). El archivo *internos.php o internos_g.php* ejecuta la función *nuevo()* al recibir estas variables, esta función recupera y revisa si los campos vacíos fueron ingresados correctamente y procede a insertar el registro en la base de datos, si los campos están vacíos o mal ingresados devuelve el error pertinente por medio de una ventana de alerta de JavaScript. Si no se produjo algún error se llama a la función *mostrar()* como en las secciones anteriores.

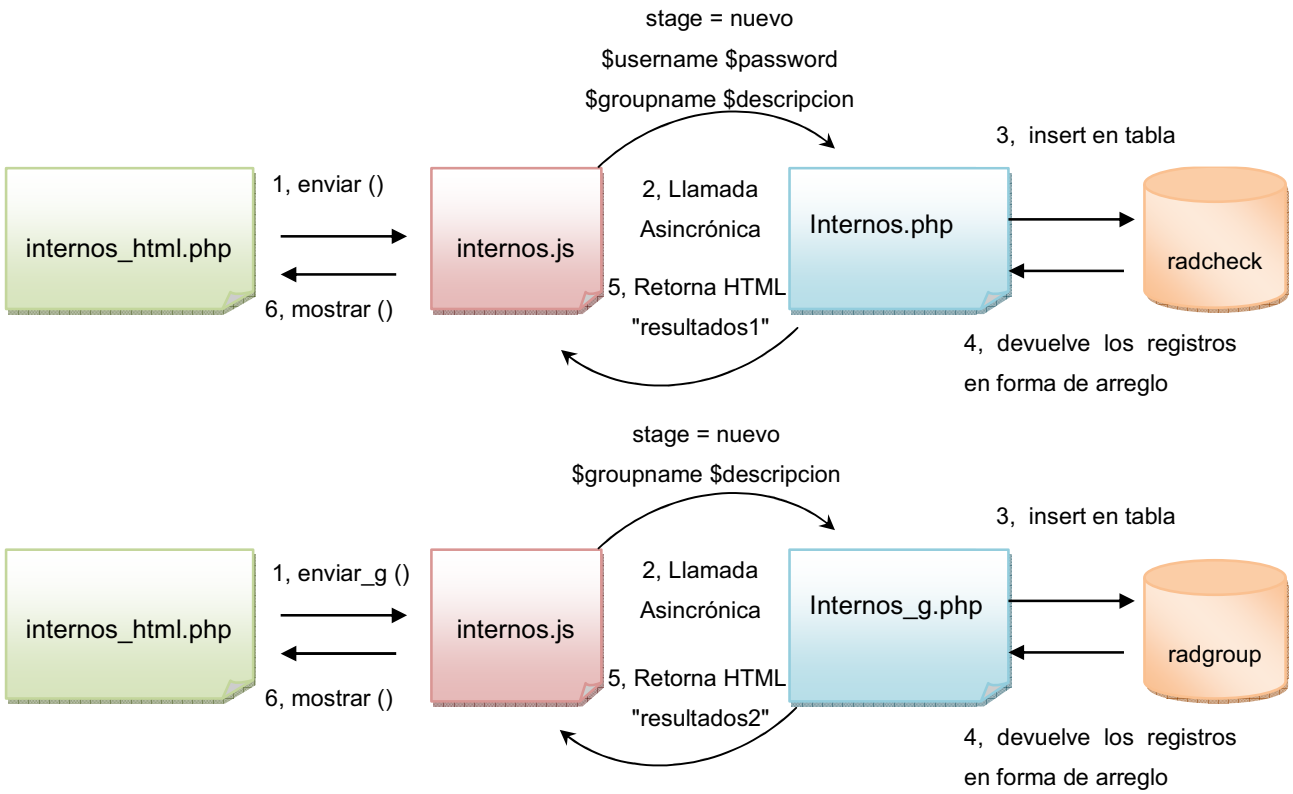


Figura 2.71 Lógica de la acción: Aceptar

Se sigue una lógica similar cuando el usuario de la aplicación Web requiere modificar algún registro y presiona el botón *Actualizar*, esta lógica se describe en la Figura 2.72.

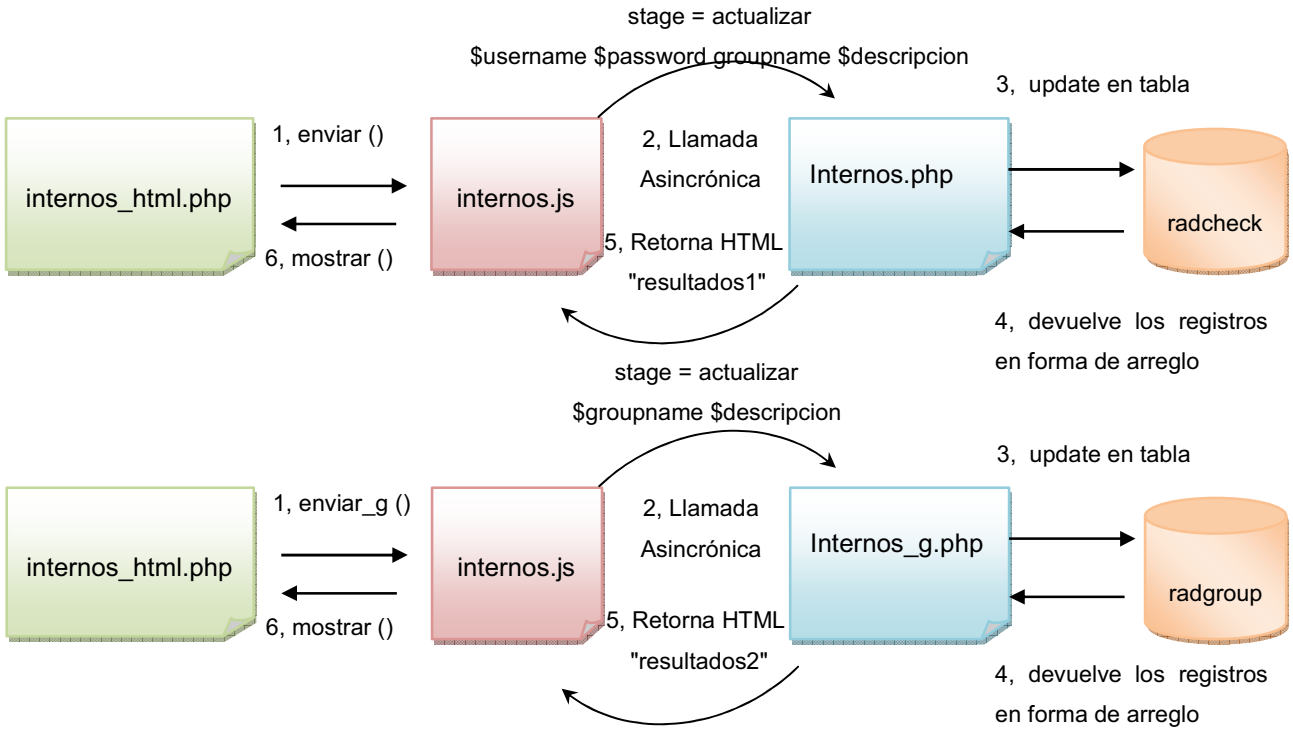


Figura 2.72 Lógica de la acción: Actualizar

La Figura 2.73 describe la lógica que se ejecuta en la aplicación Web cuando el usuario presiona el botón *Eliminar* en la página Usuarios Internos. El archivo *internos_html.php* llama a la función *borrar(),* si se lo hizo desde la pestaña Ingreso de Usuarios, o *borrar_g(),* si se lo hizo desde la pestaña Grupos de Usuarios, alojadas en el código JavaScript del archivo *internos.js.* Esta función despliega una ventana de JavaScript de confirmación preguntando si desea o no borrar el registro, si la respuesta es afirmativa realiza una llamada asincrónica al archivo *internos.php o* al archivo *internos_g.php* enviando a través del método POST el valor "borrar" en la variable stage.

Cuando esto sucede, en el archivo *internos.php o internos_g.php* se ejecuta la función borrar*()*, si es un grupo el que se desea borrar esta función revisa si este contiene usuarios; si contiene usuarios envía una ventana de alerta caso contrario procede a eliminar el registro. Si no se produce ningún error es llamada la función *mostrar()* como en las secciones anteriores.

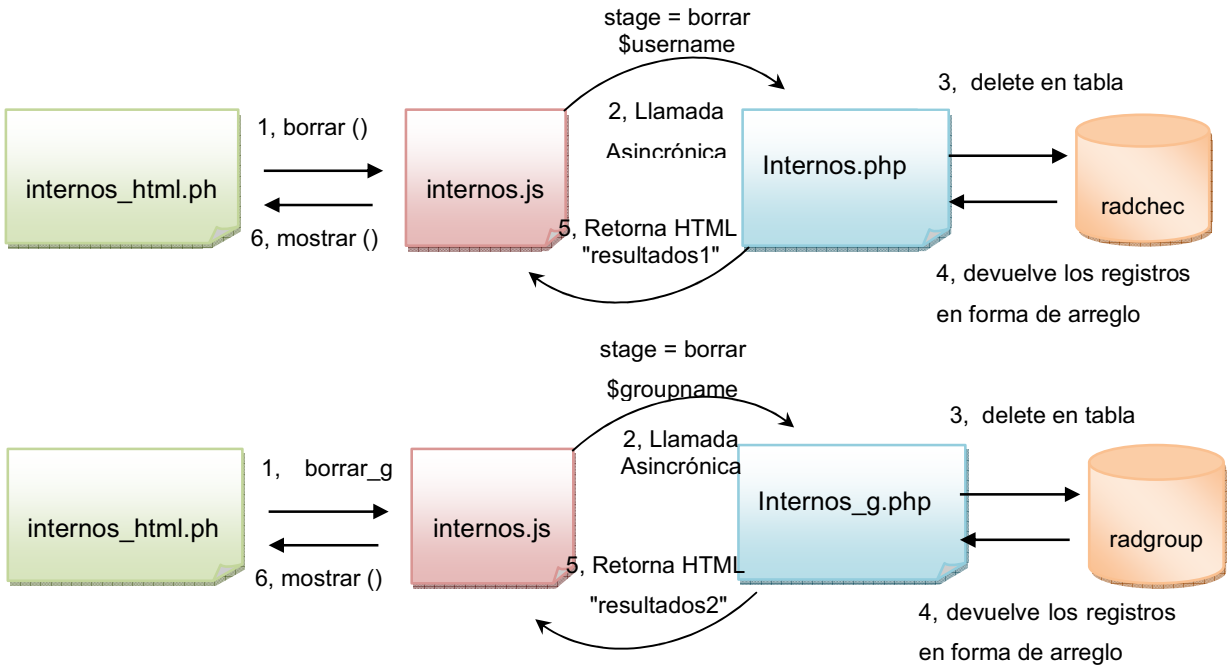


Figura 2.73 Lógica de la acción: Eliminar

2.5.4.2. Equipos Internos

Esta página se encuentra dividida en dos pestañas, la primera recupera los datos de los equipos internos, el grupo al que pertenecen y una descripción del equipo de la base de datos, mientras que la segunda recupera los nombres de los grupos existentes y su descripción correspondiente. Estos equipos y grupos pertenecen al repositorio interno dedicado únicamente a la autenticación por RADIUS específicamente se refieren al método de autenticación usando la dirección MAC. Estas dos pestañas se pueden observar en las Figuras 2.74 y 2.75.

Figura 2.74 Página Equipos Internos del servidor "SAAAS!", pestaña Ingreso de Equipos

Figura 2.75 Página Equipos Internos del servidor "SAAAS!" pestaña Grupos de Equipos

2.5.4.2.1. Lógica de la sección

La página Equipos Internos maneja dos páginas con código php: *mac.php* y *mac_g.php,* las cuales manejan de manera independiente las pestañas Ingreso de Equipos y Grupos de Equipos respectivamente, de igual forma recuperan y manipulan los datos de las tablas correspondientes en la base de datos, estas son: *radmaccheck* para los equipos y *radmacgroup* para los grupos (Ver Figura 2.76), el

caso de uso para la manipulación de equipos internos se lo presenta en el Anexo B20 y el de manipulación de grupos de equipos internos se lo presenta en el Anexo B21.

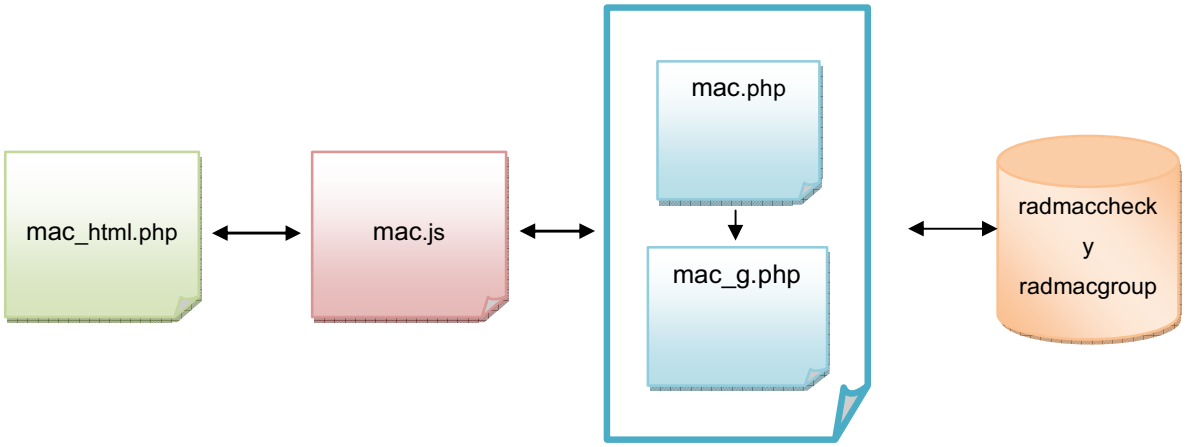


Figura 2.76 Esquema general de la página Equipos Internos

La Figura 2.77 ilustra la lógica que se ejecuta en la aplicación Web cuando la página Equipos Internos se carga por primera vez. El archivo *mac_html.php* llama a la función *inicio()* alojada en el código JavaScript del archivo *mac.js.* Esta función realiza una llamada asincrónica al archivo *mac.php* y al archivo *mac_g.php* enviando a través del método POST el valor "null" en la variable stage. Al recibir esta variable, en el archivo *mac.php* se ejecuta la función por defecto *mostrar()*, esta función genera el código HTML necesario para crear la tabla de visualización de los equipos, previamente realiza una consulta a la tabla *radmaccheck* de la base de datos obteniendo así los primeros diez registros, y genera el menú inferior apuntando a las funciones JavaScript correspondientes para la manipulación de estos registros.

De igual forma si la variable stage es "null"; en el archivo *mac_g.php* se ejecuta la función por defecto *mostrar()*, esta función genera el código HTML necesario para crear la tabla de visualización de los grupos, previamente realiza una consulta a la tabla *radmacgroup* de la base de datos obteniendo así los primeros diez registros, y genera el menú inferior apuntando a las funciones JavaScript correspondientes para

la manipulación de estos registros.

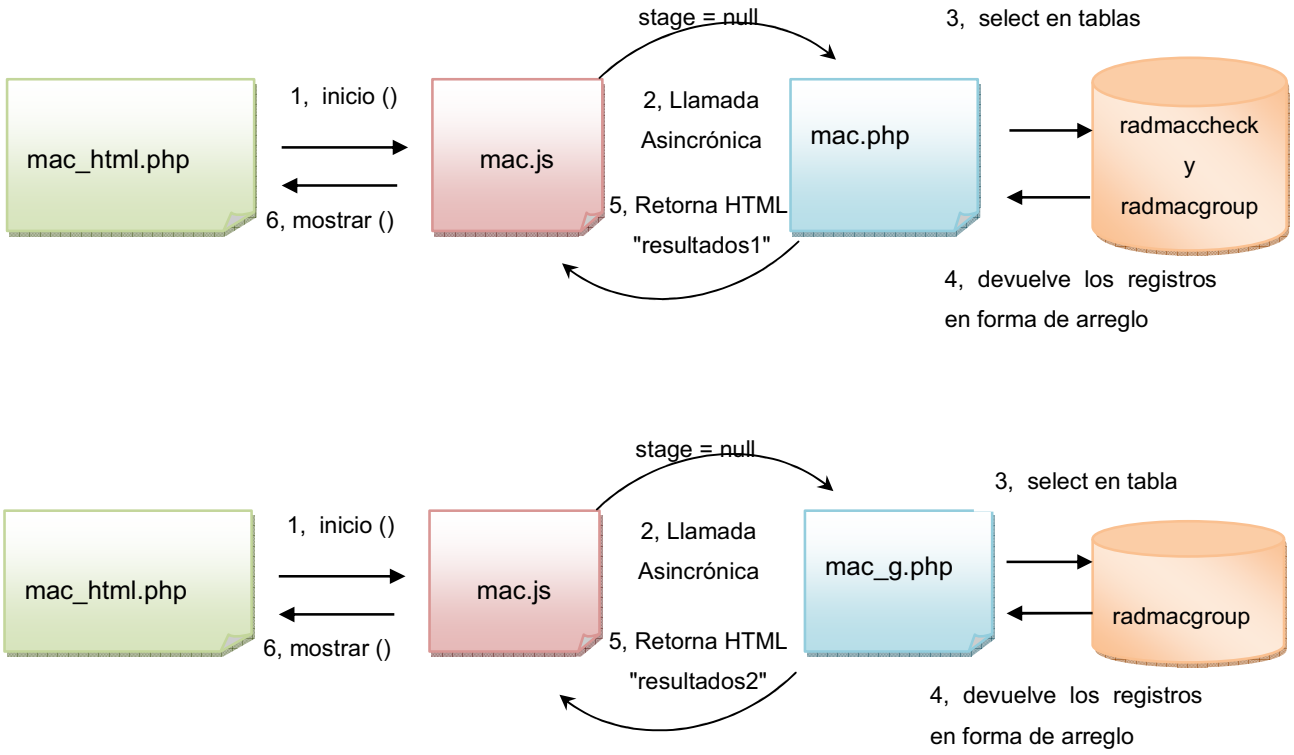


Figura 2.77 Lógica de la acción por defecto

La Figura 2.78 ilustra la lógica que se ejecuta en la aplicación cuando el usuario presiona el botón *Buscar* en las pestañas de la página Equipos Internos. La página *mac_html.php* llama a la función *buscar(),* si se lo hizo desde la pestaña Ingreso de Equipos, o *buscar_g(),* si se lo hizo desde la pestaña Grupos de Equipos, alojadas en el código JavaScript del archivo *mac.js.* Esta función despliega una ventana de JavaScript mostrando un campo vacío para ingresar la dirección MAC parcial o completa del equipo o nombre del grupo a buscar. Una vez ingresado este campo se realiza una llamada asincrónica al archivo *mac.php* o al archivo *mac_g.php* enviando a través del método POST el valor “null” en la variable stage y el valor ingresado por el usuario en la variable buscar.

Ya que la variable stage es “null”, en el archivo *mac.php o mac_g.php* se ejecuta la función por defecto *mostrar()*, y se genera el código HTML necesario para crear la tabla de visualización como se lo indicó en la sección anterior, con la diferencia que la consulta a realizarse está condicionada al valor ingresado, lo que mostraría todas las posibles coincidencias del valor ingresado con los registros almacenados.

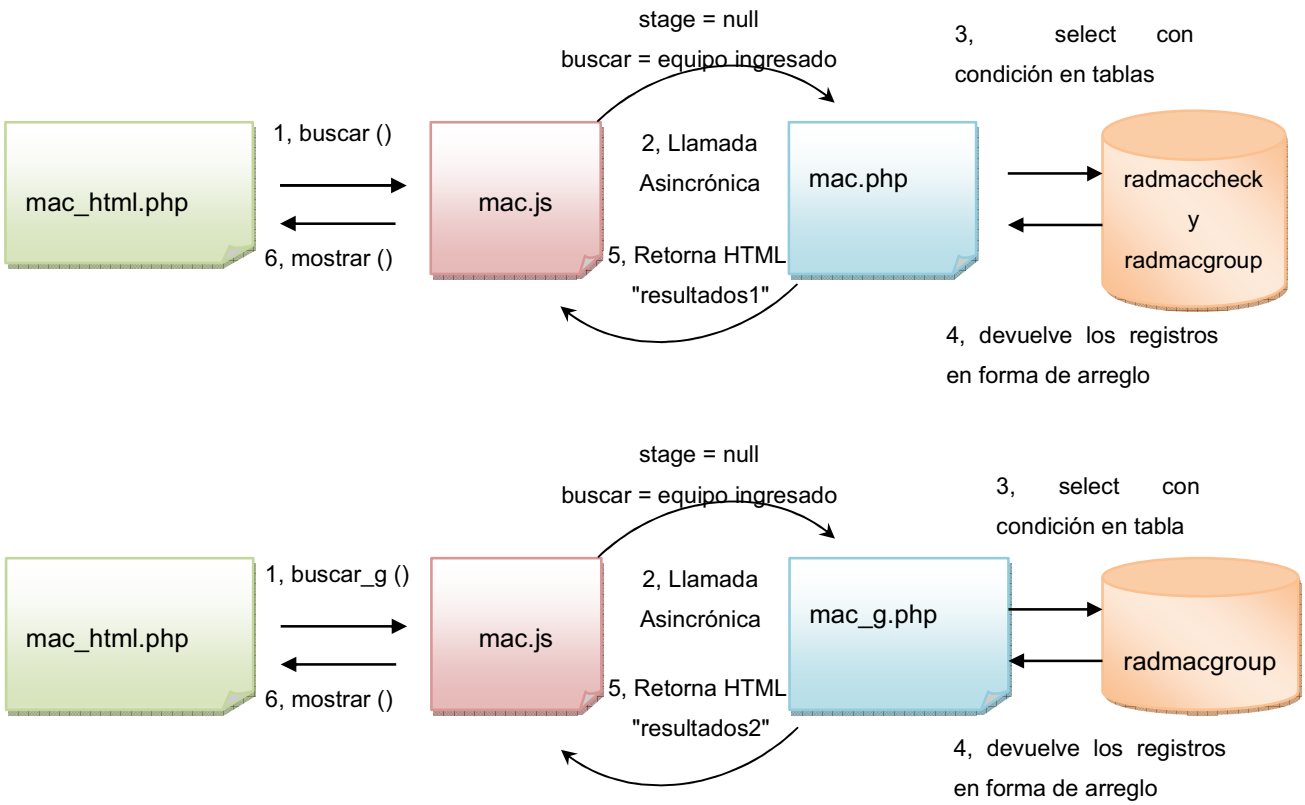


Figura 2.78 Lógica de la acción: Buscar

Al presionar en el botón *Nuevo*, la página *mac_html.php* llama a la función *enviar(),* si se lo hizo desde la pestaña Ingreso de Equipos, o *enviar_g(),* si se lo hizo desde la pestaña Grupos de Equipos, alojadas en el código JavaScript del archivo *mac.js.* Esta función realiza una llamada asincrónica al archivo *mac.php o al archivo mac_g.php* enviando a través del método POST el valor “null” en la variable stage y los valores “nuevoequipo” o “nuevogrupo” en la variable puntero. Al recibir estas

variables el archivo *mac.php o mac_g.php* ejecuta la función por defecto *mostrar()*, generando el código HTML necesario para crear la tabla de visualización como se lo indicó en las secciones anteriores, con la diferencia de que al final de la tabla de los registros almacenados se muestra una fila de campos vacíos para que el usuario ingrese los datos del nuevo equipo o grupo de equipos a crearse (ver Figura 2.79).

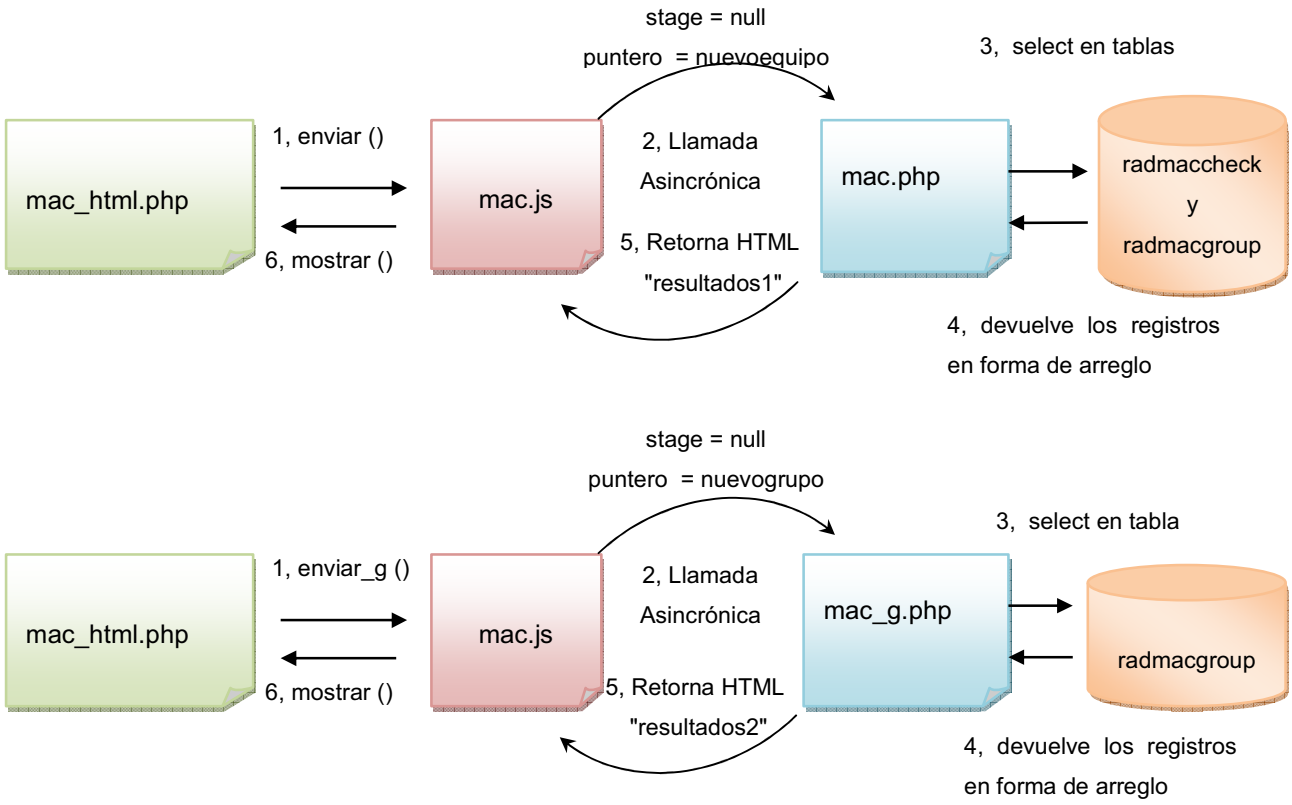


Figura 2.79 Lógica de la acción: Nuevo

Para guardar el nuevo registro el usuario debe hacer clic en el botón *Aceptar*. La Figura 2.80 ilustra los procesos que tienen lugar en la aplicación Web cuando esto sucede. La página *mac_html.php* llama a la función *enviar(),* si se lo hizo desde la pestaña Ingreso de Equipos, o *enviar_g(),* si se lo hizo desde la pestaña Grupos de Equipos, alojadas en el código JavaScript del archivo *mac.js.* Esta función realiza una llamada asincrónica al archivo *mac.php* o al archivo *mac_g.php* enviando a través del método POST el valor “nuevo” en la variable stage. Cuando esto sucede,

el archivo *mac.php o mac_g.php* ejecuta la función *nuevo()*, esta función recupera y revisa si los campos vacíos fueron ingresados correctamente y procede a insertar el registro en la base de datos, si los campos están vacíos o mal ingresados devuelve el error pertinente por medio de una ventana de alerta de JavaScript. Si no se produjo algún error se llama a la función *mostrar()* como en las secciones anteriores.

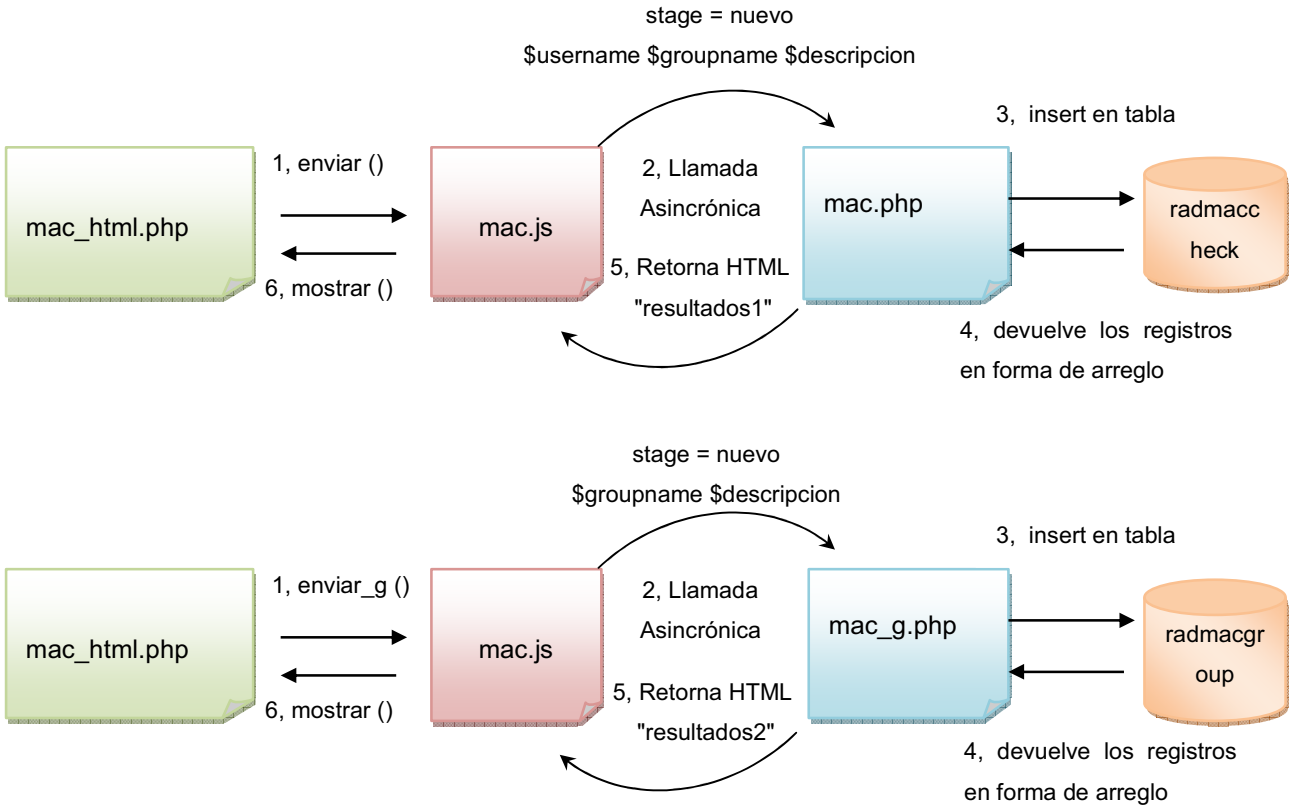


Figura 2.80 Lógica de la acción: Aceptar

El usuario también puede modificar los registros de equipos clientes y grupos de clientes existentes. La Figura 2.81 muestra la lógica que se ejecuta en la aplicación Web cuando esto sucede. Al presionar el botón *Actualizar*, la página *mac_html.php* llama a la función *enviar(),* si se lo hizo desde la pestaña Ingreso de Equipos, o *enviar_g(),* si se lo hizo desde la pestaña Grupos de Equipos, alojadas en el código JavaScript del archivo *mac.js.* Esta función realiza una llamada asincrónica al archivo

mac.php o al archivo *mac_g.php* enviando a través del método POST el valor de la variable stage con el valor “actualizar”.

Si la variable stage es “actualizar”, en el archivo *mac.php o mac_g.php* se ejecuta la función *actualizar()*, esta función recupera y revisa si los campos fueron ingresados correctamente y procede a actualizar el registro en la base de datos, si los campos están mal ingresados devuelve el error pertinente por medio de una ventana de alerta de JavaScript. Si no se produjo algún error se llama a la función *mostrar()* como en las secciones anteriores.

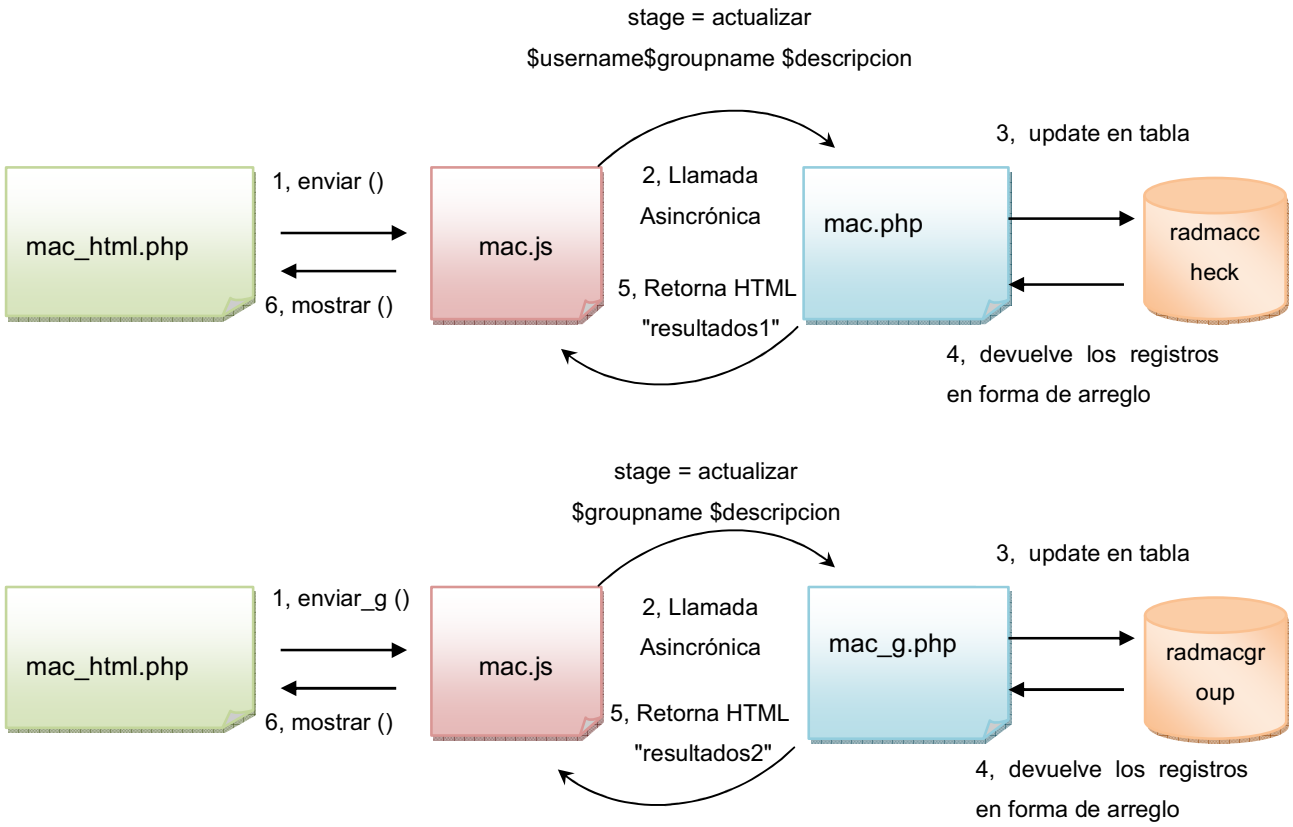


Figura 2.81 Lógica de la acción: Actualizar

Al presionar el botón *Eliminar*, la página *mac_html.php* llama a la función *borrar(),* si se lo hizo desde la pestaña Ingreso de Equipos, o *borrar_g(),* si se lo hizo desde la

pestaña Grupos de Equipos, alojadas en el código JavaScript del archivo *mac.js.* Esta función despliega una ventana de JavaScript de confirmación preguntando si desea o no borrar el registro, si la respuesta es afirmativa realiza una llamada asincrónica al archivo *mac.php* o al archivo *mac_g.php* enviando a través del método POST el valor "borrar" en la variable stage. Al recibir esta variable, el archivo *mac.php o mac_g.php* ejecuta la función borrar*()*, si es un grupo el que se desea borrar esta función revisa si este contiene equipos; si contiene equipos envía una ventana de alerta caso contrario procede a eliminar el registro. Si no se produjo algún error es llamada la función *mostrar()* como en las secciones anteriores (ver Figura 2.82).

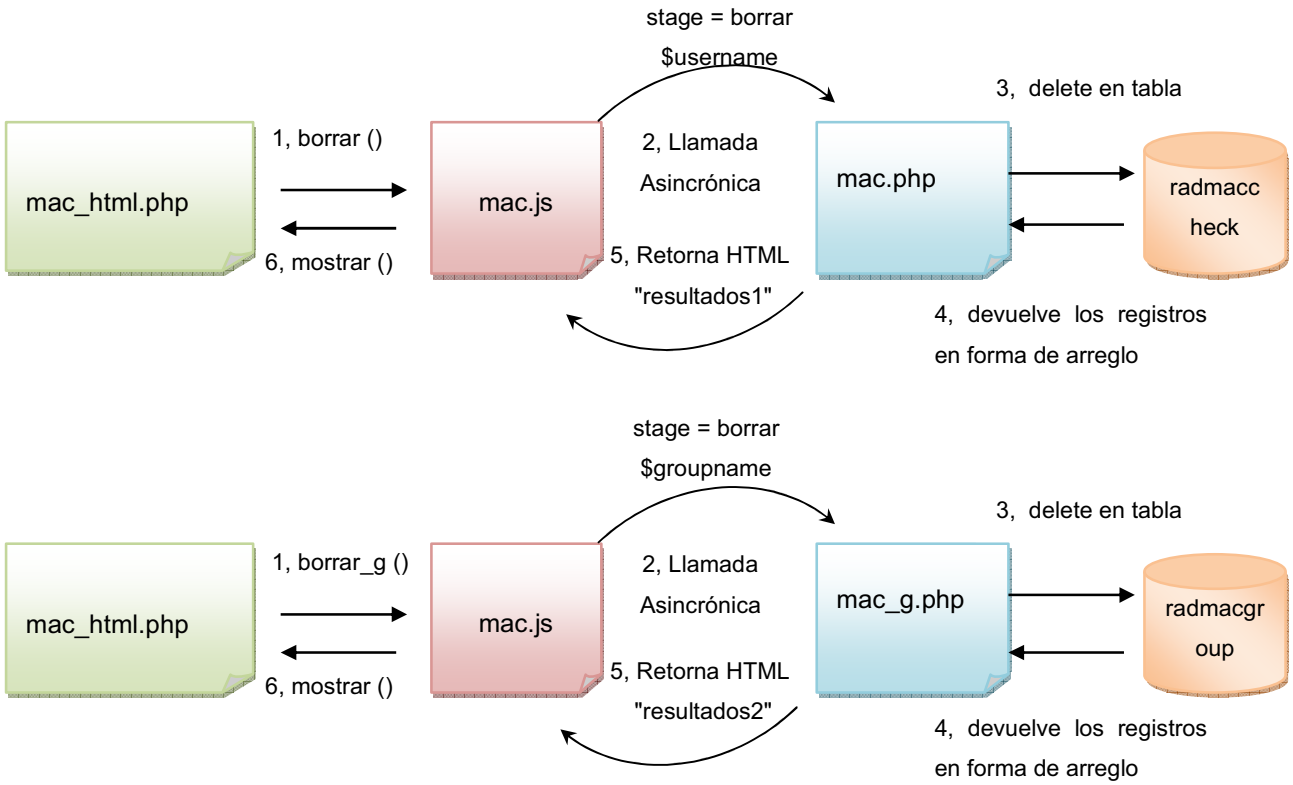


Figura 2.82 Lógica de la acción: Eliminar

2.5.4.3. Active Directory

Esta página se encuentra dividida en tres pestañas, la primera recupera las credenciales almacenadas para la integración con el Active Directory adicionalmente recupera el dominio y servidor DNS configurado en servidor, si la integración es satisfactoria se muestran las dos siguientes pestañas (ver Figura 2.83). La segunda muestra los grupos del Active Directory (ver Figura 2.84) y la tercera muestra dos cuadros de texto para verificar si el usuario y la clave (ver Figura 2.85).

Figura 2.83 Página Active Directory del servidor "SAAAS!" pestaña Configuración

Figura 2.84 Página Active Directory del servidor "SAAAS!" pestaña Grupos de Dominio

Figura 2.85 Página Active Directory del servidor "SAAAS!" pestaña Test Usuarios

2.5.4.3.1. Lógica de la sección

Como se observa en la Figura 2.86, la página Active Directory no manipula datos de la base de datos por lo que solo trabaja con el esquema general para las páginas antes descrito, la página con código php *ad.php* maneja los archivos de configuración tanto del servicio samba, de kerberos y del archivo de configuración del módulo *ldap* de FreeRADIUS.

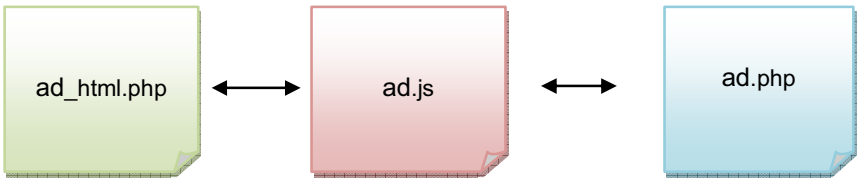


Figura 2.86 Esquema general de la página Active Directory

Como se ilustra en la Figura 2.87, al cargarse por primera vez la página *ad_html.php* llama a la función *inicio()* alojada en el código JavaScript del archivo *ad.js.* Esta función realiza una llamada asincrónica al archivo ad.*php* enviando a través del método POST el valor "null" en la variable stage. Al recibir esta variable el archivo *ad.php* ejecuta la función por defecto *mostrar()*, esta función genera el código HTML necesario para crear cuatro cuadros de texto, adicionalmente recupera la información de dominio y del servidor DNS y los muestra en dos de estos cuadros, si existe una configuración previa (archivo *config_ad.php*) recupera las credenciales para la integración del servidor con el Active Directory.

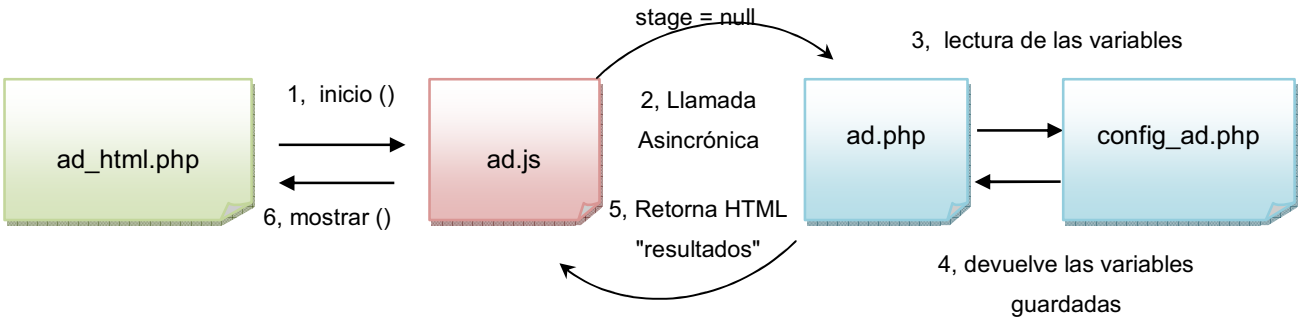


Figura 2.87 Lógica de la acción por defecto

La Figura 2.88 ilustra la lógica que se ejecuta en la aplicación Web cuando el usuario presiona el botón *Verificar* en la pestaña de Configuración de la página Active Directory. El archivo *ad_html.php* llama a la función *verificar(),* alojada en el código JavaScript del archivo *ad.js.* Se realiza una llamada asincrónica al archivo *ad.php* enviando a través del método POST el valor de las variables "verificar" en la variable stage y los valores de dominio, usuario, contraseña y ads en otras variables. Ya que

la variable stage es “verificar”, en el archivo *ad.php* se ejecuta la función *verificar()*, que edita los siguientes archivos en la carpeta etc: *krb5.conf*, *samba/smb.conf*, *freeradius/modules/da*, los primeros dos archivos sirven para la integración del servidor con el dominio, el siguiente contiene la configuración ldap entre el servidor “SAAAS!” y Active Directory. Adicionalmente, se realiza una comprobación del estado de conectividad con el Active Directory, este resultado se envía como código HTML hacia la pestaña *Configuración*, si la conexión es exitosa se envía adicionalmente el código HTML necesario para crear dos pestañas adicionales que permiten visualizar la información de los grupos de dominio y realizar un test de usuarios, finalmente se ejecuta la función mostrar como en secciones anteriores, el caso de uso para esta acción se lo presenta en el Anexo B22.

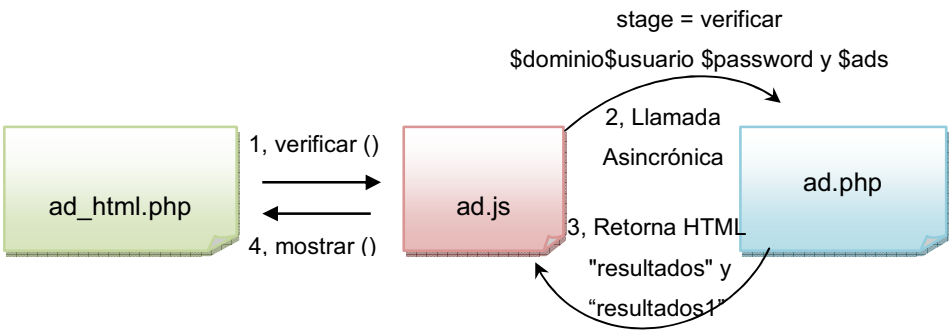


Figura 2.88 Lógica de la acción: Verificar

Cuando el usuario presiona el botón *Guardar*, la página *ad_html.php* llama a la función *guardar(),* alojada en el código JavaScript del archivo *ad.js.* Esta función realiza una llamada asincrónica al archivo *ad.php* enviando a través del método POST el valor “guardar” en la variable stage. Al recibir esta variable, el archivo *ad.php* ejecuta la función *guardar()*, que edita el archivo de configuración *config_ad.php.* En este archivo se almacenan las contraseñas de la integración con el directorio activo, de igual manera se ejecuta la verificación y si la conexión es exitosa se envía el código HTML necesario para crear dos pestañas adicionales que permiten visualizar los grupos de dominio y realizar un test de usuarios, finalmente se

ejecuta la función *mostrar()* como en secciones anteriores (ver Figura 2.89), el caso de uso para esta acción se lo presenta en el Anexo B23.

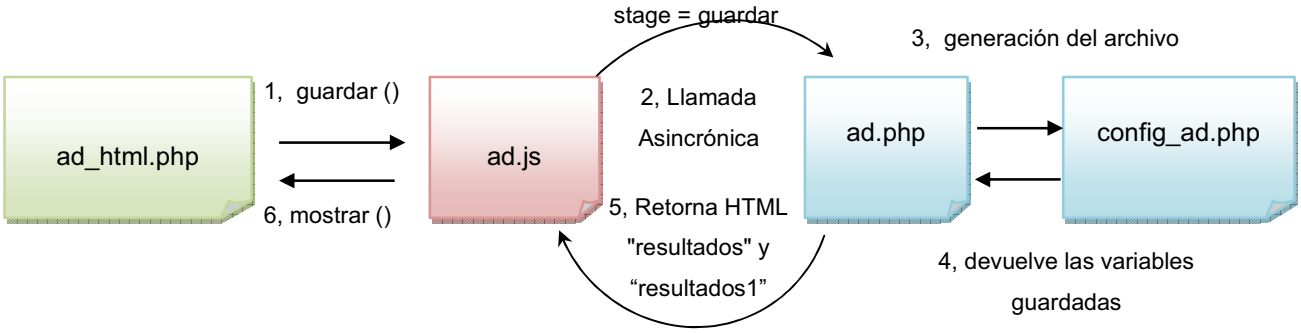


Figura 2.89 Lógica de la acción: Guardar

Cuando el usuario presiona el botón *Buscar* en la pestaña Test de Usuarios de la página Active Directory, la aplicación Web ejecuta la lógica que se ilustra en la Figura 2.90 y se describe a continuación. La página *ad_html.php* llama a la función *test(),* alojada en el código JavaScript del archivo *ad.js.* Se realiza una llamada asincrónica al archivo *ad.php* enviando a través del método POST el valor “buscaru” en la variable stage y el valor indicado por el usuario en la variable usuariot. Cuando el archivo *ad.php* recibe estas variables, ejecuta la función *test()*, que hace una consulta LDAP hacia el Active Directory; primero comprueba que el usuario exista, si es así hace una consulta de los grupos de dominio a los que pertenece caso contrario devuelve el mensaje “Usuario no Encontrado”, el caso de uso para esta acción se lo presenta en el Anexo B24.

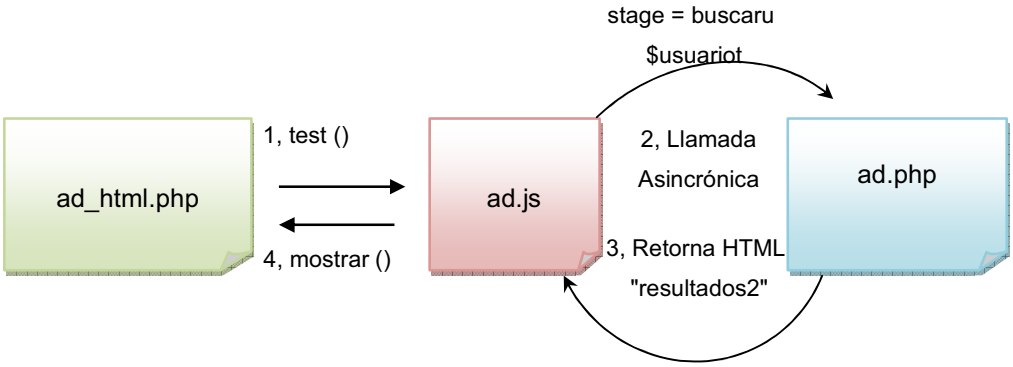


Figura 2.90 Lógica de la acción: Buscar

Al presionar el botón *Verificar* en la pestaña Test de Usuario, la página *ad_html.php* en la pestaña “Test Usuarios” llama a la función *test(),* alojada en el código JavaScript del archivo *ad.js.* Se realiza una llamada asincrónica al archivo *ad.php* enviando a través del método POST el valor *buscarup* en la variable stage y los datos del nombre de usuario y contraseña en otras variables. Cuando el archivo *ad.php* recibe esta variable, ejecuta la función *testp()*, que hace un requerimiento de autenticación al Active Directory; primero comprueba que el usuario exista, si es así comprueba que la contraseña haya sido ingresada correctamente caso contrario devuelve el mensaje “Usuario no Encontrado” o “Contraseña Incorrecta” (ver Figura 2.91).

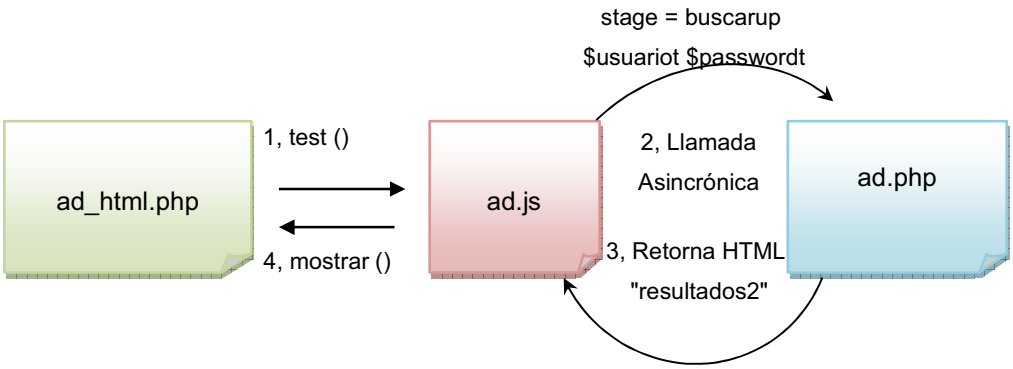


Figura 2.91 Lógica de la acción: Verificar Usuario

2.5.4.4. LDAP

Esta página se encuentra dividida en tres pestañas, la primera recupera las credenciales y parámetros almacenados para la integración con LDAP (ver Figura 2.92), si la integración es satisfactoria aparece la segunda pestaña que muestra los grupos de LDAP (ver Figura 2.93) y la tercera pestaña que muestra dos cuadros de texto para verificar usuarios y claves en el servidor LDAP (ver Figura 2.94).

Figura 2.92 Página LDAP del servidor "SAAAS!", pestaña Configuración

Figura 2.93 Página LDAP del servidor "SAAAS!" pestaña Grupos de Dominio

Figura 2.94 Página LDAP del servidor "SAAAS!" pestaña Test Usuarios

2.5.4.4.1. Lógica de la sección

La página LDAP no manipula datos de la base de datos y solo trabaja con el esquema general para las páginas antes descrito. La página con código php *ldap.php* maneja el archivo de configuración del módulo LDAP de FreeRADIUS (ver Figura 2.95).

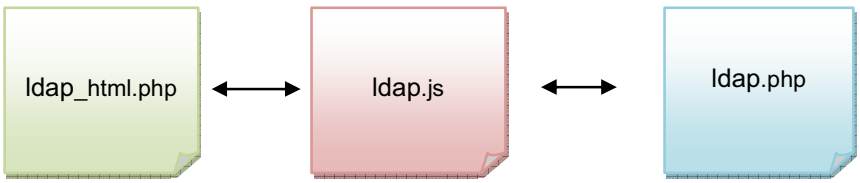


Figura 2.95 Esquema general de la página LDAP

Como se observa en la Figura 2.96, al cargarse por primera vez la página *ldap_html.php* llama a la función *inicio()* alojada en el código JavaScript del archivo *ldap.js.* Esta función realiza una llamada asincrónica al archivo *ldap.php* enviando a través del método POST el valor “null” en la variable stage; el código del archivo *ldap.php* se ejecuta de acuerdo al tipo de acción solicitada. Cuando el archivo *ldap.php* recibe esta variable ejecuta la función por defecto *mostrar()*, esta función genera el código HTML necesario para crear varios cuadros de texto, si existe una

configuración previa (archivo *config_ldap.php*) recupera las credenciales para la integración del servidor con LDAP.

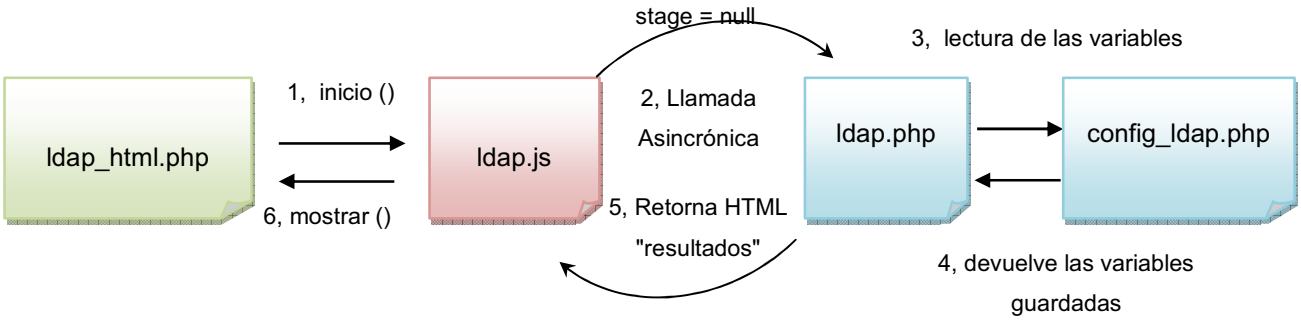


Figura 2.96 Lógica de la acción por defecto

Si el usuario desea verificar la conexión de “SAAAS!” con el servidor LDAP debe presionar el botón *Verificar*. Cuando esto sucede la página *ldap_html.php* llama a la función *verificar(),* alojada en el código JavaScript del archivo *ldap.js.* Se realiza una llamada asincrónica al archivo *ldap.php* enviando a través del método POST el valor “verificar” en la variable stage y los datos necesarios para verificar la conexión con el servidor LDAP. Cuando el archivo *ldap.php* recibe esta variable ejecuta la función *verificar()*, que edita el archivo *etc/freeradius/modules/ld*, este archivo se encarga de la configuración de la comunicación ldap entre el servidor “SAAAS!” y el LDAP. Adicionalmente, se realiza una comprobación sobre el estado de conectividad con LDAP, este resultado es enviado como código HTML hacia la pestaña Configuración, si la conexión es exitosa se envía adicionalmente el código HTML necesario para crear dos pestañas que permiten la visualización de los grupos de dominio y un test de usuarios, luego se ejecuta la función mostrar como en secciones anteriores (ver Figura 2.97), el caso de uso para esta acción se lo presenta en el Anexo B25.

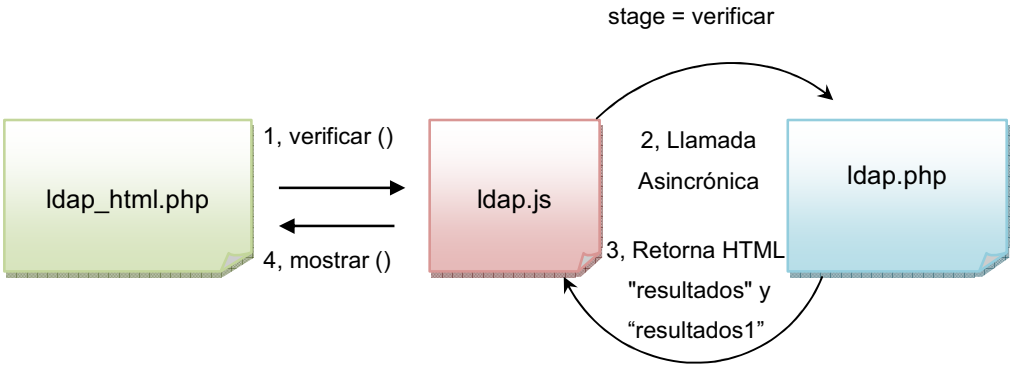


Figura 2.97 Lógica de la acción: Verificar

La lógica que se ejecuta en la aplicación Web cuando el usuario quiere guardad la configuración para la conexión con el servidor LDAP se ilustra en la Figura 2.98 y se describe a continuación. Al presionar el botón *Guardar*, la página *ldap_html.php* llama a la función *guardar(),* alojada en el código JavaScript del archivo *ldap.js.* Esta función realiza una llamada asincrónica al archivo *ldap.php* enviando a través del método POST el valor “guardar” en la variable stage y las variables necesarias para la configuración del servicio.

Cuando el archivo *ldap.php* recibe estas variables, se ejecuta la función *guardar()*, que edita el archivo de configuración *config_ldap.php,* en este archivo se almacenan las contraseñas de la integración con el LDAP; de igual manera se ejecuta la verificación y si la conexión es exitosa se envía adicionalmente el código HTML necesario para crear dos pestañas adicionales que permiten la visualización de los grupos de dominio y un test de usuarios, luego se ejecuta la función mostrar como en secciones anteriores, el caso de uso para esta acción se lo presenta en el Anexo B26.

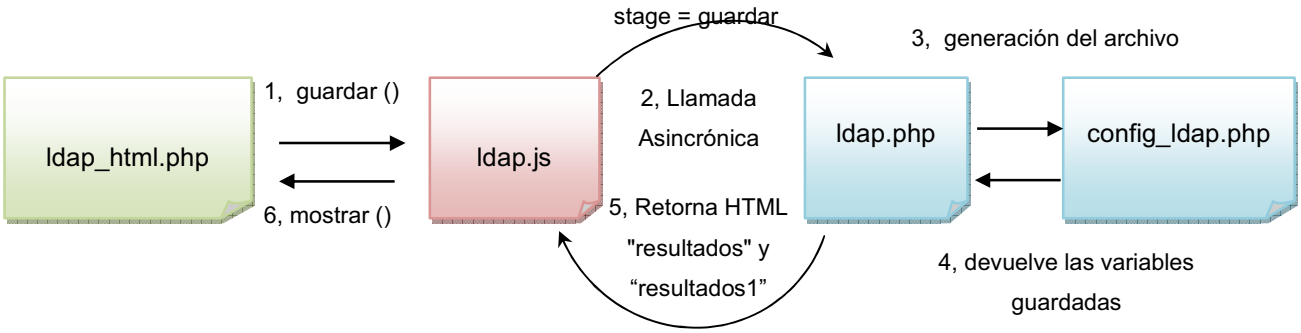


Figura 2.98 Lógica de la acción: Guardar

Al presionar el botón *Buscar*, la página *ldap_html.php* en la pestaña Test Usuarios llama a la función *test(), alojada* en el código JavaScript del archivo *ldap.js.* Se realiza una llamada asincrónica al archivo *ldap.php* enviando a través del método POST el valor “buscaru” en la variable stage y el valor ingresado por el usuario en la variable usuariot. Al recibir estas variables el archivo *ldap.php* ejecuta la función *test()*, que primero comprueba que el usuario exista en el repositorio de credenciales, si es así hace una consulta de los grupos de dominio a los que pertenece caso contrario devuelve el mensaje “Usuario no Encontrado” (ver Figura 2.99) , el caso de uso para esta acción se lo presenta en el Anexo B27.

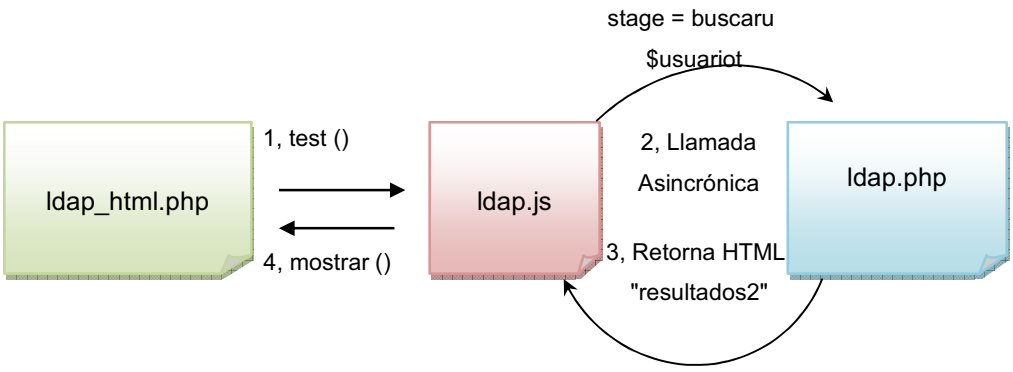


Figura 2.99 Lógica de la acción: "Buscar"

La Figura 2.100 ilustra la lógica que se ejecuta en la aplicación Web cuando el usuario quiere realizar el Test de Usuarios. Al presionar el botón *Verificar* en la

pestaña Test de Usuarios, la página *ldap_html.php* llama a la función *test(), alojada* en el código JavaScript del archivo *ldap.js.* Se realiza una llamada asincrónica al archivo *ldap.php* enviando a través del método POST el valor "buscarup" en la variable stage y los valores de nombre de usuario y contraseña que ingresó el usuario. Al recibir estas variables, el archivo *ldap.php* ejecuta la función *testp()*, la que comprueba que el usuario exista en el repositorio LDAP, si es así comprueba que la contraseña haya sido ingresada correctamente caso contrario devuelve el mensaje "Usuario no Encontrado" o "Contraseña Incorrecta".

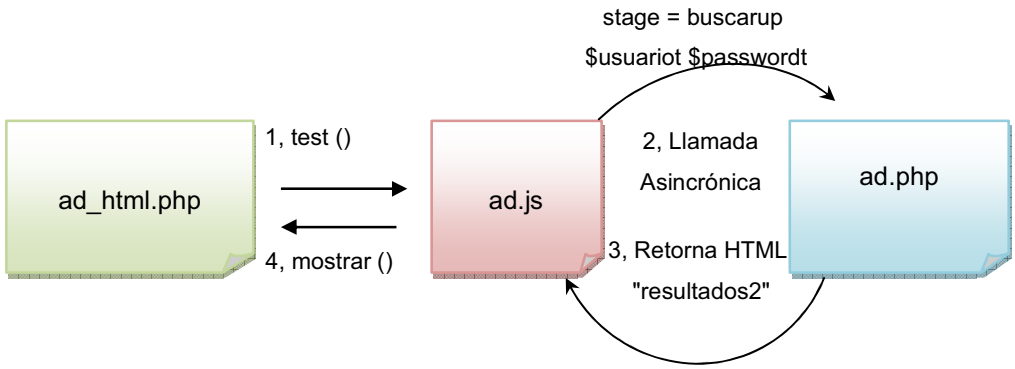


Figura 2.100 Lógica de la acción: Verificar Usuario

2.5.5. POLÍTICAS DE ACCESO

2.5.5.1. Reglas de Autenticación

Esta página se encuentra dividida en dos pestañas: Repositorios y Edición de Reglas. La pestaña Repositorios se encarga de recuperar la configuración de los repositorios de credenciales asociados a cada una de las reglas de autenticación permitiendo editar esta asociación aumentando o disminuyendo los repositorios (ver Figura 2.101). La pestaña Edición de Reglas recupera las condiciones asociadas a cada una de las reglas de autenticación permitiendo manipular estas condiciones adicionalmente permite el ingreso de nuevas reglas (ver Figura 2.102).

Figura 2.101 Página Reglas de Autenticación del servidor "SAAAS!" pestaña Repositorios

Figura 2.102 Página Reglas de Autenticación del servidor "SAAAS!" pestaña Edición de Reglas.

2.5.5.1.1. Lógica de la sección

La página Reglas de Autenticación utiliza dos archivos con código php reglas.*php* y *reglas_1.php* los cuales manejan de manera independiente las pestañas Repositorios y Edición de Reglas respectivamente, de igual forma recuperan y manipulan los datos de las tablas temporales, estas tablas se destruyen y se crean al momento de

cargar el modulo Reglas de Autenticación, correspondientes en la base de datos, éstas son: *auxiliar1* para los repositorios y *auxiliar2* para las condiciones (ver Figura 2.103).

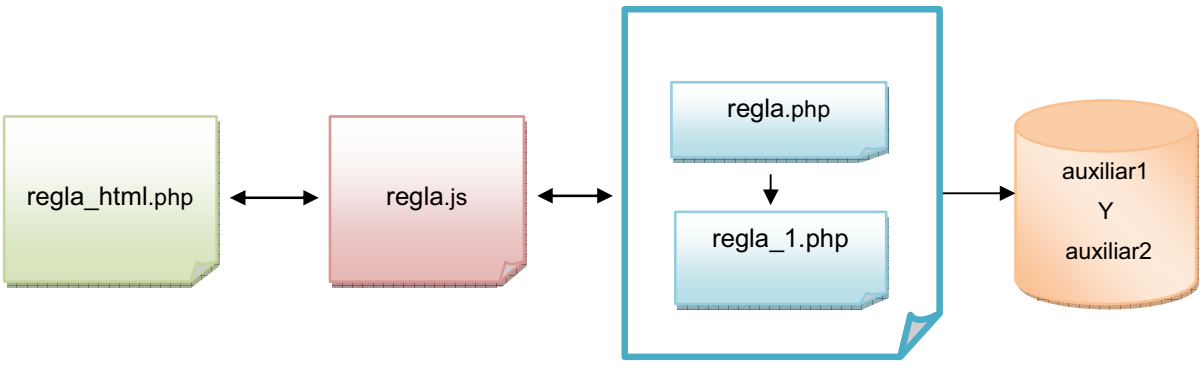


Figura 2.103 Esquema general de la página Reglas de Autenticación.

Al cargarse por primera vez, la página *reglas_html.php* llama a la función *inicio()* alojada en el código JavaScript del archivo *reglas.js.* Esta función realiza una llamada asincrónica al archivo *reglas.php* y al archivo *reglas_1.php* enviando a través del método POST el valor "null" en la variable stage=null; Al recibir esta variable los archivos *reglas.php y reglas_1.php* ejecutan la función por defecto *lectura()*, esta función lee el archivo de configuración de FreeRADIUS */etc/freeradius/sites-enabled/authorize* recuperando las reglas, las condiciones y los repositorios asociados. Estos datos son almacenados en las dos tablas temporales creadas en ese momento. En la tabla *auxiliar1* se almacena los repositorios y en la tabla *auxiliar2* las condiciones, una vez realizado este proceso se procede a leer las dos tablas y a generar el código HTML para visualizar la información de las políticas mediante la función *mostrar().* Esta lógica se ilustra en la Figura 2.104.

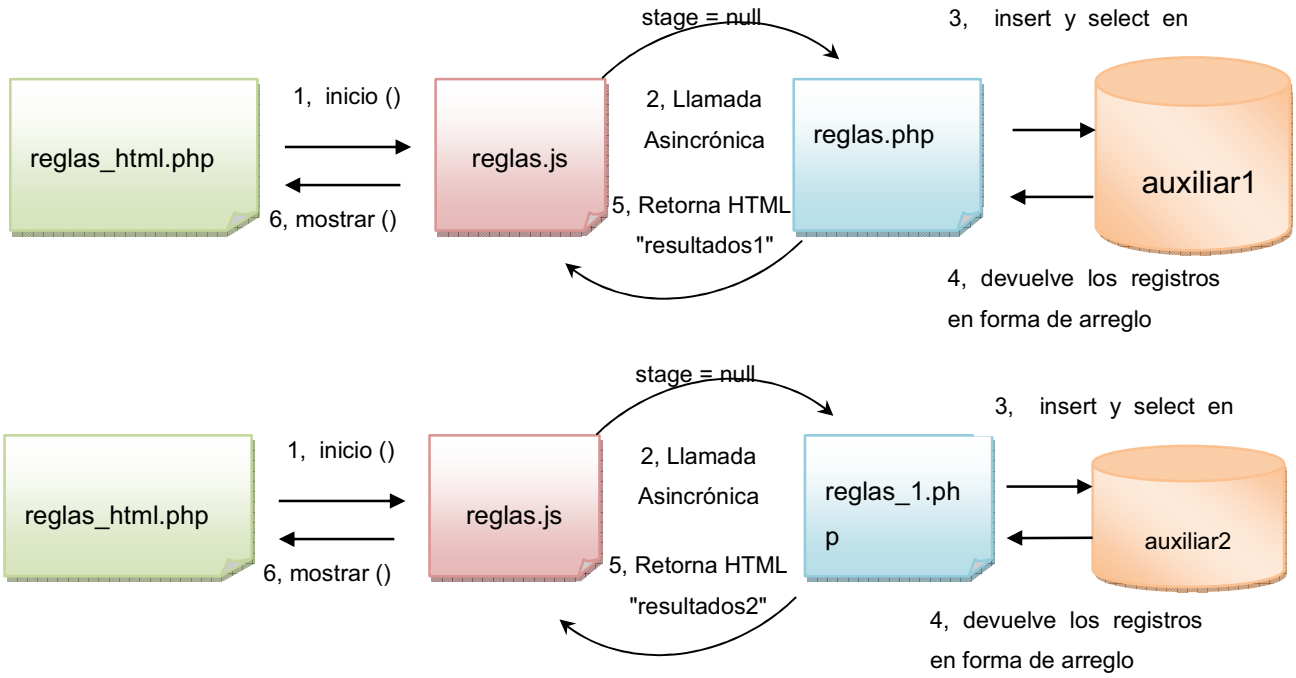


Figura 2.104 Lógica de la acción por defecto

El usuario puede visualizar el detalle de las reglas existentes al presionar sobre ellas en la pestaña Repositorios o en la pestaña Edición de Reglas. La lógica que se ejecuta en la aplicación Web cuando esto sucede se ilustra en la Figura 2.105 y se describe a continuación. La página *reglas_html.php* llama a la función *enviar()* o *enviar1()* dependiendo de la pestaña en la que se solicitó la acción. Estas funciones se encuentran en el archivo *reglas.js* y realiza una llamada asincrónica al archivo *reglas.php o reglas_1.php*, respectivamente, enviando a través del método POST el valor “mostrar” en la variable stage.

Al recibir esta variable el archivo *reglas.php* ejecuta la función *mostrar()*, esta función lee los registros de la tabla auxiliar1 y de acuerdo a la regla que fue presionada se genera el código HTML para visualizar los datos asociados a esa regla. El archivo *reglas_1.php* en cambio, lee los registros de la tabla auxiliar2 y de acuerdo a la regla que fue presionada se genera el código HTML para visualizar los datos asociados a

esa regla, el caso de uso para esta acción se lo presenta en el Anexo B28.

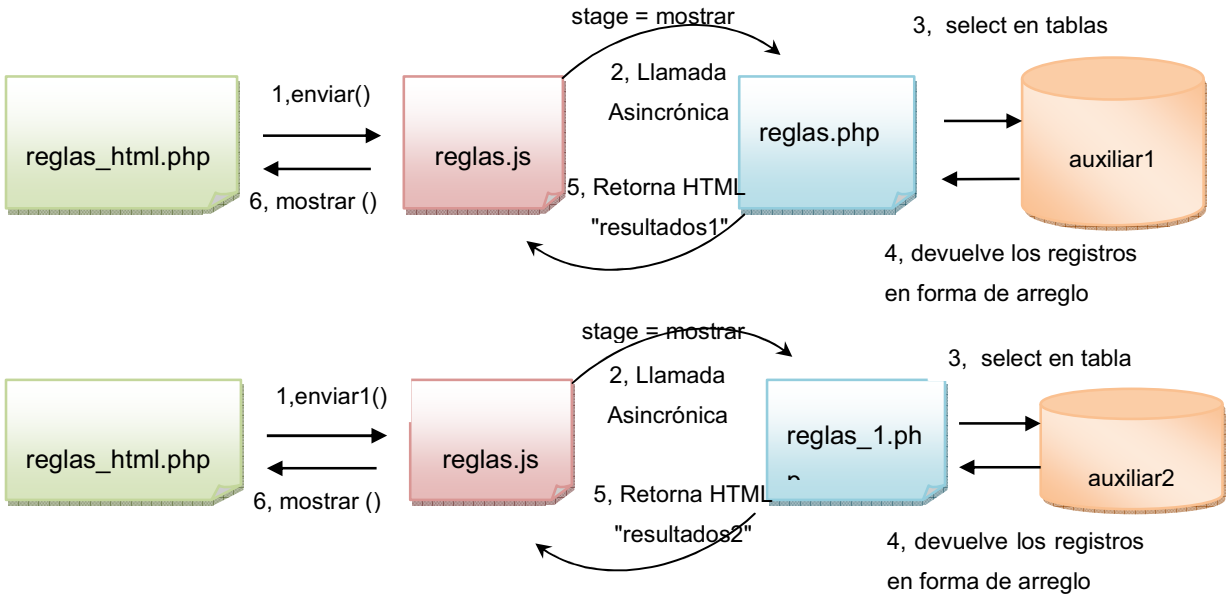


Figura 2.105 Lógica de la acción Seleccionar Regla

La opción *Aplicar* de la pestaña Repositorios permite guardar los cambios realizados en la configuración de FreeRADIUS. La página *reglas_html.php* llama a la función *enviar()* alojada en el código JavaScript del archivo *reglas.js.* Esta función realiza una llamada asincrónica al archivo *reglas.php* enviando a través del método POST el valor “aplicar” en la variable stage. Al recibir esta variable el archivo *reglas.php* ejecuta la función *aplicar()*, esta función lee los registros de la tabla auxiliar1 y auxiliar2, en base a estos registros reconstruye el archivo de configuración y lo remplaza por el de FreeRADIUS, luego llama a la función por defecto *lectura()* (ver Figura 2.106).

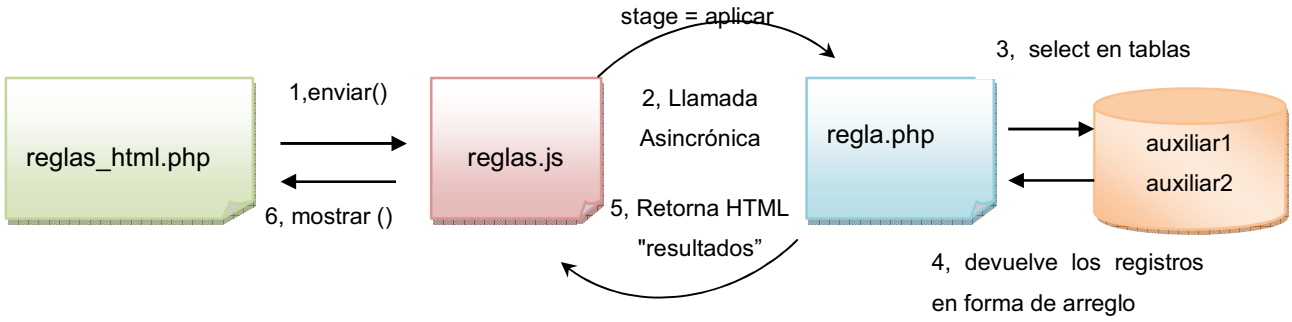


Figura 2.106 Lógica de la acción Aplicar

En la Figura 2.107 se ilustra la lógica que se ejecuta en la aplicación Web cuando un usuario intenta crear una nueva regla en la página Reglas de Autenticación. Al presionar la opción *Nueva Regla*, la página *reglas_html.php* llama a la función *enviar1()* alojada en el código JavaScript del archivo *reglas.js.* Esta función realiza una llamada asincrónica al archivo *reglas_1.php* enviando a través del método POST el valor "nuevo" en la variable stage. Al recibir esta variable el archivo *reglas_1.php* ejecuta la función *nuevo()*, esta función muestra un cuadro de selección para el tipo de condición a ingresar y un cuadro de texto para el valor de ésta, el caso de uso para la manipulación de las Reglas de Autenticación se lo presenta en el Anexo B29.

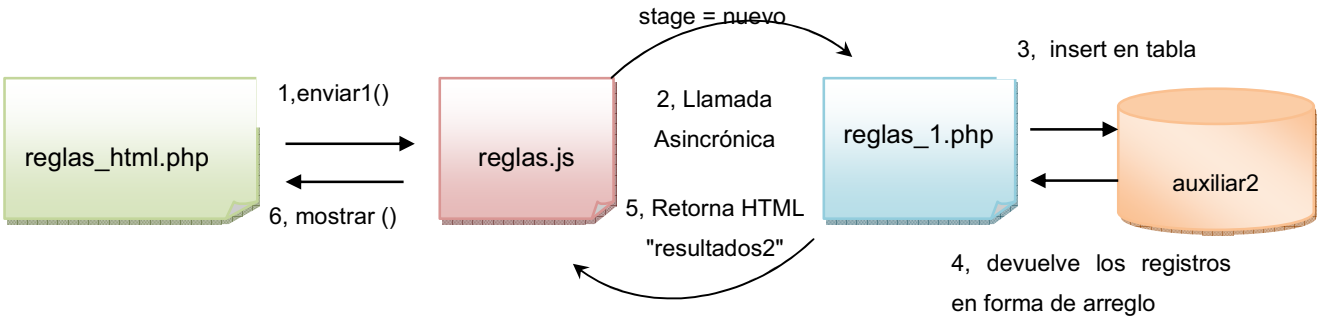


Figura 2.107 Lógica de la acción Nueva Regla

Al presionar sobre la condición y luego presionar la opción *Agregar Condición*, la página *reglas_html.php* llama a la función *enviar1()* alojada en el código JavaScript

del archivo *reglas.js.* Esta función realiza una llamada asincrónica al archivo *reglas_1.php* enviando a través del método POST el valor “condición” en la variable stage. Al recibir esta variable el archivo *reglas_1.php* ejecuta la función condicion*()*, esta función recupera el ID de la regla a la que se le va a adicionar la condición, luego muestra un cuadro de selección para agregar el tipo de condición y un cuadro de texto para el valor de esta (ver Figura 2.108)

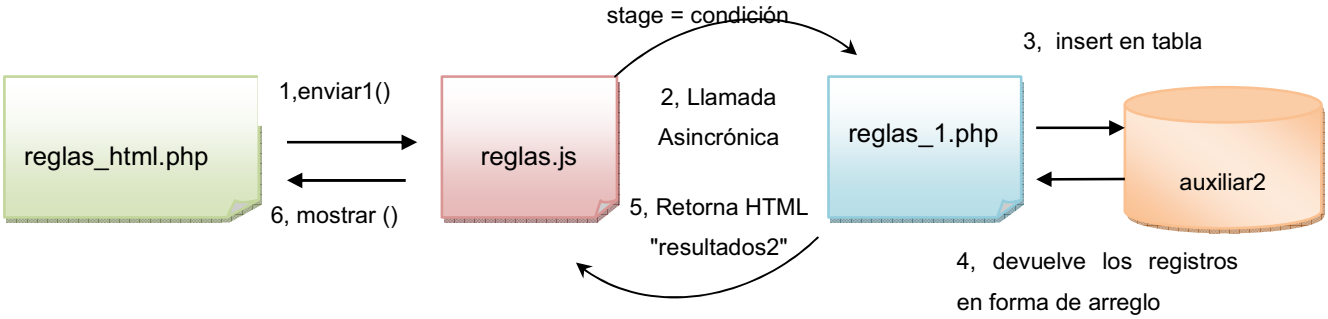


Figura 2.108 Lógica de la acción Agregar Condición

Al presionar sobre la condición y luego presionar la opción *Eliminar*, la página *reglas_html.php* llama a la función *enviar1()* alojada en el código JavaScript del archivo *reglas.js.* Esta función discrimina si se va a eliminar una regla o una condición, luego muestra una ventana de confirmación, si la respuesta es afirmativa realiza una llamada asincrónica al archivo *reglas_1.php* enviando a través del método POST el valor “eliminar” en la variable stage. Si la variable stage es: eliminar*;* en el archivo *reglas_1.php* se ejecuta la función eliminar*()*, de acuerdo al objeto seleccionado (regla o condición) esta función procede eliminar los registros de la tabla *auxiliar2* de la base de datos. Esta lógica se resume en la Figura 2.109.

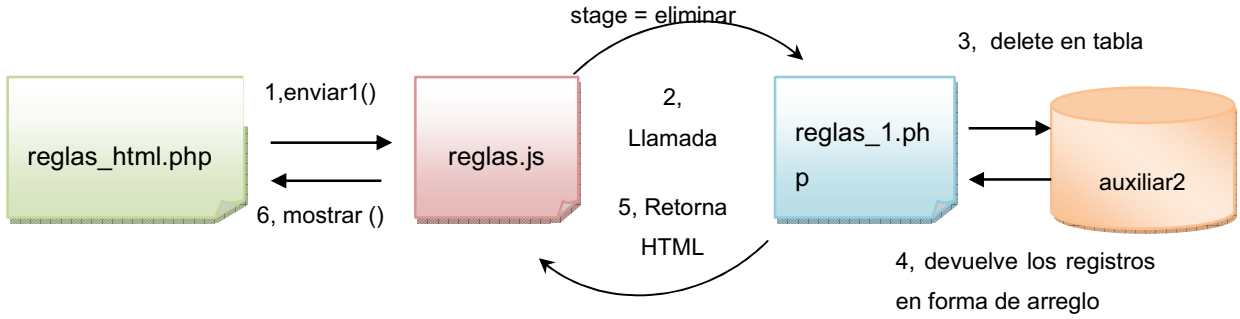


Figura 2.109 Lógica de la acción Eliminar

Al presionar sobre la condición y luego presionar la opción *modificar*, la página *reglas_html.php* llama a la función *enviar1()* alojada en el código JavaScript del archivo *reglas.js.* Esta función realiza una llamada asincrónica al archivo *reglas_1.php* enviando a través del método POST el valor "modificar" en la variable stage. Al recibir esta información el archivo *reglas_1.php* ejecuta la función *modificar(),* primero recupera el ID de la condición a modificar, luego realiza la actualización del nuevo valor en la tabla *auxiliar2* de la base de datos (ver Figura 2.110).

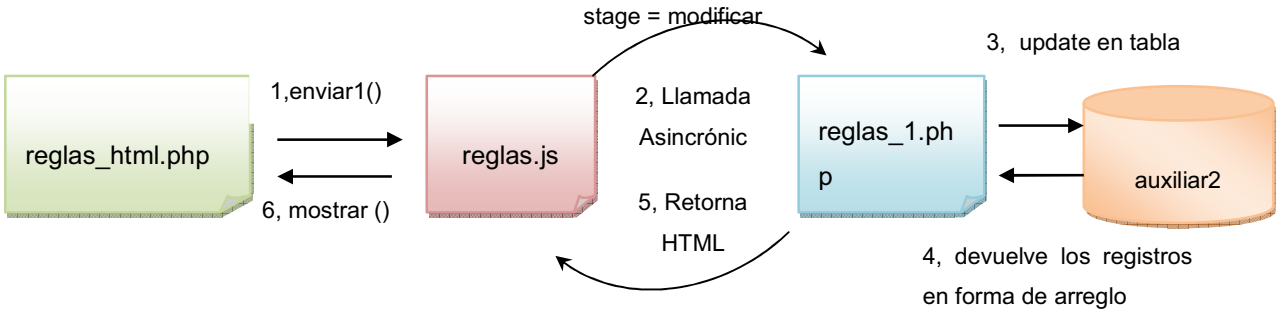


Figura 2.110 Lógica de la acción Modificar

2.5.5.2. Perfiles de Red

Esta página se encuentra dividida en dos pestañas: Perfiles y Diccionario de Atributos. La pestaña Perfiles se encarga de recuperar la configuración de los perfiles de autorización ingresados en la configuración de FreeRADIUS permitiendo editar los atributos asociados a cada perfil y crear nuevos perfiles (ver Figura 2.111). La pestaña Diccionario de Atributos recupera los archivos del diccionario interno de FreeRADIUS, estos archivos son de solo lectura y permiten al administrador consultar los atributos definidos en el RFC de RADIUS y otros atributos propietarios para utilizarlos en los perfiles de autorización (ver Figura 2.112).

Figura 2.111 Página Perfiles de Red del servidor "SAAAS!" pestaña Perfiles

Figura 2.112 Página Perfiles de Red del servidor "SAAAS!" pestaña Diccionario de Atributos.

2.5.5.2.1. Lógica de la sección

Como se ve en la Figura 2.113, la página Perfiles de Red maneja una tabla temporal de la base de datos de nombre *auxiliar* que almacena la información de los perfiles y los atributos asociados.

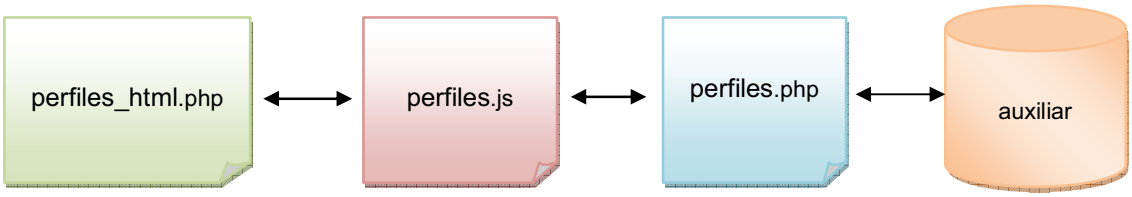


Figura 2.113 Esquema general de la página Perfiles de Red.

Al cargarse por primera vez, la página *perfiles_html.php* llama a la función *inicio()* alojada en el código JavaScript del archivo *perfiles.js.* Esta función realiza una llamada asincrónica al archivo *perfiles.php* enviando a través del método POST el valor "null" en la variable stage. Al recibir esta variable, el archivo *perfiles.php* ejecuta la función por defecto lectura(), esta función lee el archivo de configuración de FreeRADIUS *"/etc/freeradius/policy.conf"* recuperando los perfiles y los atributos asociados. Estos datos son almacenados en una tabla temporal (creada en ese momento), una vez realizado este proceso se procede a leer estos registros y a

generar el código HTML para visualizar los datos en forma de tabla esto lo realiza la función *mostrar().* Estos procedimientos se ilustran en la Figura 2.114.

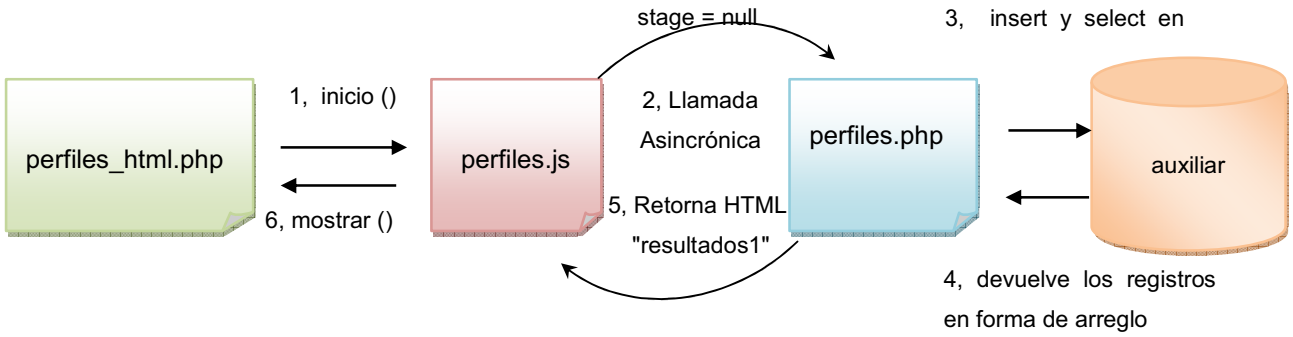


Figura 2.114 Lógica de la acción por defecto

Al presionar sobre uno de los perfiles en la pestaña Perfiles, la página *perfiles_html.php* llama a la función *enviar()* alojada en el código JavaScript del archivo *perfiles.js.* Esta función realiza una llamada asincrónica al archivo *perfiles.php* enviando a través del método POST el valor “mostrar” en la variable stage. Al recibir esta información, el archivo *perfiles.php* ejecuta la función *mostrar()*, esta función lee los registros de la tabla auxiliar y de acuerdo al perfil que fue presionado se genera el código HTML para visualizar los datos asociados a ese perfil (ver Figura 2.115).

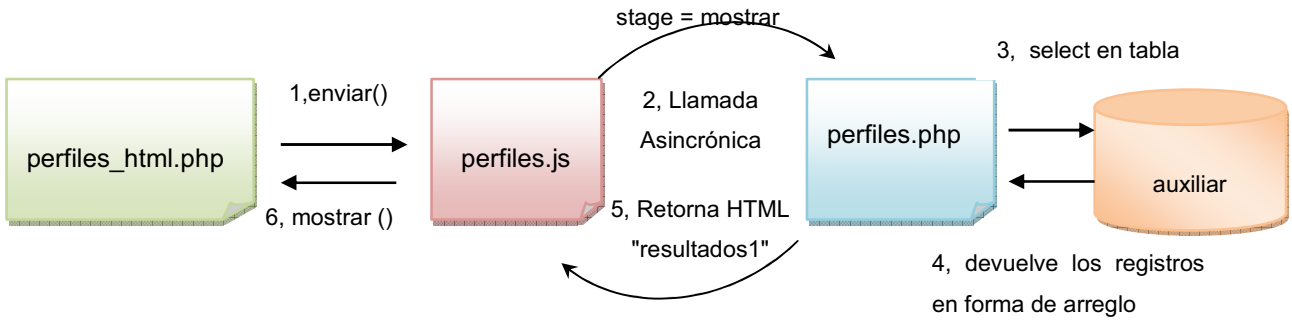


Figura 2.115 Lógica de la acción Seleccionar Perfil

La Figura 2.116 ilustra la lógica que se ejecuta en la aplicación web cuando el

usuario aplica los cambios realizados en la configuración de perfiles a la configuración de FreeRADIUS. Al presionar la opción *Aplicar* de la pestaña Perfiles, la página *perfiles_html.php* llama a la función *enviar()* alojada en el código JavaScript del archivo *perfiles.js.*

Esta función realiza una llamada asincrónica al archivo *perfiles.php* enviando a través del método POST el valor "aplicar" en la variable stage. Al recibir esta variable, el archivo *perfiles.php* se ejecuta la función *aplicar()*, esta función lee los registros de la tabla auxiliar, reconstruye el archivo de configuración de acuerdo a estos registros y lo remplaza por el de FreeRADIUS luego llama a la función por defecto *lectura()*, el caso de uso para la manipulación de perfiles se lo presenta en el Anexo B30.

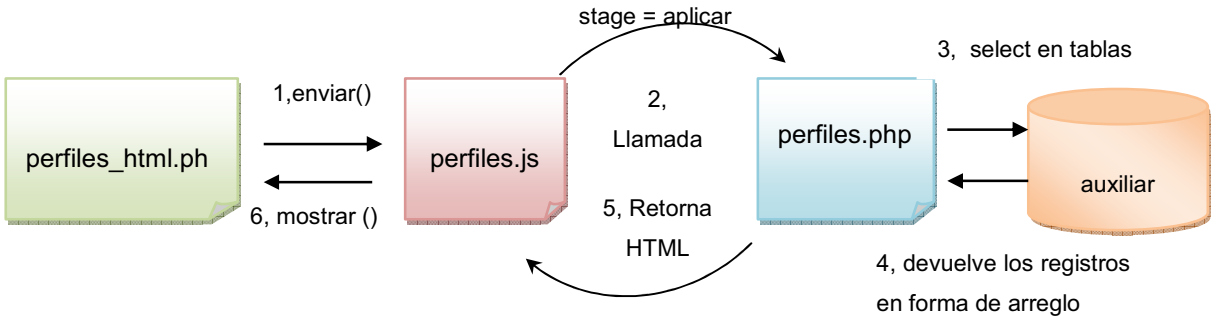


Figura 2.116 Lógica de la acción Aplicar

Al presionar la opción *Nuevo Perfil*, la página *perfiles_html.php* llama a la función *enviar()* alojada en el código JavaScript del archivo *perfiles.js.* Esta función realiza una llamada asincrónica al archivo *perfiles.php* enviando a través del método POST el valor "nuevo" en la variable stage. Si la variable stage es "nuevo"*;* en el archivo *perfiles.php* se ejecuta la función *nuevo()*, esta función muestra un cuadro de selección para el tipo de atributo a ingresar y un cuadro de texto para el valor de éste (ver Figura 2.117).

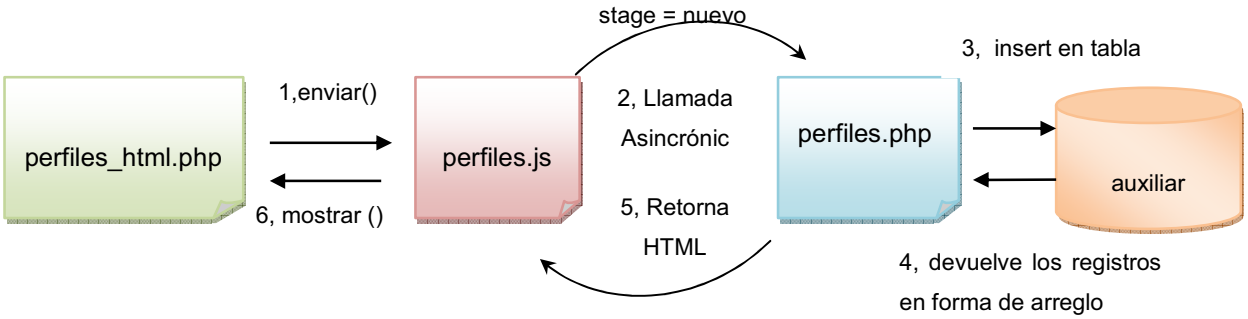


Figura 2.117 Lógica de la acción Nuevo Perfil

La Figura 2.118 ilustra la lógica que se ejecuta en la aplicación Web cuando el usuario presionar la opción *Agregar Atributo*. La página *perfiles_html.php* llama a la función *enviar()* alojada en el código JavaScript del archivo *perfiles.js.* Esta función realiza una llamada asincrónica al archivo *perfiles.php* enviando a través del método POST el valor "atributo" de la variable stage. Al recibir esta variable el archivo *perfiles.php* ejecuta la función *atributo(),* esta función recupera el ID del perfil que se le va a adicionar el atributo, luego muestra un cuadro de selección para el tipo de atributo a ingresar y un cuadro de texto para el valor de éste.

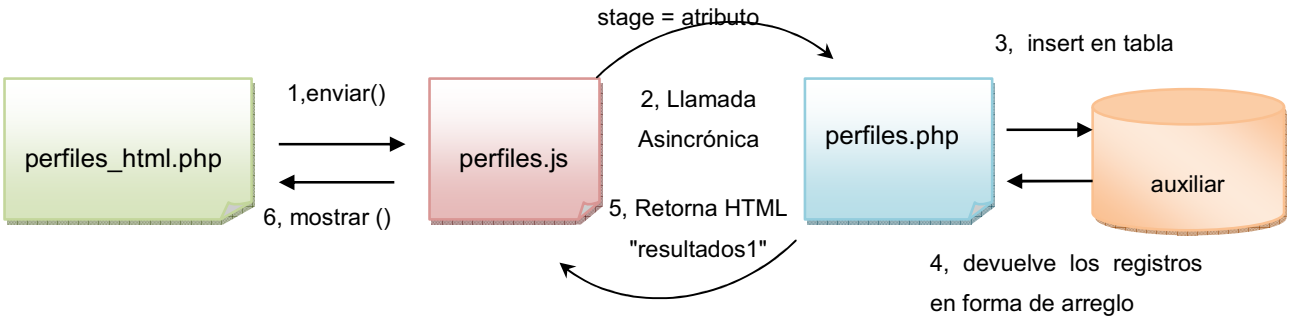


Figura 2.118 Lógica de la acción Agregar Atributo

Al presionar la opción Eliminar, la página perfiles_html.php llama a la función *enviar()* alojada en el código JavaScript del archivo *perfiles.js.* Esta función discrimina si se va a eliminar un perfil o un atributo, luego muestra una ventana de confirmación si la respuesta es afirmativa realiza una llamada asincrónica al archivo *perfiles.php* enviando a través del método POST el valor de una variable llamada

“stage=eliminar”; el código del archivo *perfiles.php* se ejecuta de acuerdo al tipo de acción solicitada. Si la variable stage es: eliminar*;* en el archivo *perfiles.php* se ejecuta la función eliminar*()*, de acuerdo al objeto seleccionado (perfil o atributo) esta función procede eliminar los registros de la tabla *auxiliar* de la base de datos.

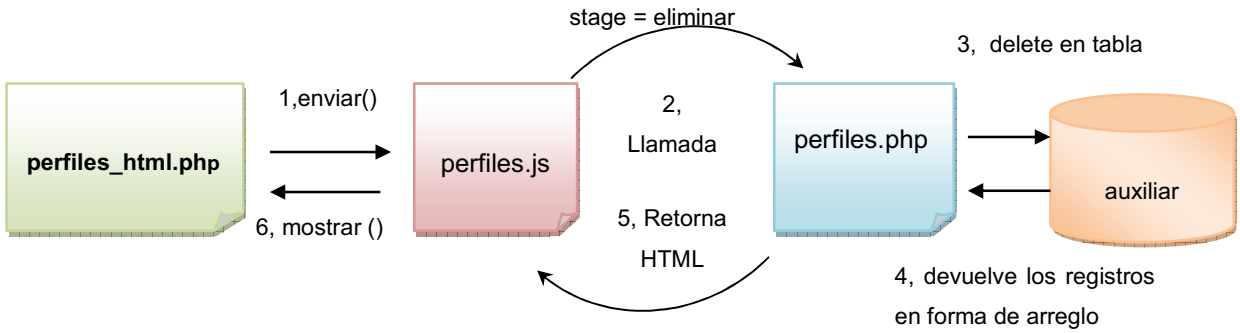


Figura 2.119 Lógica de la acción Eliminar

Al presionar sobre el atributo y luego presionar la opción *modificar*, la página *perfiles_html.php* llama a la función *enviar()* alojada en el código JavaScript del archivo *perfiles.js.* Esta función realiza una llamada asincrónica al archivo *perfiles.php* enviando a través del método POST el valor “modificar” en la variable stage; el código del archivo *perfiles.php* se ejecuta de acuerdo al tipo de acción solicitada.

Si la variable stage es: “modificar”*;* en el archivo *perfiles.php* se ejecuta la función *modificar(),* primero recupera el ID del atributo a modificar, luego realiza la actualización del nuevo valor en la tabla *auxiliar* de la base de datos (ver Figura 2.120).

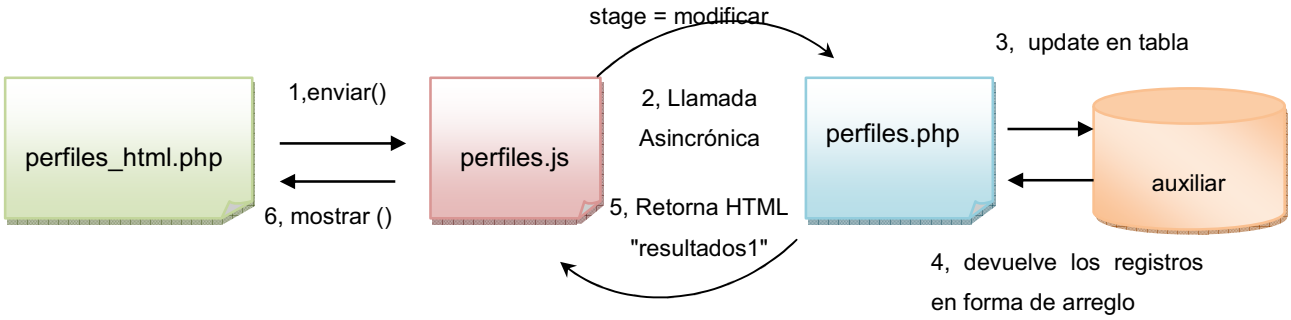


Figura 2.120 Lógica de la acción Modificar

Al cambiar el valor del menú desplegable de la pestaña Diccionario de Atributos, la página *perfiles_html.php* llama a la función *archivo()* alojada en el código JavaScript del archivo *perfiles.js.* Esta función realiza una llamada asincrónica al archivo *perfiles.php* enviando a través del método POST el valor "diccionario" en la variable stage; el código del archivo *perfiles.php* se ejecuta de acuerdo al tipo de acción solicitada. Si la variable stage es: diccionario*;* en el archivo *perfiles.php* se ejecuta la función *diccionario()*, primero recupera el valor del menú desplegable, de acuerdo a este valor recupera el archivo correspondiente y muestra su contenido (ver Figura 2.121) , el caso de uso para esta acción se lo presenta en el Anexo B31.

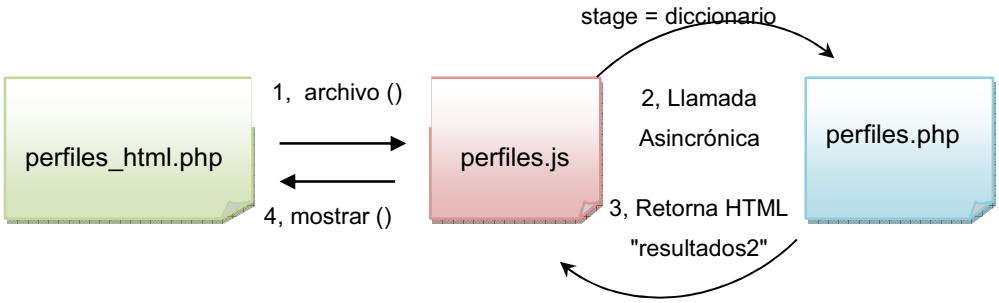


Figura 2.121 Lógica de la acción Modificar

2.5.5.3. Reglas de Autorización

Esta página se encuentra dividida en dos pestañas: Perfiles y Edición de Reglas. La

pestaña Perfiles se encarga de recuperar la configuración de los perfiles asociados a cada una de las reglas de autenticación permitiendo asociar o desasociar más perfiles (ver Figura 2.122). La pestaña Edición de Reglas recupera las condiciones asociadas a cada una de las reglas de autorización permitiendo manipular estas condiciones, adicionalmente permite el ingreso de nuevas reglas (ver Figura 2.123).

Figura 2.122 Página Reglas de Autorización del servidor "SAAAS!" pestaña Perfiles

Figura 2.123 Página Reglas de Autorización del servidor "SAAAS!" pestaña Edición de Reglas.

2.5.5.3.1. Lógica de la sección

Como se observa en la Figura 2.124, la página Reglas de Autorización maneja dos páginas con código php *autorización.php* y *autorizacion_1.php* las cuales manejan de manera independiente las pestañas Perfiles y Edición de Reglas respectivamente, de igual forma recuperan y manipulan los datos de las tablas temporales correspondientes en la base de datos, éstas son: *auxiliar3* para los perfiles y *auxiliar4* para las condiciones.

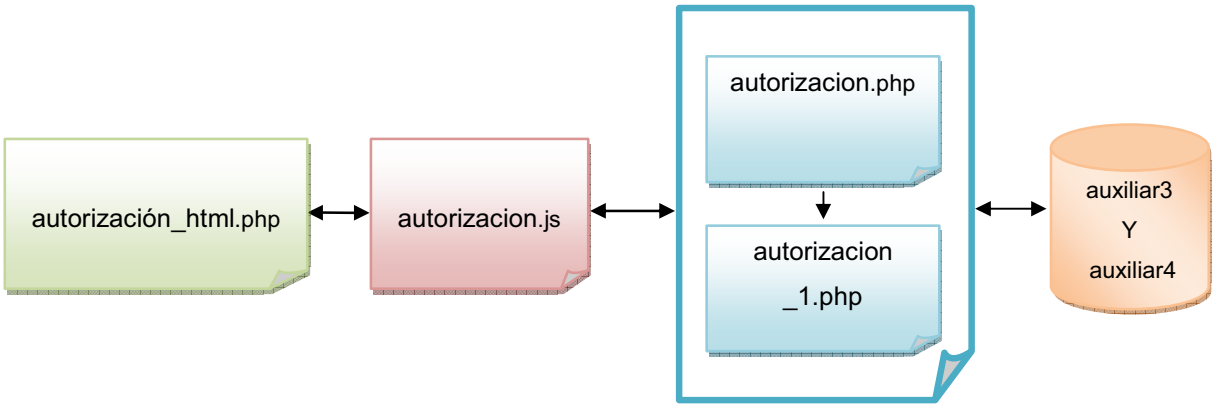


Figura 2.124 Esquema general de la página Reglas de Autorización.

La Figura 2.125 ilustra la lógica que se ejecuta en la aplicación Web cuando la página Reglas de Autorización se carga por primera vez. La página *autorizacion_html.php* llama a la función *inicio()* alojada en el código JavaScript del archivo *autorizacion.js.* Esta función realiza una llamada asincrónica al archivo *autorizacion.php* y al archivo *autorización_1.php* enviando a través del método POST el valor "null" de la variable stage; entonces el código del archivo *autorizacion.php y autorización_1.php* se ejecuta de acuerdo al tipo de acción solicitada.

Si la variable stage es: "null"*;* en el archivo *autorizacion.php y autorización_1.php* se ejecuta la función por defecto *lectura()*, esta función lee el archivo de configuración de FreeRADIUS */etc/freeradius/sites-enabled/post-auth* recuperando las reglas, las condiciones y los perfiles asociados. Estos datos son almacenados en las tablas

temporales creadas en ese momento en la tabla *auxiliar3* se almacena los repositorios y en la tabla *auxiliar4* las condiciones, una vez realizado este proceso se procede a leer las tablas y a generar el código HTML para visualizar los datos en forma de tabla esto lo realiza la función *mostrar().*

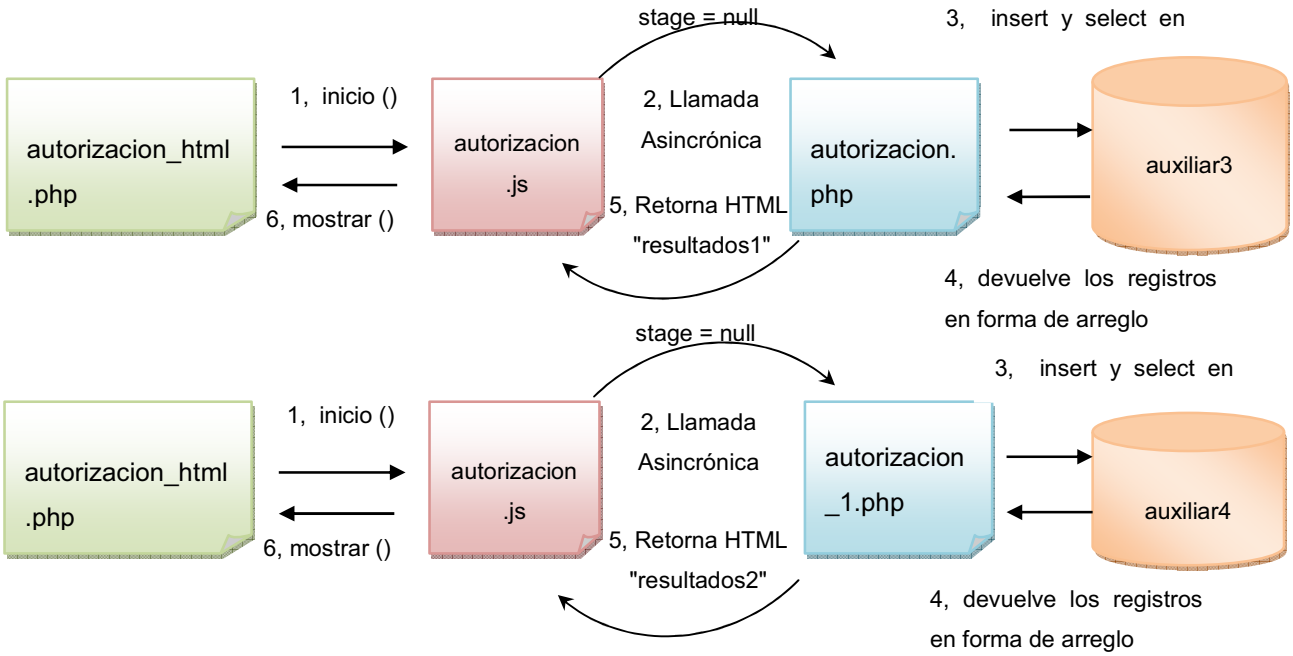


Figura 2.125 Lógica de la acción por defecto

Al presionar sobre una de las reglas en la pestaña Perfiles, la página *autorizacion_html.php* llama a la función *enviar();* si lo realiza desde la pestaña Edición de Reglas llama a la función *enviar1(),* alojadas en el código JavaScript del archivo *autorizacion.js.* Esta función realiza una llamada asincrónica al archivo *autorizacion.php o autorizacion_1.php* enviando a través del método POST el valor "mostrar" en la variable llamada stage; el código del archivo *autorizacion.php* o *autorizacion_1.php* se ejecuta de acuerdo al tipo de acción solicitada.

Si la variable stage es "mostrar"*;* en el archivo *autorizacion.php* se ejecuta la función *mostrar()*, esta función lee los registros de la tabla *auxiliar3* y de acuerdo a la regla que fue presionada se genera el código HTML para visualizar los datos asociados a

esa regla en forma de tabla. Si la variable stage es “mostrar” en el archivo *autorizacion_1.php* se ejecuta la función *mostrar()*, esta función lee los registros de la tabla *auxiliar4* y de acuerdo a la regla que fue presionada se genera el código HTML para visualizar los datos asociados a esa regla en forma de tabla (ver Figura 2.126).

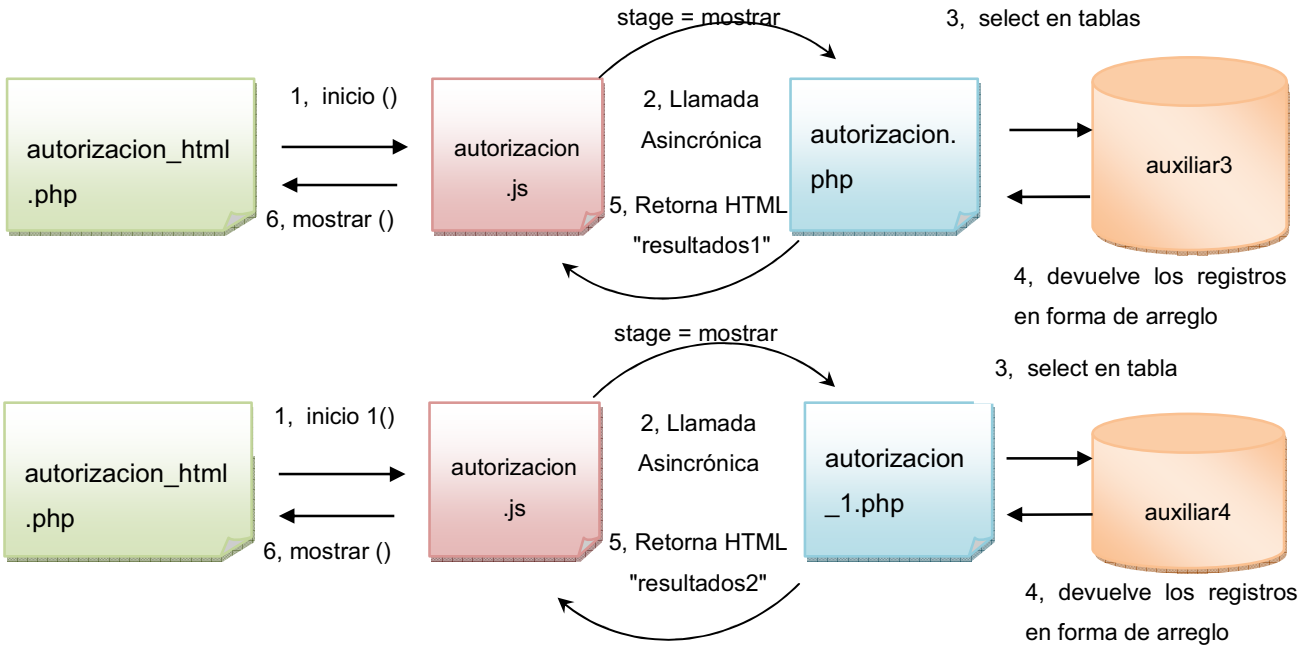


Figura 2.126 Lógica de la acción Seleccionar Regla

La Figura 2.127 ilustra las acciones que se llevan a cabo en la aplicación cuando el usuario aplica los cambios en la configuración de Perfiles. Al presionar sobre la opción *Aplicar* de la pestaña Perfiles, la página *autorizacion_html.php* llama a la función *enviar()* alojada en el código JavaScript del archivo *autorizacion.js.* Esta función realiza una llamada asincrónica al archivo *autorizacion.php* enviando a través del método POST el valor “aplicar” en la variable stage. Al recibir esta variable, el archivo *autorizacion.php* ejecuta la función *aplicar()*, esta función lee los registros de la tabla *auxiliar3* y *auxiliar4,* en base a estos registros reconstruye el archivo de configuración y lo remplaza por el de FreeRADIUS, luego llama a la función por defecto *lectura()*, el caso de uso para esta acción se lo presenta en el Anexo B32.

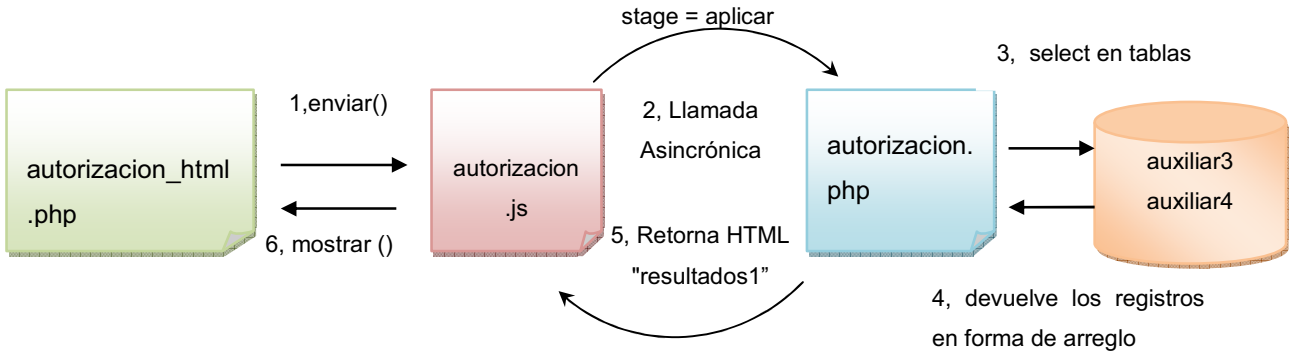


Figura 2.127 Lógica de la acción Aplicar

Al presionar la opción *Nueva Regla*, la página *autorizacion_html.php* llama a la función *enviar1()* alojada en el código JavaScript del archivo *autorizacion.js.* Esta función realiza una llamada asincrónica al archivo *autorizacion_1.php* enviando a través del método POST el valor “nuevo” en la variable stage. Al recibir este valor el archivo *autorizacion_1.php* ejecuta la función *nuevo()*, esta función muestra un cuadro de selección para el tipo de condición a ingresar y un cuadro de texto para el valor de ésta (ver Figura 2.128) , el caso de uso para la manipulación de Reglas de Autorización se lo presenta en el Anexo B33.

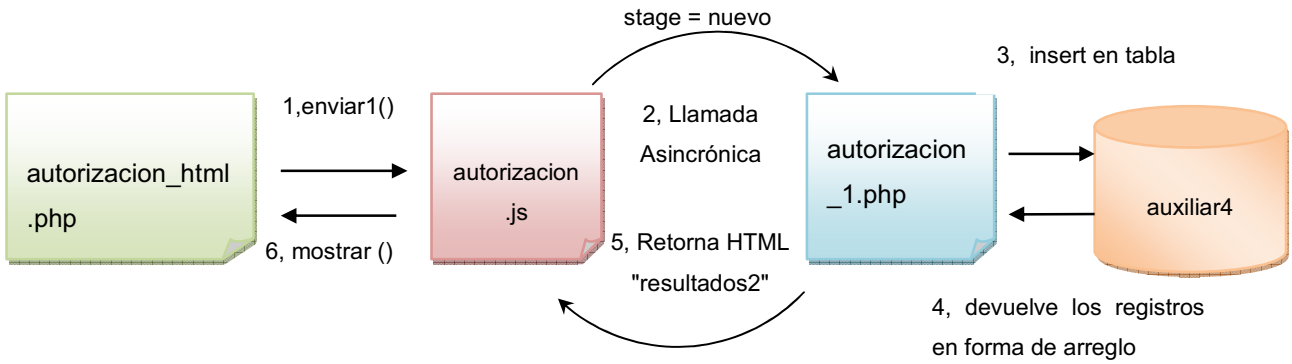


Figura 2.128 Lógica de la acción Nueva Regla

La Figura 2.129 ilustra la lógica que se ejecuta en la aplicación Web al presionar la opción *Agregar Condición*. La página *autorizacion_html.php* llama a la función *enviar1()* alojada en el código JavaScript del archivo *autorizacion.js.* Esta función realiza una llamada asincrónica al archivo *autorizacion_1.php* enviando a través del método POST el valor "condicion" en la variable stage. Al recibir esta variable el archivo *autorización_1.php* ejecuta la función condicion*()*, esta función recupera el ID de la regla a la que se le va a adicionar la condición, luego muestra un cuadro de selección para el tipo de condición a ingresar y un cuadro de texto para el valor de ésta.

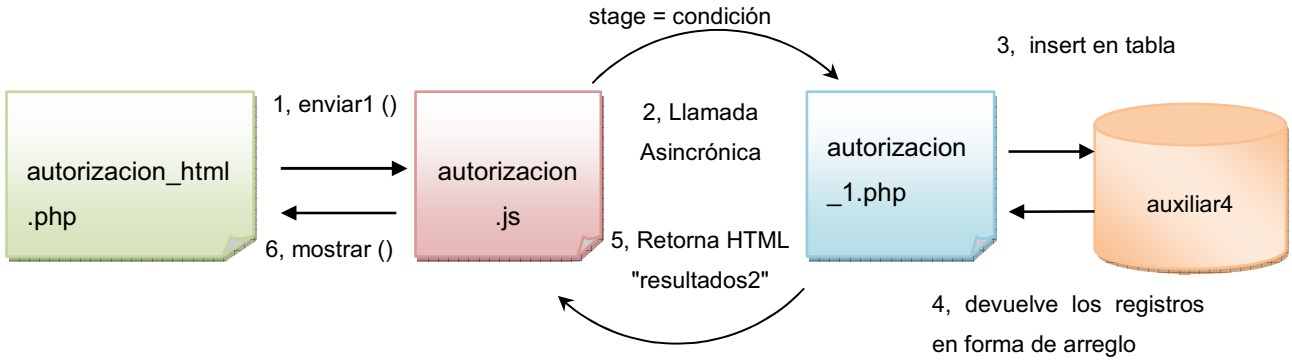


Figura 2.129 Lógica de la acción Agregar Condición

Al presionar la opción *Eliminar*, la página *autorizacion_html.php* llama a la función *enviar1()* alojada en el código JavaScript del archivo *autorizacion.js.* Esta función discrimina si se va a eliminar una regla o una condición, luego muestra una ventana de confirmación si la respuesta es afirmativa realiza una llamada asincrónica al archivo *autorizacion_1.php* enviando a través del método POST el valor "eliminar" en la variable stage. Al recibir esta variable el archivo *autorizacion_1.php* ejecuta la función *condicion(),* de acuerdo al objeto seleccionado (regla o condición) esta función procede a eliminar los registros de la tabla *auxiliar4* de la base de datos (ver Figura 2.130).

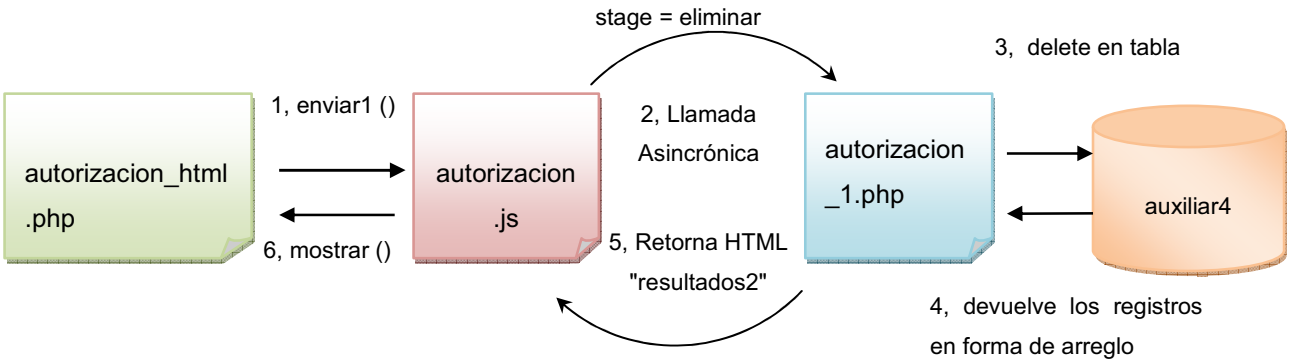


Figura 2.130 Lógica de la acción Eliminar

Al presionar sobre la condición y luego presionar la opción *modificar*, la página *autorizacion_html.php* llama a la función *enviar1()* alojada en el código JavaScript del archivo *autorizacion.js.* Esta función realiza una llamada asincrónica al archivo *autorizacion_1.php* enviando a través del método POST el valor "modificar" en la variable stage; el código del archivo *autorizacion_1.php* se ejecuta de acuerdo al tipo de acción solicitada.

Si la variable stage es "modificar"*;* en el archivo *autorizacion_1.php* se ejecuta la función *modificar(),* primero recupera el ID de la condición a modificar, luego realiza la actualización del nuevo valor en la tabla *auxiliar4* de la base de datos (ver Figura 2.131).

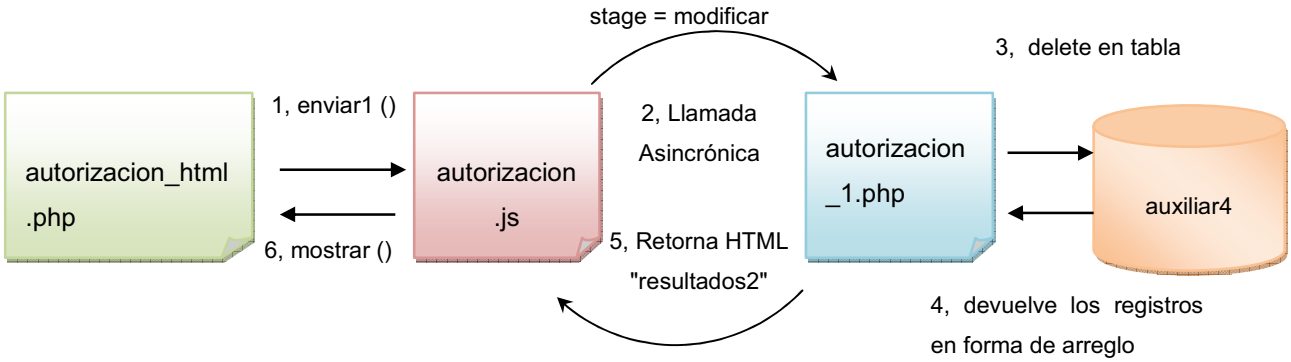


Figura 2.131 Lógica de la acción Modificar

2.5.6. VISOR DE EVENTOS

En esta sección se visualizan reportes del funcionamiento del servidor y de los procesos de autenticación y autorización de los usuarios, también se permite la creación de copias de seguridad para la recuperación de fallas mediante la restauración del sistema.

2.5.6.1. Monitoreo

Como se observa en la Figura 2.132, en esta página el usuario del sistema puede ver los mensajes de Error e Información que genera el servicio de FreeRADIUS.

Figura 2.132 Página Eventos del Sistema de la Sección Visor de Eventos de "SAAAS"

2.5.6.1.1. Lógica de la Sección

Como se observa en la Figura 2.133, al cargarse por primera vez, la página *monitoreo_html.php* llama a la función *Datos()* alojada en el código JavaScript del archivo *monitoreo.js.* Esta función obtiene los datos del rango de fechas especificado en las cajas de texto que se muestran en la interfaz Web y realiza una llamada asincrónica al archivo *monitoreo.php* enviando a través del método POST la variable acción con el valor “logSistema” y el rango de fechas.

En el archivo *monitoreo.php* se ejecutan las funciones necesarias para recuperar la información del reporte Eventos del Sistema realizando una consulta a la tabla *accounting* de la base de datos local del servidor. Esta información se entrega en formato XML al código JavaScript que inició la llamada asincrónica. Al recibir la información solicitada el código JavaScript crea objetos HTML para mostrar el reporte Eventos del Sistema, el caso de uso para esta acción se lo presenta en el Anexo B34.

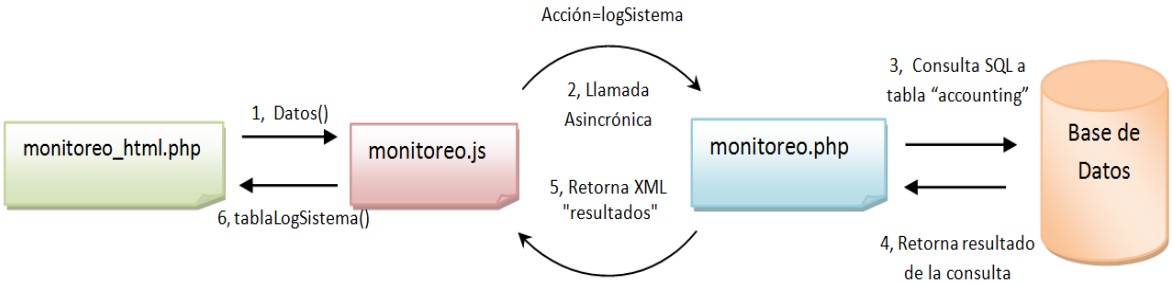


Figura 2.133 Lógica de la Sección Eventos del Sistema

Por defecto, FreeRADIUS almacena sus logs en el archivo */var/log/freeradius/radius.log.* Para optimizar el visor de eventos, "SAAAS!" utiliza una tarea programada con el demonio *crond* para ejecutar el script *cron.php* con el intérprete de comandos CURL cada cinco minutos; este script permite leer el archivo de logs y almacenar su contenido en la tabla *accounting.*

2.5.6.2. Resumen

Como se observa en la Figura 2.134, esta página muestra los reportes del proceso de autenticación y autorización de los usuarios. El usuario visualiza un resumen del registro AAA en gráficos de barra que muestran las solicitudes de autenticación realizadas al servidor por cliente NAS y por usuario y el número de sesiones de clientes activas en el servidor.

Figura 2.134 Página Resumen del Registro AAA de la sección Visor de Eventos de "SAAAS"

2.5.6.2.1. Lógica de la Sección

La página *resumen_html.php* llama a la función *inicio()* del archivo *resumen.js.* Esta función crea tres llamadas asincrónicas para obtener del servidor los datos necesarios para generar los gráficos de barra que se muestran en *resumen_html.php.*

Como se observa en la Figura 2.135, la primera llamada asincrónica envía a través del método POST la variable acción con el valor “graf1”, al recibir la información en la página *resumen.php* se realiza una consulta a la tabla *radreq* de la base de datos para conocer la dirección IP de los clientes Radius (NAS) que han recibido el mayor número de solicitudes de autenticación en las últimas veinticuatro horas. La información generada se entrega al cliente en formato XML y se guarda en el archivo *data1.xml* para ser interpretado por el código *jscharts.js*, encargado de generar el gráfico de barras en la página Web.

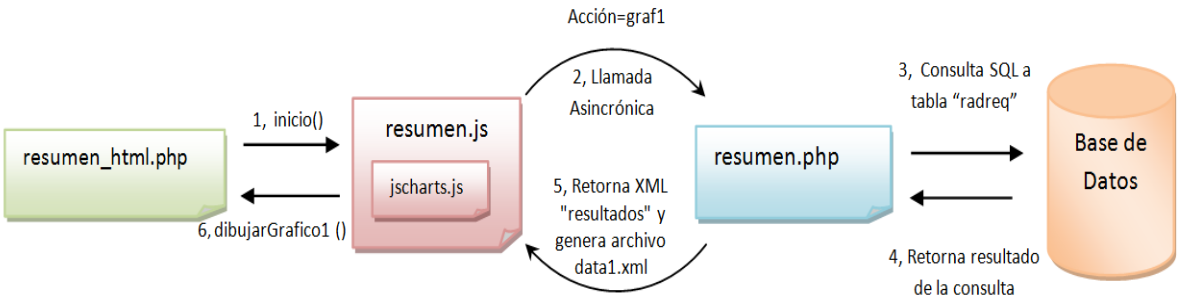


Figura 2.135 Lógica para generar gráfico Solicitudes de Autenticación vs. NAS de la página Resumen del Registro AAA de "SAAAS!"

Como se observa en la Figura 2.135, la segunda llamada asincrónica envía a través del método POST la variable acción con el valor "graf2", al recibir la información en la página *resumen.php* se realiza una consulta a la tabla *radacct* de la base de datos para conocer la dirección IP de los clientes RADIUS (NAS) que tienen sesiones abiertas al momento que se realiza la consulta. La información generada se entrega al cliente en formato XML y se guarda en el archivo *data2.xml* para ser interpretado por el código *jscharts.js* que genera el gráfico de barras en la página Web, el caso de uso para esta acción se lo presenta en el Anexo B35.

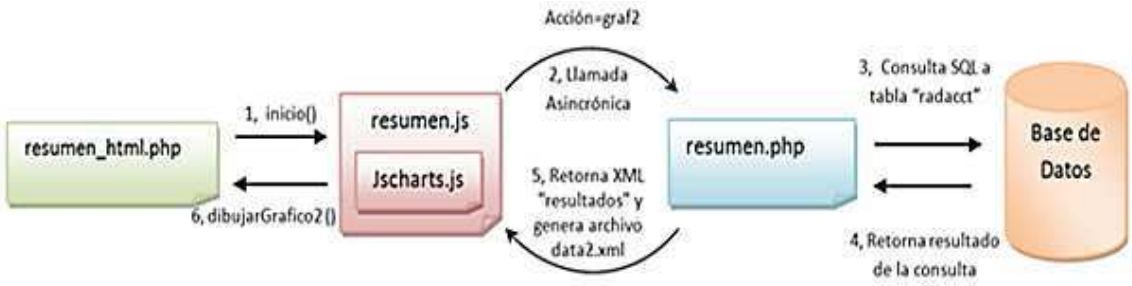


Figura 2.136 Lógica para generar gráfico Sesiones Activas vs. NAS de la página Resumen del Registro AAA de "SAAAS!"

Como se observa en la Figura 2.137, la tercera llamada asincrónica envía a través

del método POST la variable acción con el valor “graf3”, al recibir la información en la página *resumen.php* se realiza una consulta a la tabla *radreq* de la base de datos para conocer el nombre de los usuarios que han realizado mayor número de solicitudes de autenticación en las últimas veinticuatro horas. La información generada se entrega al cliente en formato XML y se guarda en el archivo *data3.xml* para ser interpretado por el código *jscharts.js* que genera el gráfico de barras en la página Web, el caso de uso para esta acción se lo presenta en el Anexo B35.

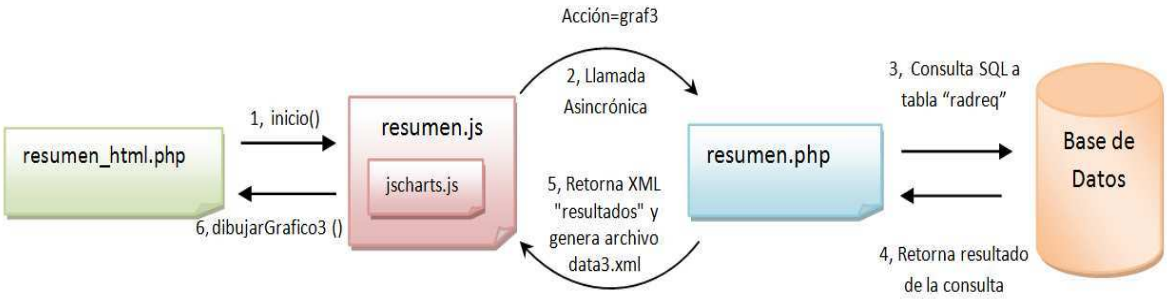


Figura 2.137 Lógica para generar gráfico Solicitudes de Autenticación vs. Usuarios de la página Resumen del Registro AAA de "SAAAS!"

2.5.6.3. Registro

La página Registro AAA (ver Figura 2.138) permite generar reportes de las solicitudes de autenticación realizadas al servidor, las sesiones exitosas según el tiempo de duración de las mismas, según la cantidad de información transmitida y las sesiones que se encuentran activas. El usuario también puede descargar los reportes generados con formato “.xls” para abrirlos en Excel.

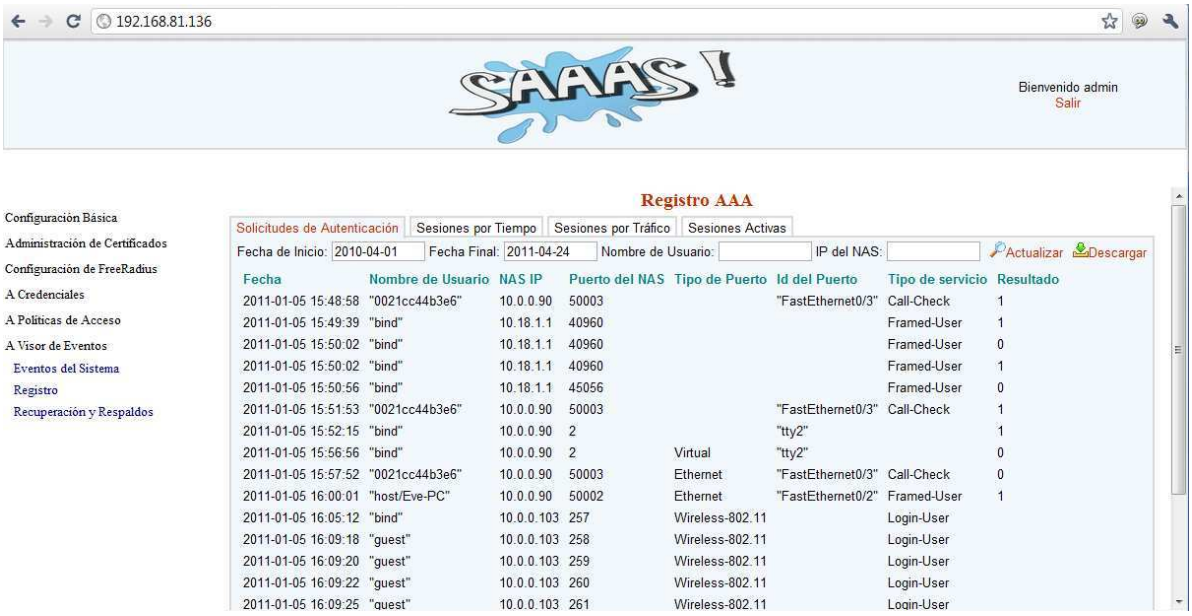
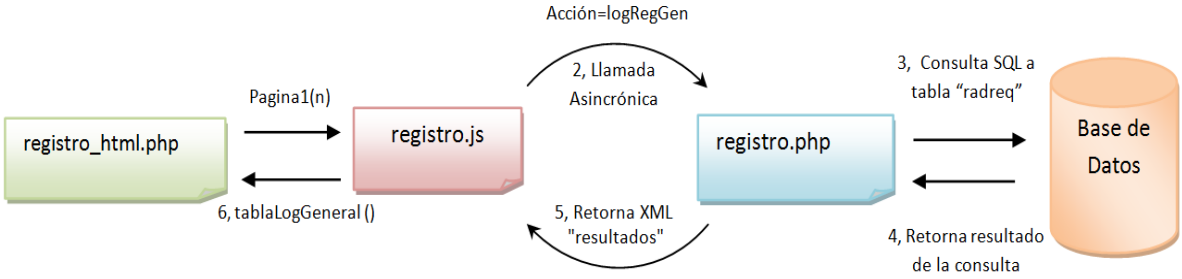


Figura 2.138 Página Registro AAA de la sección Visor de Eventos de "SAAAS"

2.5.6.3.1. Lógica de la Sección

Cada pestaña de la página Registro AAA muestra el resultado de una consulta *SQL* a las tablas *radreq* y *radacct* de la base de datos del servidor. Esta consulta se realiza de manera asincrónica a través del código JavaScript de la página *registro.js* que utiliza el método POST para enviar una solicitud a la página *registro.php* la cual retorna el resultado de la consulta a la base de datos en formato XML; estos datos son procesados en el lado del cliente y el código JavaScript lo muestra en la página Web. La lógica aquí descrita se ilustra en la Figura 2.139.



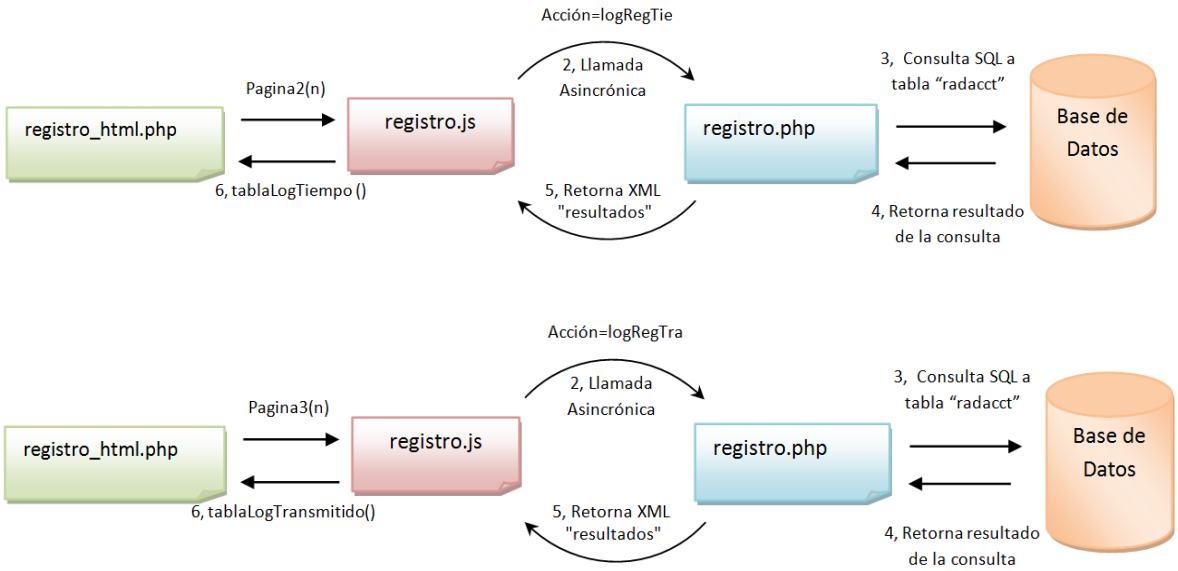


Figura 2.139 Lógica para la generación de reportes en la página Registro AAA de "SAAAS!"

La página Registro AAA también permite descargar los reportes que se muestran en la interfaz Web, la consulta a la base de datos se vuelve a realizar mediante una llamada asincrónica pero en lugar de retornar el resultado al código JavaScript para que lo procese, el archivo *descargar.php* genera un archivo de extensión *.xls* con la tabla XML del resultado (ver Figura 2.140), el caso de uso para esta acción se lo presenta en el Anexo B36.

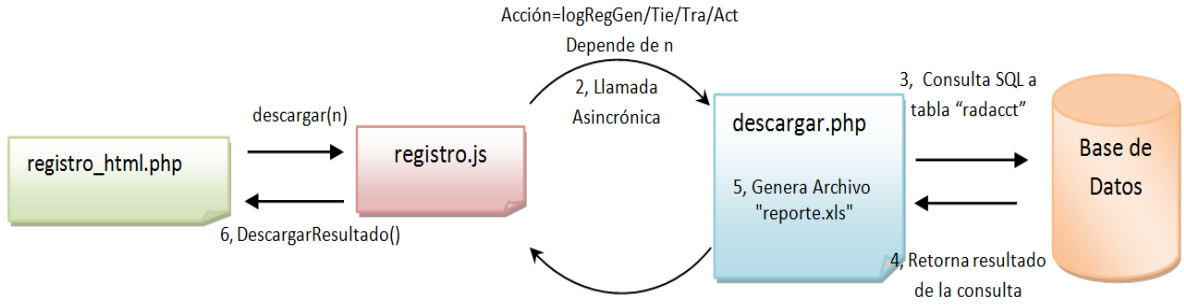


Figura 2.140 Lógica de la acción Descargar Reporte de la página Registro AAA de "SAAAS!"

El reporte Sesiones Activas muestra la lista de registros de la tabla *radacct* que tienen nulo el campo en el que se define la hora en que el usuario finalizó su sesión en la red. Sin embargo, es posible que debido a fallas con el NAS a través del cual el usuario ingresó a la red una sesión termine intempestivamente y el campo de hora de finalización no se llene. Para que el administrador pueda declarar estas sesiones como terminadas, “SAAAS!” permite establecer el campo de hora de finalización de la sesión con la hora actual del servidor al presionar el enlace *Cerrar Sesión* junto a cada registro. La lógica que se ejecuta en la aplicación Web cuando esto sucede se ilustra en la Figura 2.141, el caso de uso para esta acción se lo presenta en el Anexo B37.

Figura 2.141 Lógica de la función Cerrar Sesión del Reporte Sesiones Abiertas de la página "Registro AAA"

2.5.6.4. Recuperación y Respaldos

Como se observa en la Figura 2.142, esta página permite crear y administrar los respaldos del sistema. A través de esta interfaz se pueden crear copias de los archivos de Configuración del Servidor, eliminarlas, o utilizarlas para reemplazar la configuración actual.

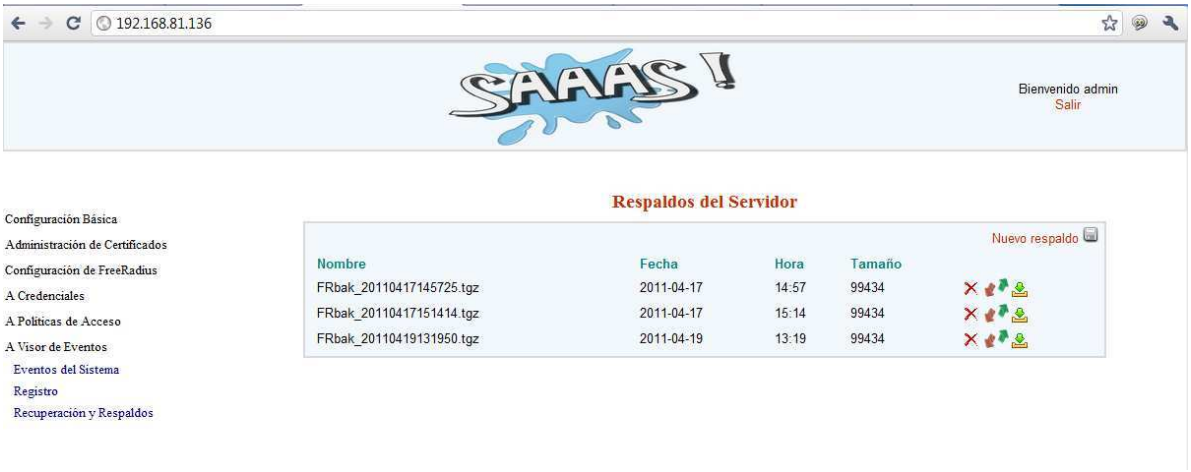


Figura 2.142 Página Recuperación y Respaldos de la Sección Visor de Eventos de "SAAAS"

2.5.6.4.1. Lógica de la Sección

Al abrirse por primera vez, el archivo *recuperación_html.php* llama a la función *datos()* del archivo *recuperación.js.* Mediante una llamada asincrónica usando el método POST, la función envía la variable acción con el valor “lista” a la página *recuperación.php,* la cual procesa la solicitud y devuelve la lista de archivos de respaldo existentes en el sistema en formato XML, el cual es procesado por el código JavaScript para mostrarlo en la interfaz Web (ver Figura 2.143).

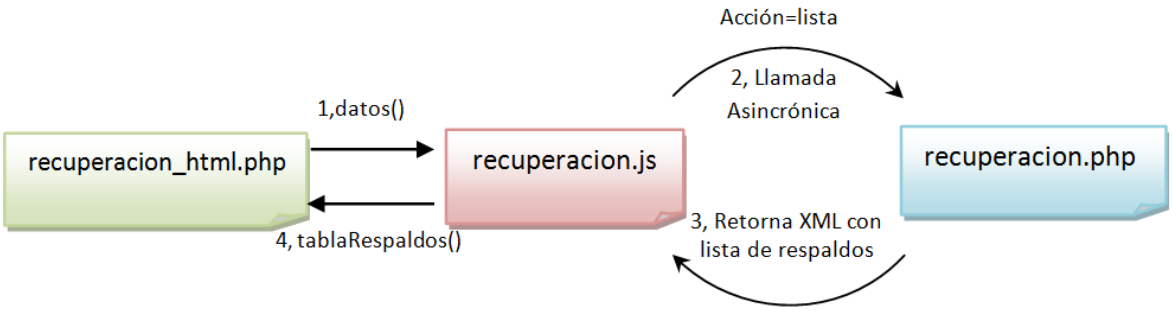


Figura 2.143 Lógica de la página Recuperación y Respaldos de "SAAAS!"

En la página Recuperación y Respaldos se puede también crear y eliminar los

archivos de respaldo del sistema. Un archivo de respaldo es una copia comprimida del directorio */etc/freeradius.* Para estas acciones se lleva a cabo una lógica similar en todos los casos, variando la función que se ejecuta en el archivo *recuperación.php* según el valor de la variable acción que envía el archivo *recuperación.js* (ver Figura 2.144), el caso de uso para esta acción se lo presenta en el Anexo B38.

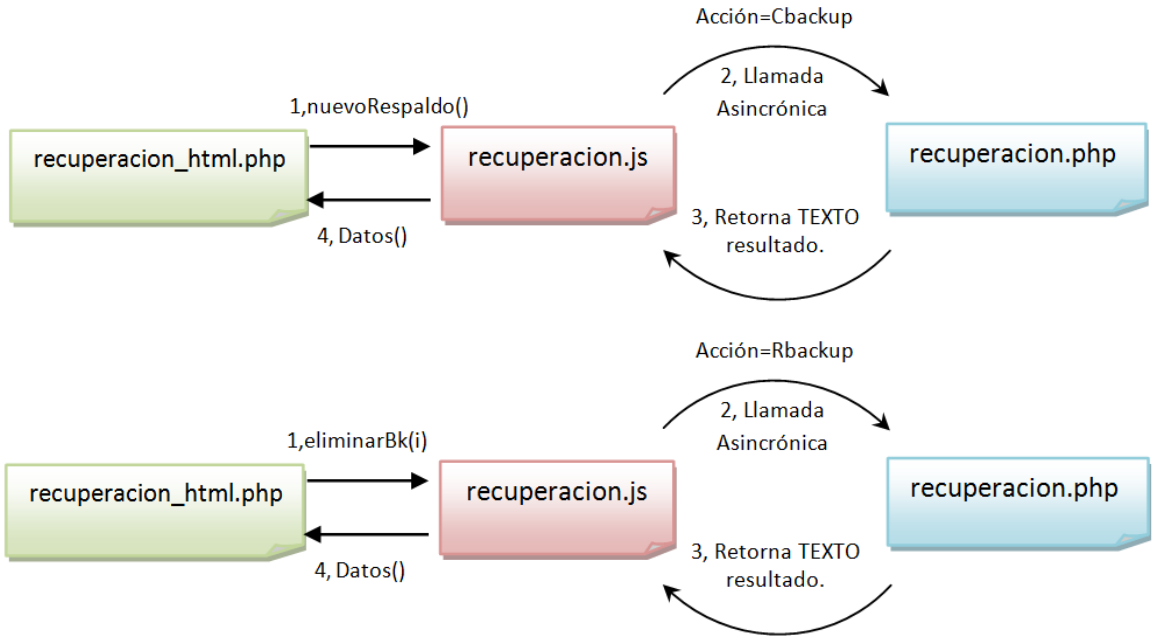


Figura 2.144 Lógica de las funciones Crear/Eliminar Respaldos de la página Recuperación y Respaldos de "SAAAS!"

Para realizar la recuperación del sistema, desde la interfaz Web el usuario llama a la función *usarBk(i),* la cual recupera el nombre del archivo de respaldo que se quiere usar y realiza una serie de llamadas asincrónicas al archivo *recuperación.php* para detener el servicio de FreeRADIUS, descomprimir el archivo de respaldo, reemplazar la configuración actual del sistema por el respaldo y reiniciar el servicio. Cada una de las llamadas asincrónicas retorna en formato de texto el resultado de las operaciones realizadas e inicia a la siguiente (ver Figura 2.145).

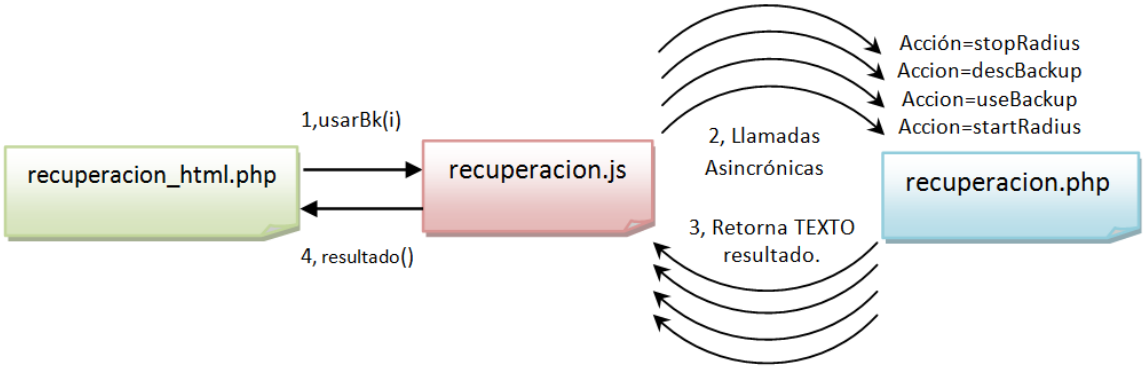


Figura 2.145 Lógica de la función Restaurar Sistema de la página Recuperación y Respaldos de "SAAAS!"

CAPÍTULO 3. IMPLEMENTACIÓN DEL PROTOTIPO DE META DISTRIBUCIÓN, PRUEBAS Y RESULTADOS.

3.1. EL PROTOTIPO DE META DISTRIBUCIÓN "SAAAS!"

3.1.1. DESCRIPCIÓN

El Servidor AAA Simplificado "SAAAS!", es un prototipo de Meta Distribución basada en Debian Squeeze, cuyo objetivo es facilitar la implementación de servicios AAA en entornos de red a través de la consola de administración de FreeRADIUS del mismo nombre. El prototipo ha sido optimizado para la implementación del servicio FreeRADIUS y el correcto funcionamiento de la aplicación Web "SAAAS!" descrita en el capítulo 2 de este documento.

3.1.1.1. Proyectos Similares

3.1.1.1.1. PHPRadmin

PhpRADmin[47] es una herramienta de software Web que permite la configuración y administración de un servidor FreeRADIUS. PhpRADmin sigue el principio de meta distribución basada en Fedora[48] y se distribuye como un LiveCD no instalable.

Este software se distribuye bajo los términos de la licencia GNU/GPL y sigue aún en desarrollo. Su código fuente utiliza las herramientas: Dialup Admin[49] y phpSysInfo, incluye módulos para administración de certificados y permite la modificación directa de los archivos de configuración de FreeRADIUS como se muestra en la Figura 3.1.

[47] Página oficial del proyecto: http://www.phpradmin.org/
[48] Distribución Linux.
[49] DialUp Admin es una herramienta de administración Web escrita en PHP que se incluye en el paquete de FreeRADIUS

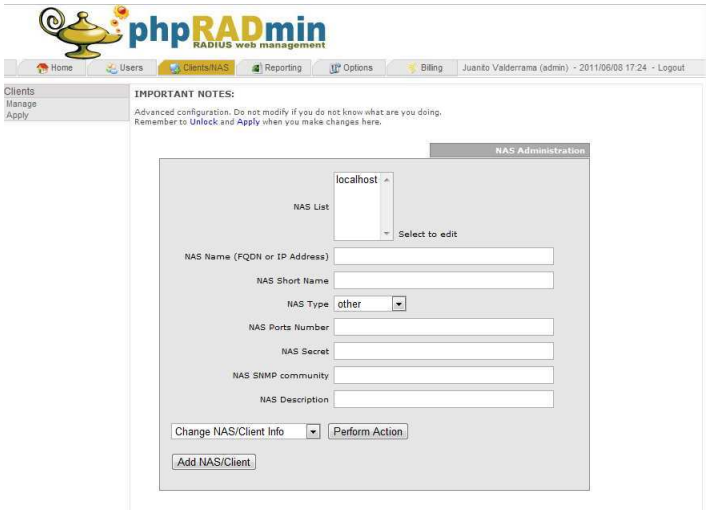


Figura 3.1 Interfaz gráfica de PhpRADmin

3.1.1.1.2. CISCO Secure ACS

Cisco Secure ACS (Access Control System), es la solución de CISCO para implementación de servicios basados en identidad en redes LAN. La plataforma funciona como PDP en la arquitectura AAA, provee el servicio RADIUS y TACACS+ e incluye una herramienta de administración Web muy completa tal como se puede apreciar en la Figura 3.2. En la tabla 3.1 se muestra una comparación entre los proyectos antes mencionados y "SAAAS!".

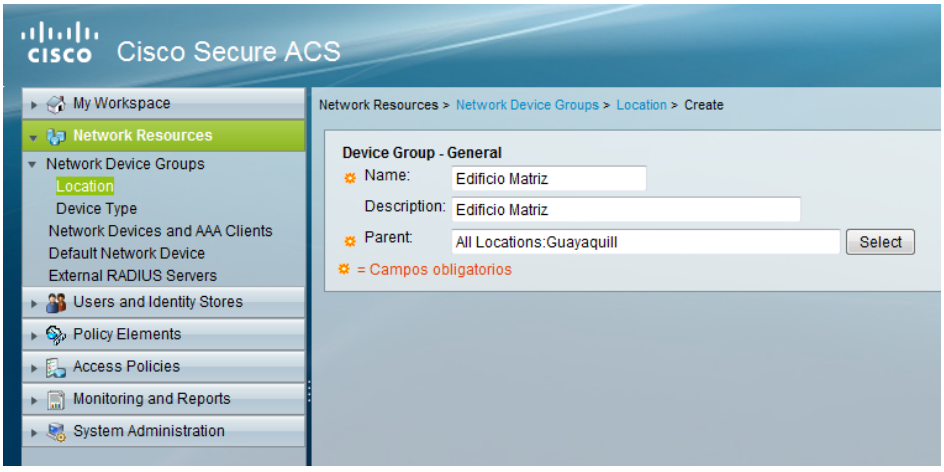


Figura 3.2 Interfaz gráfica de Cisco Secure ACS 5.x

Característica	PhpRADmin	Cisco Secure ACS	SAAAS!
Tipo de Solución	Sistema Operativo (Basado en Fedora).	Sistema Operativo (Basado en Centos)	Sistema Operativo (Basado en Debian).
Licenciamiento	Distribuido bajo Licencia GNU/GPL.	Licencia Restringida/Propietario	Distribuido bajo Licencia GNU/GPL.
Opciones de Plataforma	Cualquiera	Appliance Propietario o VMWare.	Cualquiera.
Protocolos AAA	RADIUS.	RADIUS y TACACS+	RADIUS.
Opciones para manejo de Credenciales	Base de Datos MySQL local y externa. Base de Datos PostgreSQL local y externa. Directorio LDAP.	Repositorio Local. (Base de Datos) Directorio LDAP. Active Directory de Windows.	Repositorio Local (Base de Datos MySQL). Directorio LDAP. Active Directory de Windows.
Autenticación de usuarios en la red	Con RADIUS.	Con RADIUS.	Con RADIUS.
Autenticación para la administración de dispositivos de red	Con RADIUS.	Con TACACS+.	Con RADIUS.
Políticas de Acceso	No provee interfaz de administración amigable para la configuración de políticas de acceso como asignación dinámica de VLANs.	Provee una interfaz amigable para la configuración de políticas de acceso como: asignación dinámica de VLANs, restricción por horarios. MAC Address Bypass.	Provee una interfaz amigable para la configuración de políticas de acceso como: Asignación dinámica de VLANs, MAC Address Bypass.
Certificados digitales	Soporta la creación y administración de una infraestructura de claves públicas para la creación y administración de certificados digitales.	Soporta la creación de un certificado autofirmado para ser utilizado localmente en los procesos de autenticación.	Soporta la creación y administración de una infraestructura de claves públicas para la creación y administración de certificados digitales.
Configuración de clientes NAS	Permite la configuración de clientes NAS en el archivo de configuración de FreeRADIUS.	Permite la configuración de clientes NAS y la almacena en una base de datos.	Permite la configuración de clientes NAS y la almacena en una base de datos.
Procesos de Autorización	Permite la configuración de políticas de acceso basada en usuarios.	Permite la configuración de políticas de accesos basada en reglas y condiciones.	Permite la configuración de políticas de accesos basada en reglas y condiciones.

Tabla 3.1 Comparación del prototipo de "Meta Distribución SAAAS!" con soluciones similares

3.1.2. REQUISITOS DE HARDWARE

Debian no impone requisitos de hardware más allá de los que establecen el núcleo Linux. En cualquier caso, cualquier arquitectura o plataforma para la que se haya adaptado el núcleo Linux puede ejecutar Debian y por tal razón el servidor "SAAAS!".

Debian recomienda como mínimo un procesador Pentium 4, de 1 GHz o equivalente, de acuerdo al tipo de instalación se definirá la capacidad de RAM y disco duro necesarios como se muestra en la tabla 3.2.

Tipo de instalación	RAM (mínimo)	RAM (recomendado)	Disco duro
Sin escritorio	64 Megabytes	256 Megabytes	1 Gigabyte
Con escritorio	128 Megabytes	512 Megabytes	5 Gigabytes

Tabla 3.2 Requerimientos de Hardware de Debian GNU/Linux

Es prácticamente imposible dar requisitos generales de memoria y espacio en disco para instalaciones de servidores ya que éstos dependerán en gran medida del uso del mismo. En base a los servicios que se corren en el servidor "SAAAS!" se recomienda elevar los requerimientos mínimos de la distribución Debian por lo menos en un 50%.

3.1.3. CREACIÓN DEL PROTOTIPO DE META DISTRIBUCIÓN "SAAAS!"

3.1.3.1. Personalización de la Distribución Base

El primer paso para la creación de la Meta Distribución es la instalación del Sistema Operativo base, en este caso Debian Squeeze.

Ya que el objetivo de "SAAAS!" es funcionar como un servidor AAA administrado en su totalidad a través de un interfaz Web, se omite el sistema de ventanas X y

GNOME en la instalación de la distribución base.

3.1.3.1.1. Paquetes Incluidos

"SAAAS!" necesita de los paquetes OpenSSH, OpenSSL, MySQL, cURL, Cron, Php5, Apache y Samba para funcionar correctamente, por lo que se utiliza la herramienta de gestión de paquetes *apt-get* ingresando las líneas mostradas en el Espacio de Comandos 3.1.

```
aaa:~# apt-get update
aaa:~#apt-get install ssh
aaa:~#apt-get install openssl
aaa:~#apt-get install  apache2
aaa:~#apt-get install  samba
aaa:~#apt-get install libapache2-mod-php5 php5-cli php5-common php5-cgi php5-ldap
aaa:~#apt-get install mysql-client mysql-common mysql-server php5-mysql
```

Espacio de Comandos 3.1 Instalación de paquetes necesarios para el funcionamiento de "SAAAS!"

3.1.3.1.2. Configuración de Apache

La configuración por defecto del servidor Web Apache es suficiente para levantar el servicio y visualizar las páginas alojadas en la carpeta */var/www/*, sin embargo se realizaron cambios para permitir el acceso a la aplicación Web "SAAAS!" a través de HTTPS.

Se empleó "SAAAS!" para generar el certificado autofirmado usado para la transmisión de datos usando HTTPS.

Para que el servidor utilice el certificado generado es necesario modificar los archivos */etc/apache2/sites-enabled/ssl* y */etc/apache2/mods-enabled/ssl.conf*.

En el archivo */etc/apache2/sites-enabled/ssl* se incluyó las etiquetas *SSLEngine on* para activar el modo SSL, las etiquetas *SSLCertificateFile* y *SSLCertificateKeyFile* para hacer referencia al certificado generado por "SAAAS!" como se muestra en el Espacio de Comandos 3.2.

Fragmento original del archivo *ssl*	Fragmento modificado del archivo *ssl*
… … </Directory> <FilesMatch "\.(cgi|shtml|phtml|php)$"> … … …	… … </Directory> **SSLEngine on** **SSLCertificateFile /var/www/openssl/cert/CA/newcerts/100001.pem** **SSLCertificateKeyFile /var/www/openssl/cert/CA/private/100001-key.pem** <FilesMatch "\.(cgi|shtml|phtml|php)$"> … … …

Espacio de Comandos 3.2 Cambios al archivo */etc/apache2/sites-enabled/ssl*

En el archivo */etc/apache2/mods-enabled/ssl.conf* se incluye la etiqueta *SSLPassPhraseDialog* como se muestra en el Espacio de Comandos 3.3, que hace referencia al archivo */var/www/openssl/cert/CA/ssl* que contiene la cadena de texto utilizada para descifrar el contenido del archivo que contiene la clave privada.

Fragmento original del archivo *ssl.conf*	Fragmento modificado del archivo *ssl.conf*
… # Pass Phrase Dialog: # Configure the pass phrase gathering process. # The filtering dialog program (`builtin' is a internal # terminal dialog) has to provide the pass phrase on stdout. SSLPassPhraseDialog builtin …	… # Pass Phrase Dialog: # Configure the pass phrase gathering process. # The filtering dialog program (`builtin' is a internal # terminal dialog) has to provide the pass phrase on stdout. **SSLPassPhraseDialog exec:/var/www/openssl/cert/CA/ssl** … …

Espacio de Comandos 3.3 Cambios al archivo */etc/apache2/mods-enabled/ssl.conf*

Para la configuración del host virtual en el puerto 80 se modifica el archivo */etc/apache2/sites-enabled/000-default* como se muestra en el Espacio de Comandos 3.4, la etiqueta *RewriteRule* es la encargada de realizar la redirección hacia el sitio seguro que se encuentra en el mismo servidor.

Contenido del archivo *000-default.conf*
<VirtualHost *:80> ServerAdmin Webmaster@localhost RewriteEngine on **RewriteRule /.* https://%{SERVER_NAME}$1 [L,R]** DocumentRoot /var/www/ ErrorLog /var/log/apache2/error.log LogLevel warn CustomLog /var/log/apache2/access.log combined </VirtualHost>

Espacio de Comandos 3.4 Cambios al archivo */etc/apache2/sites-enabled/000-default*

3.1.3.1.3. Instalación de FreeRADIUS

El paquete de FreeRADIUS disponible para Debian no soporta los protocolos de autenticación EAP-TLS, EAP-TTLS y EAP-PEAP debido a los problemas de

licenciamiento con OpenSSL detallados en el capítulo 2. Por esta razón, es necesario realizar algunos cambios en el código fuente de FreeRADIUS y crear un paquete de instalación propio utilizando *dpkg*[50].

Para la creación del paquete de instalación se descarga y descomprime el código fuente de la Versión 2.1.8 de FreeRADIUS de la página oficial del paquete ingresando las líneas mostradas en el Espacio de Comandos 3.5.

```
aaa:~# wget ftp://ftp.freeradius.org/pub/FreeRADIUS/FreeRADIUS-server-2.1.8.tar.gz
aaa:~# tar -vzxf freeradius-server-2.1.8.tar.gz
```

Espacio de Comandos 3.5 Descarga del paquete FreeRADIUS 2.1.8

Se descarga e instala los paquetes necesarios para la creación del paquete .deb, como se muestra en el Espacio de Comandos 3.6.

```
aaa:~# apt-get update
aaa:~# apt-get install fakeroot dpkg-dev libssl-dev quilt libtool libltdl3-dev libpam dpkg-dev
aaa:~# apt-get install libmysqlclient-dev libgdbm-dev libldap2-dev libsasl2-dev libiodbc2-dev python-dev
aaa:~# apt-get install libsnmp-dev libpq-dev libpam0g-dev libkrb5-dev libperl-dev libsnmp-dev libpq-dev
```

Espacio de Comandos 3.6 Descarga de paquetes necesarios para la creación del paquete .deb de FreeRADIUS

Para que FreeRADIUS sea compatible con EAP es necesario modificar el archivo *rules* ubicado en el directorio *debian/* del código fuente, antes de la creación del paquete e incluir las líneas que permiten la compilación del paquete con

[50] Programa base del sistema de gestión de paquetes de Debian, se utiliza para instalar, quitar, crear y proporcionar información de paquetes .deb.

compatibilidad para EAP-TLS, EAP-TTLS y EAP-PEAP. Las etiquetas *–with-rlm_eap_tls, –with-rlm_eap_ttls, –with-rlm_eap_peap* y *–with-openssl* deben ser incluidas en el archivo *rules*.

Fragmento original del archivo *rules*	Fragmento modificado del archivo *rules*
/configure $(confflags) \ … … … --config-cache \ --without-rlm_eap_tnc \ … … …	/configure $(confflags) \ … … … --config-cache \ **--with-rlm_eap_tls ** **--with-rlm_eap_ttls ** **--with-rlm_eap_peap ** **--with-openssl ** --without-rlm_eap_tnc \ … … …

Espacio de Comandos 3.7 Modificaciones al archivo *freeradius-server-2.1.8/debian/rules*

Una vez modificado el archivo *rules* se procede a la creación de los paquetes .deb para la instalación de FreeRADIUS con la herramienta *dpkg-buildpackage*[51]. Se utiliza la opción *-b,* del programa *dpkg-buildpackage* que especifica una construcción sólo binaria como se muestra en el Espacio de Comandos 3.8.

```
aaa:~# sudo dpkg-buildpackage -b
```

Espacio de Comandos 3.8 Creación de paquete .deb para la instalación de FreeRADIUS

Este comando genera los siguientes paquetes de instalación:

- ✓ freeradius_2.1.8+git_i386.deb

[51] *dpkg-buildpackage* es un programa que automatiza el proceso de construcción de un paquete de Debian

- ✓ freeradius-krb5_2.1.8+git_i386.deb
- ✓ freeradius-common_2.1.8+git_all.deb
- ✓ freeradius-ldap_2.1.8+git_i386.deb
- ✓ freeradius-utils_2.1.8+git_i386.deb
- ✓ freeradius-dbg_2.1.8+git_i386.deb
- ✓ freeradius-mysql_2.1.8+git_i386.deb
- ✓ libfreeradius2_2.1.8+git_i386.deb
- ✓ libfreeradius-dev_2.1.8+git_i386.deb
- ✓ freeradius-eaptls_2.1.8+git_i386.deb
- ✓ freeradius-eappeap_2.1.8+git_i386.deb

Los cuales se instalan utilizando el comando *dpkg* como se muestra en el Espacio de Comandos 3.9.

```
aaa:~# dpkg -i libfreeRADIUS2_2.1.8+git_i386.deb
aaa:~# dpkg -i freeRADIUS-common_2.1.8+git_all.deb
aaa:~# dpkg -i freeRADIUS_2.1.8+git_i386.deb
...
```

Espacio de Comandos 3.9 Instalación de paquetes necesarios para usar el comando "dpkg-buildpackage".

3.1.3.1.4. Configuración del archivo sudoers

Para que se puedan realizar cambios en la configuración de los servicios a través de la interfaz Web es necesario que los usuarios dueños de los servicios Apache y FreeRADIUS tengan privilegios similares al usuario root, por esta razón se modifica el archivo */etc/sudoers* utilizando el comando *visudo*, se utiliza el parametro *NOPASSWD* para independizar la programación de la aplicación Web con la contraseña de los usuarios *www-data* y *freerad* como se muestra en el Espacio de Comandos 3.10.

```
aaa:~# visudo
```

Fragmento original del archivo *sudoers*	Fragmento modificado del archivo *sudoers*
… # User privilege specification root ALL=(ALL) ALL …	… # User privilege specification root ALL=(ALL) ALL dragon ALL=(ALL) ALL www-data ALL=NOPASSWD: ALL freerad ALL=NOPASSWD: ALL …

Espacio de Comandos 3.10 Modificaciones en el archivo *sudoers*

3.1.3.1.5. Configuración del archivo crontab

Para que cada cinco minutos se ejecute el script encargado de copiar el contenido de los logs que FreeRADIUS genera en la base de datos local del sistema, es necesario configurar una tarea programada modificando el archivo */etc/crontab* como se indica en el Espacio de Comandos 3.11.

Para configurar una tarea en el archivo *crontab* se deben especificar siete campos:

- *Minuto:* El minuto de la hora en que el comando será ejecutado. El valor "*/5" indica cada cinco minutos.
- *Hora:* La hora en que el comando será ejecutado. El valor "*" indica: todas las horas.
- *Día del mes:* Día del mes en el que se quiere ejecutar el comando. El valor "*" indica todos los días del mes.
- *Mes:* Mes en el que se quiere ejecutar el comando. El valor "*" indica todos los meses.
- *Día de la semana:* Día de la semana en el que se quiere ejecutar el comando. El valor "*" indica todos los días de la semana.
- *Usuario:* Es el usuario que ejecuta el comando, en nuestro caso el usuario "root".
- *Comando:* El comando o script a ejecutarse, en nuestro caso se ejecuta el script

cron.php utilizando el intérprete de comandos cURL.

Fragmento modificado del archivo *crontab*

```
# m h dom mon dow user    command
17 *    * * *    root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *    root     test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.daily )
47 6    * * 7    root     test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.weekly )
52 6    1 * *    root     test -x /usr/sbin/anacron || ( cd / && run-parts --report
/etc/cron.monthly )
*/5 *    * * *  root   curl -k https://localhost/log/php/cron.php
```

Espacio de Comandos 3.11 Configuración del archivo *crontab*

3.1.3.1.6. Adaptación de GRUB2

Se modificó el paquete GRUB2 para que muestre el logotipo de “SAAAS!” durante el arranque del sistema, en lugar del tema por defecto de Debian Linux, lo cual se logró a través del uso de temas disponibles sólo en la versión 2 de grub como se aprecia en la Figura 3.3.

Figura 3.3 GRUB2 Modificado para la Meta Distribución "SAAAS!"

Se modificó el archivo */boot/grub/grub.cfg* para que acepte temas personalizados como se indica en el Espacio de comandos 3.12. La etiqueta *insmod gfxmenu* incluye al módulo *gfxmenu* dentro de la configuración del grub, la etiqueta *menuviewer* asigna el modulo que controla la presentación del grub en este caso *gfxmenu* y la etiqueta *theme* hace referencia al archivo de configuración del tema personalizado.

Fragmento modificado del archivo *grub.cfg*

```
…
function load_video {
insmod gfxmenu
  menuviewer="gfxmenu"
  theme="/boot/grub/themes/saaas/theme.txt"
insmod vbe
  insmod vga
  insmod video_bochs
  insmod video_cirrus
}
…
```

Espacio de Comandos 3.12 Fragmento modificado en el archivo */boot/grub/grub.cfg*

En el archivo *theme.txt* se modifica sólo dos líneas en la visualización del grub, el título que aparecerá en blanco y la imagen de fondo mediante las etiquetas *title-text* y *desktop-image* respectivamente.

Contenido del archivo */boot/grub/themes/saaas/theme.txt*

```
title-text: ""
desktop-image: "saaas.png"
```

Espacio de Comandos 3.13 Contenido del archivo *theme.txt*

3.1.3.2. Creación del LiveCD

3.1.3.2.1. Adaptación de ISOLINUX

Se modificaron los archivos de configuración de ISOLINUX, el gestor de arranque del LiveCD para personalizar el arranque "live" de la Meta Distribución "SAAAS".

El menú de arranque del LiveCD incluye las opciones:

- SAAAS!, para arrancar la meta distribución.
- Inicio desde el disco duro.

La configuración de este menú se encuentra en el archivo *menu.cfg* el cual llama al archivo *live.cfg* donde se configura la acción que se realiza al escoger los diferentes ítems del menú, y al archivo *stdmenu.cfg* en el que se maneja el entorno de visualización, los colores e imagen de fondo que se muestra al arrancar el CD.

En el archivo */etc/REMASTERSYS/isolinux/menu.cfg* se utilizan las etiquetas *menu hshift*, *menu width y menu title* para modificar el aspecto de la pantalla inicial al momento de cargar el LiveCD; como se muestra en el Espacio de Comandos 3.14, adicionalmente, se incluyen referencias hacia los archivos *live.cfg* y *stdmenu.cfg* cuyo contenido se detalla en los Espacios de Comando 3.15 y 3.16, respectivamente.

Contenido del archivo */etc/REMASTERSYS/isolinux/menu.cfg*

```
menu hshift 20
menu width 49

menu title LiveCD - SAAAS!
include stdmenu.cfg
include live.cfg
```

Espacio de Comandos 3.14 Configuración de ISOLINUX

En el archivo */etc/REMASTERSYS/isolinux/live.cfg* se utiliza la etiqueta *label* para definir una opción en el menú, la etiqueta *menu label* para asignar el texto a mostrar en esta opción y las etiqueta *kernel* y *append initrd* para indicar el kernel a cargar y las opciones para el mismo.

Contenido del archivo */etc/REMASTERSYS/isolinux/live.cfg*

```
label live
        menu label SAAAS!
        kernel /live/vmlinuz
        append initrd=/live/initrd.img boot=live config quiet splash vga=788 --
label hd
        menu label Inicio desde Disco duro
        localboot 0x80
        append -
```

Espacio de Comandos 3.15 Configuración de ISOLINUX

En el archivo */etc/REMASTERSYS/isolinux/stdmenu.cfg* se utilizan varias etiquetas referentes a la edición de la apariencia del menú, en este archivo las etiquetas relevantes son:

- *background* en la que se indica la ubicación de la imagen de fondo.
- *tabmsg* en la que se ingresa el texto a mostrar en la parte inferior del menú.

Contenido del archivo */etc/REMASTERSYS/isolinux/stdmenu.cfg*

```
menu background  /isolinux/saaas.png
menu color title        * #fa000000*
menu color border    * #00000000 #00000000 none
menu color sel                  * #fa000000 #f6a1d0ff *
menu color unsel     * #fa000000 #00000000 *
menu color hotsel    1;7;37;40 #fa000000 #f6a1d0ff *
menu color tabmsg   * #fa000000 #00000000 *
menu color help                 37;40 #fa000000 #00000000 none
```

```
menu vshift 16
menu rows 10
menu helpmsgrow 15
menu cmdlinerow 16
menu timeoutrow 16
menu tabmsgrow 18
menu tabmsg Seleccione una de las opciones
```

Espacio de Comandos 3.16 Configuración de ISOLINUX

En la Figura 3.4 se puede apreciar el resultado de la modificación de estos archivos.

Figura 3.4 Gestor de Arranque del LiveCD "SAAAS!" con ISOLINUX

3.1.3.2.2. Adaptación de REMASTERSYS

Para la creación de la Meta Distribución se modificó los scripts del paquete REMASTERSYS. Para el proceso de creación del LiveCD se utilizó el comando presentado en el Espacio de Comandos 3.17.

```
aaa:~# REMASTERSYS backup saaas.iso
```

Espacio de Comandos 3.17 Comando para la creación de la Meta Distribución utilizando REMASTERSYS.

El script */usr/bin/remastersys* es el encargado de crear la imagen squashfs del sistema operativo y convertirla en un *.iso*. En el Espacio de Comandos 3.18 se puede observar los cambios que se realizaron en este archivo. Fue necesario comentar las líneas del script que borraban el archivo */etc/hostname* y las claves generadas en el servidor para la conexión mediante SSH, opciones que generaban errores al momento de cargar el LiveCD.

Fragmento modificado del archivo *Remastersys*

```
…
…
…
sync --exclude='*.log.*' --exclude='*.pid' --exclude='*.bak' --exclude='*.[0-9].gz' --
exclude='*.deb' $VAREXCLUDES-a /var/. $WORKDIR/dummysys/var/.
rsync $VAREXCLUDES-a /etc/. $WORKDIR/dummysys/etc/.

rm -rf $WORKDIR/dummysys/etc/X11/xorg.conf*
rm -rf $WORKDIR/dummysys/etc/resolv.conf
rm -rf $WORKDIR/dummysys/etc/hosts
#rm -rf $WORKDIR/dummysys/etc/hostname
rm -rf $WORKDIR/dummysys/etc/timezone
rm -rf $WORKDIR/dummysys/etc/mtab
rm -rf $WORKDIR/dummysys/etc/fstab
rm -rf $WORKDIR/dummysys/etc/udev/rules.d/70-persistent*
#rm -rf $WORKDIR/dummysys/etc/cups/ssl/server.crt
#rm -rf $WORKDIR/dummysys/etc/cups/ssl/server.key
#rm -rf $WORKDIR/dummysys/etc/ssh/ssh_host_rsa_key
#rm -rf $WORKDIR/dummysys/etc/ssh/ssh_host_dsa_key.pub
#rm -rf $WORKDIR/dummysys/etc/ssh/ssh_host_dsa_key
#rm -rf $WORKDIR/dummysys/etc/ssh/ssh_host_rsa_key.pub
ls $WORKDIR/dummysys/var/lib/apt/lists | grep -v ".gpg" | grep -v "lock" | grep -v
"partial"…

…
…
```

Espacio de Comandos 3.18 Modificación del archivo */usr/bin/remastersys*

El script */usr/bin/remastersys-installer* se ejecuta desde el LiveCD y es el encargado

de la instalación del sistema operativo en el disco duro. Para personalizar la instalación de “SAAAS!” se modificó este script y se lo renombró de tal forma que al ejecutar el comando *setup* desde el LiveCD el usuario tiene acceso al diálogo de instalación de la Meta Distribución como se muestra en el Espacio de Comandos 3.19.

```
aaa:~# setup
```

Espacio de Comandos 3.19 Comando para la instalación en disco de "SAAAS!"

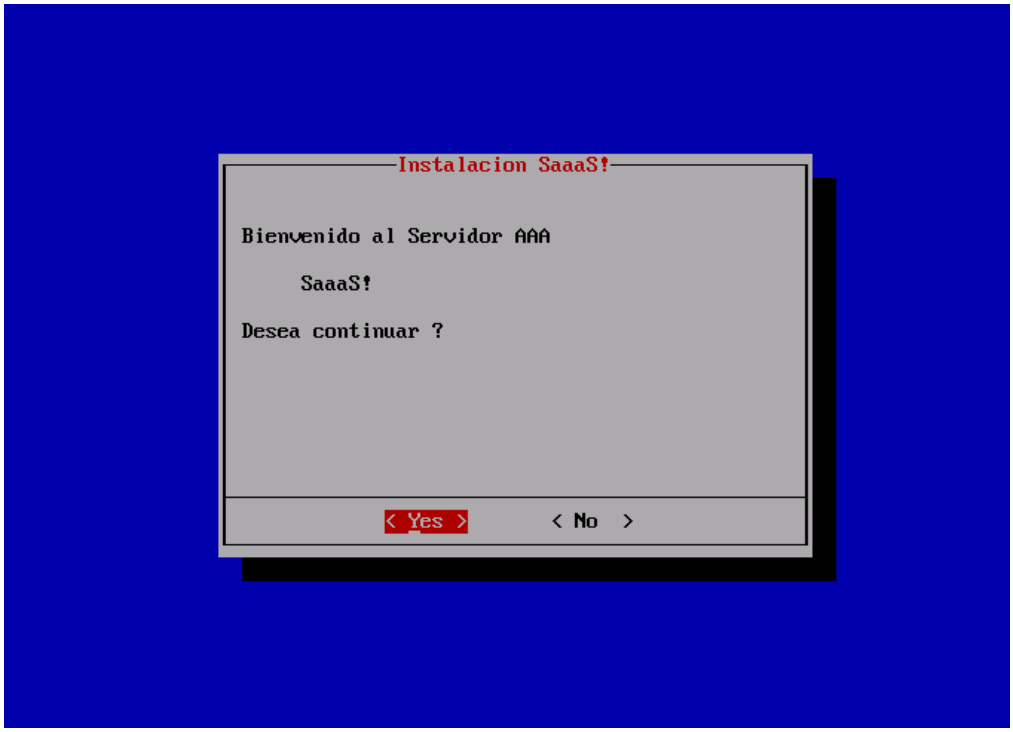


Figura 3.5 Diálogo de Instalación de la Meta Distribución "SAAAS!"

El script *setup* muestra un diálogo de instalación como se muestra en la Figura 3.5, el cual permite particionar de forma automática o manual el disco duro en el que se va a realizar la instalación, crear y montar un nuevo sistema de archivos sobre la partición destinada al sistema operativo, copiar el contenido del LiveCD a este nuevo sistema de archivos e instalar GRUB2 en el nuevo sistema para la gestión del

arranque en el disco duro.

En el archivo */usr/bin/setup* se utiliza el comando *dialog* para generar los cuadros de texto en los que el usuario elige varias opciones de acuerdo al proceso que se esté realizando, como se muestra en el Espacio de Comandos 3.20.

Fragmento del archivo */usr/bin/setup*, generación del diálogo de instalación.

```
for ((  i = 0 ;  i < 1 ; ))
do
$DIALOG $TITLE"$TITLETEXT" $MENU $TEXT"\nParticionado del Disco." $HEIGHT $WIDTH
$MENUHEIGHT Automatico "Particionamiento Automatico" Manual "Particionamiento Manual"
2>/tmp/choice.$$
if [ "$?" = "0" ]; then
        PARTTYPE=`cat /tmp/choice.$$`
        i=2
else
        $DIALOG $TITLE"$TITLETEXT" $YESNO $TEXT"\n\nDesea salir de la instalacion
?"$HEIGHT $WIDTH
        if [ "$?" = "0" ]; then
                clear
                exit 0
        fi
fi
done
rm /tmp/choice.$$
```

Espacio de Comandos 3.20 Fragmento del script generador del diálogo de instalación "SAAAS!".

Para el particionamiento automático primero se comprueba si la memoria RAM es menor a 2GB, si lo es se resta de este valor la memoria RAM y se obtiene el valor de la partición SWAP, caso contrario este valor será de 2GB. Utilizando el paquete fdisk se realizan dos particiones, una SWAP del valor calculado y una partición para archivos Linux con el resto del disco, como el paquete *fdisk* trabaja con un menú con varias opciones se le envía varios parámetros entre los que se especifica la creación de la partición de datos y de la SWAP calculada. Este procedimiento se detalla en el Espacio de Comandos 3.21.

Fragmento del archivo */usr/bin/setup*, particionamiento automático del disco mediante la herramienta *fdisk*

```
if [ $RAM -lt 2410000 ]; then
        SWAP=`expr 2048 - $RAM / 1024`
else
        SWAP=2048
Fi

$DIALOG $TITLE"$TITLETEXT" $YESNO $TEXT"\n\nConfirma la Autoparticion de $PARTDRIVE
?\
                \n\n $disco MB de Disco \n\n $SWAP MB de SWAP" $HEIGHT $WIDTH
        if [ "$?" = "0" ]; then
                disk="+$disco""M"
                fdisk /dev/$PARTDRIVE << EOF
d
1
d
2
d
3
d
4

n
p
1

$disk
n
p
2


t
2
82

m
a
1

m
w
EOF
        else
                clear
                exit 0
        fi
fi
```

Espacio de Comandos 3.21 Fragmento del script para el particionamiento automático del disco durante la instalación de "SAAAS!"

En el archivo */usr/bin/setup* se utilizan los comandos, *mount y umont* para manejar el montaje de las particiones, el comando *mkswap* para dar formato a la partición SWAP y *mke2fs* para dar formato del tipo *ext3* a la partición de datos. En el Espacio de Comandos 3.22 se puede observar el uso de estas herramientas durante el proceso de instalación de “SAAAS!”.

Fragmento del archivo */usr/bin/setup*, creación de la partición SWAP y asignación de formato a la partición de datos de “SAAAS!”

```
if [ "`mount | grep $TARGETPART`" ]; then
        umount /dev/$TARGETPART 2>&1 | echo "10" | dialog --gauge "\nDesmontando la
particion $TARGETPART para dar formato" 10 70 0
        sleep 5
fi

mkswap /dev/$SWAP 2>&1 | swapon /dev/$SWAP 2>&1 | echo "20" | dialog --gauge "\nCreando la
particion SWAP en $SWAP" 10 70 0
sleep 5

mke2fs -t ext3 /dev/$TARGETPART
mkdir -p /TARGET
sleep 5

echo "30" | dialog --gauge "\nFormateando la particion $TARGETPART como $FSTYPE" 10 70 0
sleep 5
```

Espacio de Comandos 3.22 Fragmento del script para el formateo del disco durante la instalación de "SAAAS!"

El Espacio de Comandos 3.23 muestra el fragmento del script */usr/bin/setup* en el que se monta la partición Linux en un directorio virtual llamado TARGET, sobre el que se realizará la copia del sistema de archivos del LiveCD hacia el disco duro y la modificación e instalación de grub.

Fragmento del archivo */usr/bin/setup*, montaje del nuevo sistema de archivos en la partición del disco.

```
mount /dev/$TARGETPART /TARGET -o rw
mkdir -p /TARGET/home
sleep 5

echo "35" | dialog --gauge "\nMontando la Particion $TARGETPART" 10 70 0
sleep 5
```

```
echo "Using tune2fs to prevent the forced checks on boot"
tune2fs -c 0 -i 0 /dev/$TARGETPART
rm -rf /TARGET/lost+found
sleep 5
```

Espacio de Comandos 3.23 Fragmento del script para el montaje del sistema de archivos durante la instalación de "SAAAS!"

En el archivo */usr/bin/setup* se utiliza el comando *rsync* para realizar una sincronización de todos los archivos existentes en el Live CD con la partición de datos. Los comandos utilizados para realizar esta tarea se detallan en el Espacio de Comandos 3.24.

Fragmento del archivo */usr/bin/setup*, copiado del sistema de archivos al disco duro

```
echo "40" | dialog --gauge "\nCopiando el sistema de archivos al disco duro.\nEsta tarea puede
tomar bastante tiempo." 10 70 0
sleep 5
rsync -a / /TARGET --exclude=/{TARGET,live,cdrom,mnt,proc,sys,media}
sleep 5
echo "50" | dialog --gauge "\nCopiando el sistema de archivos al disco duro.\nEsta tarea puede
tomar bastante tiempo." 10 70 0
sleep 5
mkdir -p /TARGET/{proc,mnt,sys,media/cdrom} 2>&1 | echo "60" | dialog --gauge "\nCopiado
finalizado." 10 70 0
sleep 5
```

Espacio de Comandos 3.24 Fragmento del script para copiar el sistema de archivos durante la instalación de "SAAAS!"

En el archivo */usr/bin/setup* se utiliza el script *grub-install* para generar los archivos de configuración del grub en la partición de datos y el script *update-grub2* para aplicar la personalización del menú del grub, como se observa en el Espacio de Comandos 3.25.

Fragmento del archivo */usr/bin/setup*, instalación del GRUB

```
ROOTMOUNT="/TARGET"
ROOTPART=`mount | grep " $ROOTMOUNT " | awk '{print $1}'`
chroot $ROOTMOUNT grub-install --no-floppy /dev/$PARTDRIVE
sleep 5
echo "92" | dialog --gauge "\nInstalando el Grub." 10 70 0
sleep 5
```

```
chroot $ROOTMOUNT update-grub2
sleep 5
mkdir -p /TARGET/boot/grub/themes/saaas/
cp  /usr/bin/themes/saaas/* /TARGET/boot/grub/themes/saaas/
sleep 5
```

Espacio de Comandos 3.25 Fragmento del script para la instalación del GRUB durante la instalación de "SAAAS!"

Una vez ejecutado el archivo */usr/bin/setup* se pide el reinicio de la máquina y el sistema se encuentra instalado satisfactoriamente, como la instalación de “SaaaS!” copia el sistema de archivos del LiveCD todos los servicios y configuraciones para la aplicación Web se encuentran ya instalados y funcionando por lo que no se debe realizar cambio alguno para poner en marcha el servidor.

3.2. PRUEBAS Y RESULTADOS

3.2.1. ENTORNO DE PRUEBA

3.2.1.1. Topología utilizada

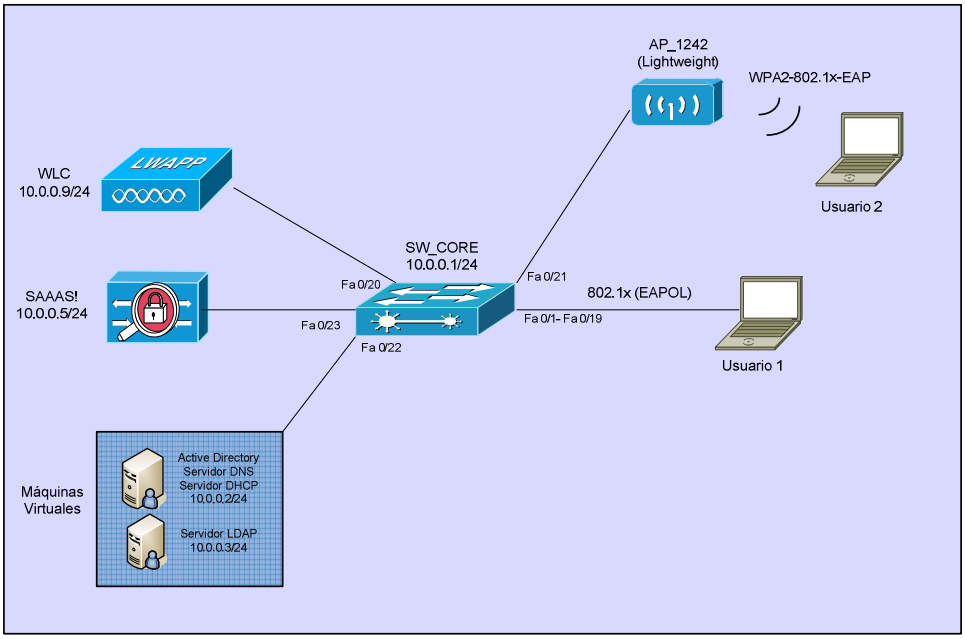


Figura 3.6 Topología de red utilizada durante las pruebas del servidor "SAAAS!"

Para las pruebas de funcionamiento del servidor “SAAAS!” se utilizó una topología acorde a la arquitectura AAA detallada en el primer capítulo de este documento. La tabla 3.3 detalla el rol de cada dispositivo en la topología de acuerdo a la arquitectura AAA y el direccionamiento IP utilizado durante las pruebas.

Dispositivo	Descripción	Rol AAA	Interfaz	Dirección IP/Máscara	Gateway
Servidor “SAAAS!”	Provee el servicio de FreeRADIUS y la interfaz de administración “SAAAS!”.	PDP, PIP y Sistema de Registro y Auditoría	eth0	10.0.0.5/24	10.0.0.1/24
Servidor Windows 2003	Provee los servicios de Active Directory, DNS y DHCP para la red de pruebas.	PIP	Ethernet	10.0.0.2/24	10.0.0.1/24
Servidor LDAP	Servidor Debian que provee el servicio LDAP embebido en Zimbra.	PIP	eth0	10.0.0.3/24	10.0.0.1/24
WLC	Controladora Inalámbrica que actúa como cliente RADIUS	PDP	Management	10.0.0.9/24	
			Manager	10.0.0.8/24	
Switch de Core	Switch capa 3 que realiza el enrutamiento entre VLANs y funciona como cliente RADIUS	PDP	VLAN 1	10.0.0.1/24	
			VLAN 2: Empleados	10.0.2.1/24	
			VLAN 3: Autoridades	10.0.3.1/24	
			VLAN 4: Consultores	10.0.4.1/24	
			VLAN 120: Falla	10.0.120.1/24	
			VLAN 121: Invitados	10.0.121.1/24	

Tabla 3.3 Tabla de direccionamiento IP utilizado en el esquema de laboratorio

3.2.1.2. Configuración de los Dispositivos Involucrados

3.2.1.2.1. Configuración básica del servidor "SAAAS!"

Después de realizar la instalación en disco desde el LiveCD de "SAAAS!", utilizando el comando *setup,* se cambió la dirección IP por defecto del servidor y se le asignó una dirección dentro del esquema de la red LAN del laboratorio.

El servicio FreeRADIUS y la autenticación utilizando diferentes métodos EAP funcionan por defecto después de la instalación de la Meta Distribución "SAAAS!", por lo que sólo es necesario agregar los clientes NAS, configurar los repositorios de credenciales a usar y personalizar los esquemas de autorización para completar la configuración del servicio AAA.

Se agregaron dos clientes NAS al servidor, el switch de core y la controladora inalámbrica. La configuración de los repositorios de credenciales y los esquemas de autorización se detallarán en cada uno de los escenarios descritos en los puntos 3.4.2 y 3.4.3

Figura 3.7 Configuración de Clientes en el Servidor "SAAAS!" para las pruebas de Laboratorio

3.2.1.2.2. Configuración del Servicio DNS y Active Directory

El proceso de instalación y puesta en marcha de los servicios de DNS y Active Directory en Windows 2003 está fuera del alcance de este documento por lo cual no será descrito, documentos relacionados a estos procesos están disponibles directamenta en la página oficial de Microsoft. La configuración necesaria en estos servicios para la integración con el servidor “SAAAS!” se describe en los siguientes párrafos:

Con el fin de realizar las correspondientes pruebas se crearon tres usuarios y tres grupos dentro de la unidad organizativa *saaas* en el dominio *saaas.ad* como se muestra en las figuras 3.8 y 3.9. Estos grupos se utilizarán durante el proceso de autorización del servicio AAA.

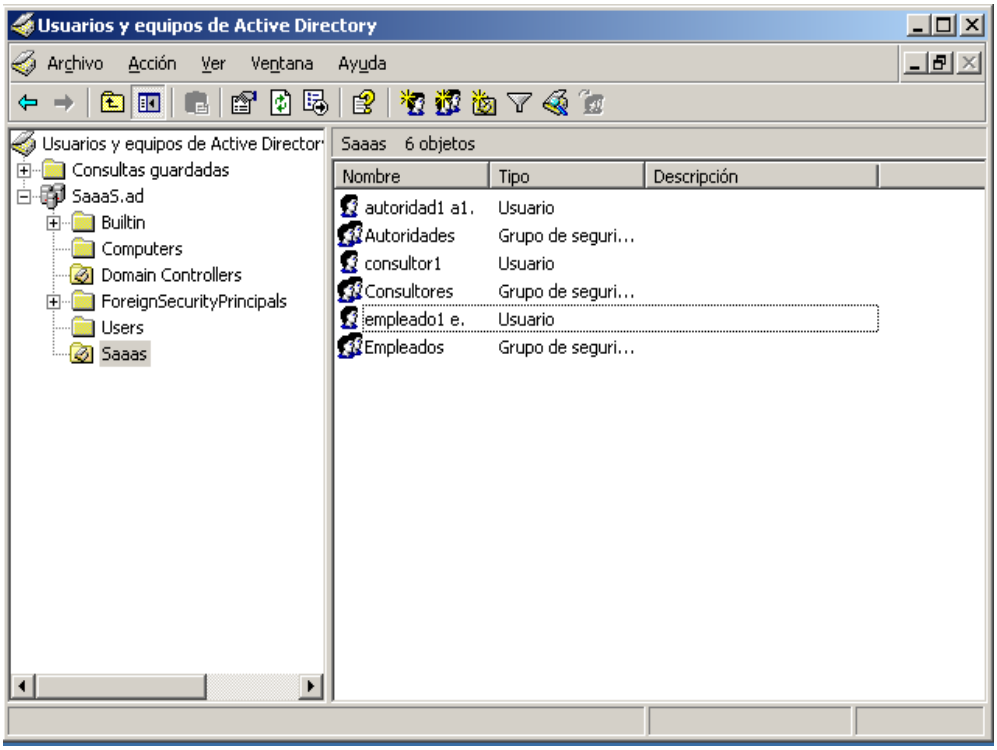


Figura 3.8 Usuarios y Equipos del dominio *saaas.ad* utilizado para las pruebas de laboratorio

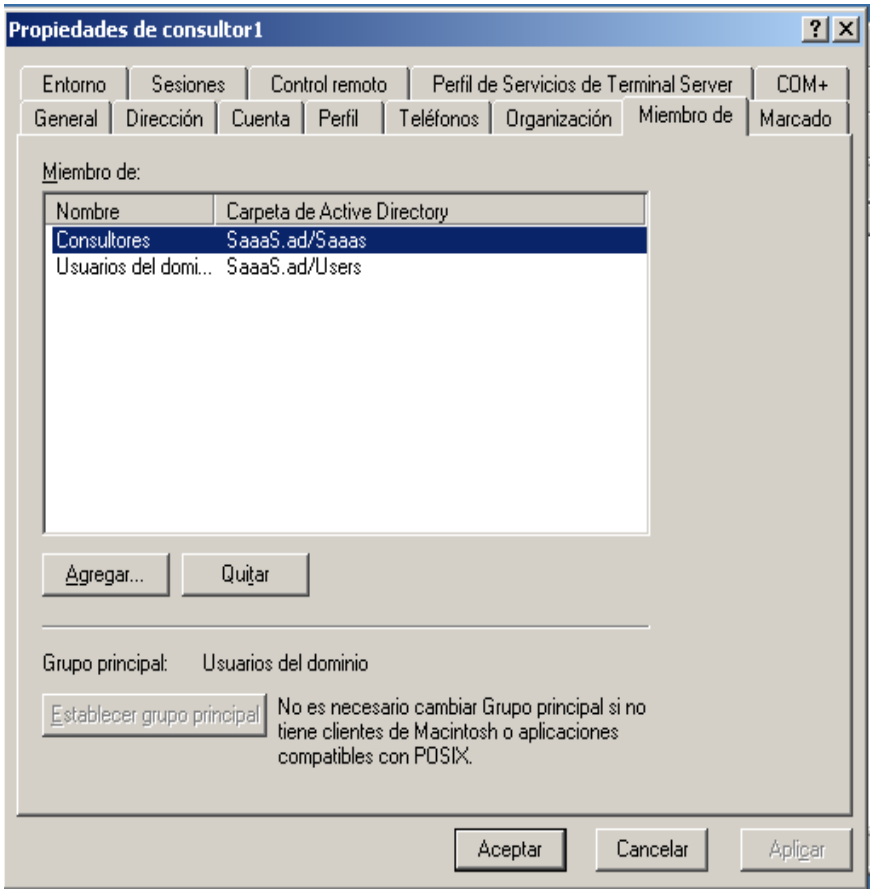


Figura 3.9 Detalle del usuario *consultor1* miembro del grupo Consultores del dominio *saaas.ad* utilizado para las pruebas de laboratorio

Para el correcto funcionamiento de la red inalámbrica es necesario crear una entrada DNS que apunte a la dirección IP de la controladora inalámbrica (Cisco WLC) como se muestra en la Figura 3.10.

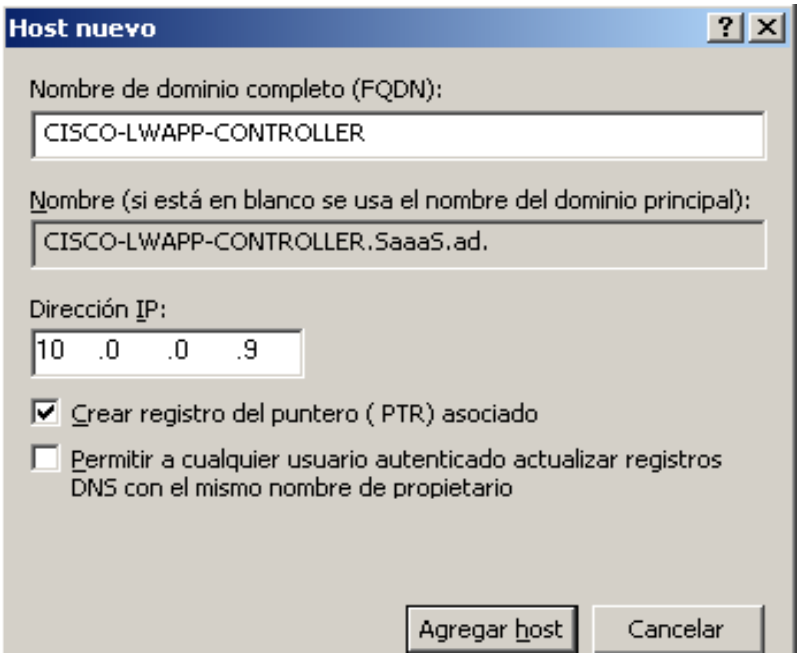


Figura 3.10 Entrada DNS de la Cisco WLC

3.2.1.2.3. Configuración del Servicio DHCP

Debido a que el esquema de pruebas utiliza diferentes VLANs para dividir a los usuarios según sus perfiles durante el proceso de autorización del servicio AAA, fue necesario crear ámbitos diferentes para la asignación dinámica de IP en cada una de las VLANs como se muestra en la Figura 3.11.

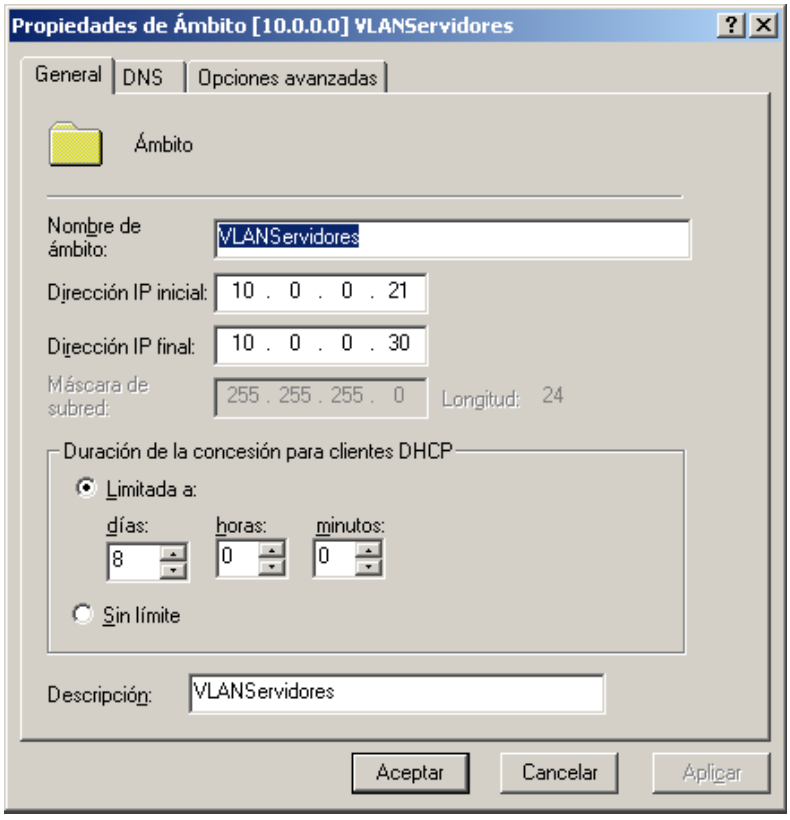


Figura 3.11 Ejemplo de ámbito DHCP creado para las pruebas de laboratorio

3.2.1.2.4. Configuración del Servidor LDAP

Para las pruebas de laboratorio se utilizó el servidor LDAP incluido en Zimbra Collaboration Server Open Source Edition[52].

Zimbra utiliza por defecto OpenLDAP para proveer un repositorio de credenciales de usuarios para el servicio de correo. Se crearon cuentas de correo en el servidor Zimbra para almacenar las credenciales de los usuarios en el servidor LDAP y también listas de correos, análogas a los grupos creados en el Active Directory como se puede apreciar en las figuras 3.12, 3.13 y 3.14.

[52] Robusto servidor de mail y colaboración de código abierto integrado con una herramienta de administración Web basada en AJAX.

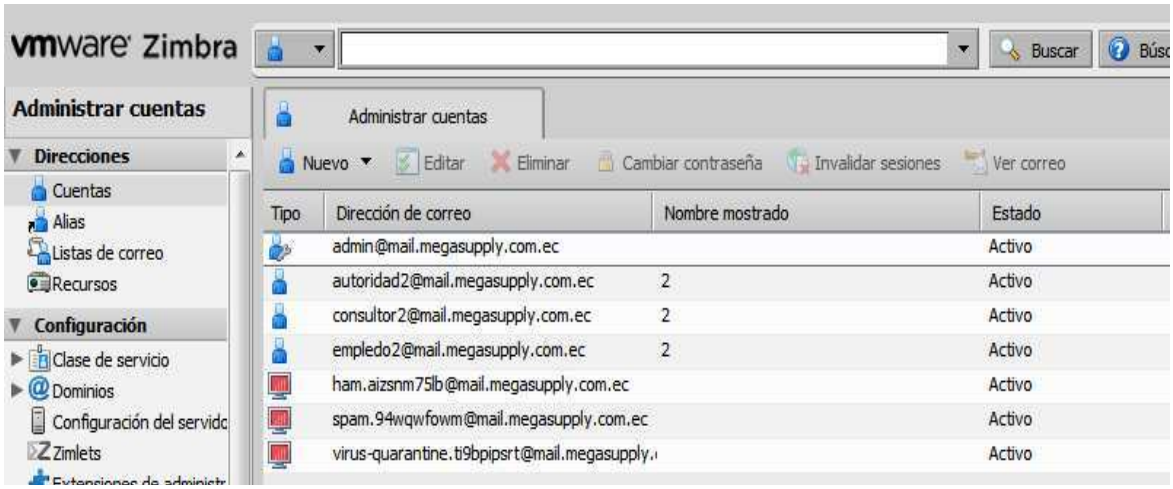


Figura 3.12 Cuentas de Correo creadas en el servidor Zimbra

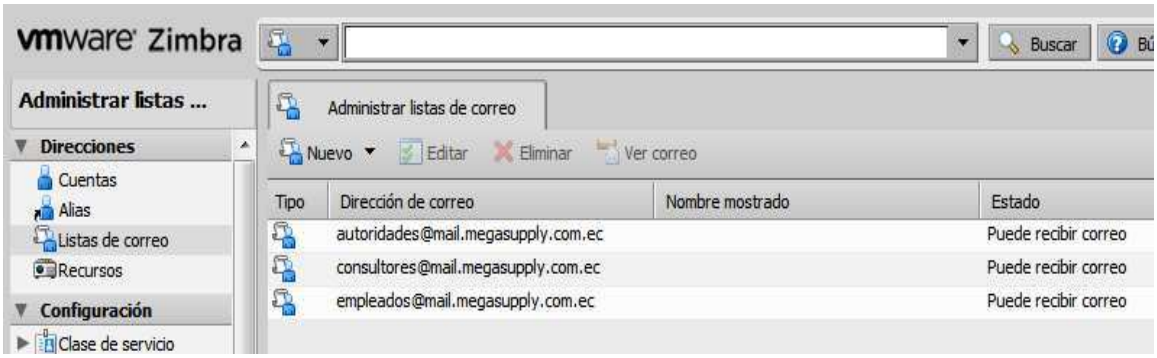


Figura 3.13 Listas de Correo creadas en el servidor Zimbra

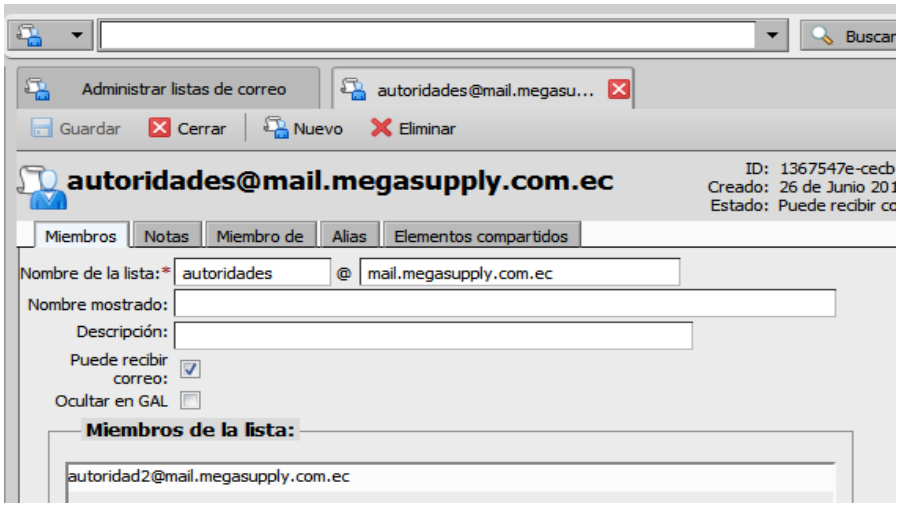


Figura 3.14 Detalle de usuarios miembros de una lista de correos en el servidor Zimbra

3.2.1.2.5. Configuración de la WLC

En la opción RADIUS de la pestaña de seguridad de la administración Web de la controladora inalámbrica se configuraron los parámetros para que el dispositivo pueda funcionar como cliente RADIUS del servidor “SAAAS!” como se muestra en la Figura 3.15.

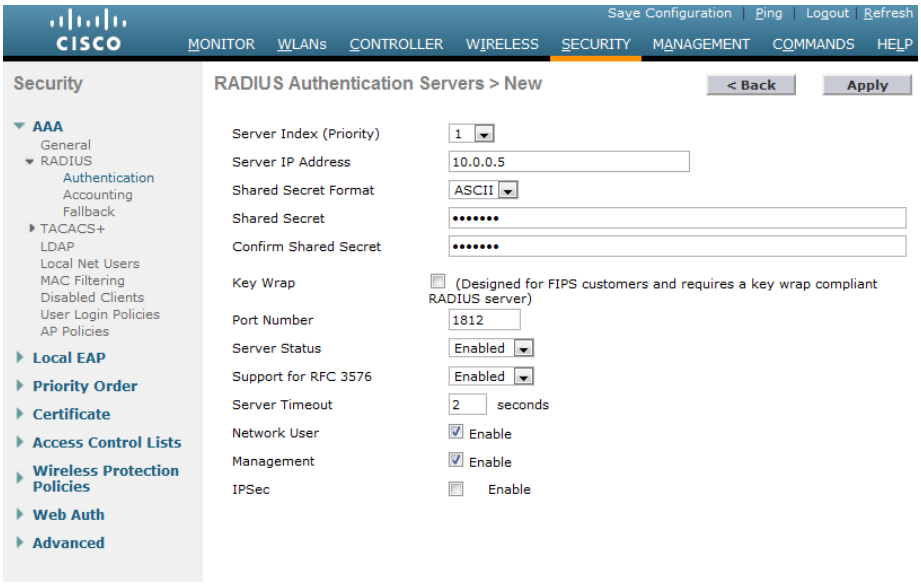


Figura 3.15 Configuración de la WLC como cliente RADIUS

Para el funcionamiento del esquema de autorización se configuraron interfaces lógicas en la controladora inalámbrica para trabajar con el mismo direccionamiento y esquema de VLANs que en la red cableada como se puede observar en las figuras 3.16 y 3.17.

Figura 3.16 Configuración de las interfaces en la WLC del esquema de laboratorio

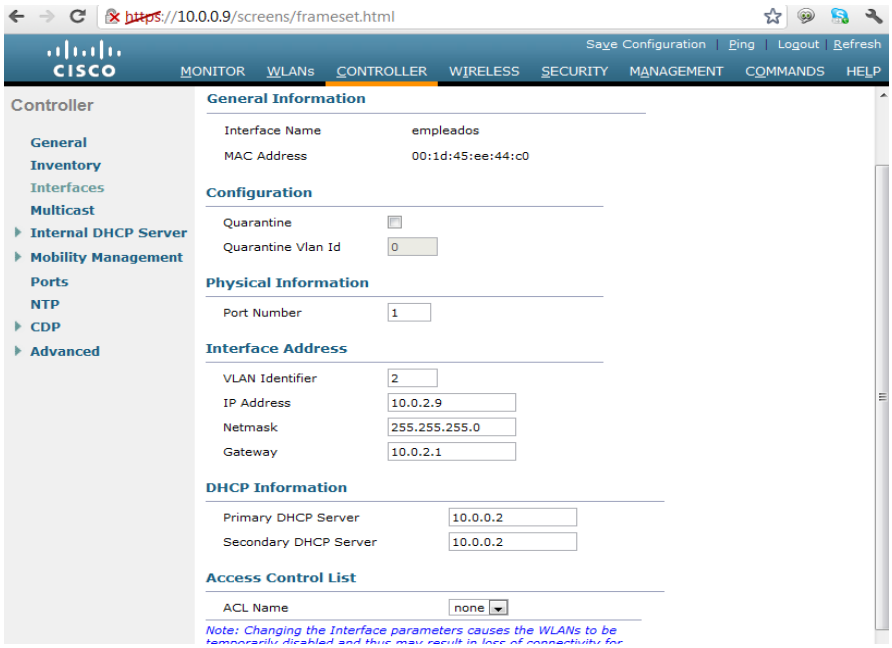


Figura 3.17 Detalle de la configuración de una interfaz en la WLC

3.2.1.2.6. Configuración del Switch de Core

Es necesario configurar un usuario local con todos los privilegios en el switch para la administración del equipo en caso de que la conexión con el servidor RADIUS falle como se muestra en el Espacio de Comandos 3.26.

Configuración de un usuario local

```
hostname SW_CORE
!
!
username cisco privilege 15 password 0 cisco
username eve privilege 15 password 0 eve
```

Espacio de Comandos 3.26 Configuración de un usuario local en el switch de pruebas de laboratorio

Se configura el esquema de VLANs para la asignación dinámica en el puerto una vez que el usuario se ha autenticado como se muestra en el Espacio de Comandos 3.27.

Configuración de VLANs

```
vlan 2
 name empleados
!
vlan 3
 name autoridades
!
vlan 4
 name consultores
!
vlan 120
 name Falla
!
vlan 121
 name Invitados

.....

interface Vlan1
 ip address 10.0.0.1 255.255.255.0
 !
interface Vlan2
ip address 10.0.2.1 255.255.255.0
ip helper-address 10.0.0.2
!
interface Vlan3
 ip address 10.0.3.1 255.255.255.0
 ip helper-address 10.0.0.2
```

Configuración de VLANs

```
!
interface Vlan4
 ip address 10.0.4.1 255.255.255.0
 ip helper-address 10.0.0.2
!
interface Vlan120
 ip address 10.0.120.1 255.255.255.0
!
interface Vlan121
 ip address 10.0.121.1 255.255.255.0
!
```

Espacio de Comandos 3.27 Configuración de VLANs en el switch de pruebas de laboratorio

Es necesario activar el modelo AAA en el switch para indicar que los procesos de autenticación, autorización y auditoría se realizarán con el protocolo RADIUS, para esta acción se utilizaron las líneas mostradas en el Espacio de Comando 3.28.

Configuración del esquema AAA

```
aaa new-model
!
!
aaa authentication login default group radius local
aaa authentication dot1x default group radius
aaa authorization exec default group radius local
aaa authorization network default group radius
aaa accounting exec default start-stop group radius
aaa accounting network default start-stop group radius
```

Espacio de Comandos 3.28 Configuración del esquema AAA en el switch para pruebas de laboratorio

Para configurar al equipo como cliente de “SAAAS!” se especifica la Vlan1 como la interfaz por la cual se deben enviar todos los paquetes RADIUS, se apunta a la dirección IP y puertos del servidor “SAAAS!” y se configura el secreto compartido entre el cliente y el servidor con las líneas mostradas en el Espacio de Comandos 3.29.

Configuración de Radius

```
ip radius source-interface Vlan1
radius-server host 10.0.0.5 auth-port 1812 acct-port 1813
radius-server key demoaaa
```

Espacio de Comandos 3.29 Configuración como cliente RADIUS en el switch de pruebas de laboratorio

Se debe configurar 802.1x en cada uno de los puertos del switch a los que se conectarán equipos de red, indicando que se realice MAC Address Bypass para que dispositivos de red como impresoras puedan realizar autenticación por MAC y definiendo las VLANs de Invitados (usuarios no autenticados), VLAN de Falla (usuarios con credenciales erróneas) y la VLAN Crítica a la cual se enviarán todos los usuarios en caso de que no se pueda establecer comunicación con el servidor RADIUS, la configuración se la realiza con las líneas demostradas en el Espacio de Comandos 3.30.

Configuración de 802.1x en los puertos de acceso

```
interface FastEthernet0/1
 switchport mode access
 dot1x mac-auth-bypass
 dot1x pae authenticator
 dot1x port-control auto
 dot1x violation-mode protect
 dot1x reauthentication
 dot1x guest-vlan 121
 dot1x auth-fail vlan 120
 dot1x critical vlan 2
 spanning-tree portfast
!
```

Espacio de Comandos 3.30 Configuración de 802.1x en los puertos de acceso del switch para pruebas de laboratorio.

Los puertos del switch a los que se conectarán los servidores y los equipos activos de red no deben tener 802.1x configurado, en el caso de la controladora es necesario

que el puerto esté configurado como troncal para que permita el paso de todas las VLANs como se muestra en el Espacio de Comandos 3.31.

Configuración de puertos de servidores.

```
interface FastEthernet0/20
 switchport trunk encapsulation dot1q
 switchport mode trunk
!
interface FastEthernet0/21
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/22
 switchport mode access
 spanning-tree portfast
!
interface FastEthernet0/23
 switchport mode access
 spanning-tree portfast
```

Espacio de Comandos 3.31 Configuración de los puertos para servidores y enlace troncal en el switch para pruebas de laboratorio

3.2.2. ESCENARIO I: ESQUEMA AAA UTILIZANDO LA BASE DE DATOS LOCAL DE "SAAAS!" COMO REPOSITORIO DE CREDENCIALES

3.2.2.1. Objetivo

Analizar los protocolos que intervienen en el proceso de Autenticación de un usuario local dentro de un esquema AAA de prueba.

3.2.2.2. Alcance

En este escenario se implementa un esquema AAA básico en los que interviene un usuario (computador portátil), un cliente (switch de acceso) y un servidor AAA (SAAAS!). El usuario se registra en la red LAN por medio de 802.1x y después de la autenticación se aplican políticas de acceso basadas en su perfil de usuario. Las credenciales que utilizan son almacenadas localmente en una base de datos MySQL

del servidor “SAAAS!”.

3.2.2.3. Análisis de Resultados

Para el análisis de los resultados se ha dividido el esquema en diferentes fases y en cada una de ellas se han estudiado los paquetes que han sido enviados.

Fase 1

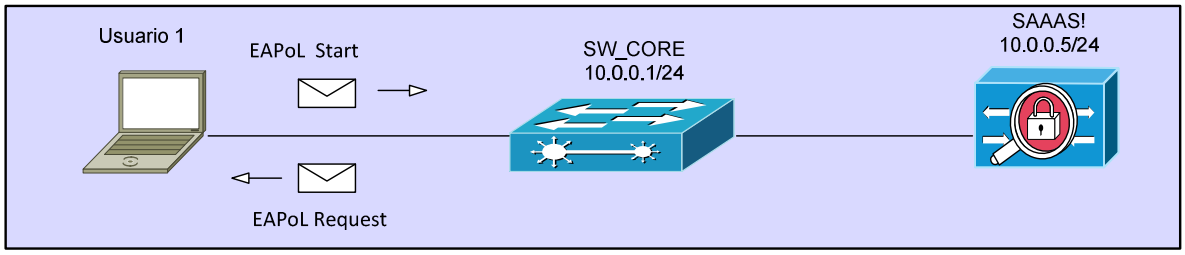


Figura 3.18 Esquema Fase 1

Como se presenta en la Figura 3.18, en la Fase 1, El usuario inicia el proceso con el paquete EAPoL, enviando en el atributo Type del paquete el valor binario 1 que equivale a la opción Start del proceso EAP como se aprecia en la captura de tráfico realizada con el aplicativo Wireshark y presentada en la Figura 3.19.

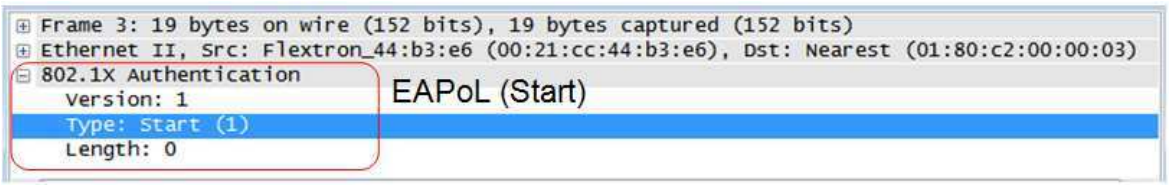


Figura 3.19 Paquete EAPoL (Start)

El Cliente responde a este paquete con un EAPoL Request Identity indicándole que existe este servicio en la red y que envíe la identidad del usuario y la variante de EAP que va a utilizar como se observa en la Figura 3.20.

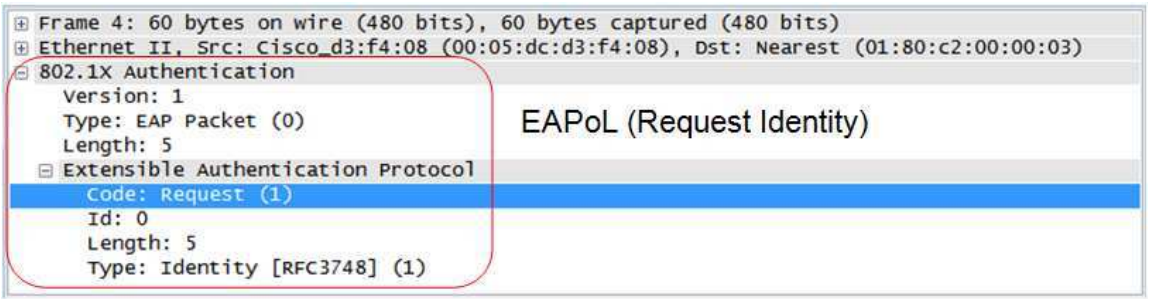


Figura 3.20 Paquete EAPoL (Request Identity)

Fase 2

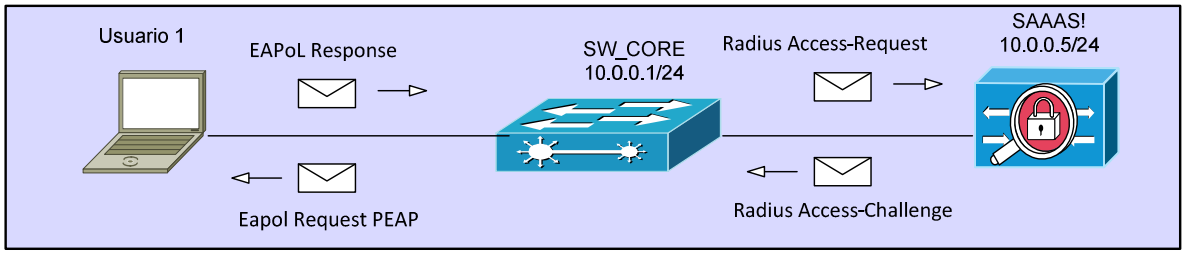


Figura 3.21 Esquema Fase 2

Como se presenta en la Figura 3.21, en la Fase 2, el usuario envía una respuesta con el paquete EAPoL Response, enviando en el paquete, en el atributo Identity, el valor del nombre de usuario para la autenticación como se muestra en la Figura 3.22.

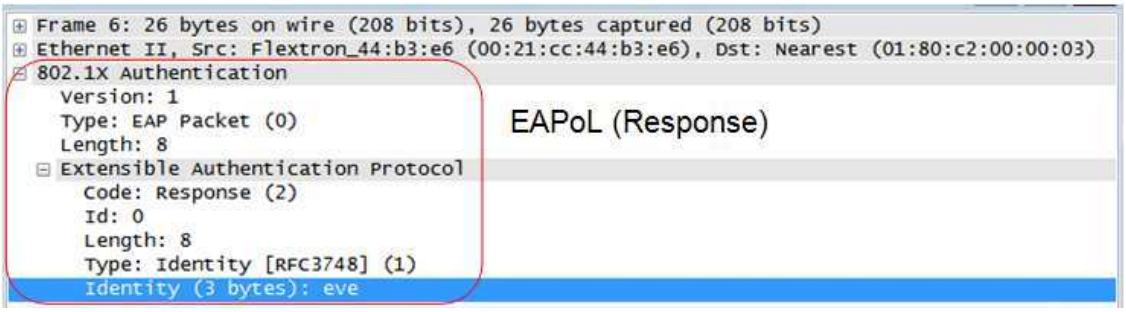


Figura 3.22 Paquete EAPoL (Response)

El cliente encapsula el paquete EAPoL Response dentro del protocolo RADIUS, adicionalmente añade atributos al paquete RADIUS correspondientes a la

información sobre el cliente, este paquete es enviado hacia el servidor “SAAAS!”

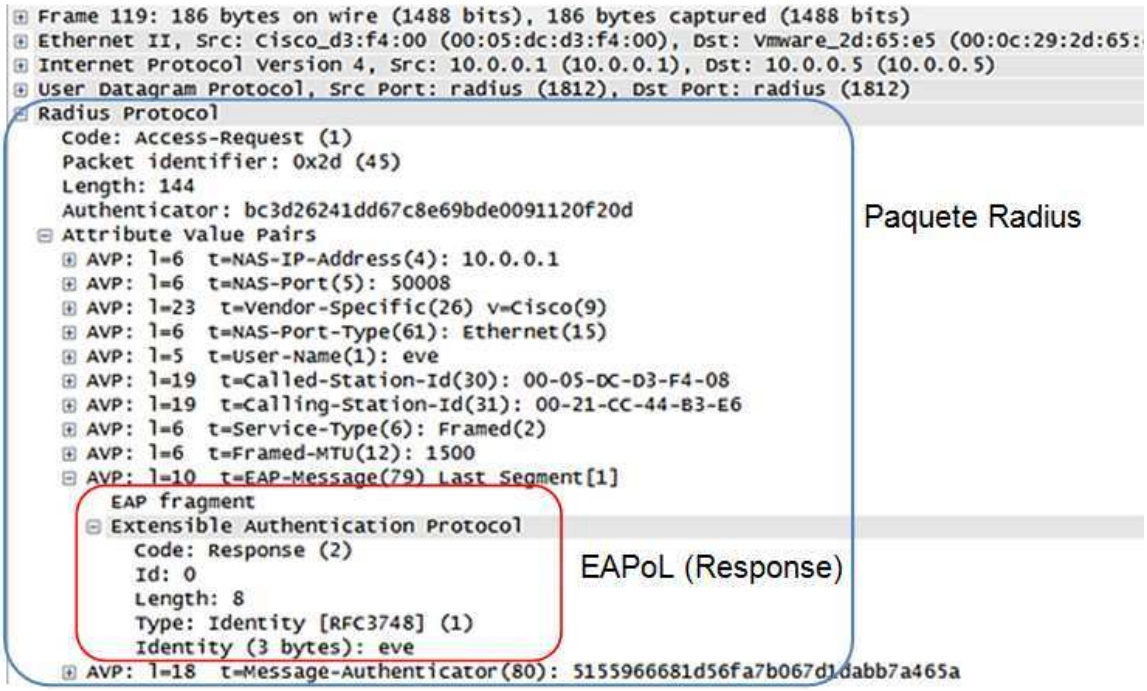


Figura 3.23 Paquete Radius Access-Request

El servidor “SAAAS!” recupera el paquete EAPoL Response y los atributos en el paquete RADIUS, lee el atributo de identidad y revisa la existencia del usuario en el repositorio interno, si el usuario existe responde con un paquete Radius Access-Challenge, en donde envía un paquete EAPoL Request PEAP con el tipo de EAP a utilizar (PEAP) y la bandera para iniciar este tipo de conversación, encapsulado en un paquete RADIUS dirigido al cliente como se muestra en la Figura 3.24.

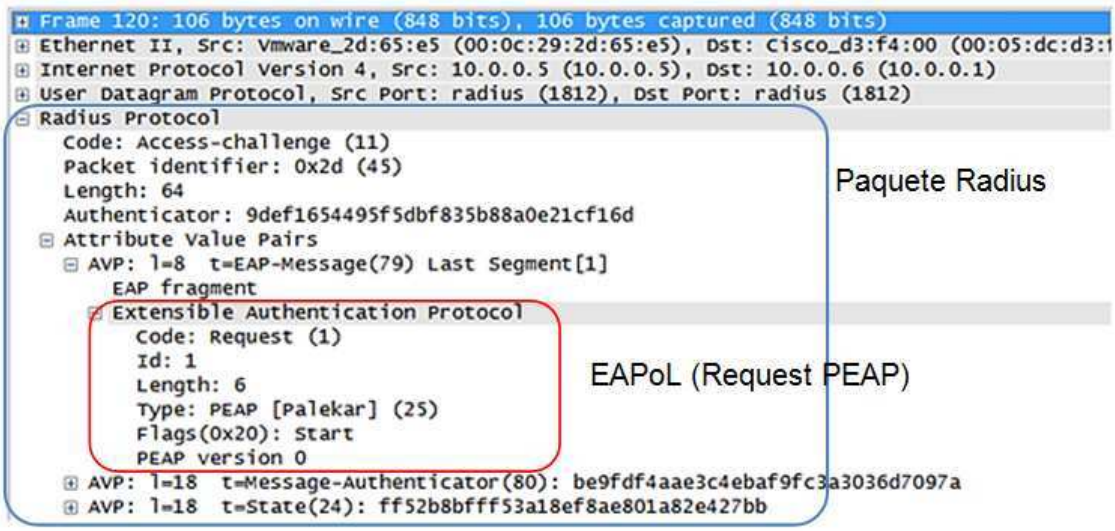


Figura 3.24 Paquete Radius Access - Challenge

El cliente recupera el paquete EAPoL Request PEAP y lo reenvía hacia el usuario como se aprecia en la Figura 3.25.

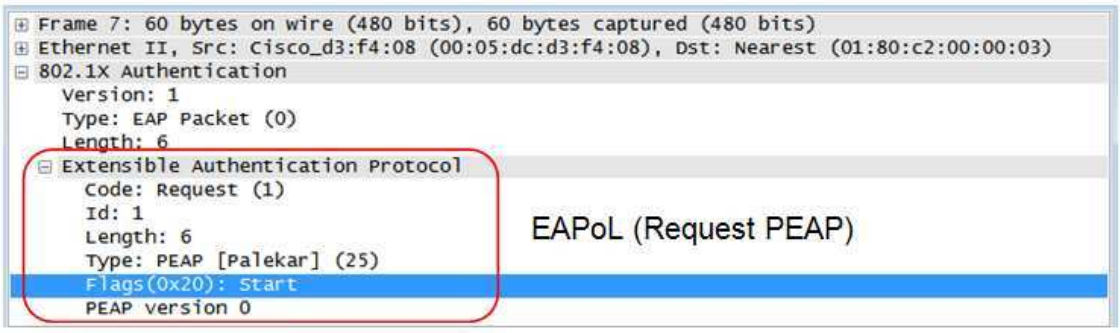


Figura 3.25 Paquete EAPoL Request PEAP

Fase 3

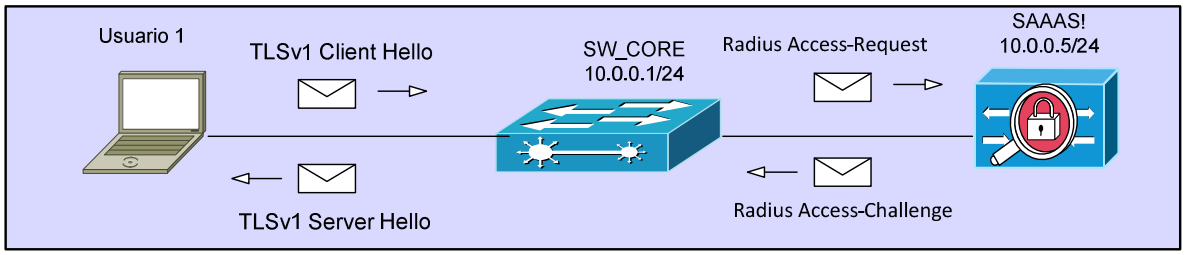


Figura 3.26 Esquema Fase 3

Como se aprecia en la Figura 3.26, en la Fase 3, el usuario inicia el intercambio de mensajes PEAP, basado en TLSv1 que se detalla en la Figura 3.27.

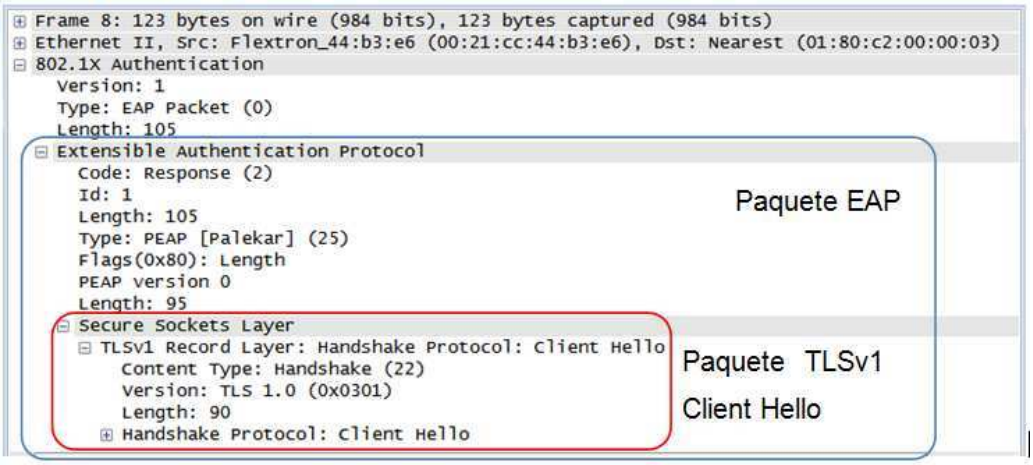


Figura 3.27 Paquete TLSv1 Client Hello

De igual manera que en la fase anterior el cliente encapsula los paquetes EAP entre el usuario y el servidor “SAAAS!” como se aprecia en las figuras 3.28 y 3.29.

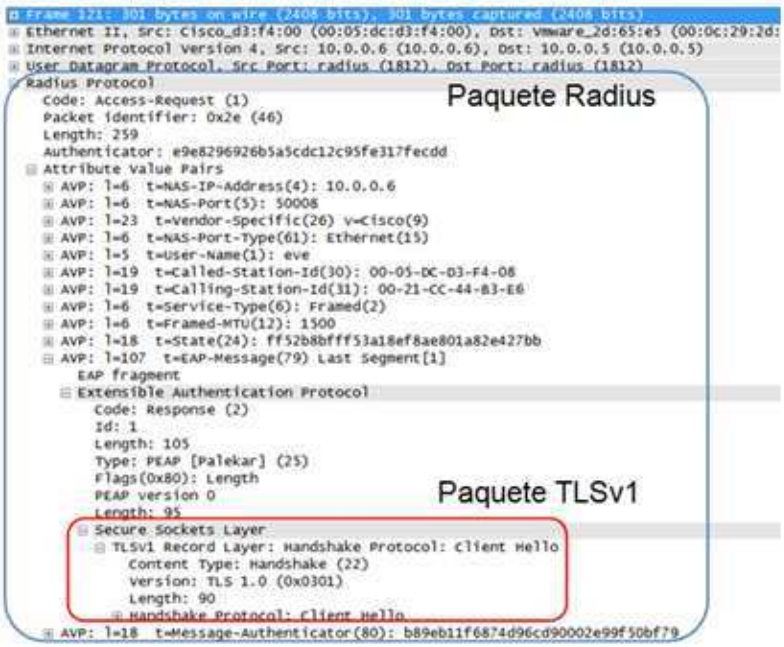


Figura 3.28 Paquete TLSv1 Encapsulado sobre Radius

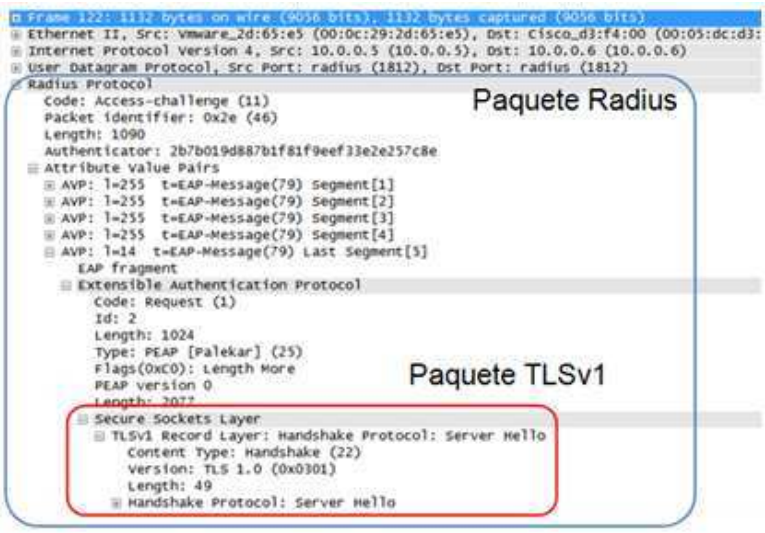


Figura 3.29 Paquete TLSv1 Encapsulado sobre Radius

El cliente recupera el paquete EAPoL TLSv1 Server Hello y una secuencia de paquetes en los que el servidor “SAAAS!” envía el certificado para iniciar el túnel encriptado, estos paquetes son reenviados hacia el usuario para su aceptación o denegación como se observa en la Figura 3.30.

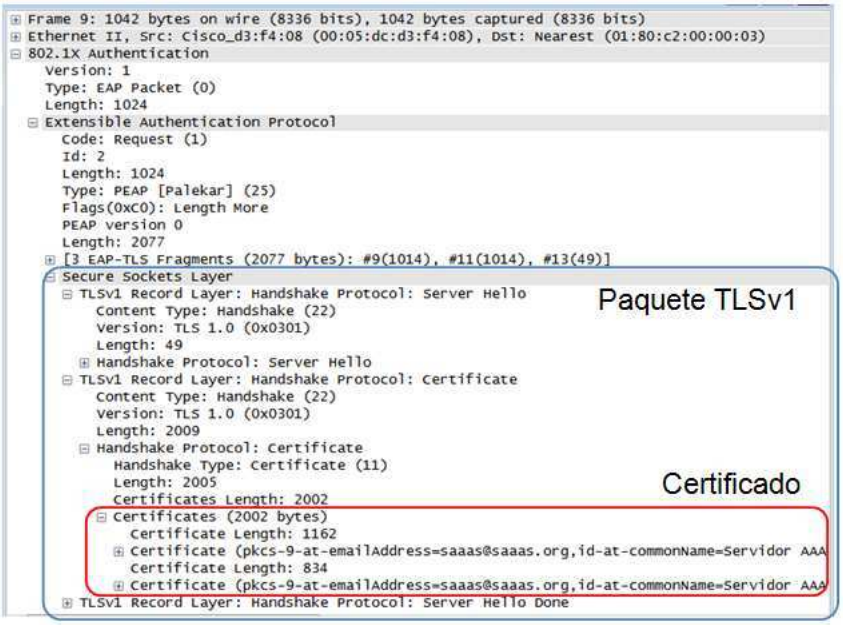


Figura 3.30 Paquete EAPoL TLSv1 Server Hello

Fase 4

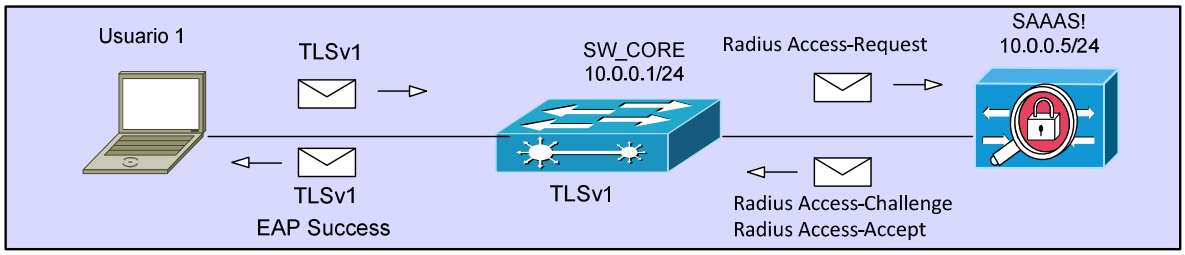


Figura 3.31 Esquema Fase 4

Como se aprecia en la Figura 3.31, en la Fase 4, el usuario intercambia los mensajes de autenticación (contraseña) a través del método de autenticación MSCHAPv2 sobre un túnel encriptado (TLSv1); este intercambio se realiza con el servidor "SAAAS!", por lo que el cliente seguirá encapsulando estos mensajes sobre el protocolo RADIUS como en las fases anteriores, como se muestra en la Figura 3.32.

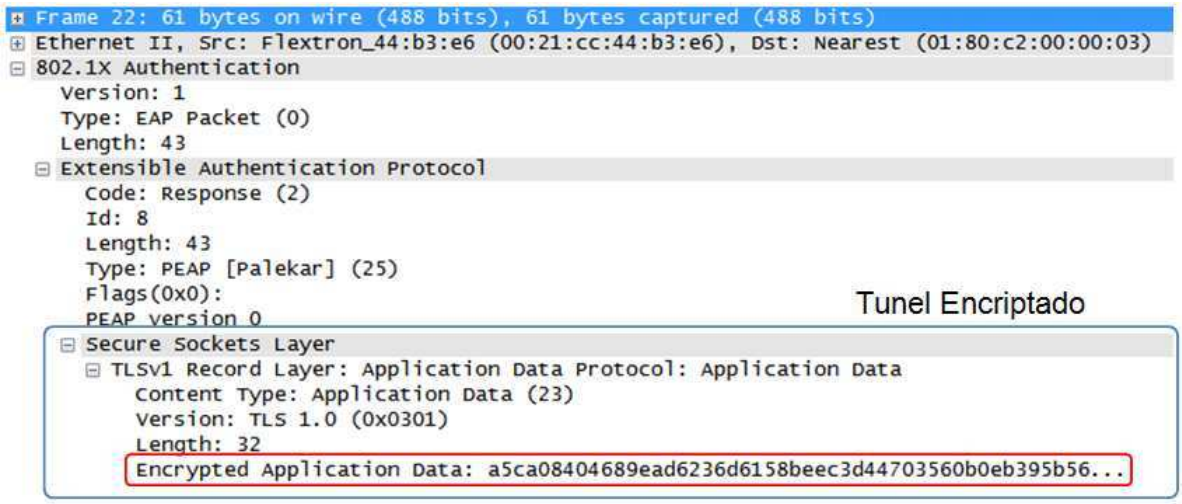


Figura 3.32 Paquete TLSv1 Túnel Encriptado

Una vez que el servidor "SAAAS!" reconozca al usuario como válido y recupere las políticas asociadas a éste, enviará un paquete Radius Access-Accept en el cual indica al cliente que el usuario está autorizado y las políticas que debe aplicarle a

éste, adicionalmente encapsula el paquete EAP-Success que será enviado al usuario estos paquetes se pueden apreciar en las figuras 3.33 y 3.34.

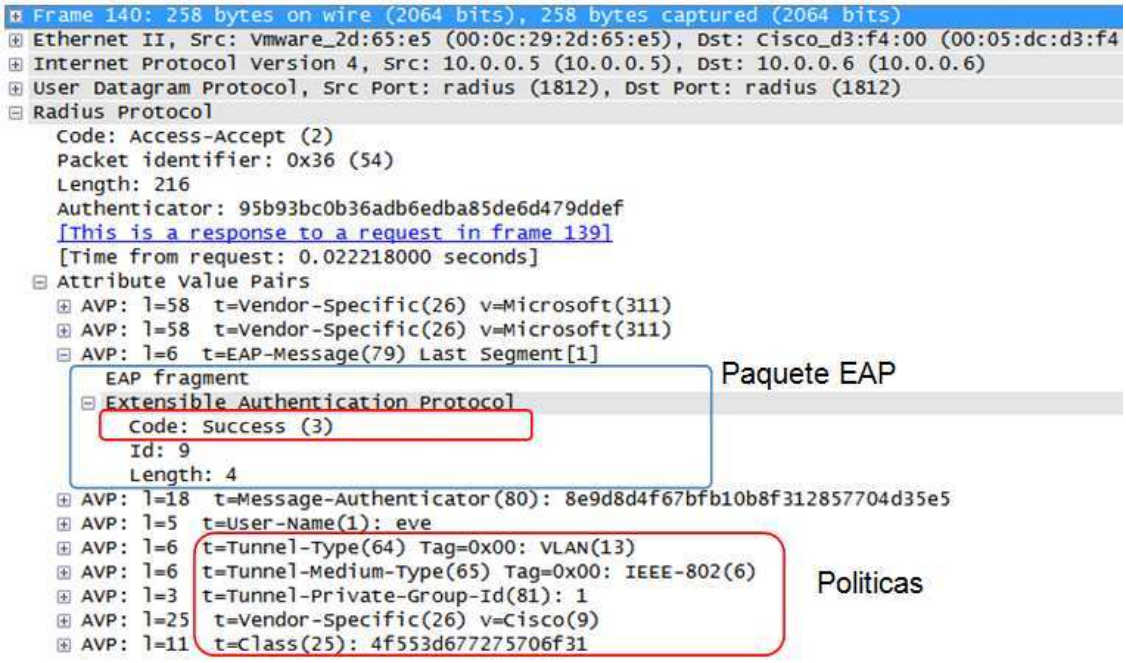


Figura 3.33 Paquete Radius Access-Accept

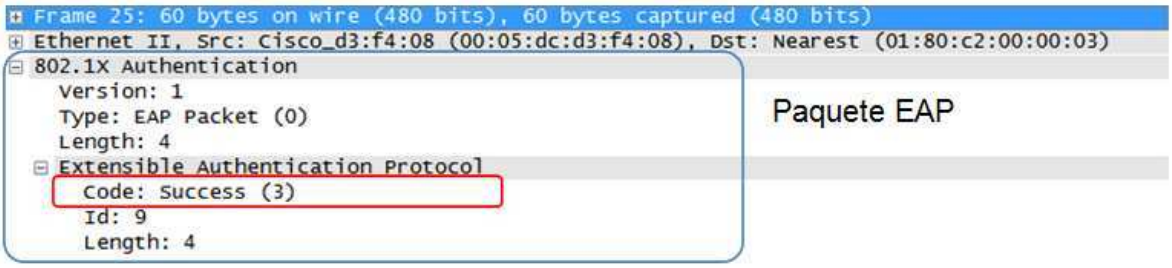


Figura 3.34 Paquete EAP-Success

3.2.3. ESCENARIO II: ESQUEMA AAA UTILIZANDO UN REPOSITORIO DE CREDENCIALES EXTERNO

3.2.3.1. Objetivo

Analizar los protocolos que interviene en el proceso de Autenticación de un usuario bajo Active Directory dentro de un esquema AAA de prueba.

3.2.3.2. Alcance

En este escenario se implementa un esquema AAA básico en los que interviene un usuario (computador portátil), un cliente (switch de acceso), un servidor de Active Directory (Windows 2003) y un servidor AAA (SAAAS!). En este escenario el usuario se registra en la red LAN por medio de 802.1x, las credenciales del usuario residen en un Active Directory.

3.2.3.3. Análisis de Resultados

Para el análisis de los resultados se ha dividido el esquema en diferentes fases y en cada una de ellas se han estudiado los paquetes que han sido enviados.

Fase 1

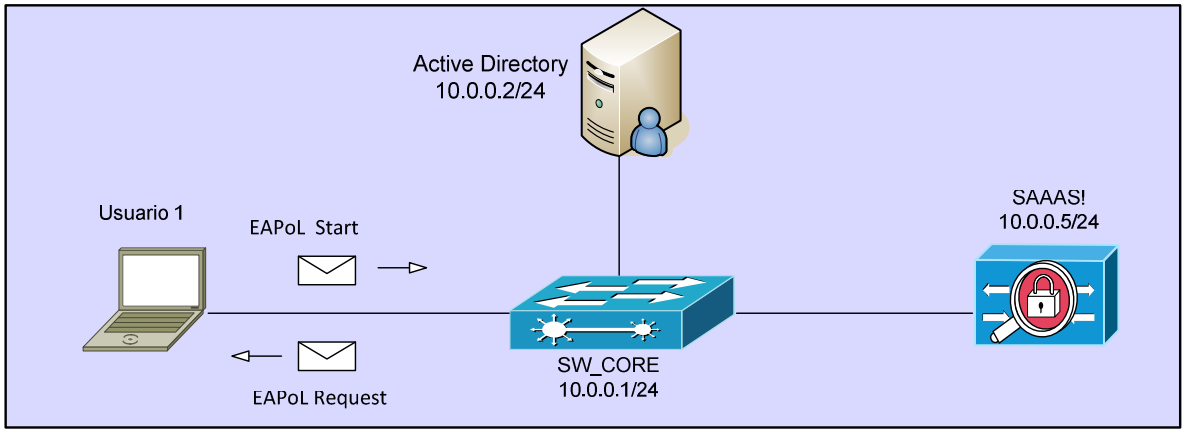


Figura 3.35 Esquema Fase 1

Como se muestra en la Figura 3.35, en la Fase 1, el usuario inicia el proceso con el paquete EAPoL, enviando en el paquete, en el atributo Type el valor binario 1 que equivale a la opción Start del proceso EAP como se presenta en la Figura 3.36.

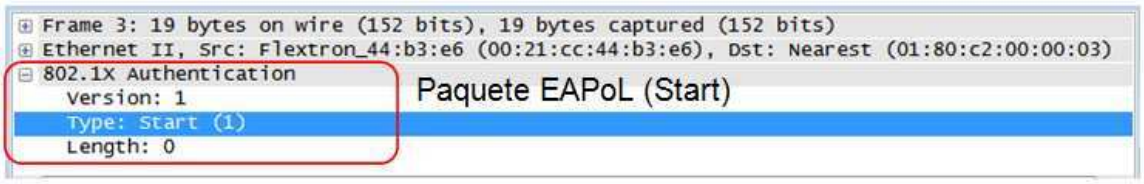


Figura 3.36 Paquete EAPoL (Start)

El cliente responde a este paquete con un EAPoL Request Identity indicándole que existe este servicio en la red e indicándole que envíe la identidad del usuario y la variante de EAP que va a utilizar como se muestra en la Figura 3.37.

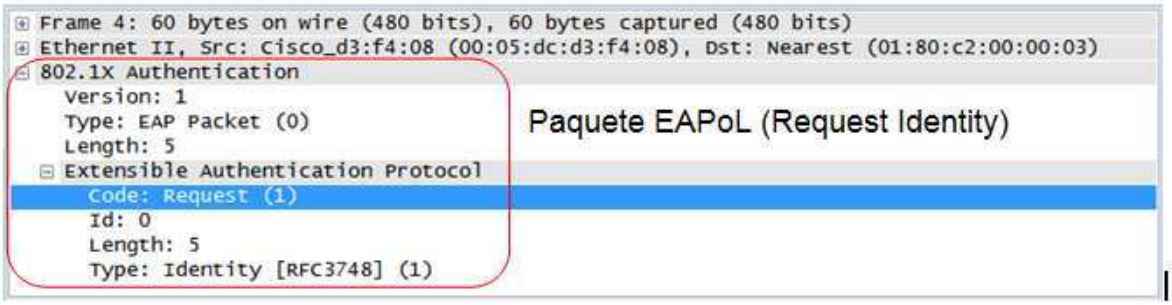


Figura 3.37 Paquete EAPoL (Request Identity)

Fase 2

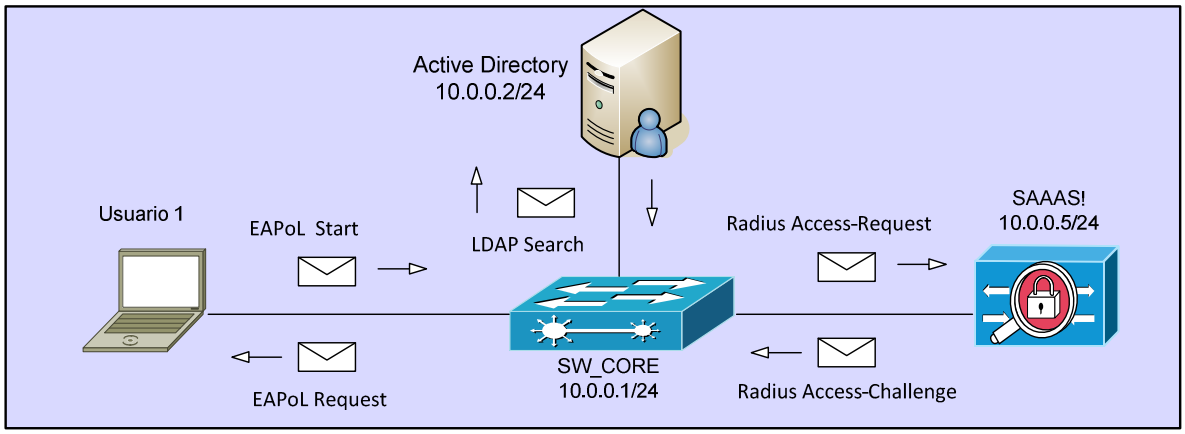


Figura 3.38 Esquema Fase 2

Como se aprecia en la Figura 3.38, en la Fase 2, el usuario envía una respuesta con el paquete EAPoL Response, enviando en el paquete, en el atributo Identity el valor del nombre de usuario para la autenticación como se presenta en la Figura 3.39.

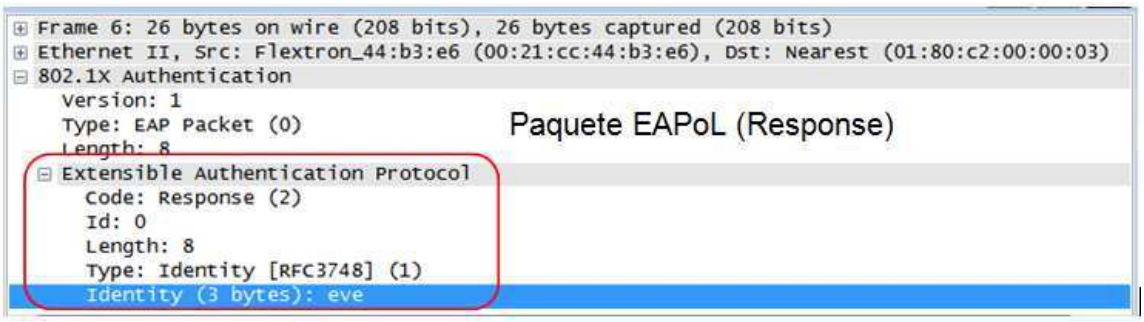


Figura 3.39 Paquete EAPoL (Response)

El cliente encapsula el paquete EAPoL Response dentro del protocolo RADIUS, adicionalmente añade atributos al paquete RADIUS correspondientes a la información sobre el cliente, este paquete es enviado hacia el servidor “SAAAS!” como se presenta en la Figura 3.40.

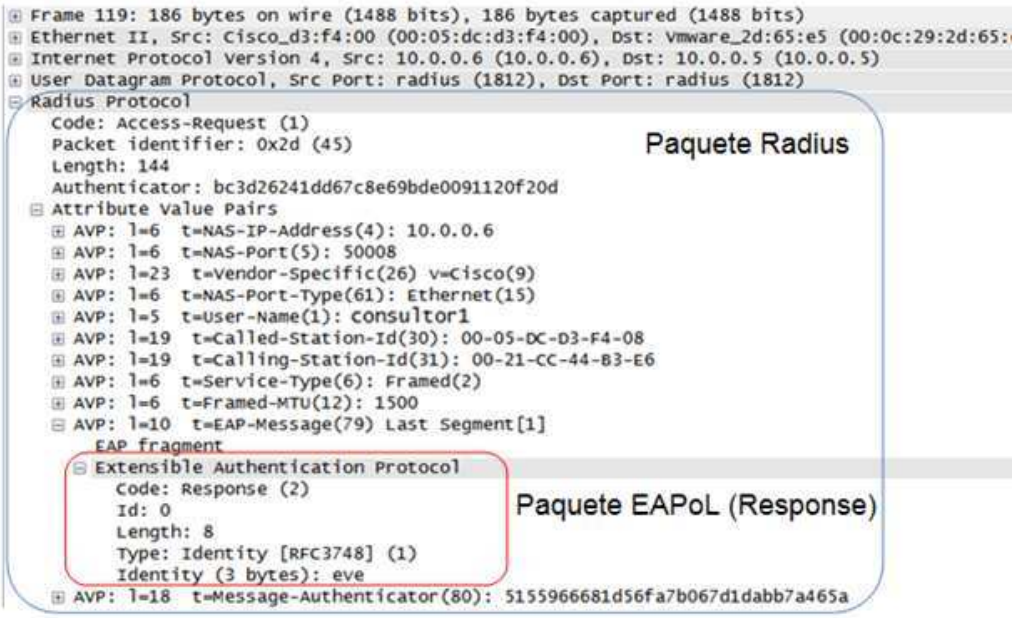


Figura 3.40 Paquete Radius Access-Request

El servidor “SAAAS!” recupera el paquete EAPoL Response y los atributos del paquete RADIUS, lee el atributo de identidad y revisa la existencia del usuario realizando una consulta a Active Directory este requerimiento lo realiza mediante el protocolo LDAP, si el usuario existe responde con un paquete Radius Access-Challenge, en donde envía un paquete EAPoL Request PEAP con el tipo de EAP a utilizar (PEAP) y la bandera para iniciar este tipo de conversación, encapsulándolo en un paquete RADIUS dirigido al cliente como se muestra en las figuras 3.41 y 3.42.

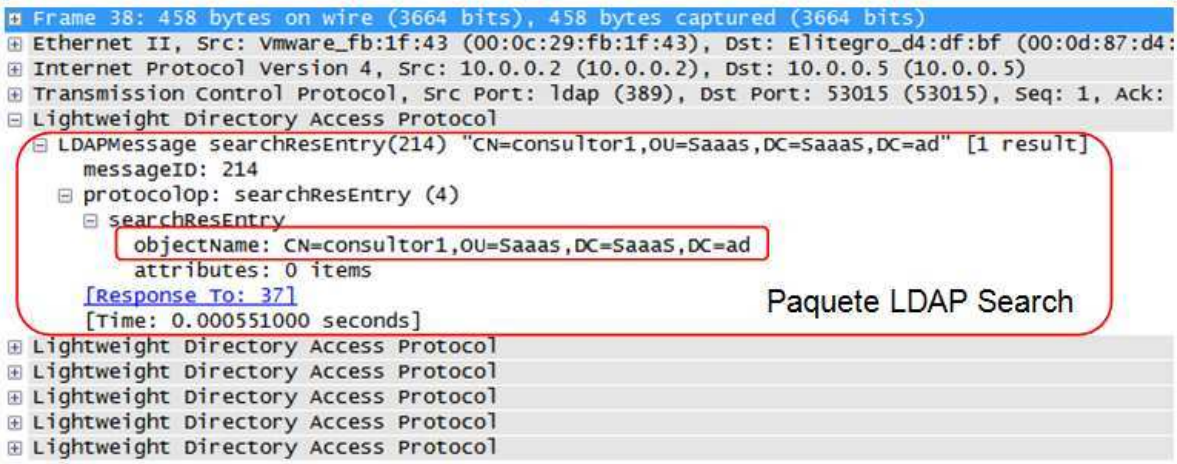


Figura 3.41 Paquete LDAP Search

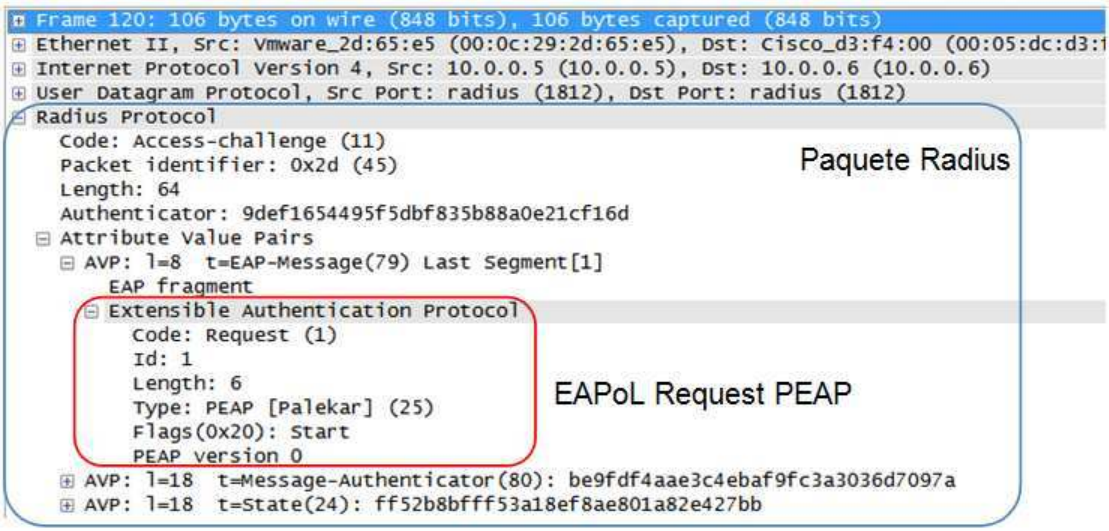


Figura 3.42 Paquete Radius Access - Challenge

El Cliente recupera el paquete EAPoL Request PEAP y lo reenvía hacia el usuario como se presenta en la Figura 3.43.

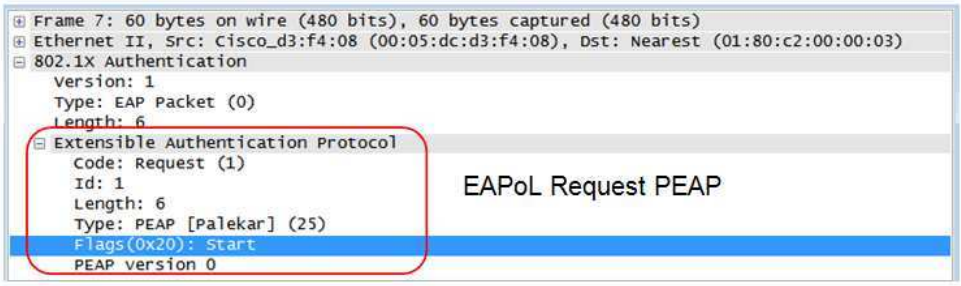


Figura 3.43 Paquete EAPoL Request PEAP

Fase 3

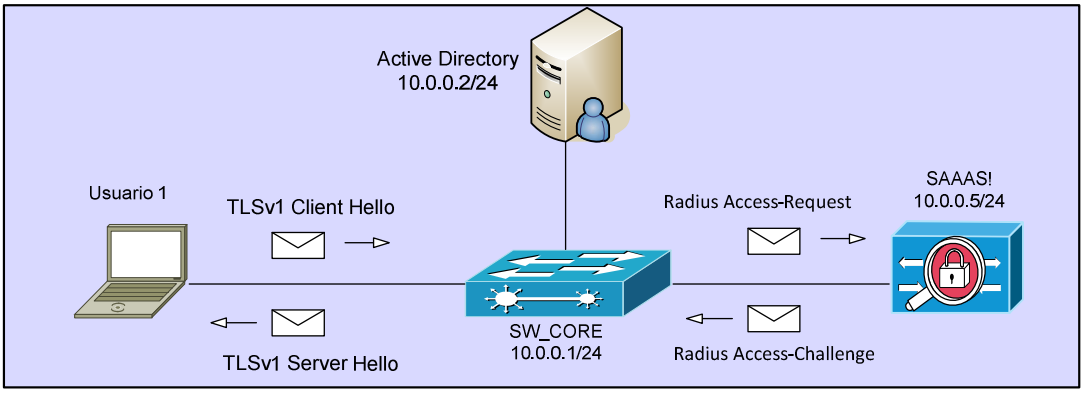


Figura 3.44 Esquema Fase 3

Como se aprecia en la Figura 3.44, en la Fase 3, el usuario inicia el intercambio de mensajes PEAP, basado en TLSv1 como se presenta en la Figura 3.45.

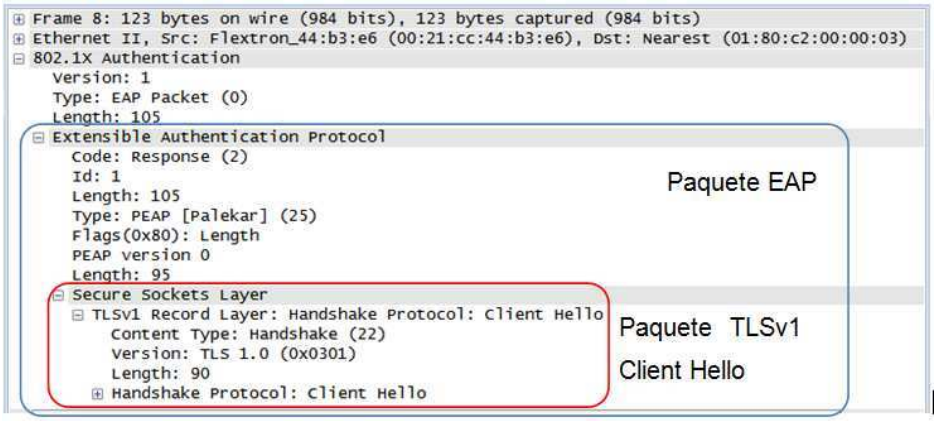


Figura 3.45 Paquete TLSv1 Client Hello

De igual manera que en la fase anterior el cliente encapsula los paquetes EAP entre el usuario y el servidor “SAAAS!” como se aprecia en la figura 3.46.

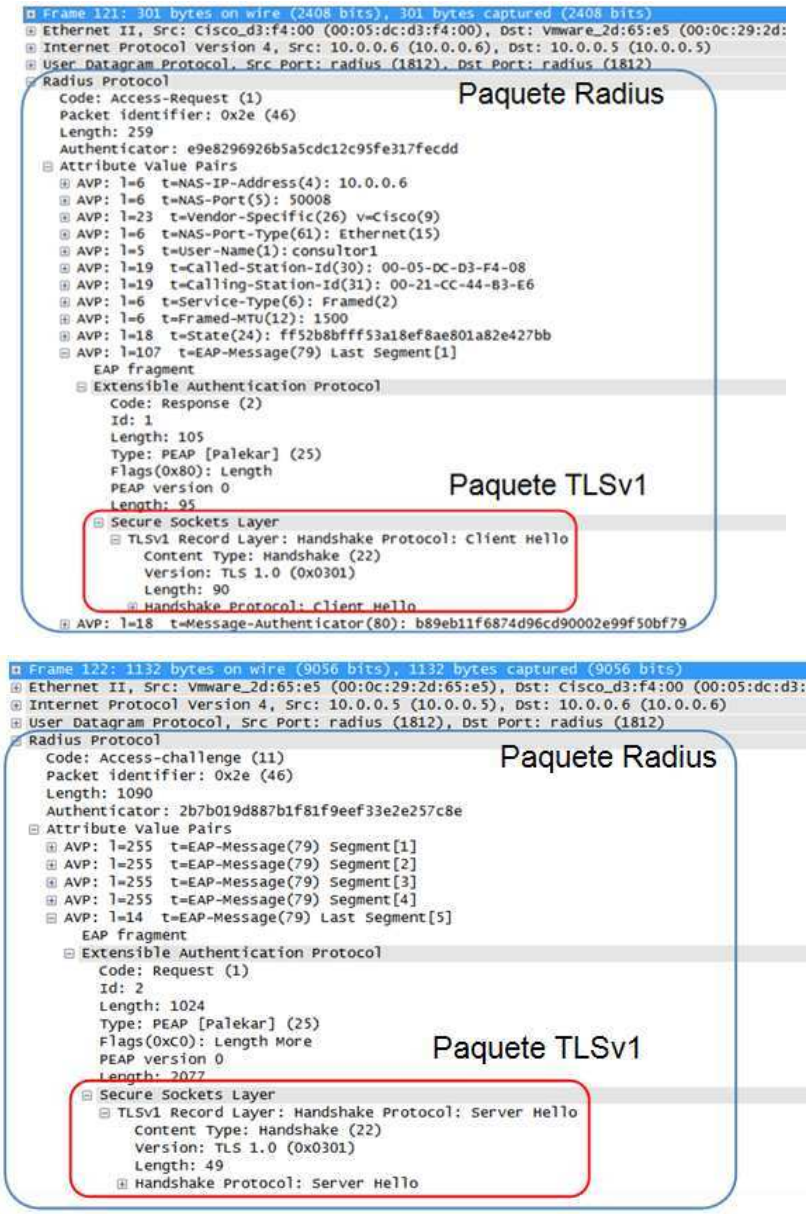


Figura 3.46 Paquetes TLSv1 Encapsulados sobre RADIUS

El cliente recupera el paquete EAPoL TLSv1 Server Hello y una secuencia de paquetes en los que el servidor “SAAAS!” envía el certificado para iniciar el túnel encriptado, estos paquetes son reenviados hacia el usuario para su aceptación o denegación como se presenta en la Figura 3.47.

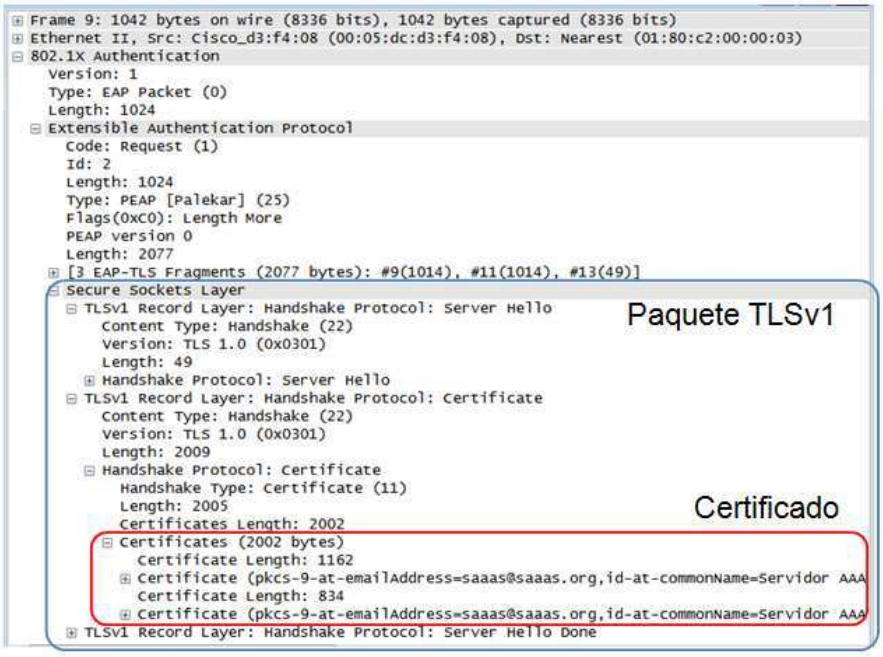


Figura 3.47 Paquete EAPoL TLSv1 Server Hello

Fase 4

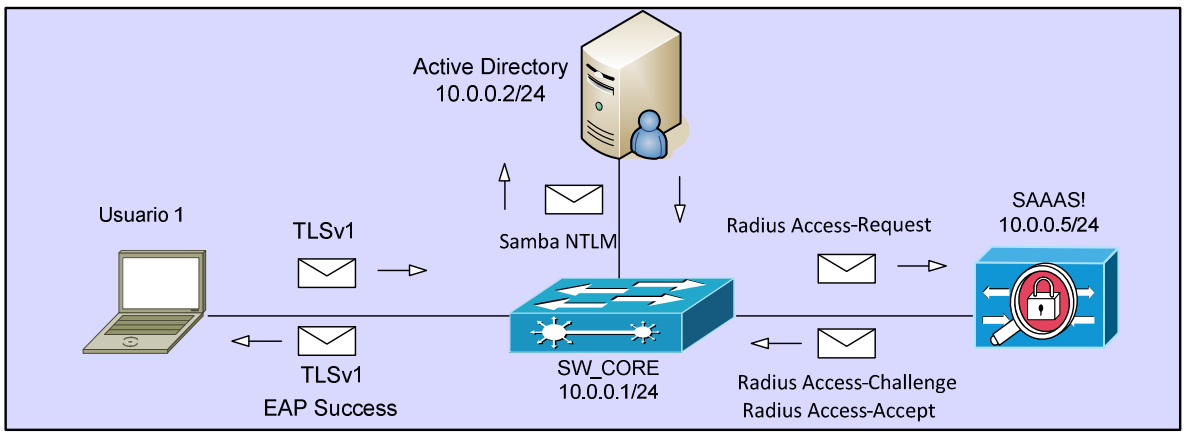


Figura 3.48 Esquema Fase 4

Como se muestra en la Figura 3.48, en la Fase 4, el usuario intercambia los mensajes de autenticación (contraseña) a través del método de autenticación MsCHAPv2 sobre un túnel encriptado (TLSv1) este intercambio se realiza con el servidor “SAAAS!” por lo que el cliente seguirá encapsulando estos mensajes sobre el protocolo RADIUS como en las fases anteriores, como se presenta en la Figura 3.49.

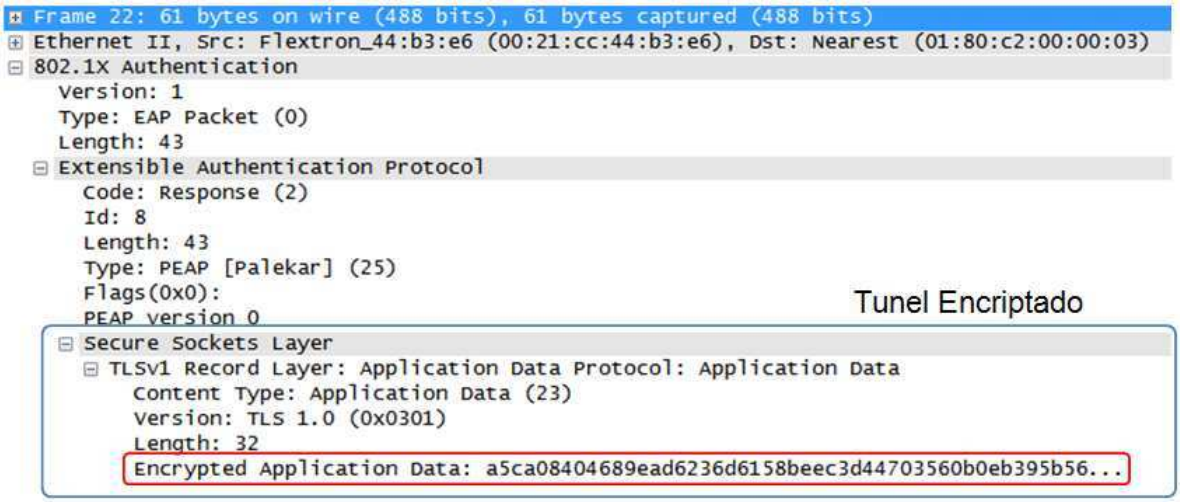


Figura 3.49 Paquete TLSv1 Túnel Encriptado

Una vez que el servidor “SAAAS!” reconozca al usuario como válido (por medio de un repositorio externo en este caso Active Directory mediante Samba) y recupere las políticas asociadas a éste, enviará un paquete Radius Access-Accept en el cual indica al cliente que el usuario está autorizado y las políticas que debe aplicarle a éste, adicionalmente encapsula el paquete EAP-Success que será enviado al usuario como se muestra en las figuras 3.50, 3.51 y 3.52.

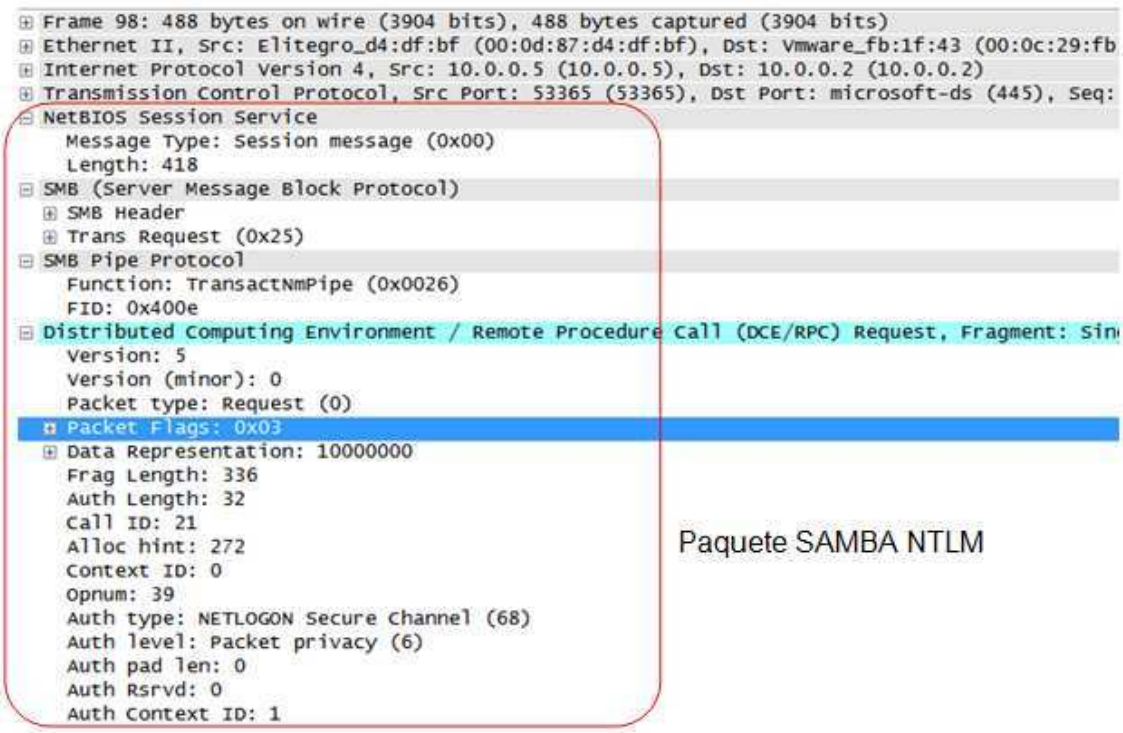


Figura 3.50 Paquete Samba NTLM

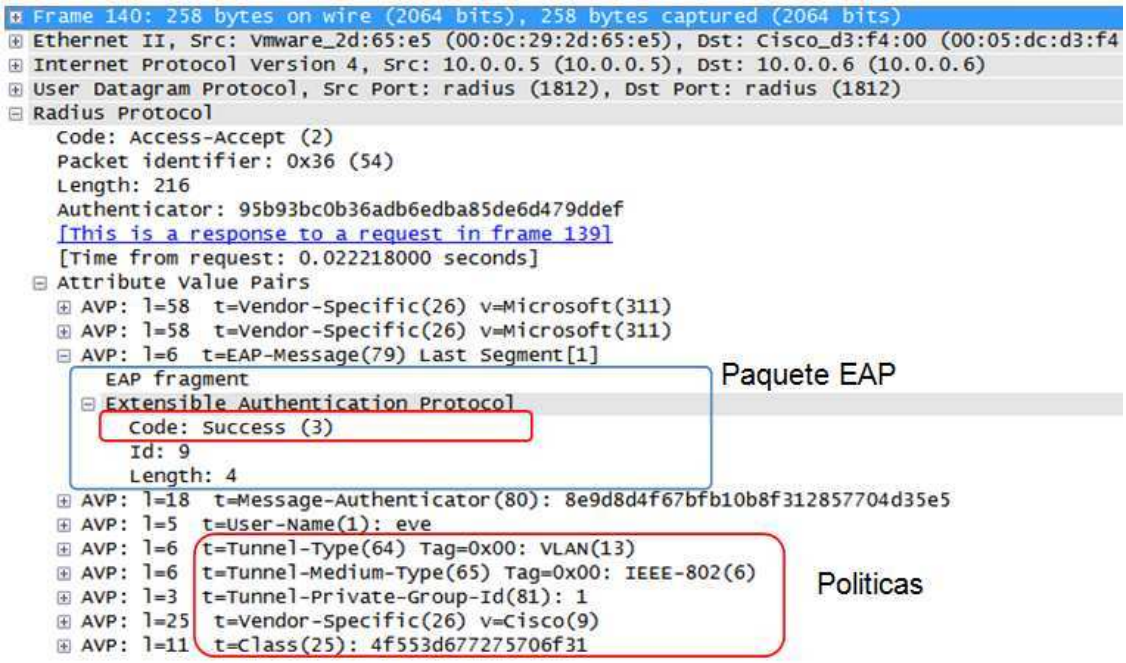


Figura 3.51 Paquete Radius Access-Accept

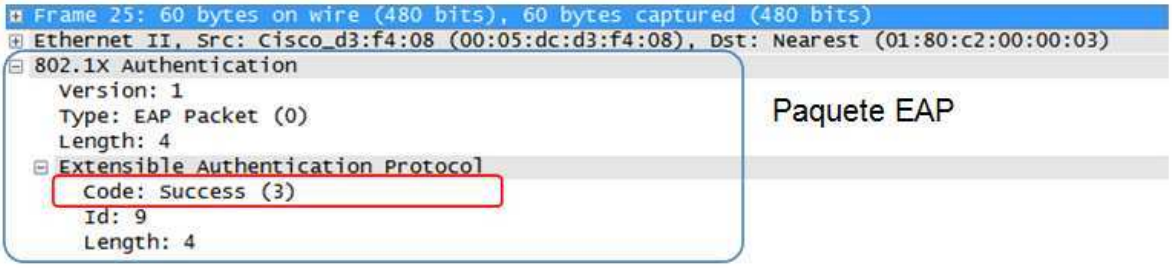


Figura 3.52 Paquete EAP-Success

CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES

En base al desarrollo e implementación del prototipo de Meta Distribución “SAAAS!” se presentan las siguientes conclusiones y recomendaciones.

4.1. CONCLUSIONES

- La tendencia actual de transmitir y almacenar información valiosa en medios electrónicos ha creado la necesidad de implementar esquemas de seguridad en las redes de datos sin importar el número de usuarios ni la cantidad de servicios a los que brinda acceso. Por esta razón, hoy en día la implementación de procesos de autenticación, autorización y registro del uso de la red se ha convertido en un paradigma para los Administradores de red.

- El resultado final de este proyecto de titulación es el Servidor de Autenticación, Autorización y Auditoría Simplificado “SAAAS!”, un prototipo de Meta Distribución Linux basada en Debian Squeeze que permite la implementación de servicios AAA mediante RADIUS en redes de datos de mediana escala que requieren soluciones avanzadas de seguridad de la red con bajos costos de implementación.

- Para el desarrollo de este proyecto se utilizó el protocolo RADIUS como base de la arquitectura AAA. La decisión de utilizar este protocolo se basó en la alta compatibilidad de los equipos de red actuales con el mismo y en la existencia de la herramienta FreeRADIUS, que es un conjunto de paquetes estable desarrollado para plataformas Linux/Unix.

- Se considera a “SAAAS!” una Meta Distribución por ser un sistema operativo basado en Debian Linux, personalizado para entornos AAA y cuyo proceso de arranque tiene lugar desde un CD-ROM (modo LIVE), si el usuario decide

instalar “SAAAS!”, esta instalación mantendrá todas configuraciones realizadas en el modo Live (cambios en la configuración de FreeRADIUS, instalación de nuevos paquetes, modificaciones en la base de datos). Además, se mantienen las herramientas necesarias para poder generar otra distribución a partir de esta instalación.

- Para el desarrollo del módulo de administración Web “SAAAS!” se usó AJAX (JavaScript Asíncrono + XML), para que la aplicación sea más interactiva y amigable para el usuario siguiendo la tendencia de la Web 2.0; ya que permite que una página Web solicite y reciba información del servidor en segundo plano de manera asíncrona sin interferir con la presentación de la página.

- El módulo de administración Web “SAAAS!” provee una solución rápida y eficaz para la implementación de servicios AAA en una red de datos al permitir un manejo flexible de repositorios externos y locales de credenciales para la autenticación de usuarios en la red. La aplicación Web ha sido probada con una base de datos local MySQL, un servidor Active Directory sobre Windows 2003 y un servidor OpenLDAP.

- Al momento de integrar soluciones de código abierto con servicios propietarios se debe tomar en consideración protocolos comunes y desarrollos de integración parcial. En el caso de este proyecto para la integración con Active Directory (Microsoft) se tuvo que ocupar dos herramientas: el servicio LDAP y el paquete NTLM controlado por el servicio Samba, que en conjunto permiten tener una gran compatibilidad con los atributos y métodos de autenticación de Active Directory.

- El servidor “SAAAS!” almacena los eventos de autenticación y autorización de los usuarios o dispositivos que han intentado acceder a la red y los muestra a

través de la aplicación Web, permitiendo que el administrador del servicio lleve a cabo un seguimiento que más tarde puede utilizarse en la administración de los recursos de red.

- Se puede implementar “SAAAS!” en redes pequeñas y medianas en las que se requiere limitar el acceso de los usuarios a información confidencial o a ciertos servicios de red, como Cooperativas de Ahorro en las que el acceso a herramientas financieras debe ser limitado para pocos usuarios, o clínicas pequeñas, en las que sólo los médicos tratantes deben tener acceso al historial de los pacientes.

4.2. RECOMENDACIONES

- La edición y lectura de archivos de configuración de Linux desde el módulo de administración Web, con varias pestañas editando el mismo archivo, puede convertirse en una tarea bastante compleja. Por esta razón al programar se recomienda crear mecanismos de organización como el uso de tablas auxiliares de una base de datos para superar estos problemas.

- La integración del servidor “SAAAS!” con el Active Directory de Microsoft es limitada debido a las restricciones de la versión actual del servicio SAMBA. Se ha anunciado que futuras versiones del paquete permitirán una mejor integración a nivel de manejo de credenciales (ejemplo: cambio directo de contraseñas utilizando EAP cuando las credenciales han caducado). Se recomienda la actualización del paquete SAMBA cuando se encuentren disponibles nuevas versiones o en su defecto, desarrollar una extensión del paquete que solucione este inconveniente.

- Los procesos de autenticación, autorización y registro del uso de la red pueden volverse una tarea no viable al momento de la implementación, debido

a las limitaciones de los equipos de red o a sistemas operativos descontinuados, por lo que se debe mantener un esquema básico de autenticación e ir escalando poco a poco de acuerdo a las funcionalidades del entorno AAA. Como se presenta en este proyecto una solución básica para los entornos AAA es la autenticación por dirección MAC.

- El servidor “SAAAS!” soporta la integración con equipos de varias marcas al tener la capacidad de manejar atributos propietarios y de ampliar su diccionario de atributos tanto como sea posible. Por esta razón, antes de iniciar una implementación se recomienda garantizar que los atributos a ser usados sean parte del estándar y los atributos propietarios sean ingresados en el diccionario del servidor.

- Cuando se realice la instalación del aplicativo Web “SAAAS!” se debe tomar en cuenta que el kernel del sistema base utilizado soporta arquitecturas de hardware comunes como i386 y amd64, por lo que si la instalación se va a realizar en equipos basados en arquitecturas diferentes se debe verificar la compatibilidad de los paquetes instalados con el aplicativo Web “SAAAS”.

- El proyecto aquí descrito es de código abierto y se distribuye bajo los términos de la licencia GNU GPL. Se recomienda a los usuarios adaptar el código fuente de esta solución a sus necesidades, utilizarlo, redistribuirlo, estudiarlo y mejorarlo de modo que toda la comunidad se beneficie, siguiendo los principios de colaboración del software libre.

- Se puede utilizar el proyecto aquí descrito como base para la generación de soluciones de software libre para la implementación de seguridad a nivel corporativo que además del servicio AAA permita la ejecución de remediación en los equipos de la red que no cumplan con las políticas de acceso, creando un esquema de control de acceso a la red (NAC) integrado con los servicios

AAA.

- Al momento de escoger una tecnología para el desarrollo Web, y principalmente el lenguaje que se ejecutará en el servidor se debe considerar que éste tenga las suficientes herramientas para interactuar con todos los servicios de la aplicación. En el caso de “SAAAS!” se escogió PHP por su gran desarrollo y el soporte de librerías para LDAP y MYSQL.

BIBLIOGRAFÍA

1. **Vollbrecht, J., y otros, y otros.** AAA Authorization Framework. *RFC 2904.* Agosto de 2000.
2. **LLoyd, B. y Simpson, W.** PPP Authentication Protocols. *RFC 1334.* Octubre de 1992.
3. **Simpson, W.** PPP Challenge Handshake Authentication Protocol (CHAP). *RFC 1994.* Agosto de 1996.
4. **Aboba, B. y Simon, D.** PPP EAP TLS Authentication Protocol. *RFC 2716.* Octubre de 1999.
5. **Cam-Winget, N., y otros, y otros.** The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST). *RFC 4851.* Mayo de 2007.
6. **Rigney, C., y otros, y otros.** Remote Authentication Dial In User Service (RADIUS). *RFC 2865.* Junio de 2000.
7. **Rigney, C.** RADIUS Accounting. *RFC 2866.* Junio de 2000.
8. **Finseth, C.** An Access Control Protocol, Sometimes Called TACACS. *RFC 1492.* Julio de 1993.
9. **Calhoun, P., y otros, y otros.** Diameter Base Protocol. *RFC 3588.* Septiembre de 2003.
10. **ITU-T Telecommunication Standarization Sector of ITU.** Information technology-Open systems interconnection-The Directory: Public-key and attribute certificate frameworks. *ITU-T Recommendation X.509:.* 11/2008.
11. **IEEE Computer Society.** IEEE Std 802.1X-2010: Port-Based Network Access Control. *IEEE Standard for Local and Metropolitan area Networks.* Nueva York : s.n., 2010.
12. **Yera, Angel Cobo.** *Estudio Científico de las Redes de Ordenadores.* Madrid : Vision Libros, 2009.

13. **Westerinen, A, y otros, y otros.** Terminology for Policy-Based Management. *RFC:3198.* 2001.
14. **Fielding, R., y otros, y otros.** Hypertext Transfer Protocol -- HTTP/1.1. *RFC: 2616.* 1999.

Páginas Web:

1. **Garrett, Jesse James.** *Ajax: A New Approach to Web Applications.* 2005, http://www.adaptivepath.com/ideas/ajax-new-approach-Web-applications [Citado el: 2 de Enero de 2011.]
2. **Tirado, Rafael Martín de Agar.** CALIU:Asociación de Usuario de GNU/Linux en Lengua Catalana. *Analizando en profundidad el proyecto METADISTROS.* [En línea] Septiembre de 2003. [Citado el: 1 de Junio de 2011.] http://ftp.caliu.cat/pub/caliu/distros/doc/sistemas_dw_0.3.0.pdf.
3. **Charles P. Wright, Erez Zadok.** Linux Journal. *Kernel Korner-Unionfs:Bringing Filesystems toguether.* [En línea] 01 de 12 de 2004. [Citado el: 04 de 06 de 2011.] http://www.linuxjournal.com/article/7714?page=0,0.
4. **González, Sergio Gonzáles.** Creando un CDROM autoejecutable con ISOLINUX. [En línea] [Citado el: 04 de 06 de 2011.] http://linuxupc.upc.es/~kampa/documentos/isolinux/index.html.
5. **Anvin, Peter.** Syslinux. *Comboot/menu.c32.* [En línea] 27 de Enero de 2011. [Citado el: 20 de Junio de 2011.] http://syslinux.zytor.com/wiki/index.php/Comboot/menu.c32.
6. **Bennett, Colin.** GNU GRUB Graphical Menu. [En línea] 29 de Noviembre de 2009. [Citado el: 21 de Enero de 2011.] http://grub.gibibit.com/Journal.
7. **ICS Yaco, S.L y Emergya S. Coop. And.** El Proyecto Guadalinex. *Infraestructura del Proyecto de Sotfware Libre impulsado por la Junta de Andalucía.* [En línea] 23 de Enero de 2004. [Citado el: 5 de Marzo de 2011.] http://speeches.ofset.org/jrfernandez/rmll2004/fuentes/proyguadalinex.pdf.
8. **Samba Team.** HowTo-Collection. *Winbind.* [En línea] [Citado el: 28 de

Diciembre de 2010.] http://www.samba.org/samba/docs/man/Samba-HOWTO-Collection/winbind.html.

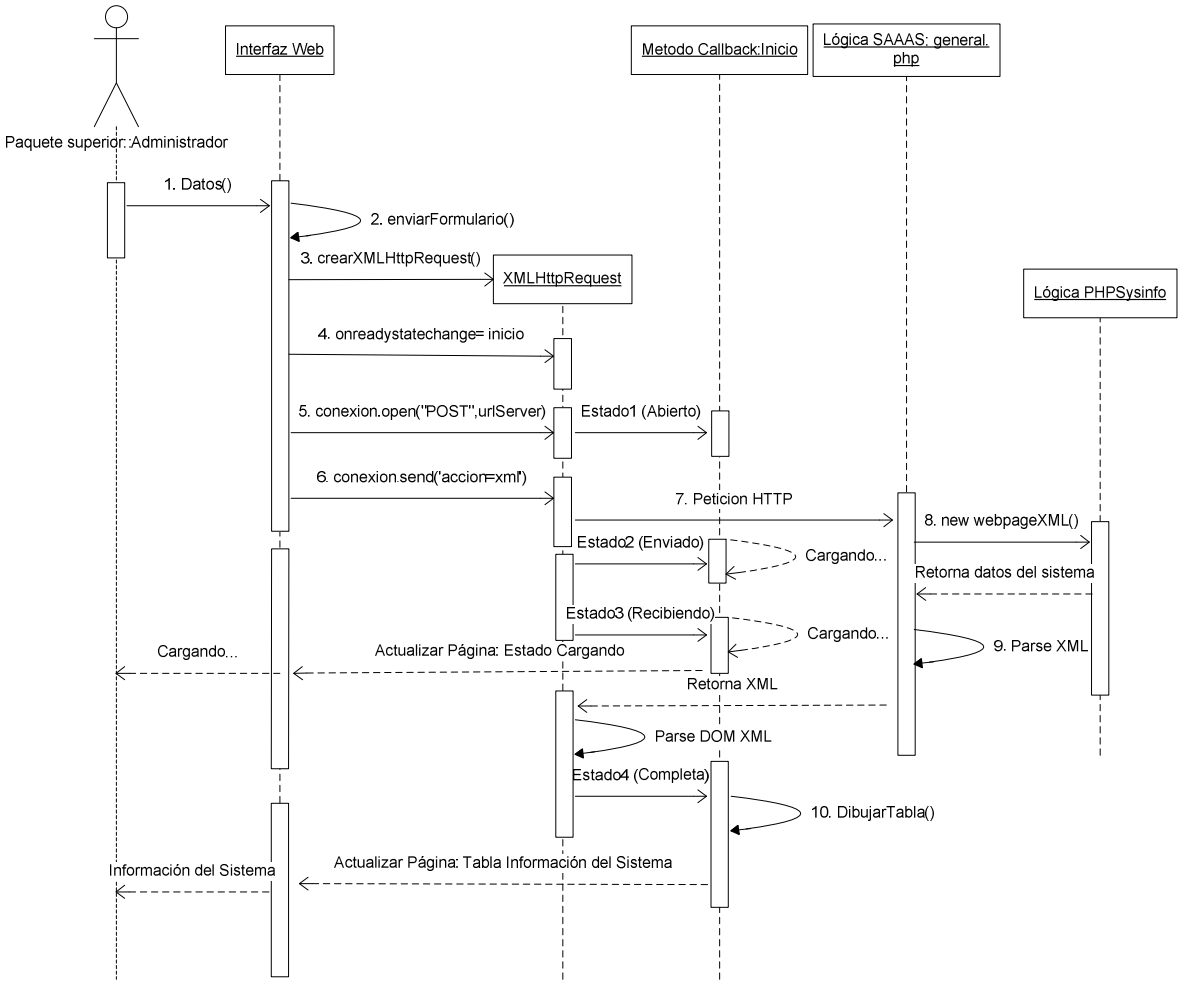
9. **Comunidad de Usuarios Guadalinex.** Initramfs:¿Qué es? [En línea] 12 de Julio de 2010. [Citado el: 4 de Junio de 2011.] http://forja.guadalinex.org/Webs/guadalinexv5/doku.php?id=initramfs:que_es.
10. **Pavlov, Artemiy I.** SquashFS HOWTO. [En línea] 24 de Julio de 2008. [Citado el: 4 de Junio de 2011.] http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/pdf/SquashFS-HOWTO.pdf.

Anexos

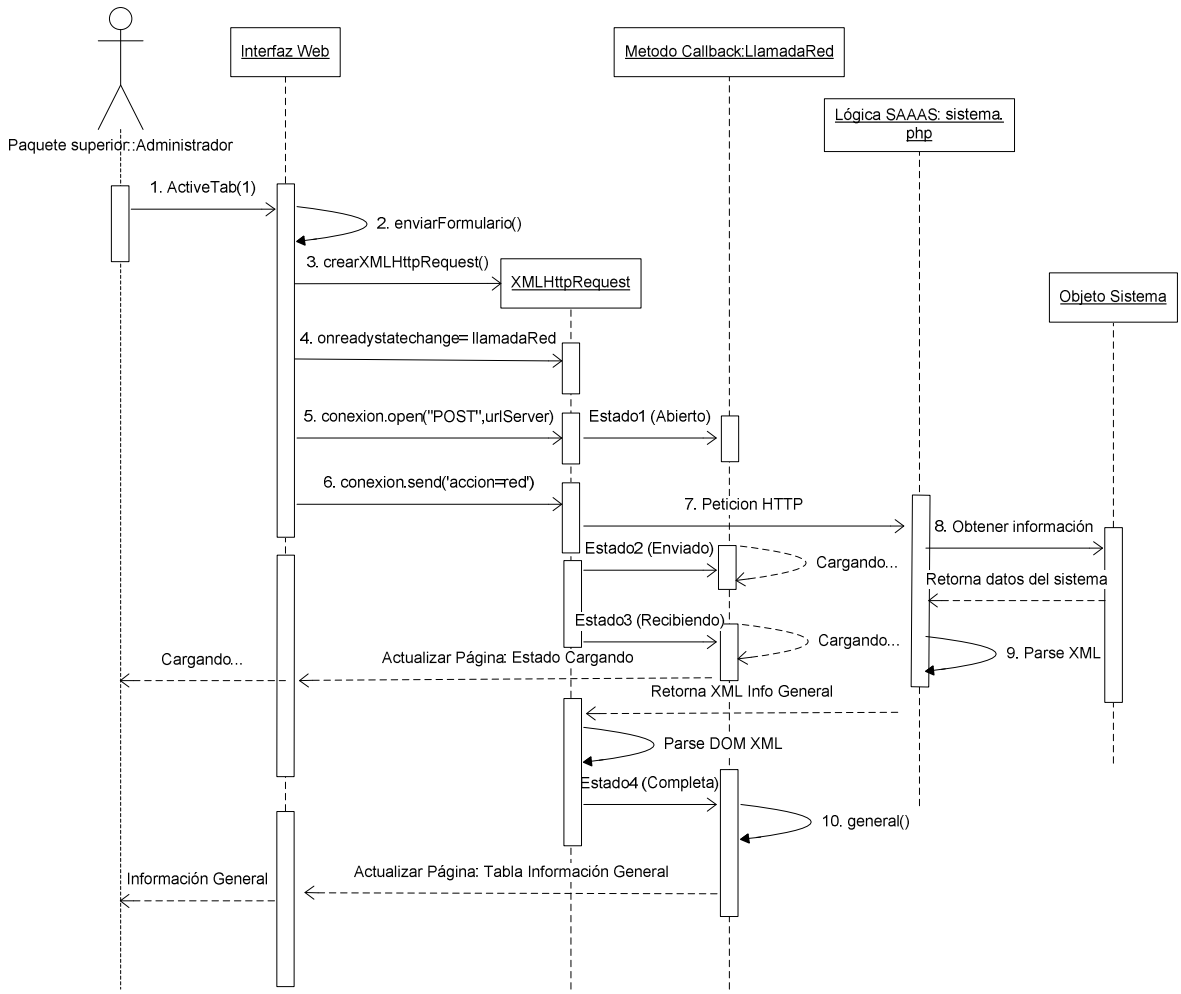
ANEXO A

Diagramas de Secuencia de la Aplicación “SAAAS!”

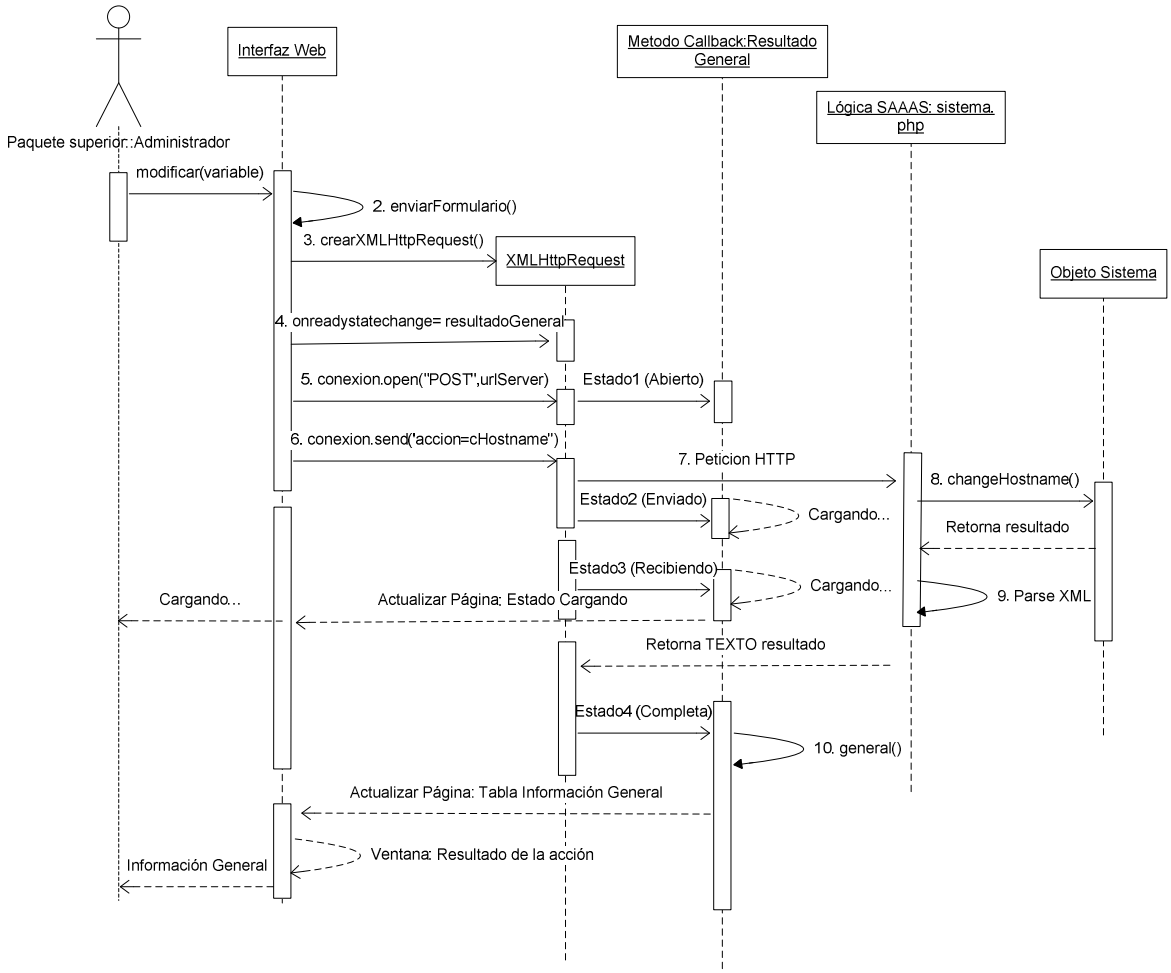
A1. Diagrama de Secuencia para la Acción: Visualizar Información Básica del Sistema



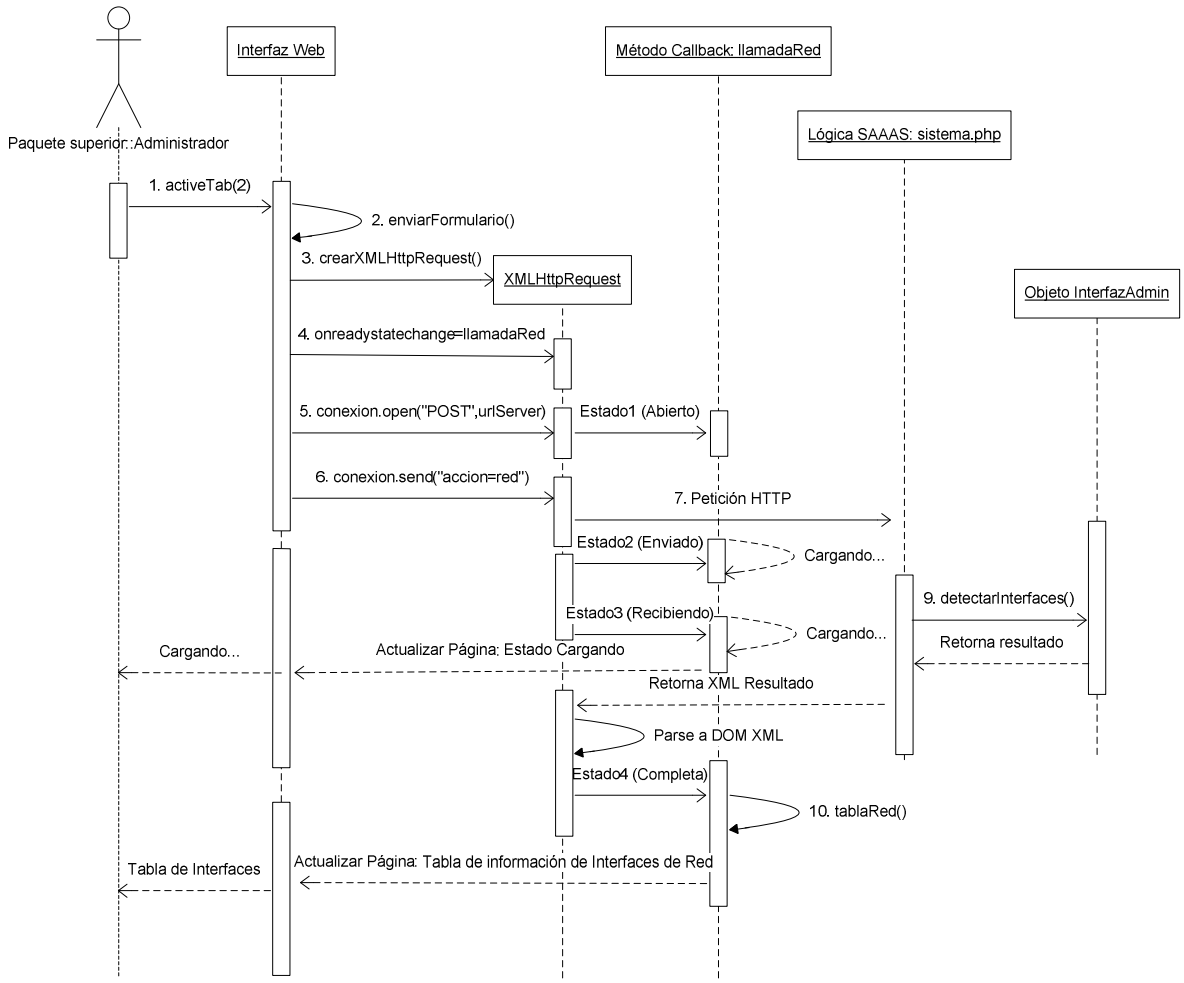
A2. Diagrama de Secuencia para la Acción: Visualizar información Básica de Red



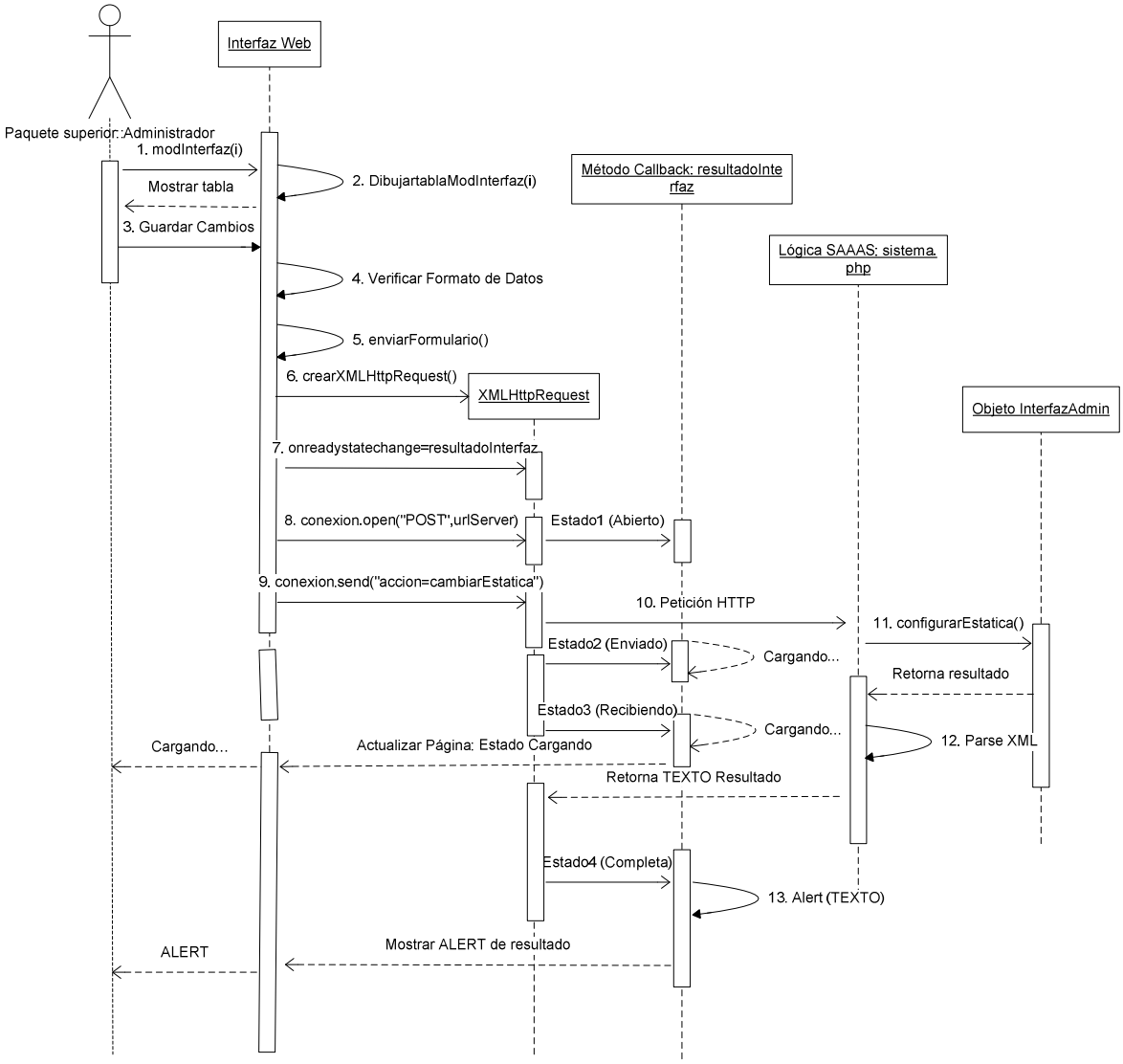
A3. Diagrama de Secuencia para la Acción: Modificar Información Básica de Red (Ejemplo: Cambiar el nombre de Host)



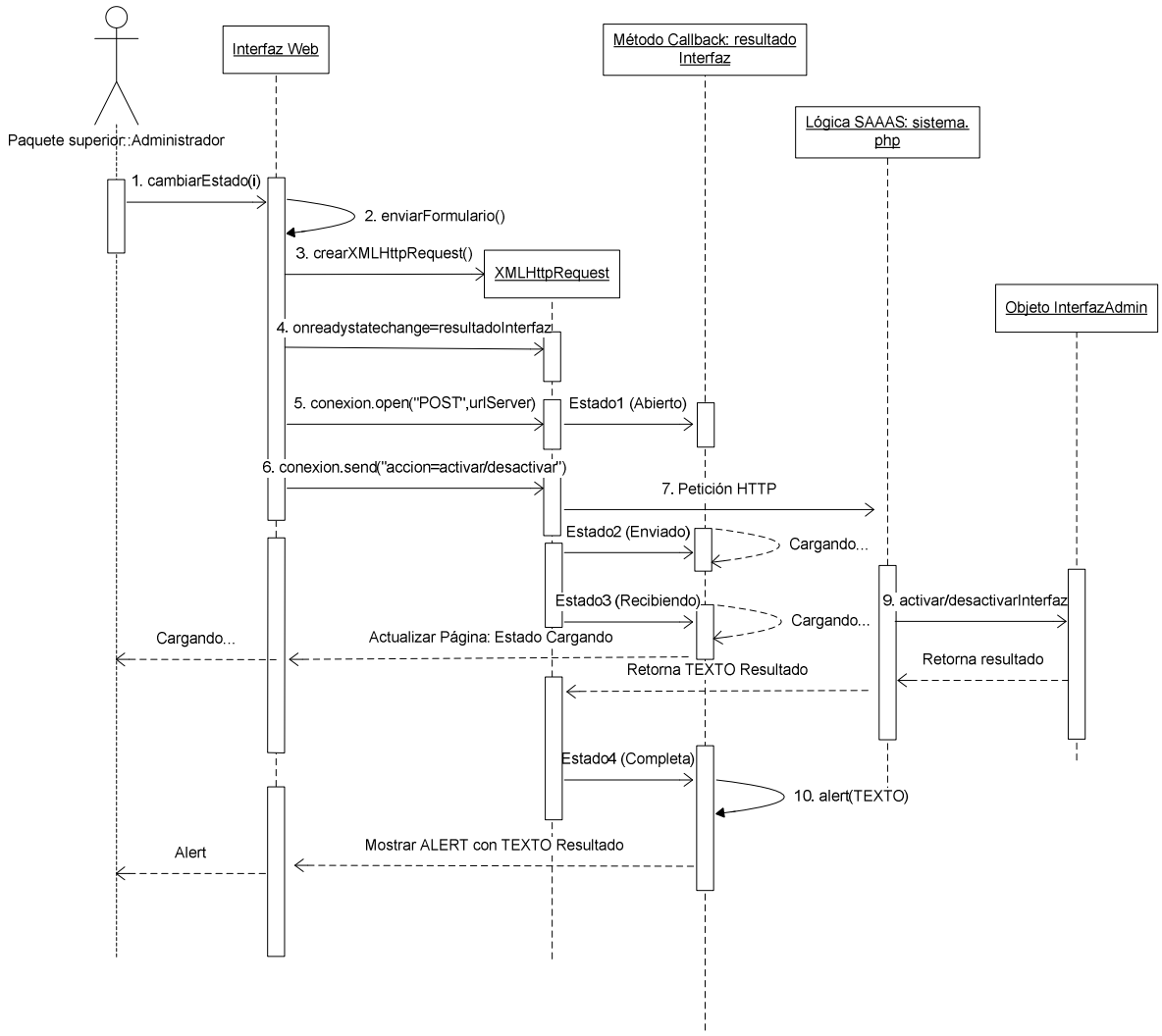
A4. Diagrama de Secuencia para la Acción: Visualizar interfaces de Red



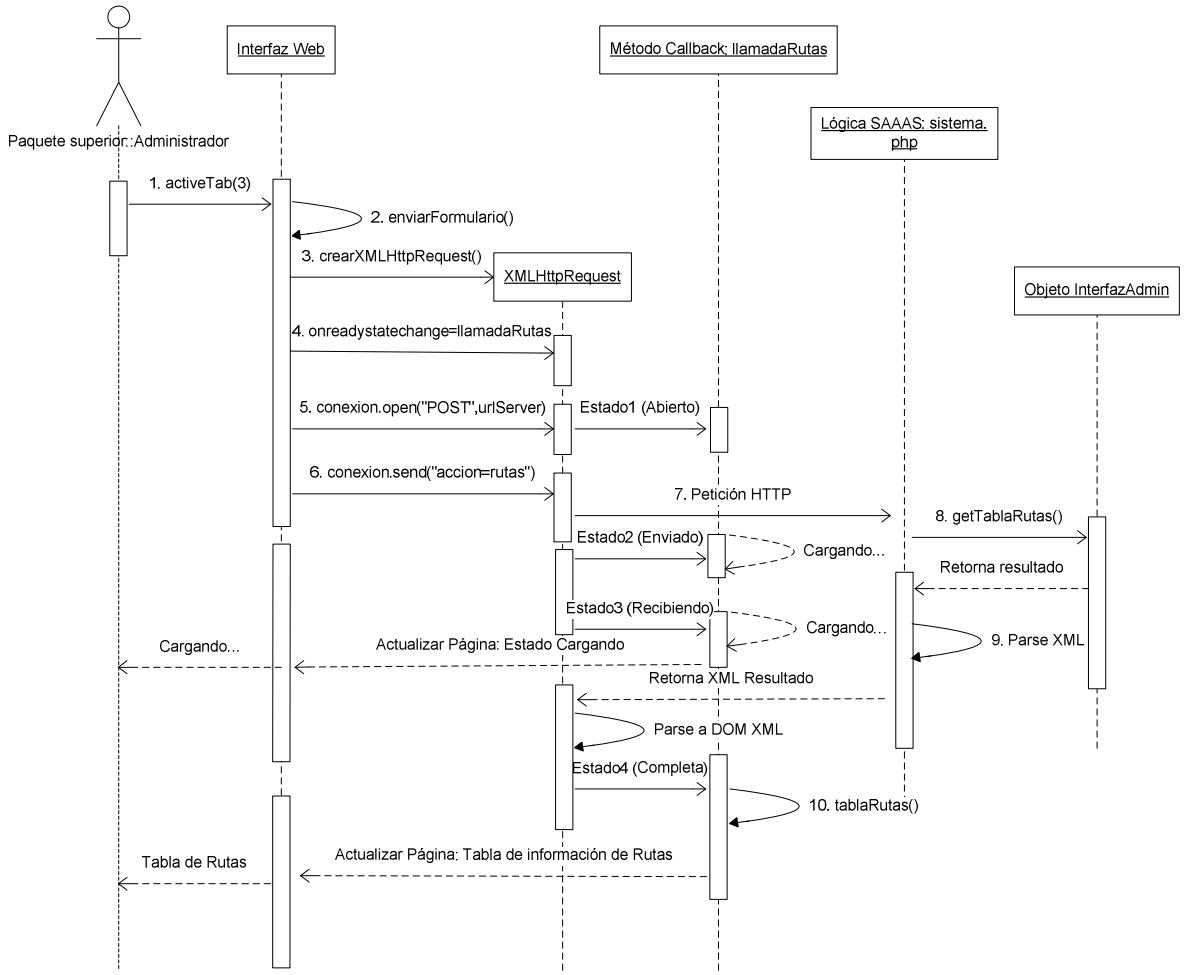
A5. Diagrama de Secuencia para la Acción: Modificar una interfaz de Red (Ejemplo: Cambiar direccionamiento estático)



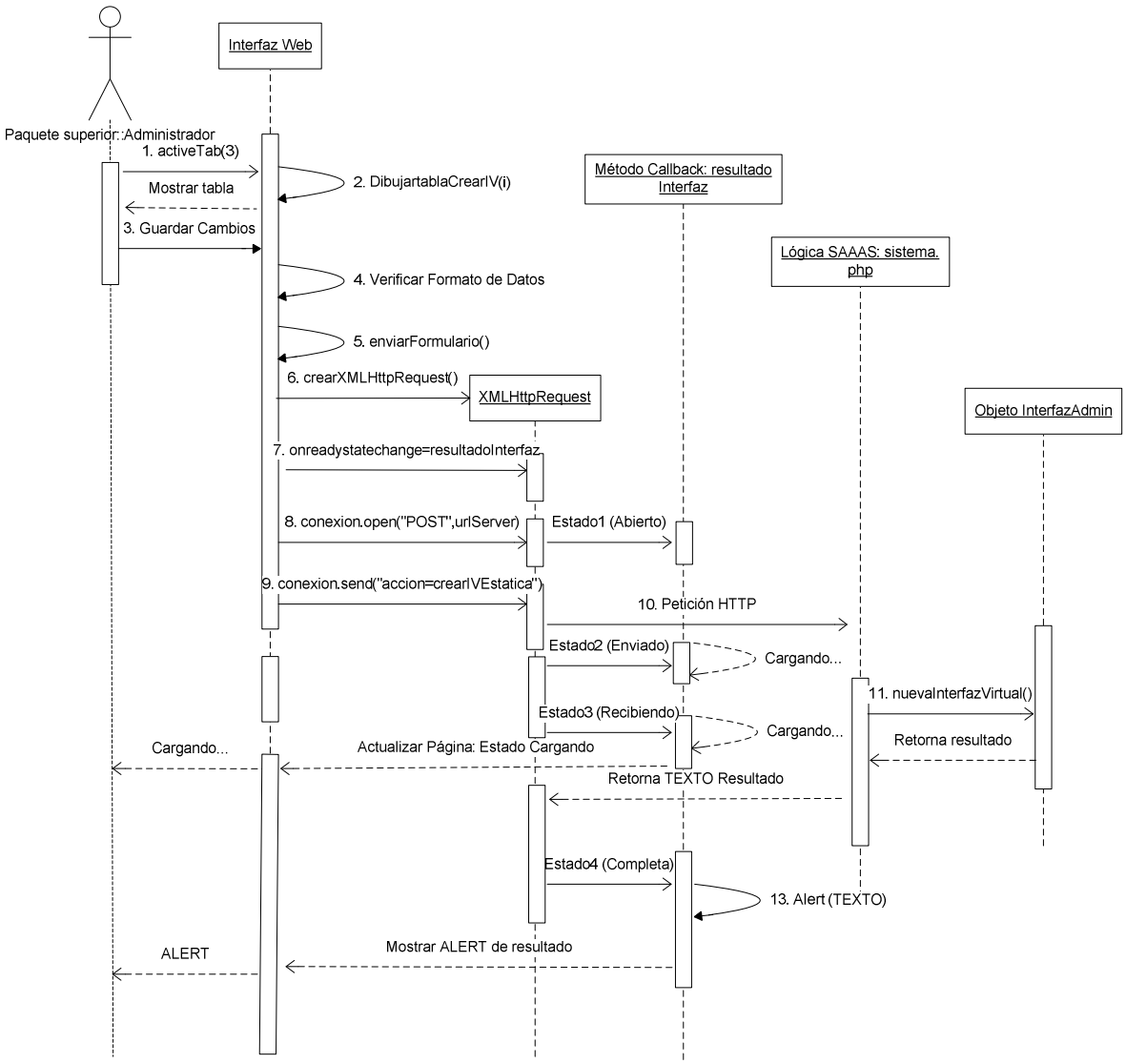
A6. Diagrama de Secuencia para la Acción: Activar/Desactivar una interfaz de Red



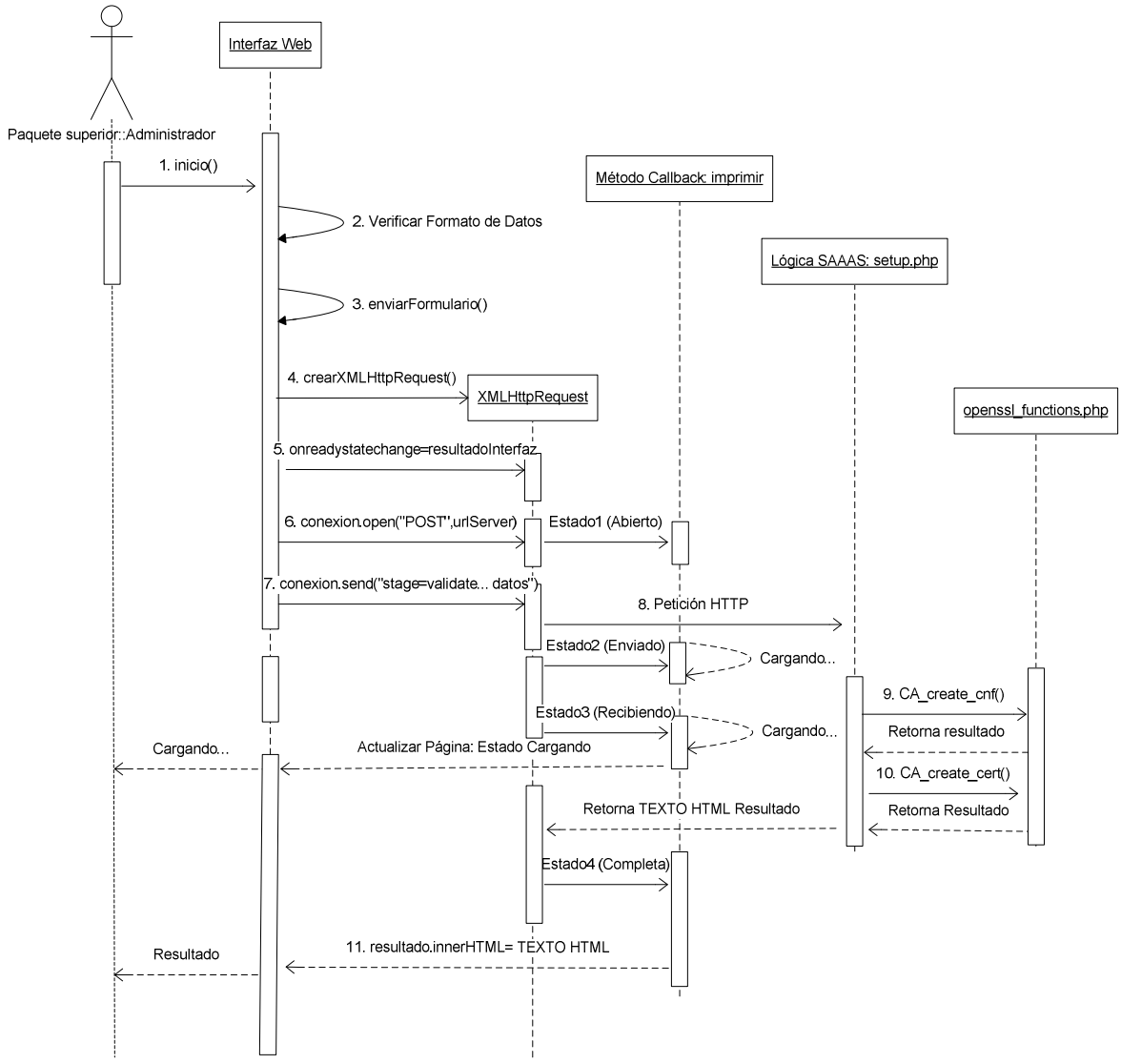
A7. Diagrama de Secuencia para la Acción: Visualizar la tabla de Rutas



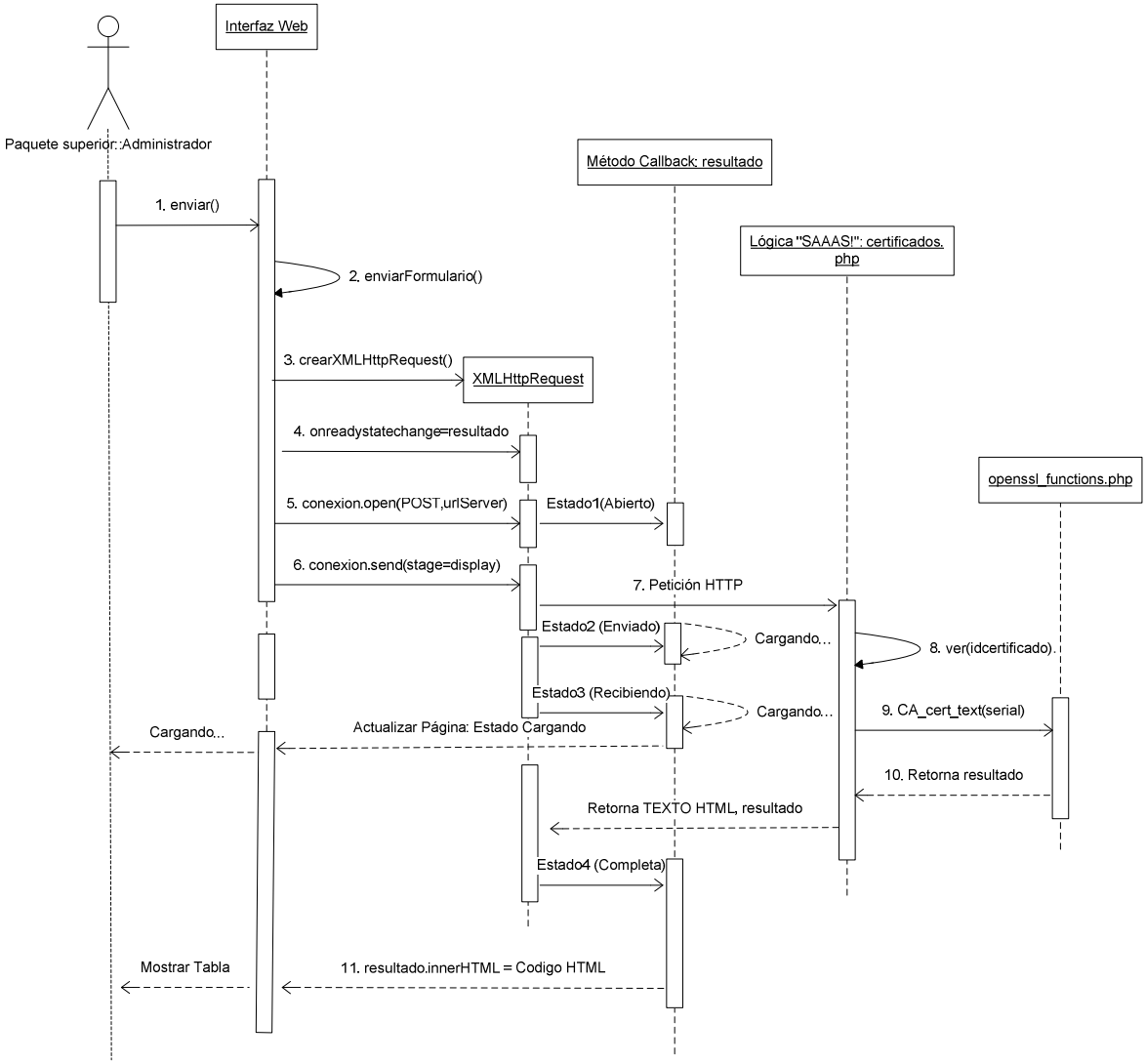
A8. Diagrama de Secuencia para la Acción: Crear una Interfaz Virtual



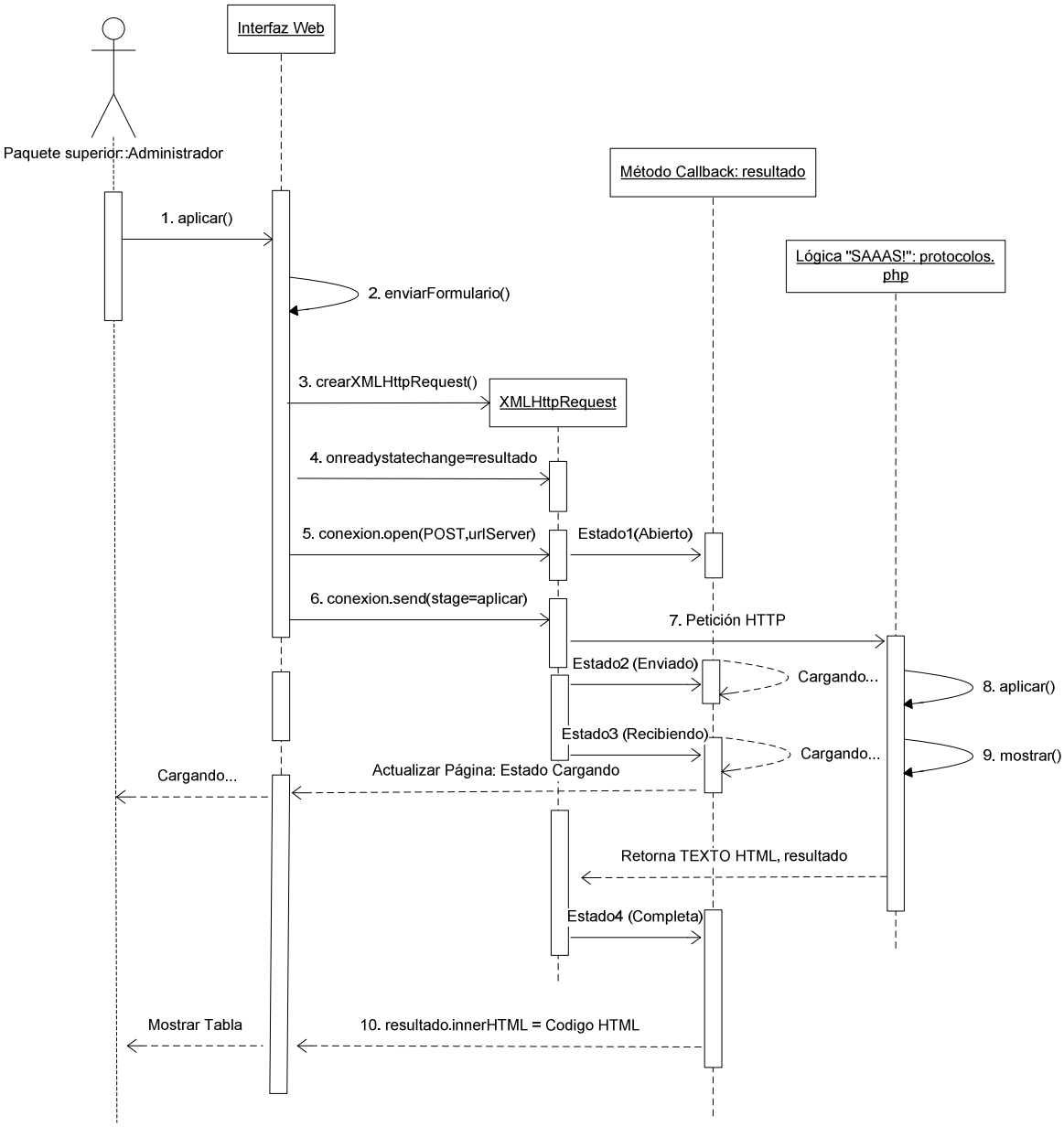
A9. Diagrama de Secuencia para la Acción: Configurar una Autoridad Certificadora



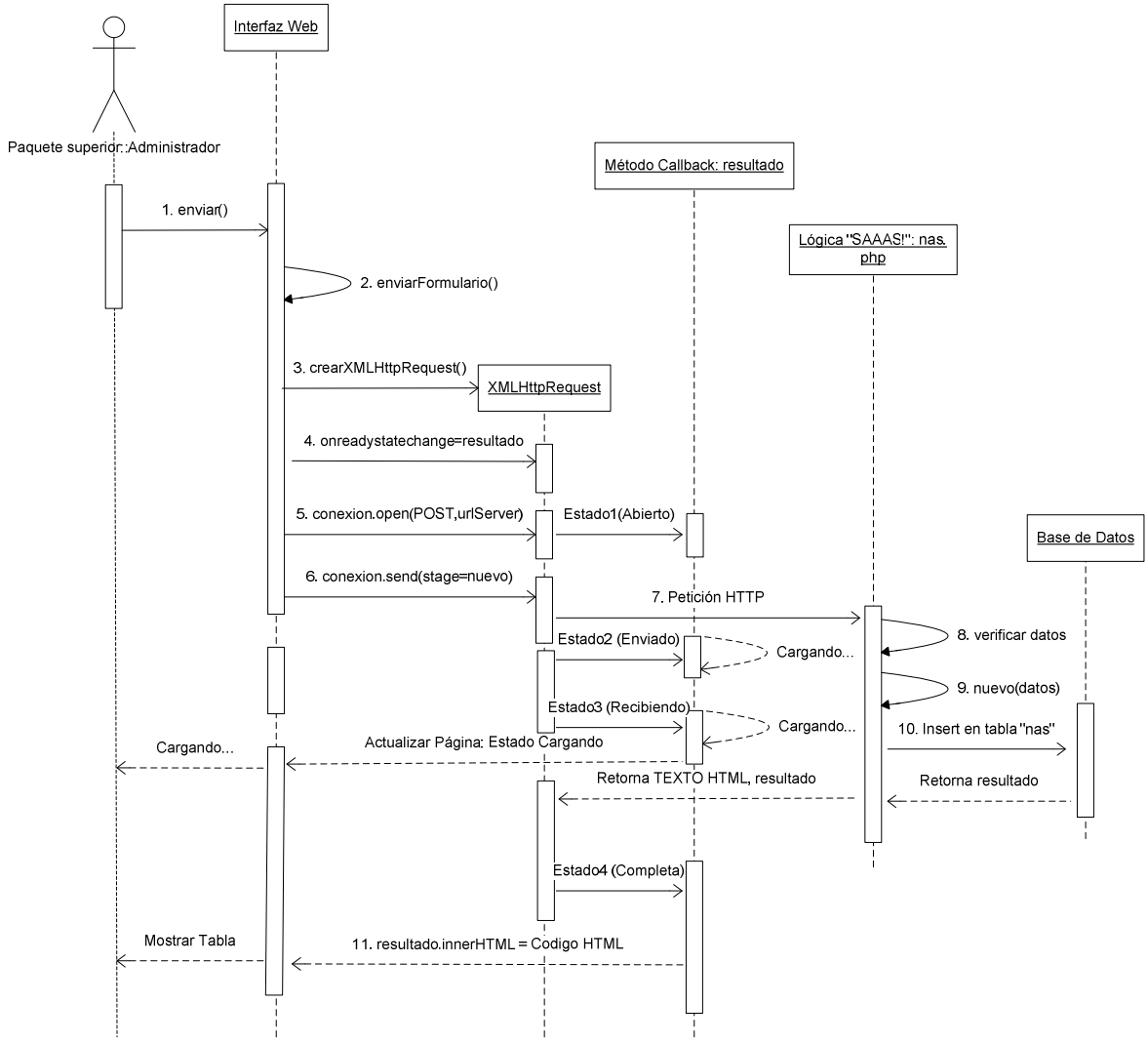
A10. Diagrama de Secuencia para la Acción: Manipulación de Certificados (Ejemplo: Ver certificado)



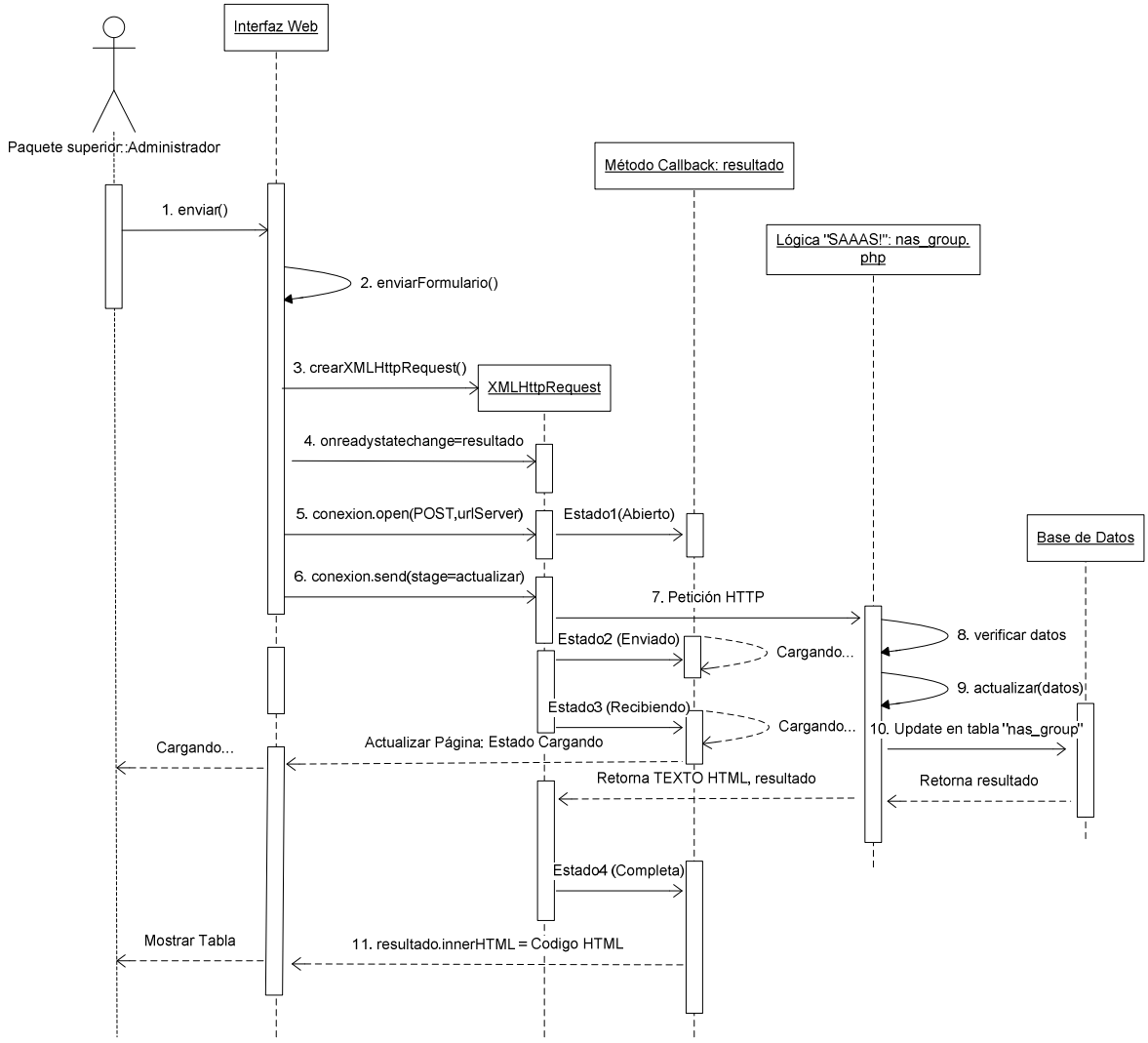
A11. Diagrama de Secuencia para la Acción: Configuración de Protocolos de Autenticación



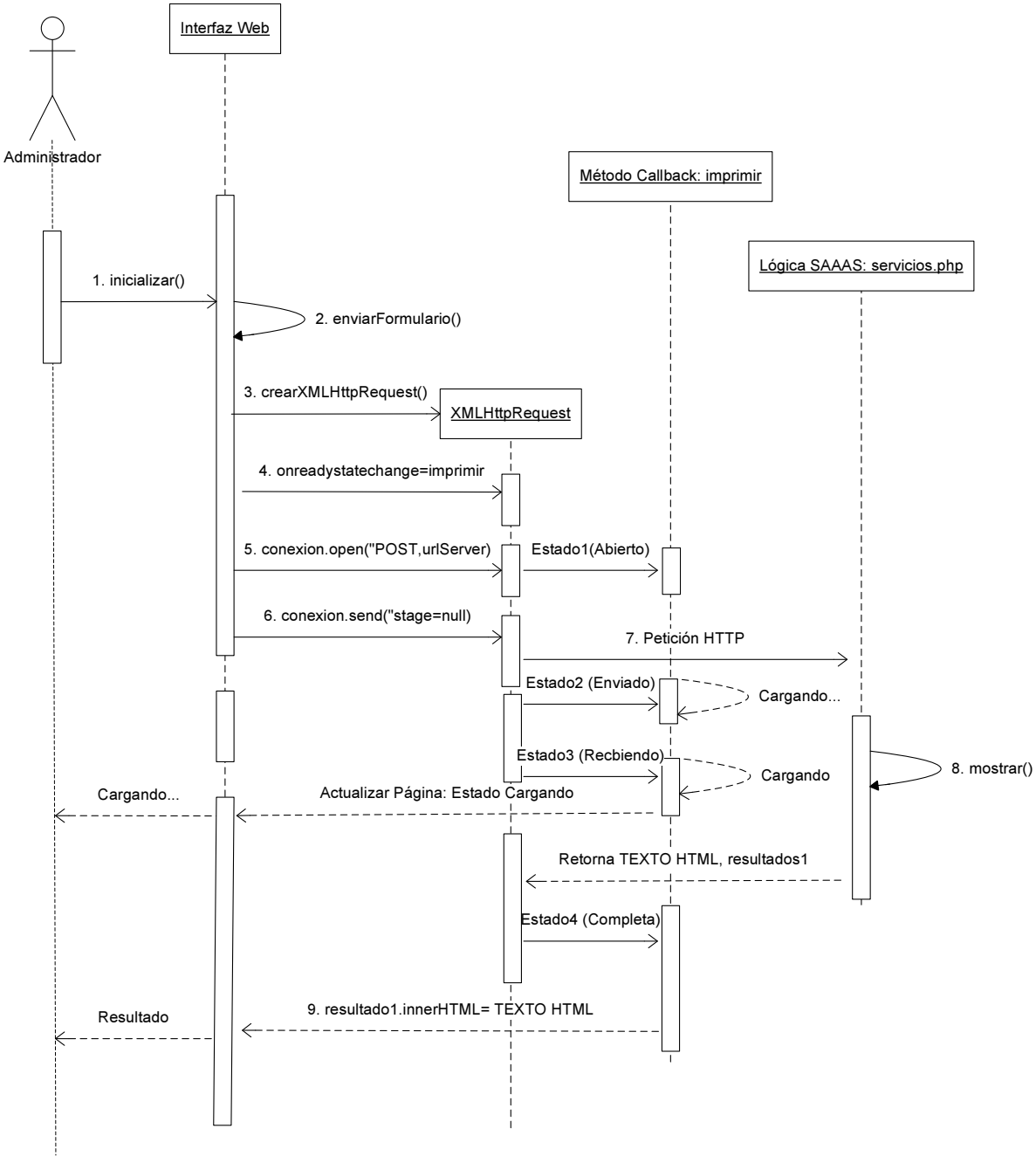
A12. Diagrama de Secuencia para la Acción: Manipulación de Clientes RADIUS (Ejemplo: Agregar Cliente)



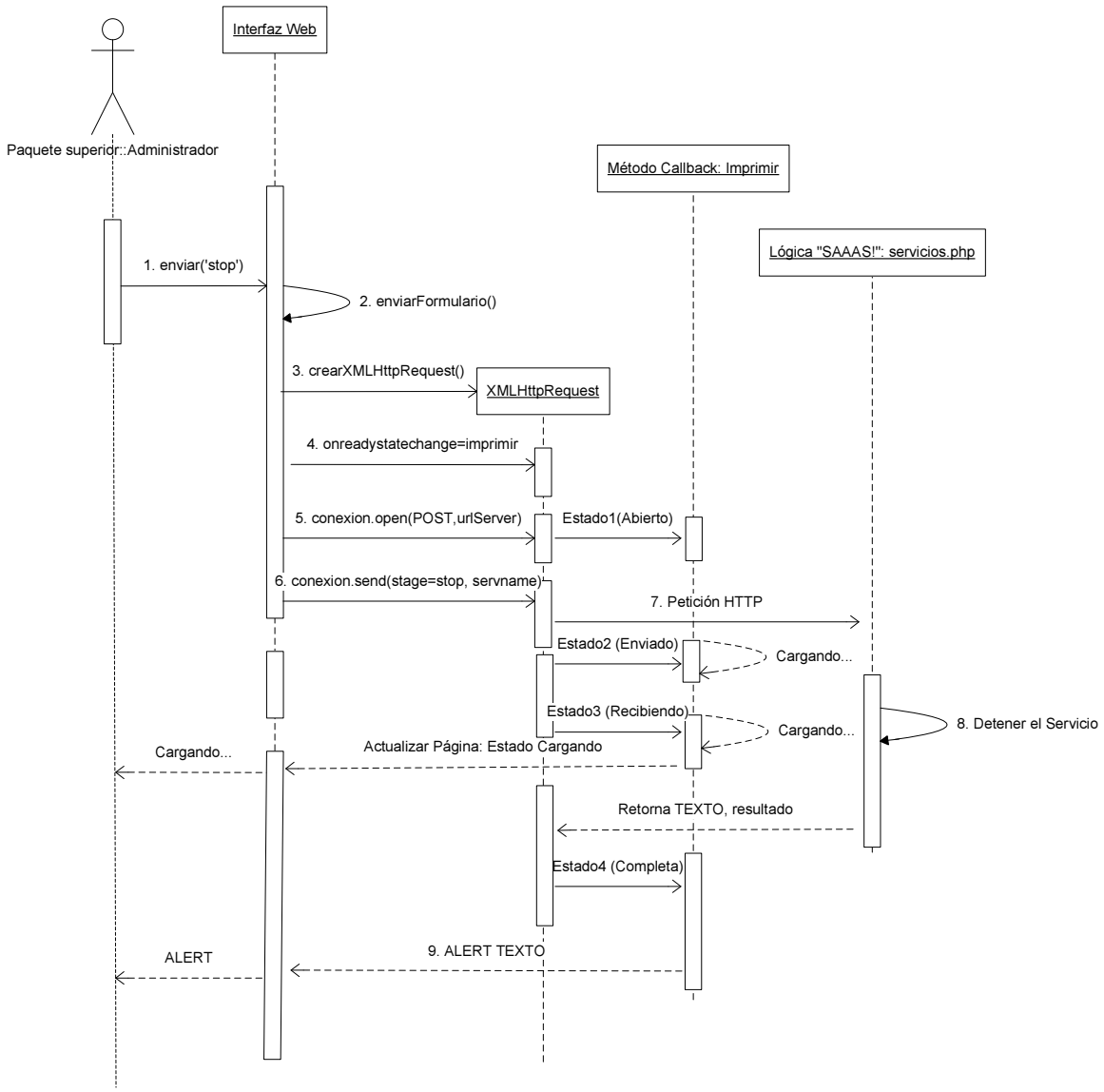
A13. Diagrama de Secuencia para la Acción: Manipulación de Grupos de Clientes RADIUS (Ejemplo: Modificar Grupo)



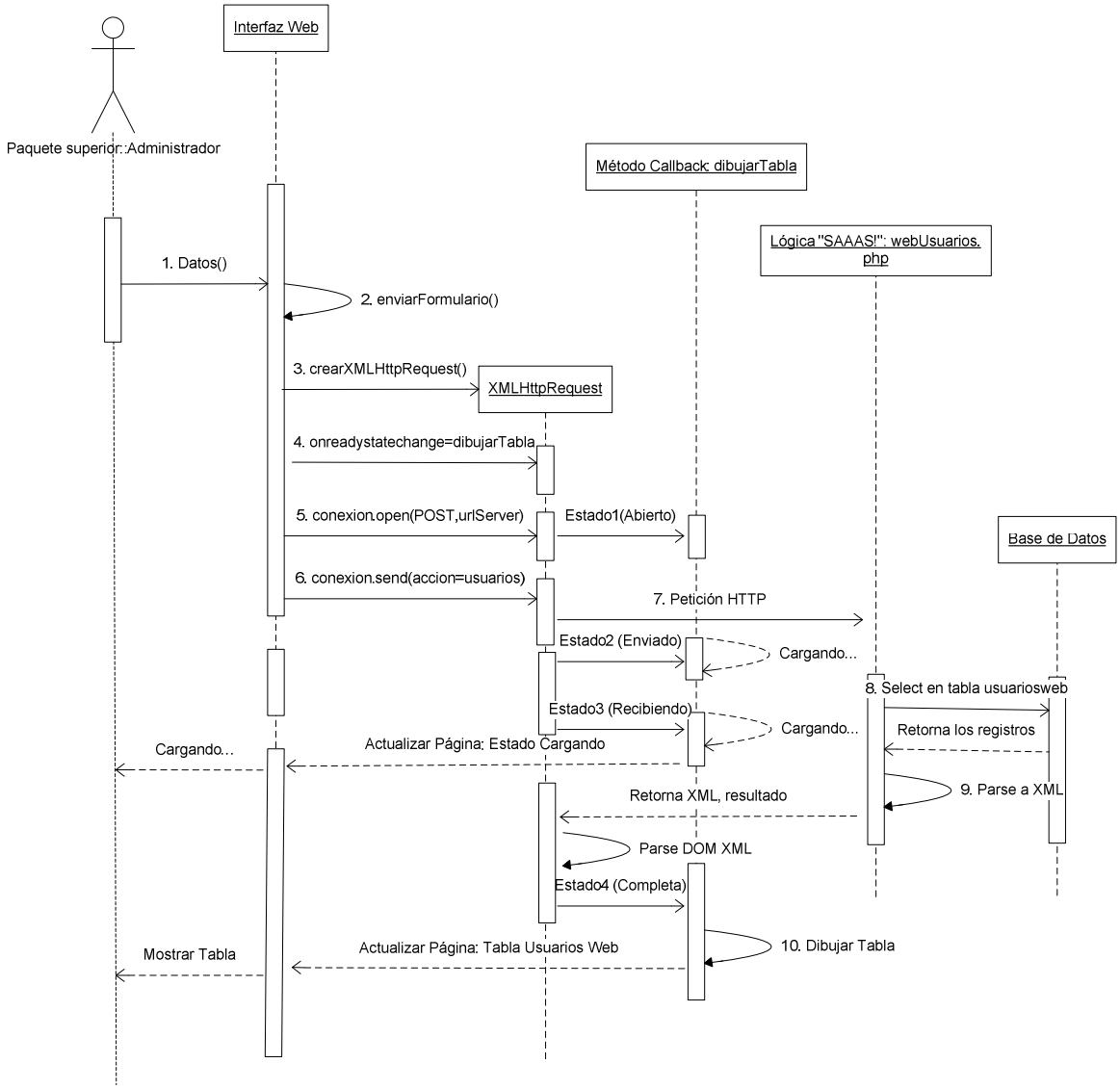
A14. Diagrama de Secuencia para la Acción: Mostrar estado de los servicios de "SAAAS!"



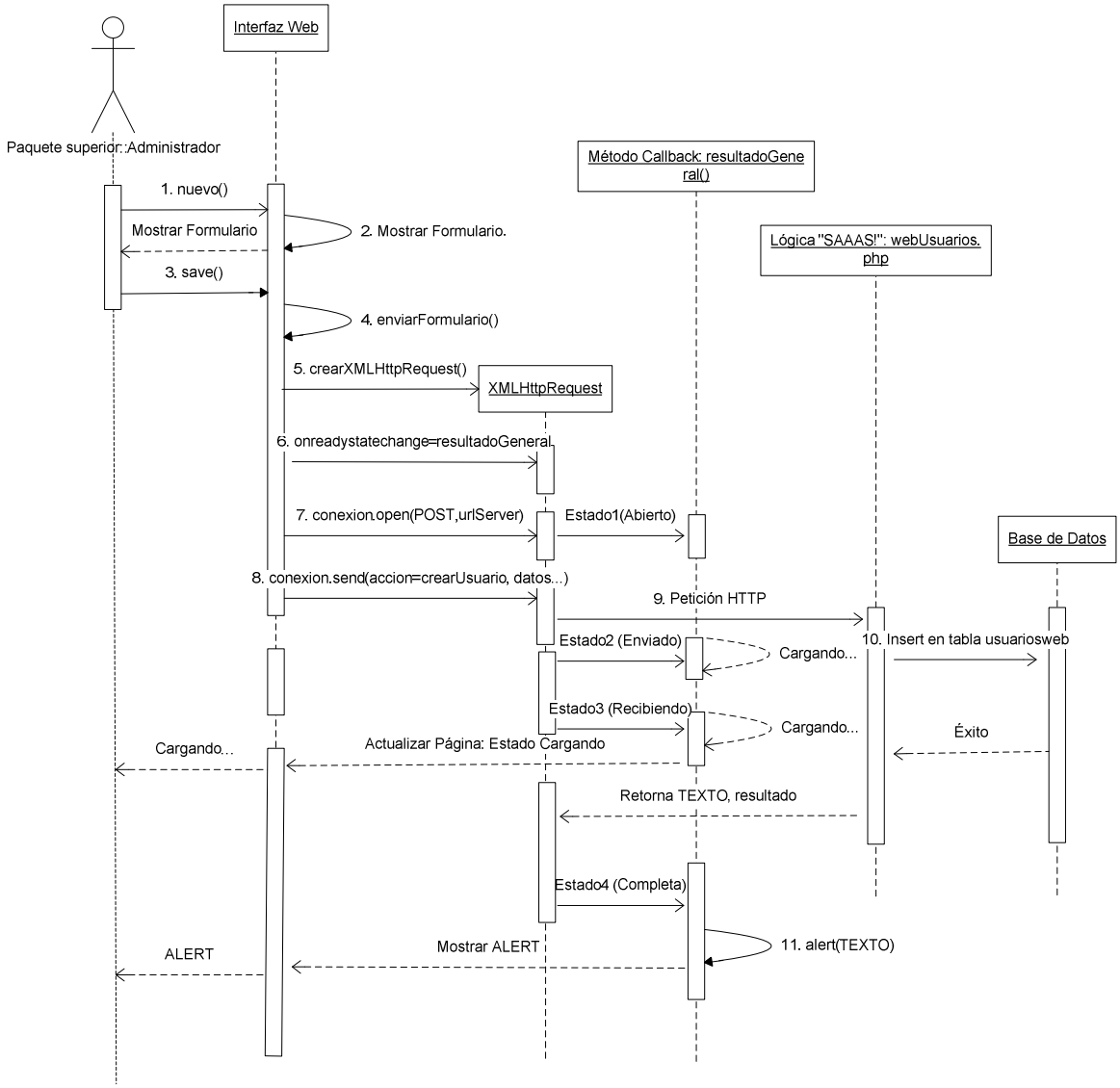
A15. Diagrama de Secuencia para la Acción: Manipular servicios de “SAAAS!” (Ejemplo: Detener un Servicio)



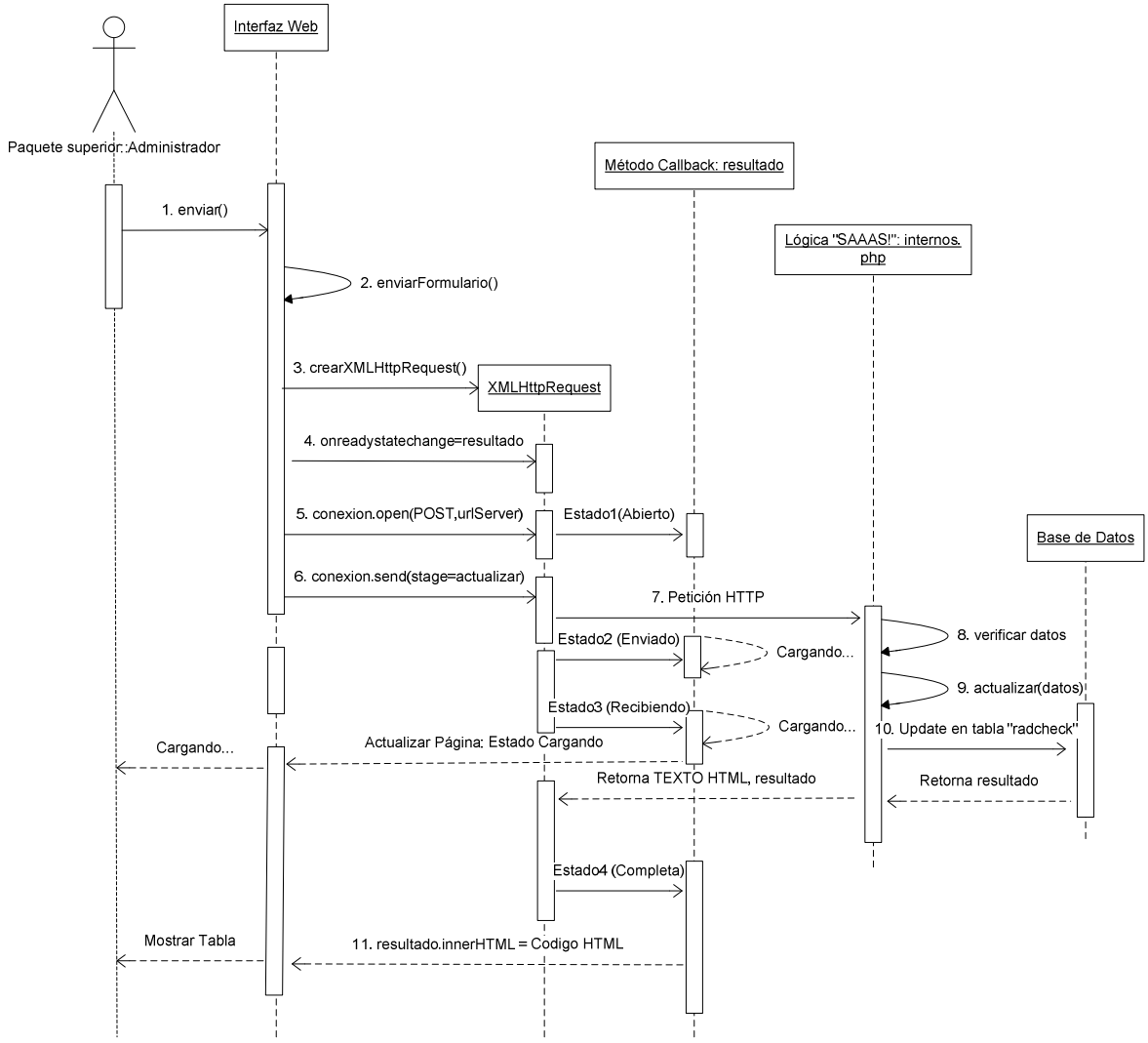
A16. Diagrama de Secuencia para la Acción: Mostrar Usuarios Web



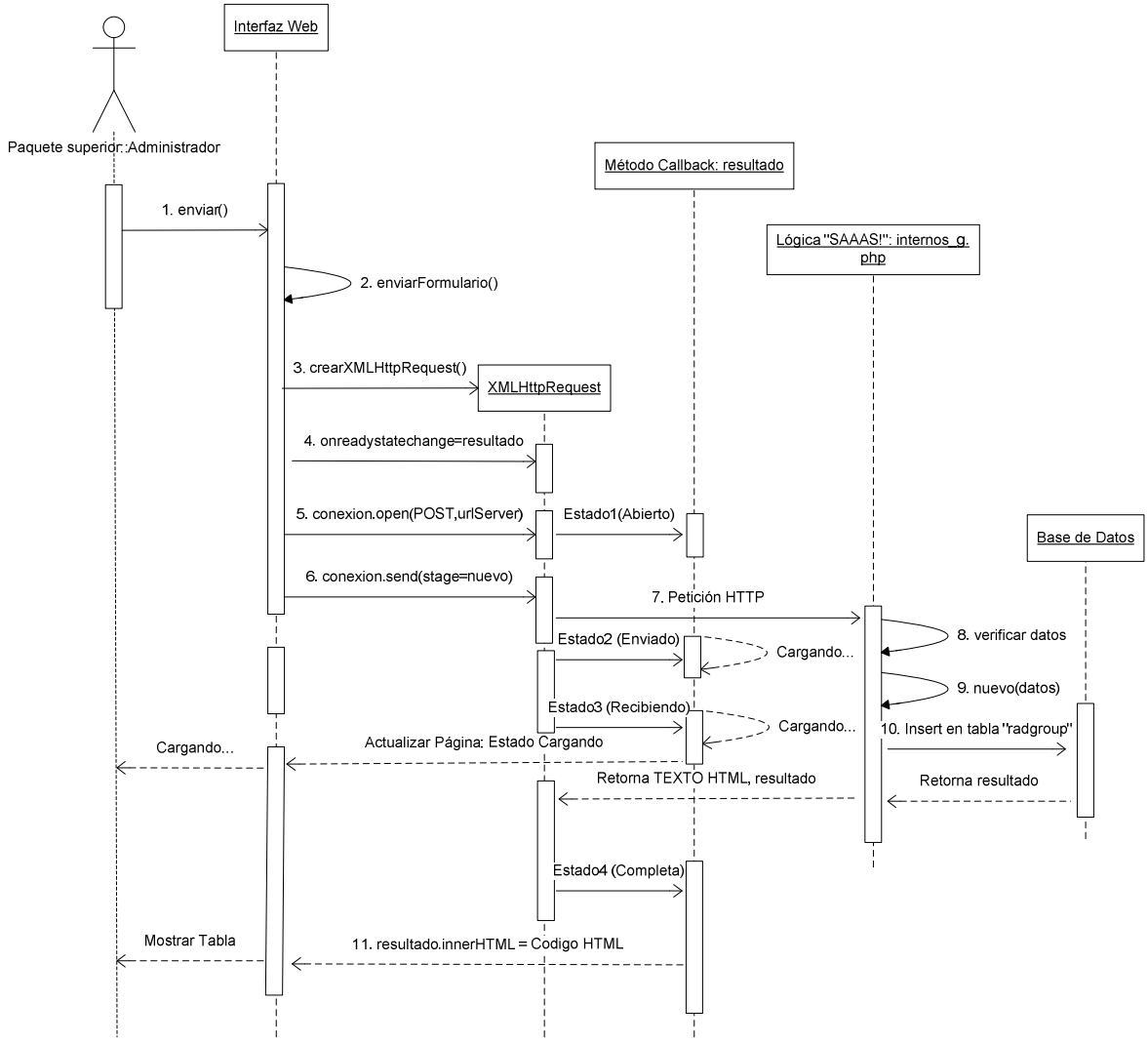
A17. Diagrama de Secuencia para la Acción: Manipular Usuarios Web (Ejemplo: Crear)



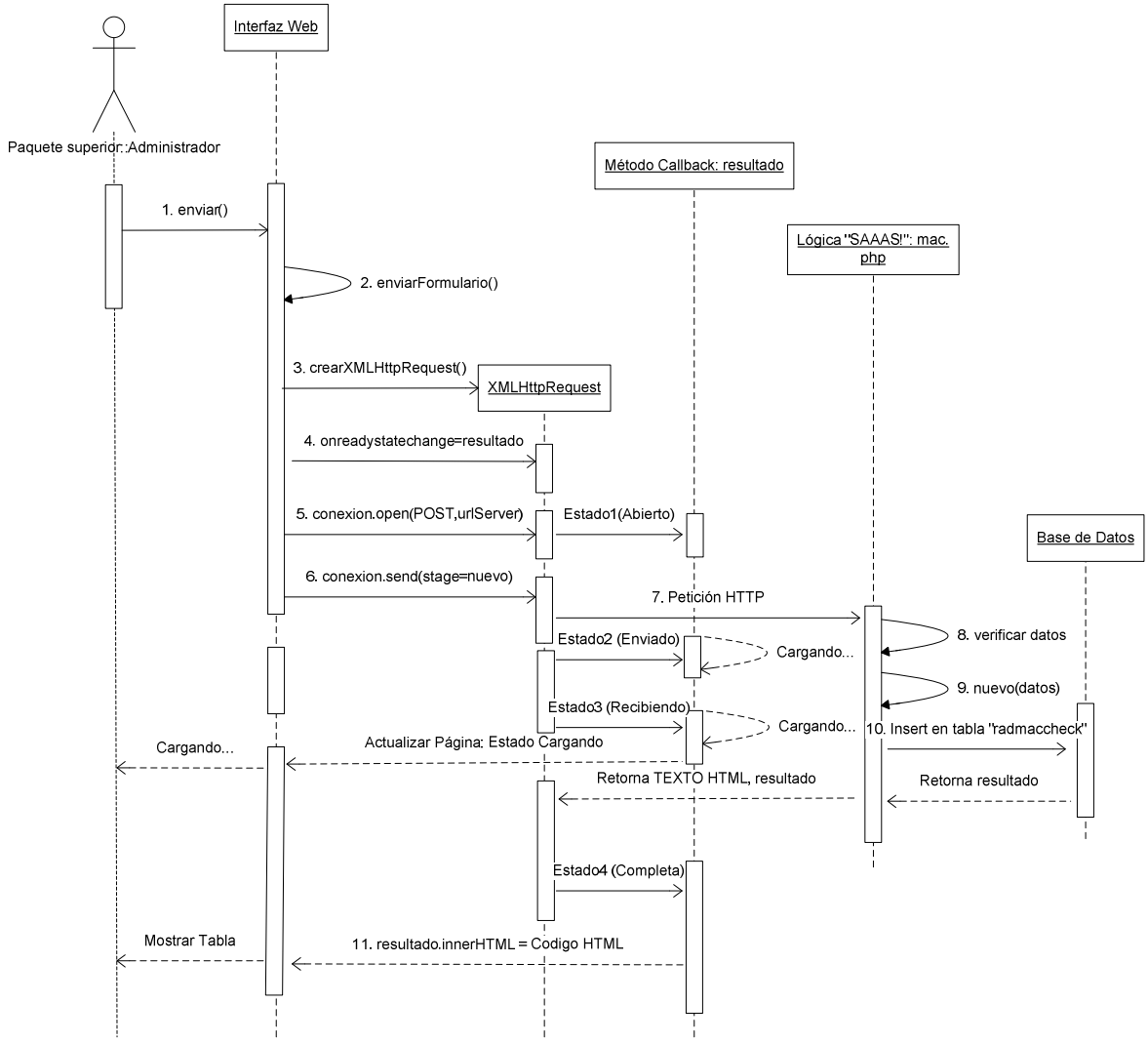
A18. Diagrama de Secuencia para la Acción: Manipulación de Usuarios locales para la autenticación (Ejemplo: Modificar Usuario local)



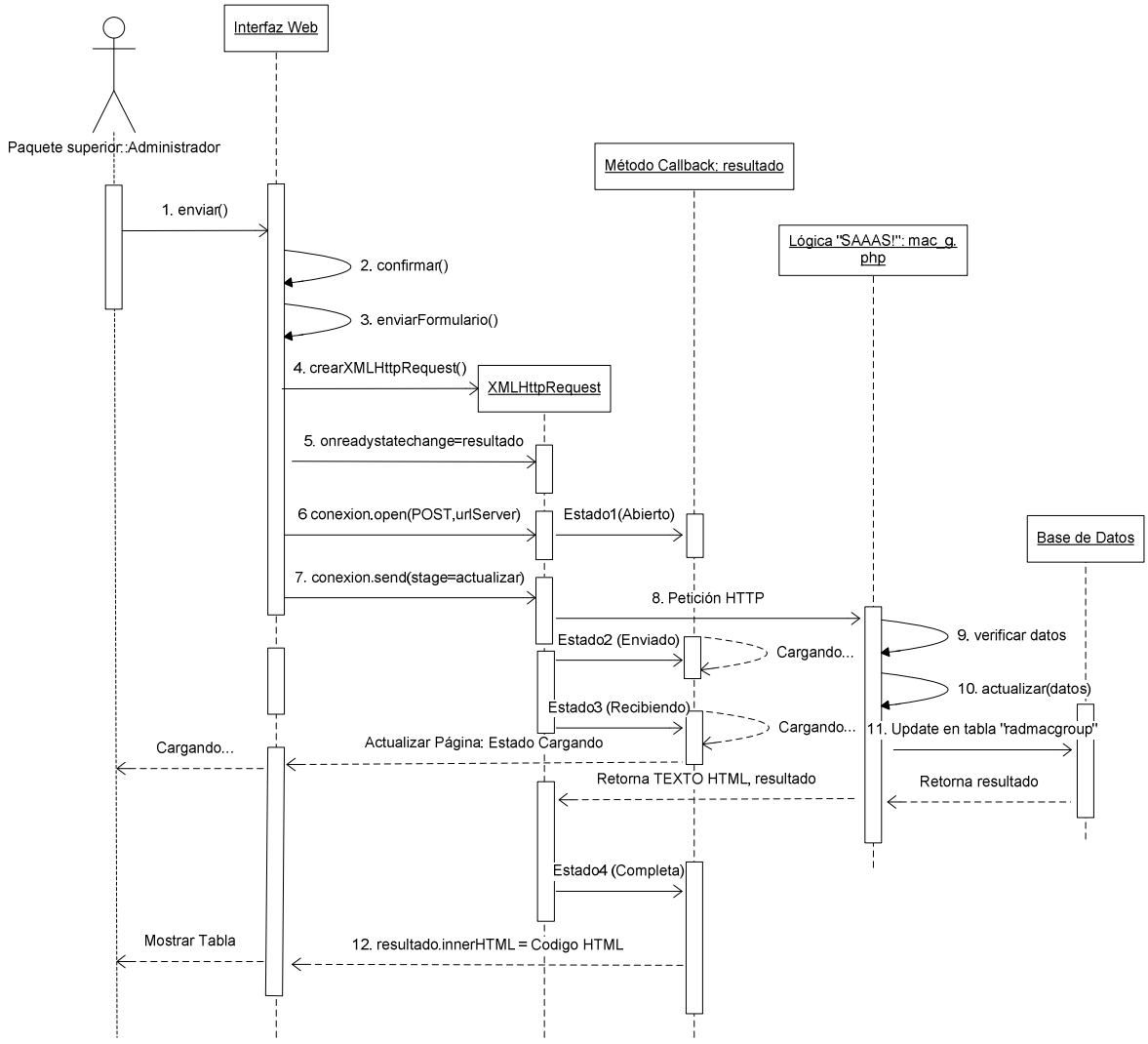
A19. Diagrama de Secuencia para la Acción: Manipulación de grupos de usuarios locales para la autenticación (Ejemplo: Nuevo Grupo local)



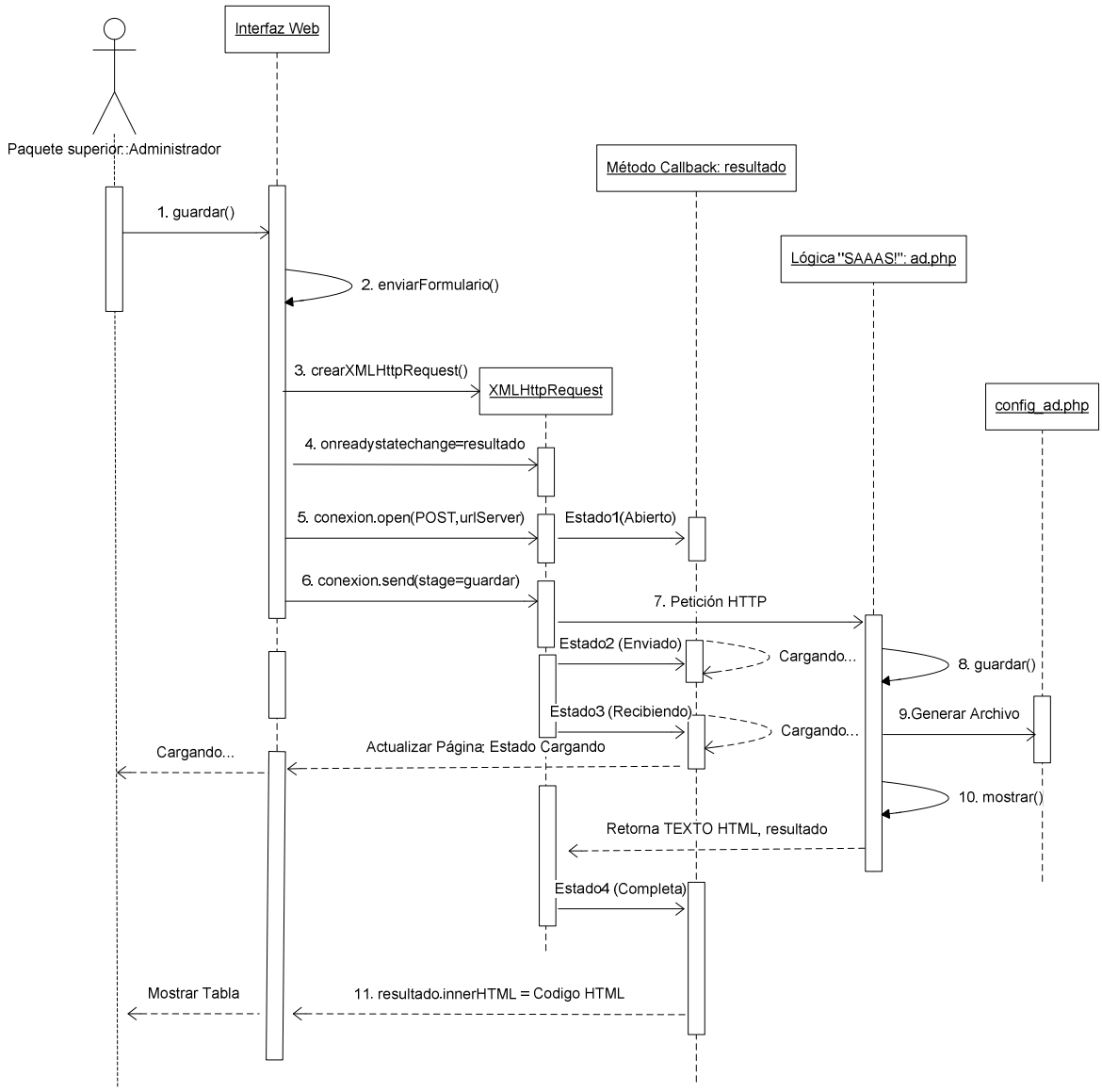
A20. Diagrama de Secuencia para la Acción: Manipulación de equipos locales para la autenticación (Ejemplo: Nuevo Equipo local)



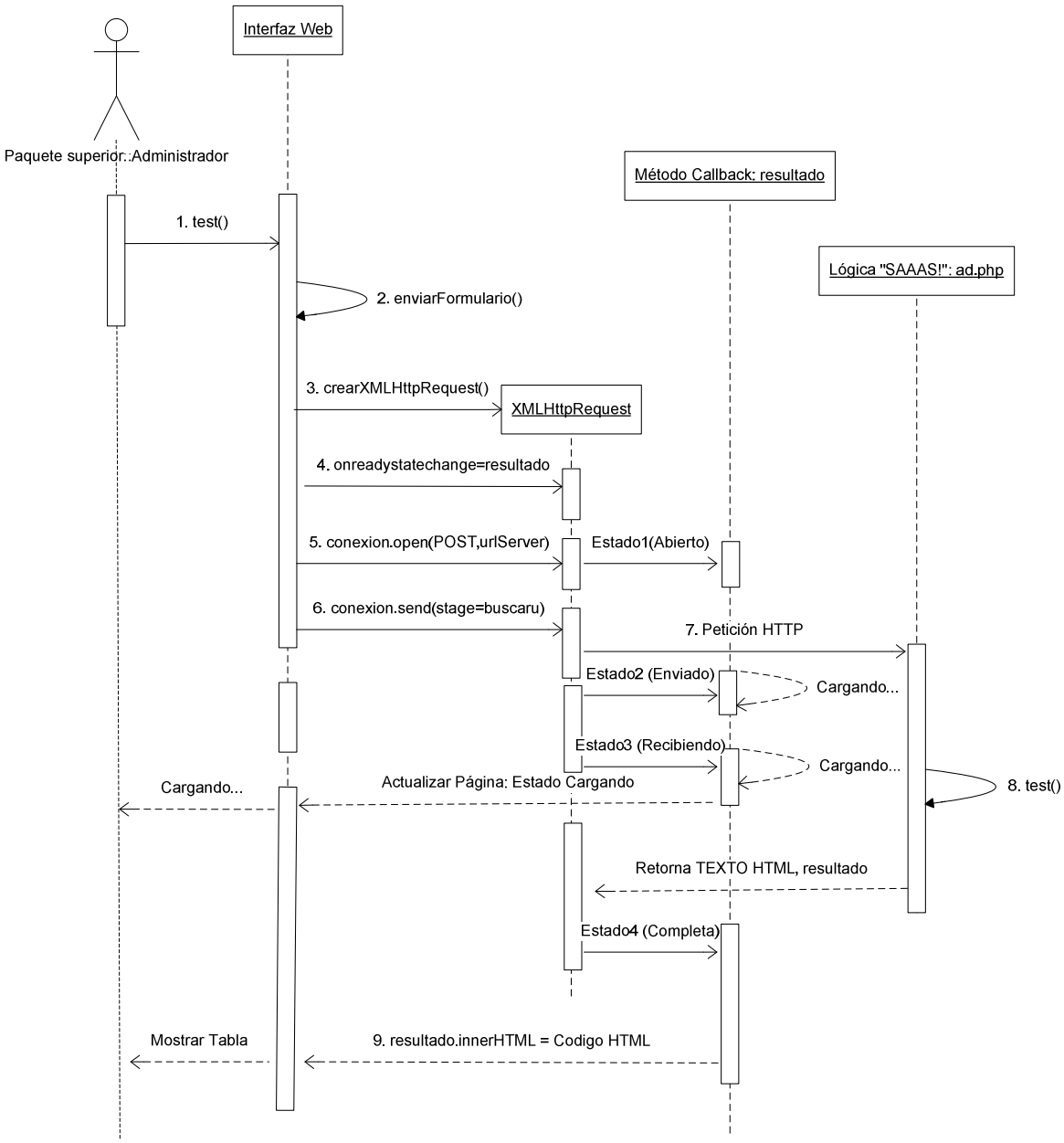
A21. Diagrama de Secuencia para la Acción: Manipulación de grupos de equipos locales para la autenticación (Ejemplo: Eliminar Grupo de equipos locales)



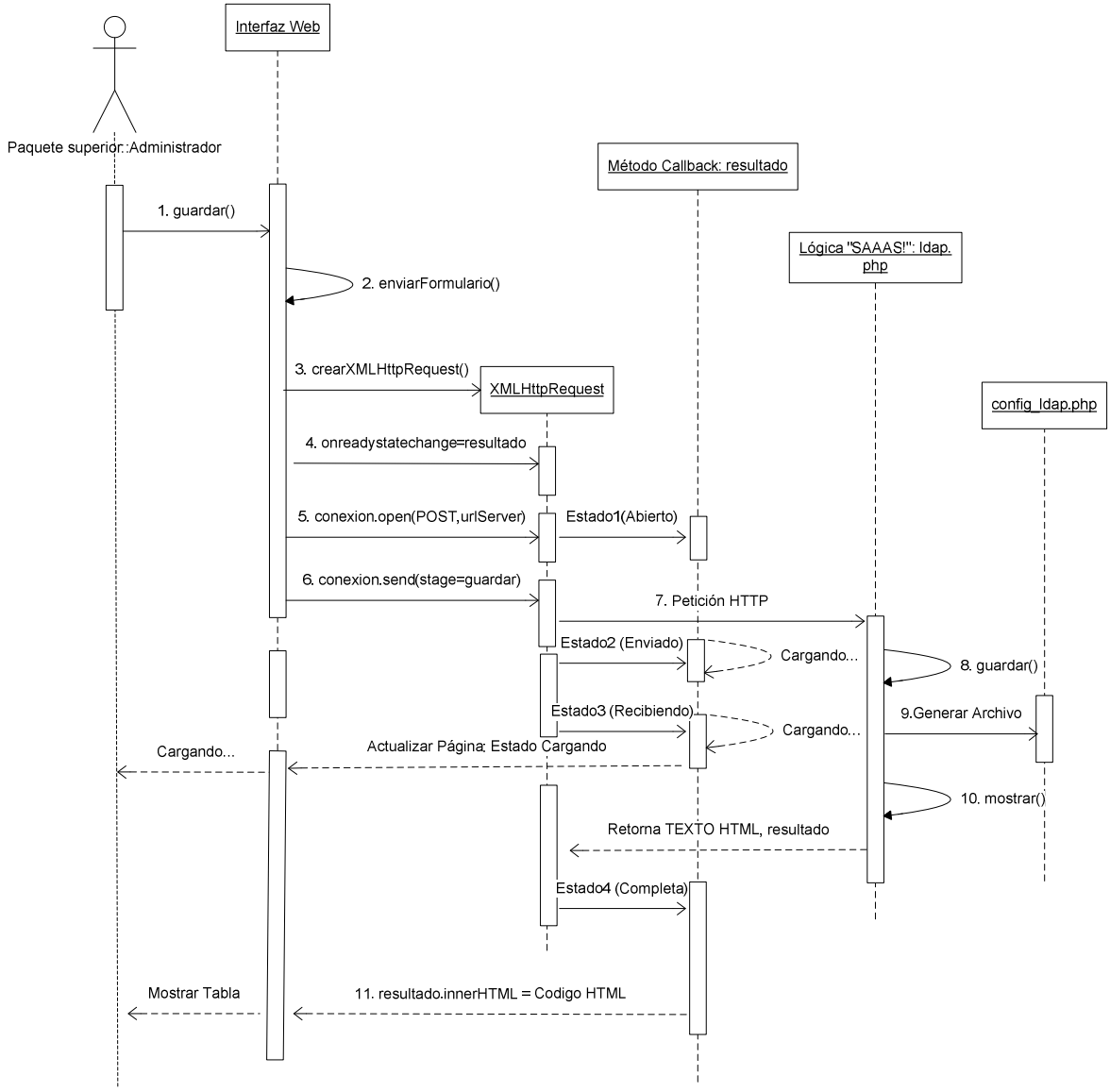
A22 Diagrama de Secuencia para la Acción: Configuración de la conexión con Active Directory para manejo de credenciales



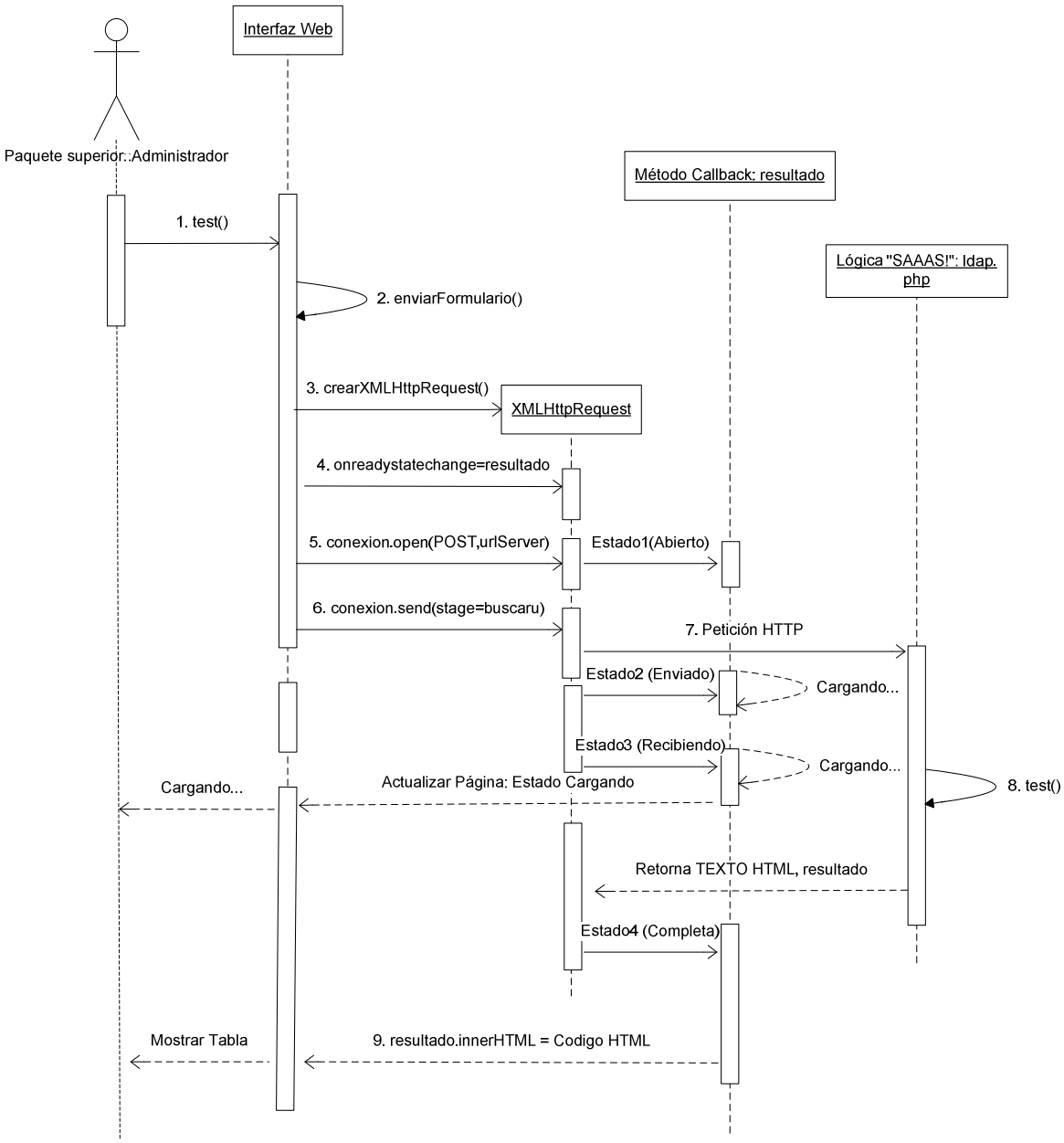
A23. Diagrama de Secuencia para la Acción: Test de usuarios del Active Directory



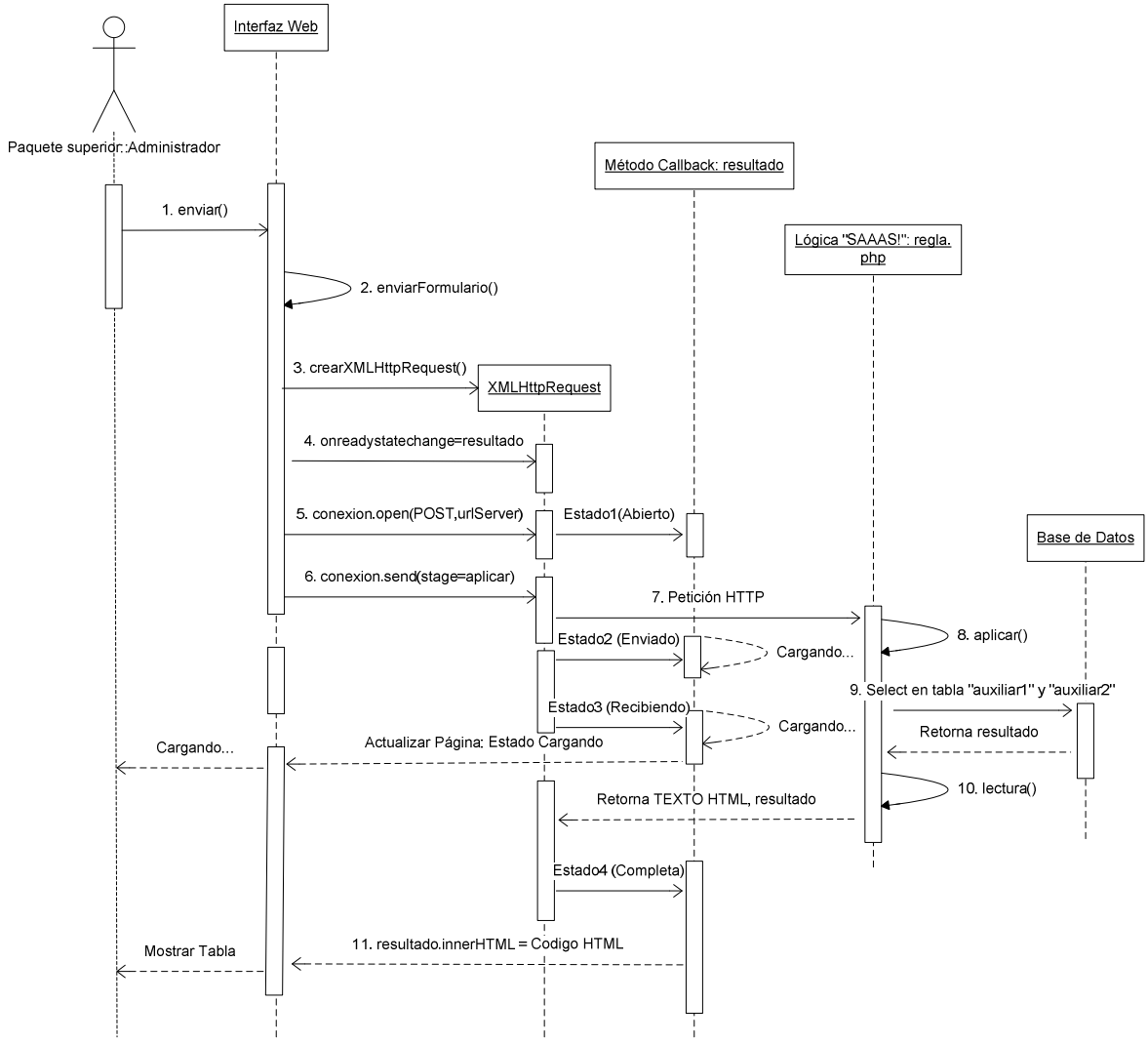
A24. Diagrama de Secuencia para la Acción: Configuración de la conexión con LDAP para manejo de credenciales



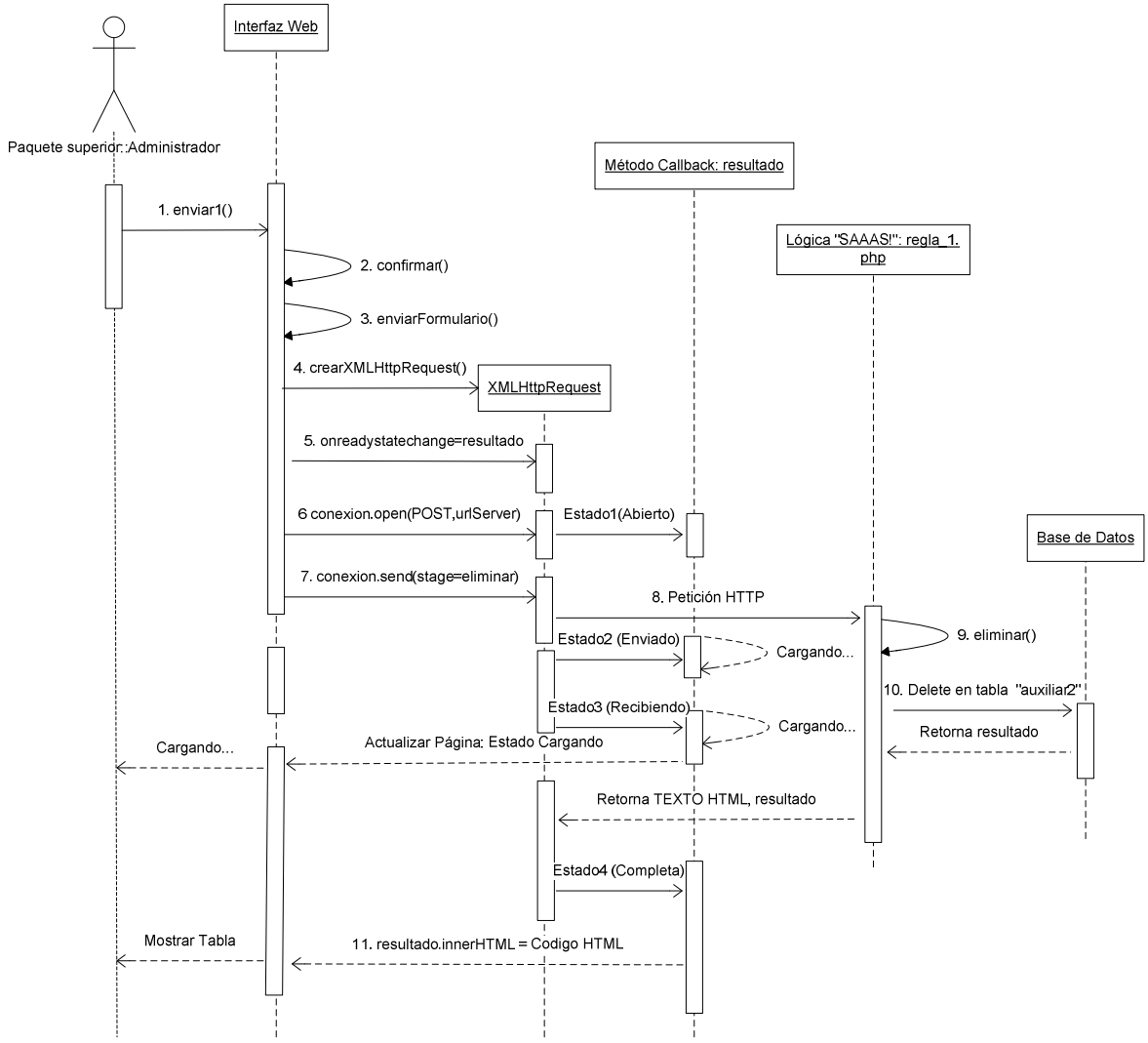
A25. Diagrama de Secuencia para la Acción: Test de usuarios de LDAP



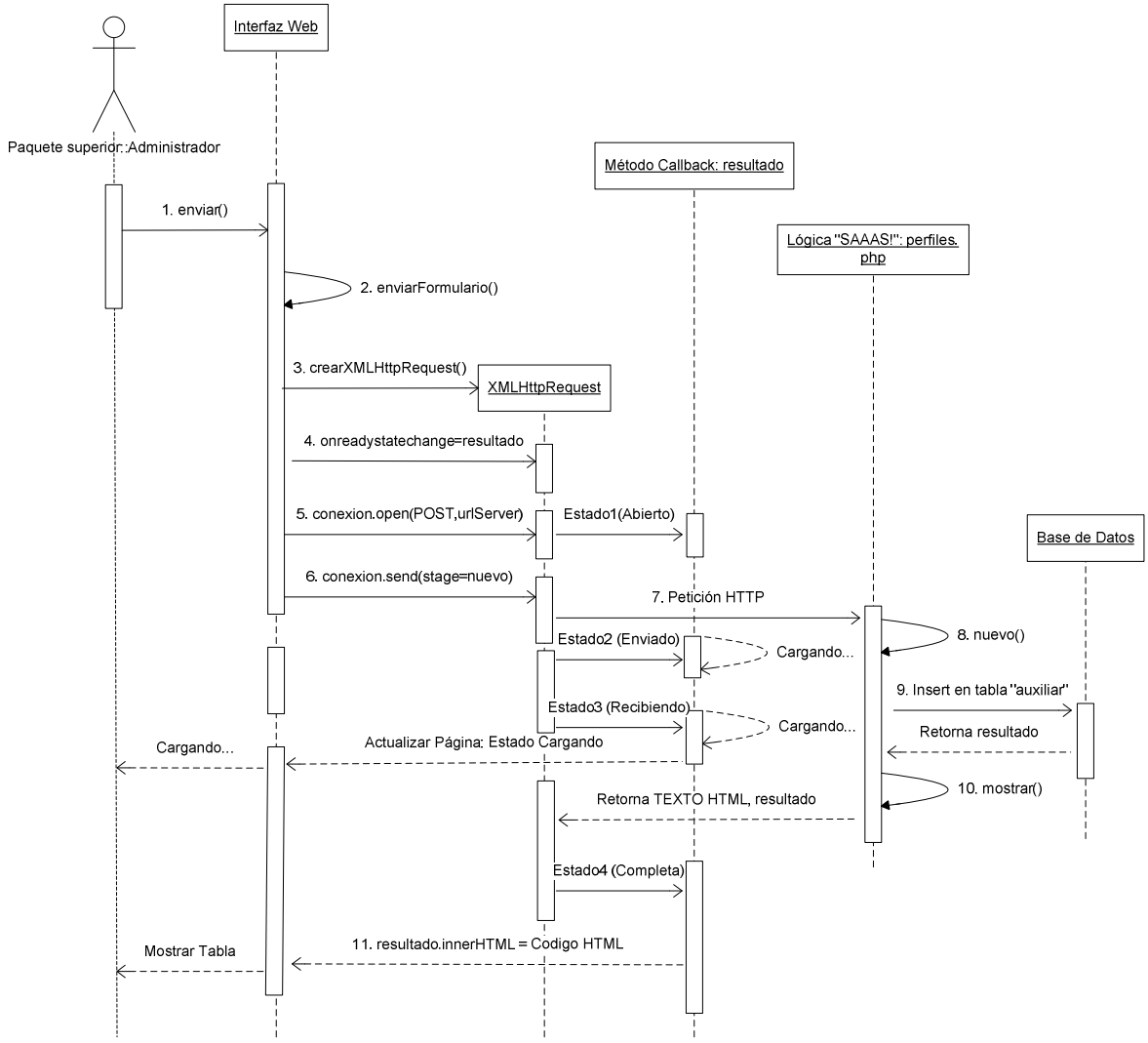
A26. Diagrama de Secuencia para la Acción: Configuración de Reglas de Autenticación (Repositorios de credenciales)



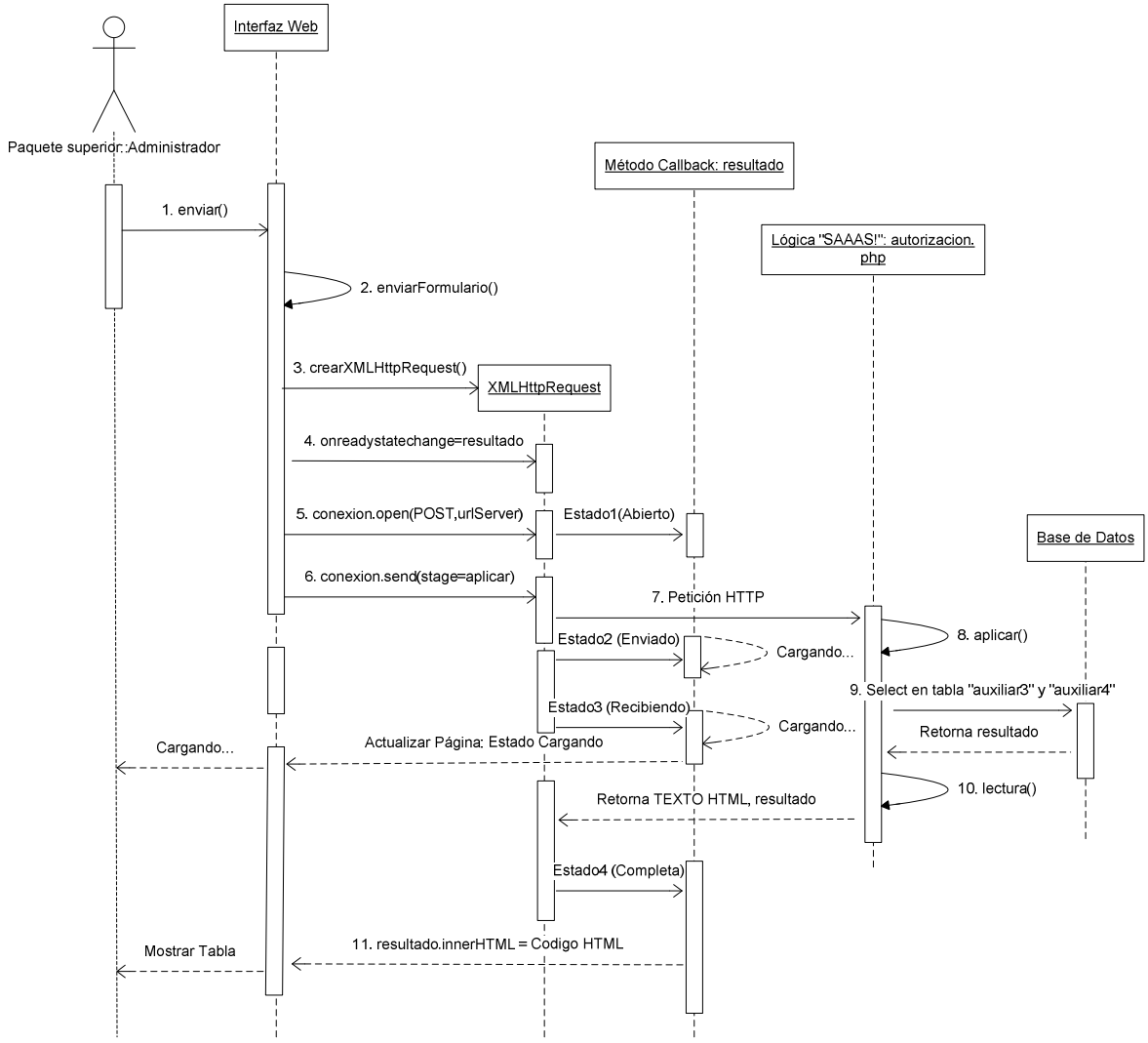
A27. Diagrama de Secuencia para la Acción: Edición de Reglas de Autenticación (Ejemplo: Borrar Regla de Autenticación)



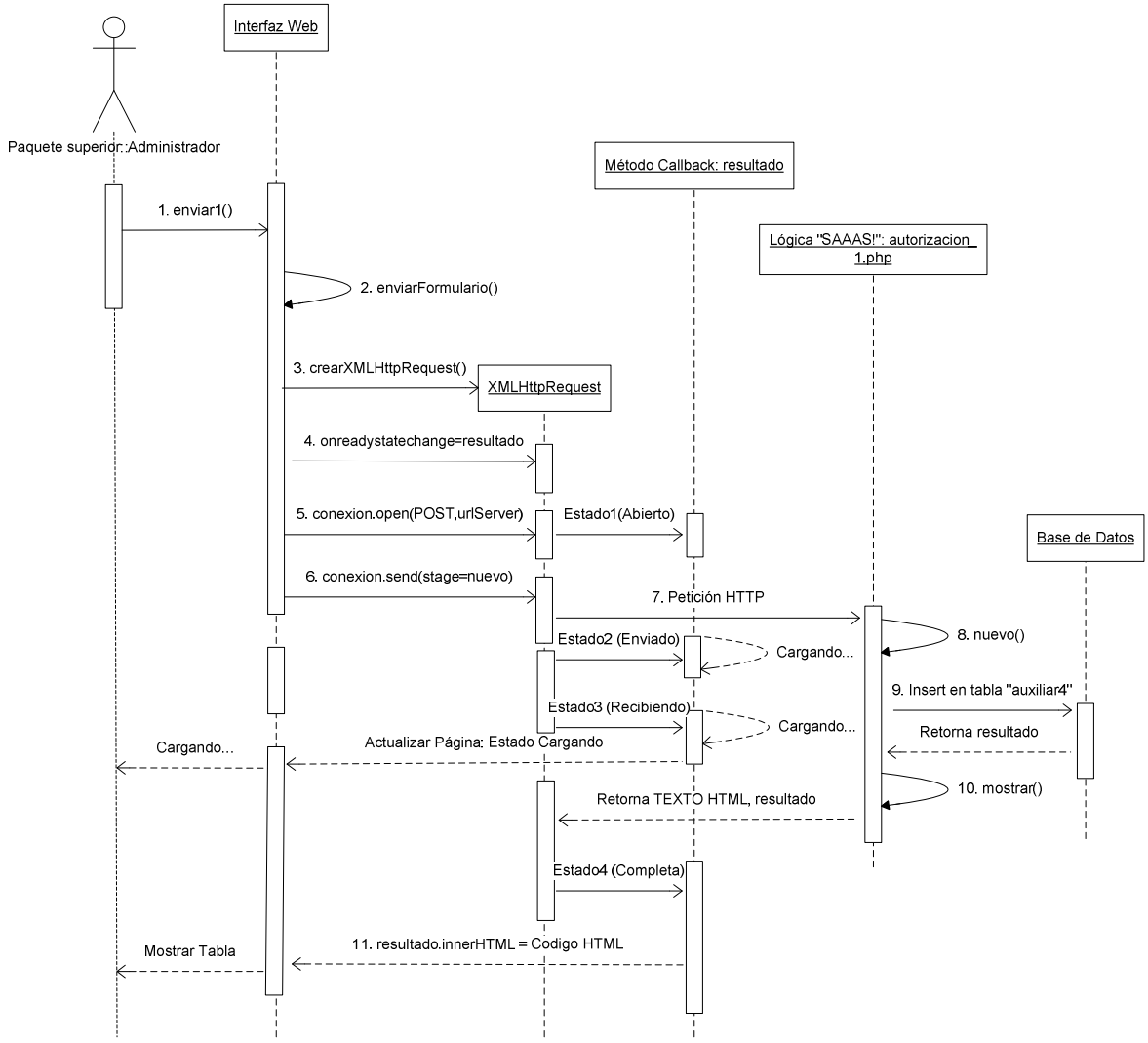
A28. Diagrama de Secuencia para la Acción: Configuración de Perfiles de Red (Ejemplo: Nuevo Perfil)



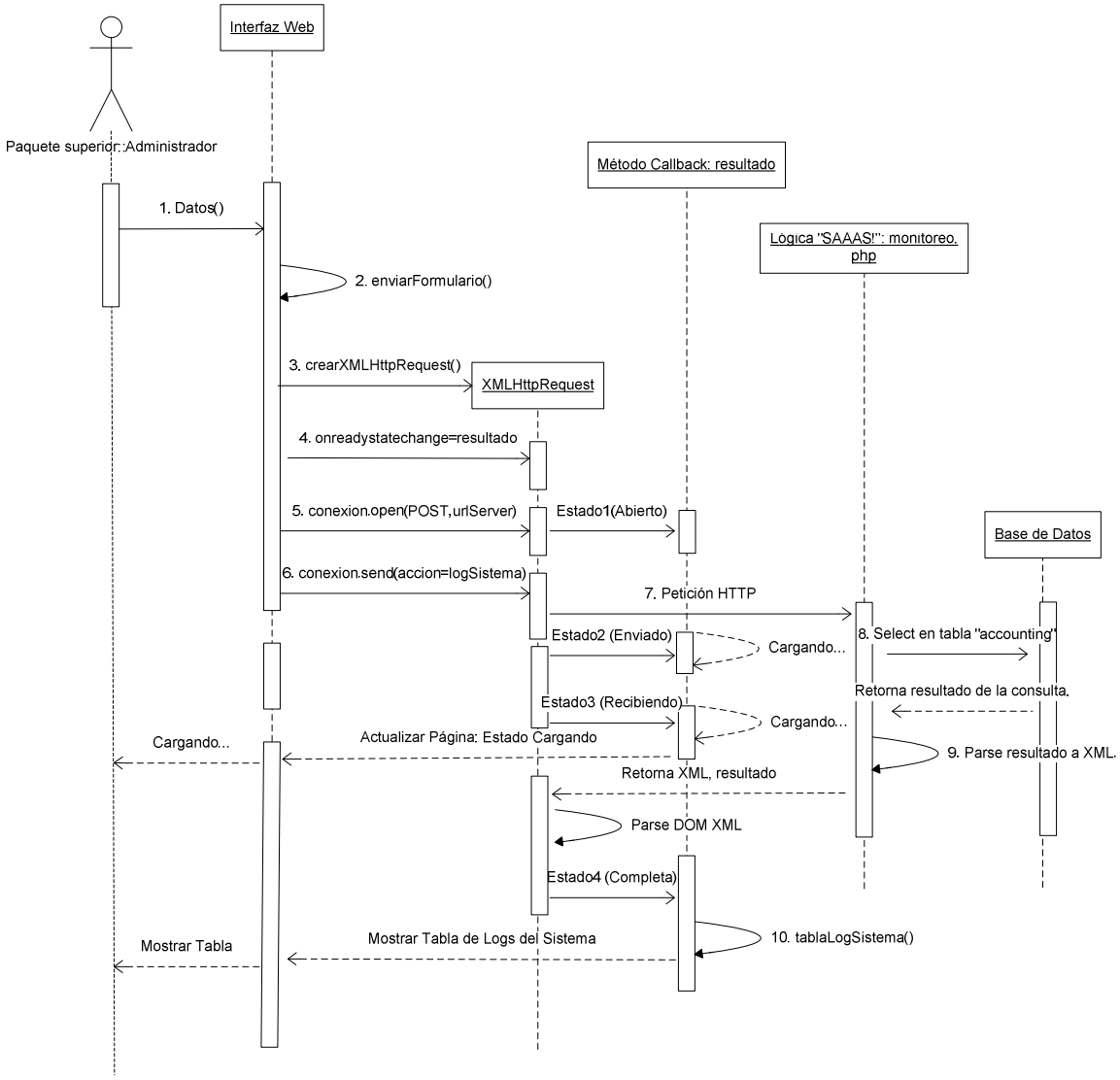
A29. Diagrama de Secuencia para la Acción: Configuración de Reglas de Autorización (Perfiles)



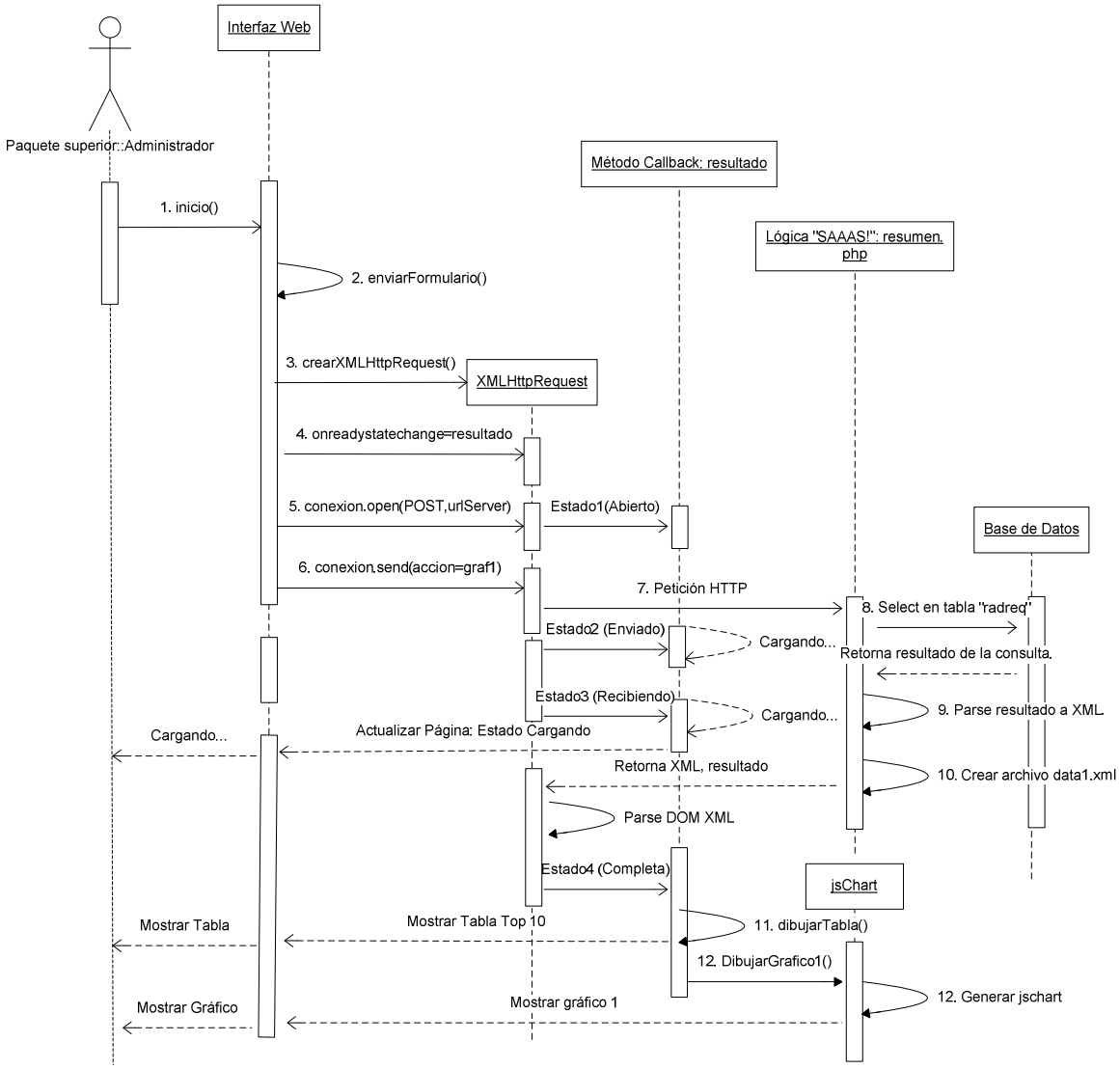
A30. Diagrama de Secuencia para la Acción: Configuración de Reglas de Autorización (Ejemplo: Nuevo Reglas de Autorización)



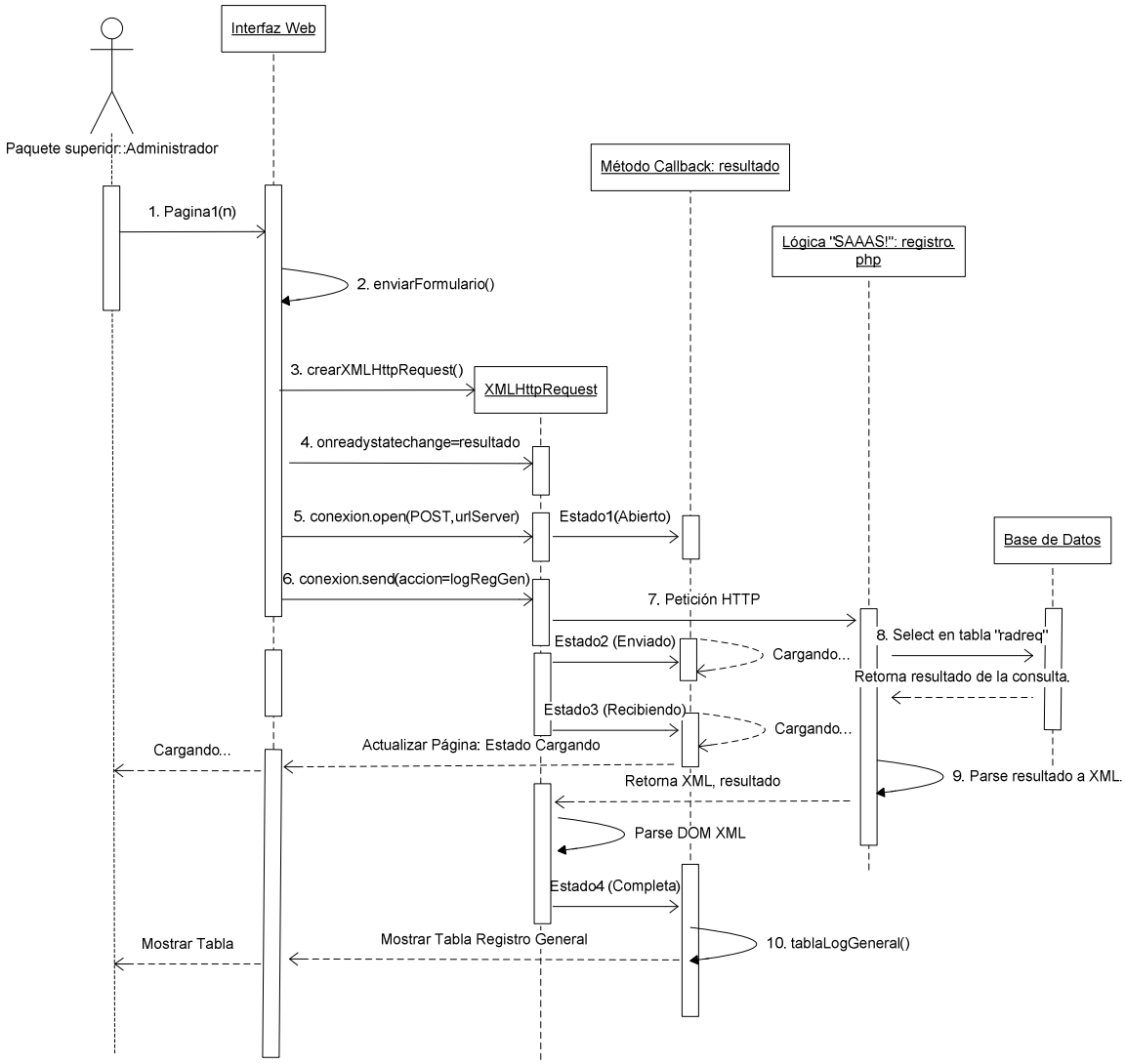
A31. Diagrama de Secuencia para la Acción: Generar Reporte de Eventos del Sistema



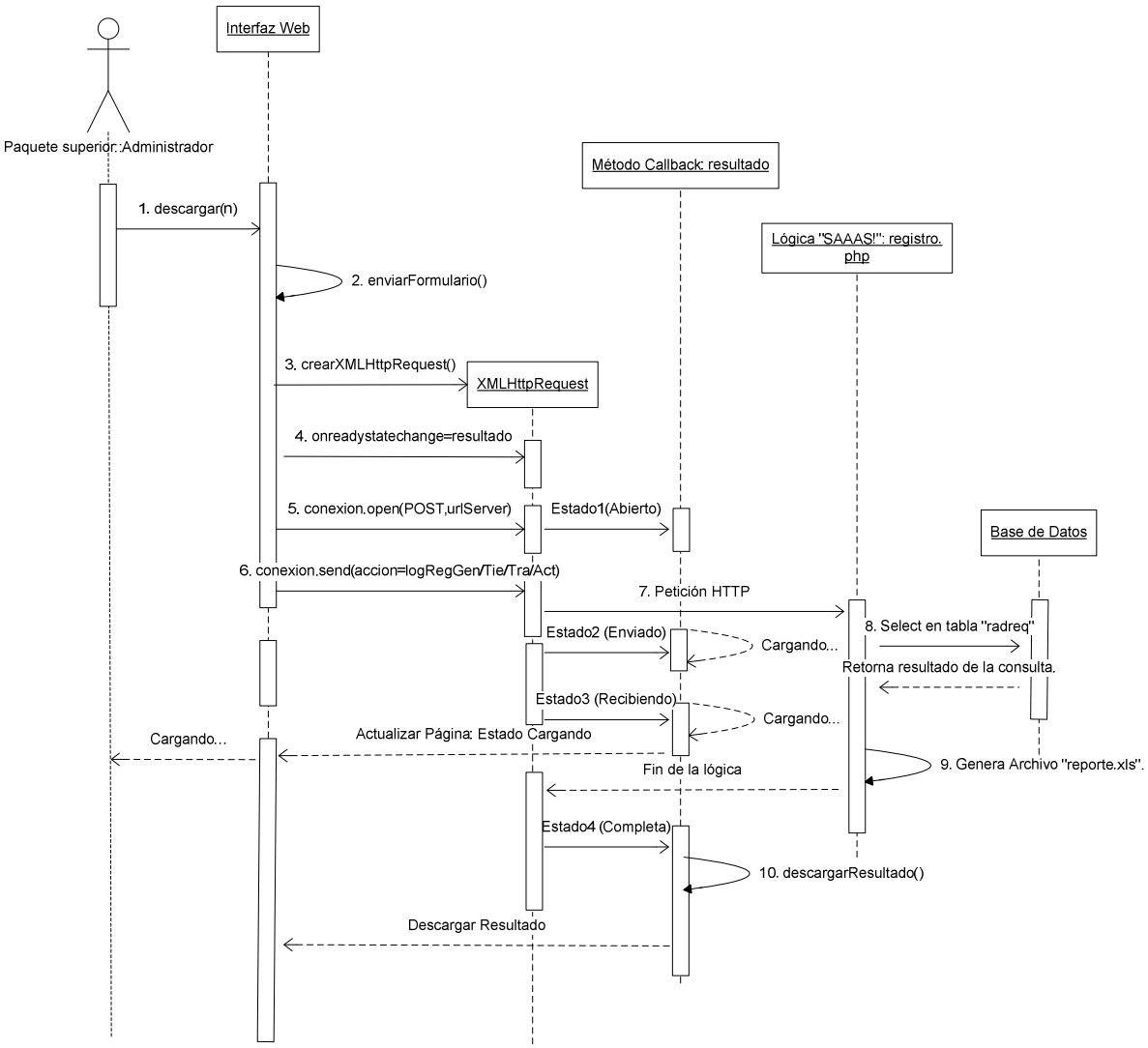
A32. Diagrama de Secuencia para la Acción: Mostrar Resumen de Registro AAA



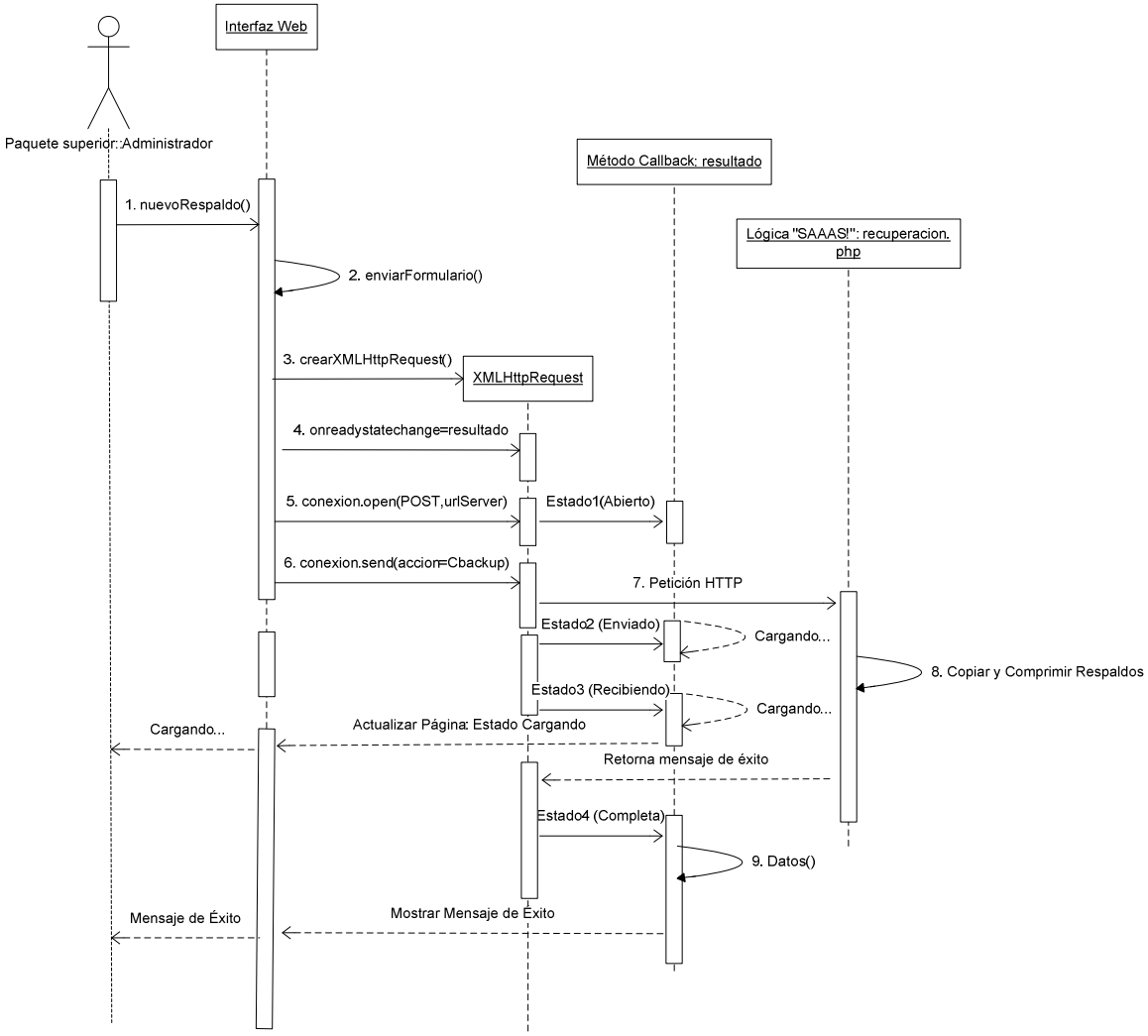
A33. Diagrama de Secuencia para la Acción: Visualizar Reportes del Registro AAA (Ejemplo: Registro General)



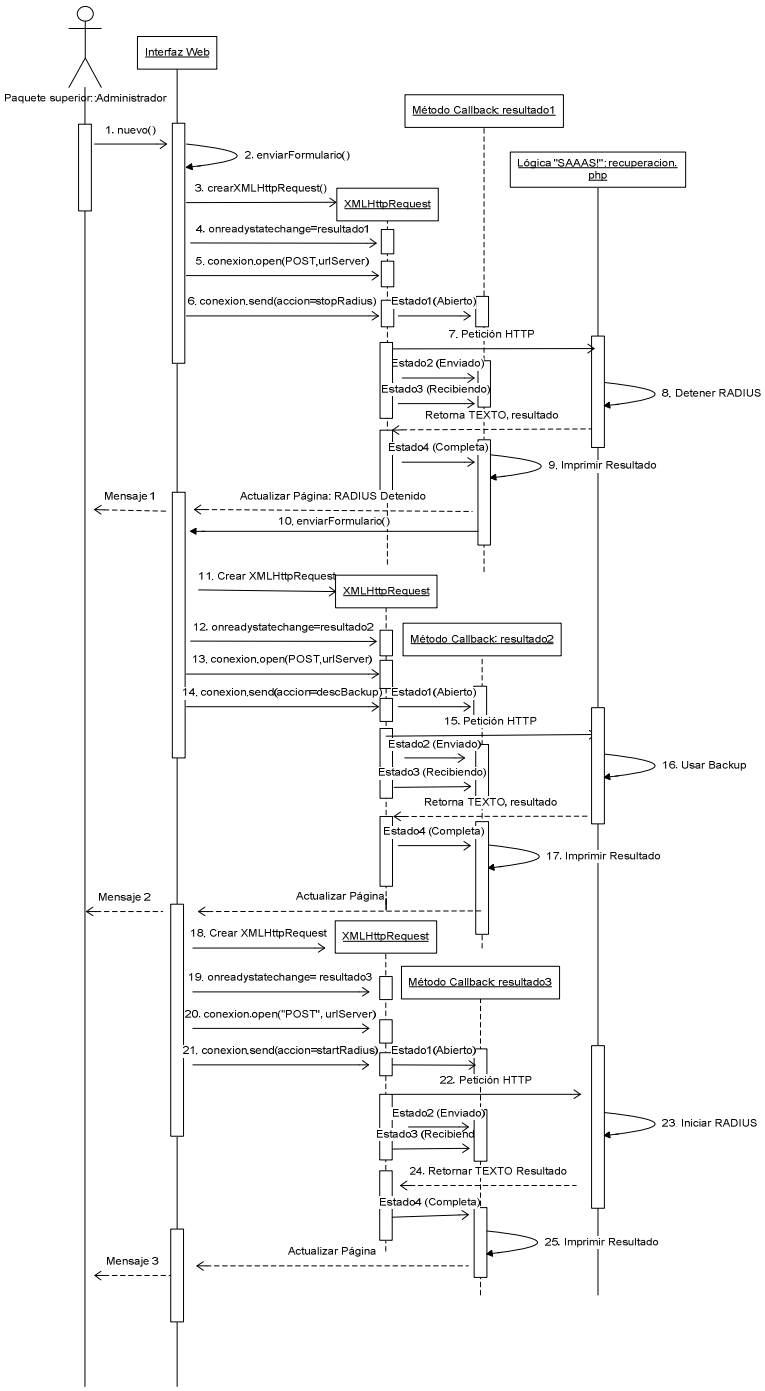
A34. Diagrama de Secuencia para la Acción: Descargar Reportes del Registro AAA



A35. Diagrama de Secuencia para la Acción: Crear Respaldos de la configuración del Servidor



A.36 Diagrama de Secuencia para la Acción: Usar Respaldos de la configuración del Servidor



B1. Caso de Uso:

ANEXO B

Casos de Uso de la Aplicación

“SAAAS!”

B1. Caso de Uso: Visualizar Información Básica del Sistema

Caso de Uso: Visualizar Información Básica del Sistema	
Descripción	Permite visualizar la información básica del servidor.
Actores	Administrador del Sistema.
Precondiciones	El usuario debe haber iniciado sesión con perfil de Administrador y haber accedido a la página Información Básica dentro de la sección de Configuración General.
Flujo Normal	El sistema muestra una tabla con los datos: Dirección IP, kernel, sistema operativo y uptime. El actor pulsa sobre la pestaña *Uso de la Red.* El sistema muestra una tabla con la información del uso de las interfaces de red activas. El actor pulsa sobre la pestaña *Hardware*. El sistema muestra una tabla con la información de las características de hardware (PCI, USB, etc) del sistema. El actor pulsa sobre la pestaña *Uso de la Memoria*. El sistema muestra una tabla con la información del porcentaje de uso de la memoria.
Flujo Alternativo	No existe flujo alternativo, la página solo muestra información.
Post Condiciones	No existen post condiciones.

B2. Caso de Uso: Modificar Interfaz de Red

Caso de Uso: Modificar Interfaz de Red	
Descripción	Permite visualizar y modificar el direccionamiento IP de las interfaces de red físicas y virtuales del sistema.
Actores	Administrador del Sistema.
Precondiciones	El usuario debe haber iniciado sesión en el sistema con perfil de Administrador y haber accedido a la página *Recursos de Red*, pestaña *Interfaces de Red,* dentro de la sección de Configuración General.
Flujo Normal	El sistema muestra una tabla con el Nombre, Tipo, Dirección MAC, Dirección IP, Máscara de Red, Clase, Estado y Gateway asignados a las interfaces físicas y virtuales del servidor. El actor pulsa sobre el botón *Modificar* junto a la información de una interfaz de red en particular. El sistema muestra la información que se puede modificar en cajas de texto: Dirección IP, Máscara de Red y Gateway y el campo Tipo en un menú desplegable. Si el campo tipo de la interfaz es *Dinámica*, el resto de campos no aparece en cajas de texto. El actor modifica los campos y pulsa sobre el botón *Guardar* junto a la información de la interfaz a modificarse. El sistema verifica que el formato de los datos ingresados sea correcto y pide confirmación al usuario para procesar la orden. El sistema reinicia los servicios de red para aplicar los cambios. El sistema retorna un mensaje de éxito al usuario.
Flujo Alternativo	Si alguno de los datos ingresados no tiene el formato correcto el sistema retorna un mensaje de error y no se procede con la acción solicitada.
Post Condiciones	La pestaña se actualiza y se pueden verificar los cambios realizados.

B3. Caso de Uso: Eliminar Interfaz de Red Virtual

Caso de Uso: Eliminar Interfaz de Red Virtual	
Descripción	Permite visualizar y modificar el direccionamiento IP de las interfaces de red físicas y virtuales del sistema.
Actores	Administrador del Sistema.
Precondiciones	El usuario debe haber iniciado sesión en el sistema con perfil de Administrador y haber accedido a la página *Recursos de Red,* pestaña *Interfaces de Red*, dentro de la sección de Configuración General.
Flujo Normal	El sistema muestra una tabla con el Nombre, Tipo, Dirección MAC, Dirección IP, Máscara de Red, Clase, Estado y Gateway asignados a las interfaces físicas y virtuales del servidor. Junto a las interfaces virtuales aparece un botón *Eliminar*. El actor pulsa sobre el botón *Eliminar* junto a la información de la interfaz de red virtual que desea borrar. El sistema pide confirmación al usuario para procesar la orden. El sistema elimina la interfaz de red deseada y reinicia los servicios de red para aplicar los cambios. El sistema retorna un mensaje de éxito.
Flujo Alternativo	Si el usuario no confirma, no se realiza ninguna acción.
Post Condiciones	La pestaña se actualiza y se pueden verificar los cambios realizados.

B4. Caso de Uso: Activar/Desactivar una Interfaz de Red

Caso de Uso: Activar/Desactivar una Interfaz de Red	
Descripción	Permite activar o desactivar un interfaz de red física.
Actores	Administrador del Sistema.
Precondiciones	El usuario debe haber iniciado sesión en el sistema con perfil de Administrador y haber accedido a la página *Recursos de Red*, pestaña *Interfaces de Red*, dentro de la sección de Configuración General.
Flujo Normal	El sistema muestra una tabla con el Nombre, Tipo, Dirección MAC, Dirección IP, Máscara de Red, Clase, Estado y Gateway asignados a las interfaces físicas y virtuales del servidor. El actor pulsa sobre el menú desplegable *Estado* de una interfaz en particular y elige entre la opción *Activa* o *Inactiva*. El sistema pide confirmación al usuario para procesar la orden. El sistema cambia el estado de la interfaz de red deseada y reinicia los servicios de red para que se visualicen los cambios. El sistema retorna un mensaje de éxito.
Flujo Alternativo	Si el usuario no confirma, no se realiza ninguna acción.
Post Condiciones	La pestaña se actualiza y se pueden verificar los cambios realizados.

B5. Caso de Uso: Visualizar Tabla de Rutas

Caso de Uso: Visualizar Tabla de Rutas	
Descripción	Permite visualizar la tabla de rutas que maneja el servidor.
Actores	Administrador del Sistema.
Precondiciones	El usuario debe haber iniciado sesión en el sistema con perfil de Administrador y haber accedido a la página *Recursos de Red*, pestaña *Rutas*, dentro de la sección de Configuración General.
Flujo Normal	El sistema muestra una tabla de rutas con el Destino, Gateway, Máscara y la Interfaz física por la que se envía el tráfico.

Flujo Alternativo	No aplica.
Post Condiciones	No aplica.

B6. Caso de Uso: Crear una Interfaz Virtual

Caso de Uso: Crear una Interfaz Virtual	
Descripción	Permite crear una interfaz virtual.
Actores	Administrador del Sistema.
Precondiciones	El usuario debe haber iniciado sesión en el sistema con perfil de Administrador y haber accedido a la página *Recursos de Red*, pestaña *Nueva Interfaz Virtual*, dentro de la sección de Configuración General.
Flujo Normal	El sistema muestra una tabla con un menú desplegable para elegir la interfaz física sobre la cual se desea crear la interfaz virtual; y cuadros de texto para especificar la dirección IP y la máscara de red que se quiere asignar a la nueva interfaz virtual. El actor llena los campos y pulsa sobre el botón *Guardar*. El sistema verifica que el formato de los datos ingresados sea correcto y pide confirmación al usuario para procesar la orden. El sistema reinicia los servicios de red para aplicar los cambios. El sistema retorna un mensaje de éxito al usuario.
Flujo Alternativo	Si alguno de los datos ingresados no tiene el formato correcto el sistema retorna un mensaje de error y no se procede con la acción solicitada.
Post Condiciones	No existen.

B7. Caso de Uso: Configurar Autoridad Certificadora

Caso de Uso: Configurar Autoridad Certificadora	
Descripción	Permite crear una Autoridad Certificadora encargada de firmar los certificados generados localmente.
Actores	Administrador del Sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Configuración Básica* en la sección *Administración de Certificados*.
Flujo Normal	El usuario completa los campos obligatorios del formulario para crear una CA. El usuario presiona el botón *Enviar*. El sistema verifica la validez de los datos ingresados en el formulario. El sistema crea una nueva estructura de certificados. El sistema muestra un mensaje de éxito al usuario.
Flujo Alternativo	Los datos ingresados en el formulario no son correctos. El sistema informa al usuario los errores.
Post Condiciones	No aplican.

B8. Caso de Uso: Caso de Uso: Manipulación de certificados

Caso de Uso: Manipulación de certificados	
Descripción	Permite crear, visualizar, descargar, revocar o renovar un certificado firmado por la Autoridad Certificadora Local.
Actores	Administrador del Sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Certificados* en la sección *Administración de Certificados*.
Flujo Normal	El usuario presiona el botón *Nuevo.* El usuario llena el formulario desplegado para la creación de un nuevo certificado. El usuario presiona el botón *Enviar.* El sistema verifica la validez de los datos ingresados en el formulario. El sistema crea un nuevo certificado autofirmado. El sistema muestra un mensaje de éxito al usuario. El usuario hace clic sobre cualquier certificado de la lista que se muestra. El sistema muestra las opciones de Descargar/Ver/Revocar o Renovar el certificado. El hace clic sobre la opción *Descargar*. El certificado se descarga del servidor al equipo local. El hace clic sobre la opción *Ver*. El servidor muestra los datos del certificado en el explorador. El hace clic sobre la opción *Revocar.* El servidor solicita confirmación de la acción. El usuario acepta. El certificado es revocado. El hace clic sobre la opción *Renovar.* El servidor despliega un formulario de renovación del certificado. El usuario llena los campos clave del certificado y el tiempo de vida del certificado. El certificado es renovado.
Flujo Alternativo	Los datos ingresados en el formulario no son correctos. El sistema informa al usuario los errores.
Post Condiciones	No aplican.

B9. Caso de Uso: Configuración de Protocolos de Autenticación

Caso de Uso: Configuración de Protocolos de Autenticación	
Descripción	Permite configurar los protocolos de EAP.
Actores	Administrador del Sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Protocolos de Autenticación* en la sección *Administración de Certificados*.
Flujo Normal	El usuario elige el tipo de EAP que se aplicará por defecto en el proceso de autenticación. (MD5, LEAP, GTC, TLS, TTLS, PEAP). El usuario elige el número máximo de conexiones que monitoreará el servidor (Para reducir la posibilidad de ataques de DoS). El usuario marca los protocolos de autenticación que aceptará el servidor RADIUS. Si el usuario marca el modo de autenticación EAP-TLS aparece la pestaña certificados para que el usuario configure el certificado digital a usarse en este tipo de autenticación. El usuario presiona el botón aceptar. El servidor aplica los cambios en el archivo de configuración *eap.conf* de FreeRADIUS y reinicia el servicio.
Flujo Alternativo	No aplican
Post Condiciones	No aplican.

B10. Caso de Uso: Visualizar y Manipular Clientes Radius

Caso de Uso: Visualizar y Manipular Clientes Radius	
Descripción	Permite visualizar, modificar y eliminar clientes Radius.
Actores	Administrador del sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador u Operador y haber accedido a la página *Clientes Radius* pestaña *Ingreso de Clientes* dentro de la sección de *Configuración del Servidor*.
Flujo Normal	El sistema muestra una tabla con los diez primeros clientes y sus datos de dirección IP, nombre, secreto compartido, grupo al que pertenece el cliente y descripción del cliente. El actor pulsa sobre la opción *Buscar*. El sistema muestra una ventana emergente donde el actor puede ingresar el nombre de cliente a buscar y se mostrará el resultado. El actor pulsa sobre la opción *Nuevo*. El sistema muestra una fila con campos en blanco para ingresar los datos requeridos, adicionalmente muestra la opción aceptar para confirmar el ingreso del cliente. El actor pulsa sobre la opción *Siguiente* o *Anterior*. El sistema muestra una tabla con los diez siguientes o anteriores clientes respectivamente. El actor pulsa sobre el nombre de los Clientes. El sistema resalta la fila sobre la que el actor pulsó y muestra campos editables. El sistema muestra la opción actualizar o eliminar para confirmar los cambios o eliminar al cliente.
Flujo Alternativo	No existe flujo alternativo.
Post Condiciones	No existen post condiciones.

B11. Caso de Uso: Visualizar y Manipular Grupos de Clientes RADIUS

Caso de Uso: Visualizar y Manipular Grupos de Clientes Radius	
Descripción	Permite visualizar, modificar y eliminar grupos de clientes Radius.
Actores	Administrador del Sistema
Precondiciones	El cliente debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Clientes Radius* pestaña *Grupos de Clientes* dentro de la sección de *Credenciales*.
Flujo Normal	El sistema muestra una tabla con los diez primeros grupos y sus datos de nombre y descripción del grupo de clientes. El actor pulsa sobre la opción *Buscar*. El sistema muestra una ventana emergente donde el actor puede ingresar el nombre de grupo a buscar y se mostrara su resultado. El actor pulsa sobre la opción *Nuevo*. El sistema muestra una fila con campos en blanco para ser ingresados los datos requeridos, adicionalmente muestra la opción aceptar para confirmar el ingreso del grupo. El actor pulsa sobre la opción *Siguiente* o *Anterior*. El sistema muestra una tabla con los diez siguientes o anteriores grupos respectivamente. El actor pulsa sobre el nombre de los Grupos. El sistema muestra una fila resaltada con campos editables para poder modificar los datos deseados, adicionalmente muestra la opción actualizar o eliminar, para confirmar los cambios o eliminar al grupo.
Flujo Alternativo	No existe flujo alternativo
Post Condiciones	No existen post condiciones.

B12. Caso de Uso: Reiniciar Apache, Cron, FreeRADIUS, MySQL, nmbd, smbd, ssh y winbind

Caso de Uso: Reiniciar Apache, Cron, FreeRADIUS, MySQL, nmbd, smbd, ssh y winbind	
Descripción	Permite reiniciar los servicios de “SAAAS!”
Actores	Administrador del sistema.
Precondiciones	El usuario debe haber ingresado en el sistema con perfil de Administrador y haber accedido a la página *Servicios* en la sección de *Configuración del Servidor*. El servicio debe estar corriendo.
Flujo Normal	El usuario presiona el enlace *Reiniciar* junto al servicio. El sistema muestra un mensaje de alerta solicitando al usuario confirmación para proceder. El usuario confirma. El sistema envía un mensaje al servidor para que proceda a reiniciar el servicio especificado.
Flujo Alternativo	El usuario no confirma, la acción no se realiza.
Post Condiciones	No existen post condiciones.

B13. Caso de Uso: Detener Cron, FreeRADIUS, MySQL, nmbd, smbd, ssh y winbind.

Caso de Uso: Detener Cron, FreeRADIUS, MySQL, nmbd, smbd, ssh y winbind.	
Descripción	Permite detener los servicios de “SAAAS!”
Actores	Administrador del sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Servicios* en la sección de “Configuración del Servidor”. El servicio debe estar corriendo.
Flujo Normal	El usuario presiona el enlace *Detener* junto al servicio. El sistema muestra un mensaje de alerta solicitando al usuario confirmación para proceder. El usuario confirma. El sistema envía un mensaje al servidor para que proceda a detener el servicio especificado.
Flujo Alternativo	El usuario no confirma, la acción no se realiza.
Post Condiciones	El estado del servicio se cambia a *Detenido* y solo se muestra la opción *Iniciar*.

B14. Caso de Uso: Iniciar Cron, FreeRADIUS, MySQL, nmbd, smbd, ssh y winbind.

Caso de Uso: Iniciar Cron, FreeRADIUS, MySQL, nmbd, smbd, ssh y winbind.	
Descripción	Permite iniciar los servicios de “SAAAS!”
Actores	Administrador del sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Servicios* en la sección de *Configuración del Servidor*. El servicio debe estar detenido.
Flujo Normal	El usuario presiona el enlace *Iniciar* junto al servicio. El sistema muestra un mensaje de alerta solicitando al usuario confirmación para proceder. El usuario confirma. El sistema envía un mensaje al servidor para que proceda a iniciar el servicio especificado.
Flujo Alternativo	El usuario no confirma, la acción no se realiza.
Post Condiciones	El estado del servicio se cambia a *Corriendo* y se muestra la opción *Reiniciar* y *Detener*.

B15. Caso de Uso: Eliminar Usuario Web

Caso de Uso: Eliminar Usuario Web	
Descripción	Permite eliminar un usuario Web del servidor “SAAAS!”
Actores	Administrador del sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Usuarios Web* en la sección de *Configuración del Servidor*.
Flujo Normal	El usuario presiona el botón *Eliminar* junto al nombre de usuario. El sistema muestra un mensaje de alerta solicitando al usuario confirmación para proceder. El usuario confirma. El sistema elimina al usuario especificado de la base de datos.
Flujo Alternativo	El usuario no confirma, la acción no se realiza.
Post Condiciones	No existen Post Condiciones.

B16. Caso de Uso: Modificar Usuario Web

Caso de Uso: Modificar Usuario Web	
Descripción	Permite modificar la contraseña y el tipo de perfil de un usuario Web del servidor “SAAAS!”
Actores	Administrador del sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Usuarios Web* en la sección de *Configuración del Servidor*.
Flujo Normal	El usuario presiona el botón *Modificar* junto al nombre de usuario. El sistema muestra un formulario para la alteración del usuario. El usuario llena el formulario y presiona el botón *Guardar*. El sistema muestra un mensaje solicitando confirmación al usuario. El usuario confirma. El sistema guarda los cambios especificados en la base de datos.
Flujo Alternativo	El usuario no confirma, la acción no se realiza. El usuario presiona el botón *Cancelar*, la acción no se realiza y el sistema retorna al listado de usuarios.
Post Condiciones	No existen Post Condiciones.

B17. Caso de Uso: Crear Usuario Web

Caso de Uso: Crear Usuario Web	
Descripción	Permite crear un nuevo usuario Web del servidor “SAAAS!”
Actores	Usuario validado en el sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página “Usuarios Web” en la sección de “Configuración del Servidor”.
Flujo Normal	El usuario presiona el botón “Nuevo”. El sistema muestra un formulario para la creación de un nuevo usuario. El usuario llena el formulario y presiona el botón “Guardar”. El sistema muestra un mensaje solicitando confirmación al usuario. El usuario confirma. El sistema guarda los datos del nuevo usuario en la base de datos.
Flujo Alternativo	El usuario no confirma, la acción no se realiza.
Post Condiciones	No existen Post Condiciones.

B18. Caso de Uso: Visualizar y Manipular Usuarios del Repositorio Interno

Caso de Uso: Visualizar y Manipular Usuarios del Repositorio Interno	
Descripción	Permite visualizar, modificar y eliminar usuarios del repositorio de credenciales interno.
Actores	Administrador del sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador u Operador y haber accedido a la página *Usuarios Internos* pestaña *Usuarios Internos* dentro de la sección de *Credenciales*.
Flujo Normal	El sistema muestra una tabla con los diez primeros usuarios y sus datos: nombre, contraseña, grupo al que pertenece el usuario y descripción del usuario. El actor pulsa sobre la opción *Buscar*. El sistema muestra una ventana emergente donde el actor puede ingresar el nombre de usuario a buscar y se mostrará su resultado. El actor pulsa sobre la opción *Nuevo*. El sistema muestra una fila con campos en blanco para ser ingresados los datos requeridos, adicionalmente muestra la opción aceptar para confirmar el ingreso del usuario. El actor pulsa sobre la opción *Siguiente* o *Anterior*. El sistema muestra una tabla con los diez siguientes o anteriores usuarios respectivamente. El actor pulsa sobre el nombre de los Usuarios. El sistema muestra una fila resaltada con campos editables para ser modificados los datos deseados, adicionalmente muestra la opción actualizar o eliminar para confirmar los cambios o eliminar al usuario.
Flujo Alternativo	No existe flujo alternativo.
Post Condiciones	No existen post condiciones.

B19. Caso de Uso: Visualizar y Manipular Grupos de Usuarios Radius del Repositorio Interno.

Caso de Uso: Visualizar y Manipular Grupos de Usuarios Radius del Repositorio Interno	
Descripción	Permite visualizar, modificar y eliminar grupos de usuarios Radius del repositorio de credenciales interno.
Actores	Administrador del sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Usuarios Internos* pestaña *Grupos de Usuarios* dentro de la sección de *Credenciales*.
Flujo Normal	El sistema muestra una tabla con los diez primeros grupos y sus datos de nombre y descripción del grupo de usuarios. El actor pulsa sobre la opción *Buscar*. El sistema muestra una ventana emergente donde el actor puede ingresar el nombre de grupo a buscar y se mostrará su resultado. El actor pulsa sobre la opción *Nuevo*. El sistema muestra una fila con campos en blanco para ser ingresados los datos requeridos, adicionalmente muestra la opción aceptar para confirmar el ingreso del grupo. El actor pulsa sobre la opción Siguiente o Anterior. El sistema muestra una tabla con los diez siguientes o anteriores grupos respectivamente. El actor pulsa sobre el nombre de los Grupos. El sistema muestra una fila resaltada con campos editables para ser modificados los datos deseados, adicionalmente muestra la opción actualizar o eliminar para confirmar los cambios o eliminar al grupo.
Flujo Alternativo	No existe flujo alternativo
Post Condiciones	No existen post condiciones.

B20. Caso de Uso: Visualizar y Manipular Equipos del Repositorio Interno

Caso de Uso: Visualizar y Manipular Equipos del Repositorio Interno	
Descripción	Permite visualizar, modificar y eliminar equipos del repositorio de credenciales interno.
Actores	Administrador del Sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador u Operador y haber accedido a la página *Equipos Internos* pestaña *Ingreso de Equipos* dentro de la sección de *Credenciales.*
Flujo Normal	El sistema muestra una tabla con los diez primeros registros y sus datos de dirección MAC, grupo al que pertenece y descripción del equipo. El actor pulsa sobre la opción *Buscar*. El sistema muestra una ventana emergente donde el actor puede ingresar la dirección MAC a buscar y se mostrará su resultado. El actor pulsa sobre la opción *Nuevo*. El sistema muestra una fila con campos en blanco para ser ingresados los datos requeridos, adicionalmente muestra la opción aceptar para confirmar el ingreso del equipo. El actor pulsa sobre la opción *Siguiente* o *Anterior*. El sistema muestra una tabla con los diez siguientes o anteriores equipos respectivamente. El actor pulsa sobre la dirección MAC de los Equipos. El sistema muestra una fila resaltada con campos editables para ser modificados los datos deseados, adicionalmente muestra la opción actualizar o eliminar para confirmar los cambios o eliminar al equipo.
Flujo Alternativo	No existe flujo alternativo.
Post Condiciones	No existen post condiciones.

B21. Caso de Uso: Visualizar y Manipular Grupos de Equipos del Repositorio Interno

Caso de Uso: Visualizar y Manipular Grupos de Equipos del Repositorio Interno	
Descripción	Permite visualizar, modificar y eliminar grupos de equipos del repositorio de credenciales interno.
Actores	Administrador del sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Equipos Internos* pestaña *Grupos de Equipos* dentro de la sección de *Credenciales*.
Flujo Normal	El sistema muestra una tabla con los diez primeros grupos y sus datos de nombre y descripción del grupo de equipos. El actor pulsa sobre la opción *Buscar*. El sistema muestra una ventana emergente donde el actor puede ingresar el nombre de grupo a buscar y se mostrará su resultado. El actor pulsa sobre la opción *Nuevo*. El sistema muestra una fila con campos en blanco para ser ingresados los datos requeridos, adicionalmente muestra la opción aceptar para confirmar el ingreso del grupo. El actor pulsa sobre la opción *Siguiente* o *Anterior.* El sistema muestra una tabla con los diez siguientes o anteriores grupos respectivamente. El actor pulsa sobre el nombre de los Grupos. El sistema muestra una fila resaltada con campos editables para ser modificados los datos deseados, adicionalmente muestra la opción actualizar o eliminar para confirmar los cambios o eliminar al grupo.
Flujo Alternativo	No existe flujo alternativo
Post Condiciones	No existen post condiciones.

B22. Caso de Uso: Configuración de la Integración del Servidor con el Active Directory

Caso de Uso: Configuración de la Integración del Servidor con el Active Directory	
Descripción	Permite verificar y guardar las credenciales ingresadas para la integración del servidor con un directorio activo.
Actores	Administrador el sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador u Operador y haber accedido a la página *Active Directory* pestaña *Configuración* dentro de la sección de *Credenciales*.
Flujo Normal	El sistema muestra el dominio y el servidor DNS del servidor que a su vez pueden ser editados. Se muestra dos campos vacíos para el ingreso del usuario y contraseña con las cuales se va a realizar la integración. El actor pulsa sobre la opción *Verificar*. El sistema realiza la petición y muestra un mensaje en la página sobre el estado de la conexión, si la integración es satisfactoria se muestran dos pestañas adicionales (Grupos de Dominio y Test Usuarios). El actor pulsa sobre la opción *Guardar*. El sistema actualiza la configuración de la integración y muestra el estado de la conexión.
Flujo Alternativo	Si la conexión con el Active Directory no es satisfactoria, la página muestra un mensaje de error.
Post Condiciones	Si la integración es satisfactoria se muestran dos pestañas adicionales (Grupos de Dominio y Test Usuarios).

B23. Caso de Uso: Visualizarlos Grupos de Domino del Active Directory.

Caso de Uso: Visualizarlos Grupos de Domino del Active Directory	
Descripción	Permite visualizar los grupos de dominio del Active Directory.
Actores	Administrador del Sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Active Directory* pestaña Grupos de Dominio dentro de la sección de *Credenciales*. Integración satisfactoria con el Active Directory.
Flujo Normal	El sistema muestra una tabla con los veinte primeros grupos. El actor pulsa sobre la opción *Siguiente* o *Anterior*. El sistema muestra una tabla con los veinte siguientes o anteriores grupos respectivamente.
Flujo Alternativo	No existe flujo alternativo
Post Condiciones	No existen post condiciones.

B24. Caso de Uso: Comprobar los Usuarios de Domino del Active Directory

Caso de Uso: Comprobar los Usuarios de Domino del Active Directory	
Descripción	Permite comprobar los usuarios de dominio del Active Directory.
Actores	Administrador del Sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Active Directory* pestaña *Grupos de Dominio* dentro de la sección de *Credenciales*. Integración satisfactoria con el Active Directory.
Flujo Normal	El sistema muestra dos campos vacíos para ingresar el usuario y la contraseña. El actor pulsa sobre la opción *Buscar*. El sistema muestra una tabla con los grupos de dominio a los que pertenece el usuario. El actor pulsa sobre la opción *Verificar*. El sistema muestra un mensaje indicando si el usuario existe o no y si la contraseña es correcta.

Caso de Uso: Comprobar los Usuarios de Domino del Active Directory	
Flujo Alternativo	No existe flujo alternativo
Post Condiciones	No existen post condiciones.

B25. Caso de Uso: Configuración de la Integración del Servidor con un servidor LDAP

Caso de Uso: Configuración de la Integración del Servidor con un servidor LDAP	
Descripción	Permite verificar y guardar las credenciales y parámetros ingresados para la integración del servidor con un servidor LDAP.
Actores	Administrador del sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador u Operador y haber accedido a la página *LDAP* pestaña *Configuración* dentro de la sección de *Credenciales*.
Flujo Normal	El sistema muestra varios cuadros de texto, si existe una configuración previa, los mostrará con los datos almacenados anteriormente los mismos que pueden ser editados. El actor pulsa sobre la opción *Verificar*. El sistema realiza la petición y muestra un mensaje en la página sobre el estado de la conexión, si la integración es satisfactoria se muestran dos pestañas adicionales (Grupos de Dominio y Test Usuarios). El actor pulsa sobre la opción *Guardar*. El sistema actualiza la configuración de la integración y muestra el estado de la conexión.
Flujo Alternativo	Si la integración falla, el sistema muestra un mensaje de error al usuario.
Post Condiciones	Si la integración es satisfactoria se muestran dos pestañas adicionales (Grupos de Dominio y Test Usuarios).

B26. Caso de Uso: Visualizar los Grupos de Domino del servidor LDAP

Caso de Uso: Visualizar los Grupos de Domino del servidor LDAP	
Descripción	Permite visualizar los grupos de dominio del servidor LDAP.
Actores	Administrador del Sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *LDAP* pestaña *Grupos de Dominio* dentro de la sección de *Credenciales*. Integración satisfactoria con el servidor LDAP.
Flujo Normal	El sistema muestra una tabla con los veinte primeros grupos. El actor pulsa sobre la opción *Siguiente* o *Anterior.* El sistema muestra una tabla con los veinte siguientes o anteriores grupos respectivamente.
Flujo Alternativo	No existe flujo alternativo
Post Condiciones	No existen post condiciones.

B27. Caso de Uso: Comprobar los Usuarios de Domino del servidor LDAP

Caso de Uso: Comprobar los Usuarios de Domino del servidor LDAP	
Descripción	Permite comprobar los usuarios de dominio del servidor LDAP.
Actores	Administrador del Sistema
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *LDAP* pestaña *Grupos de Dominio* dentro de la sección de *Credenciales*. Integración satisfactoria con el servidor LDAP.
Flujo Normal	El sistema muestra dos campos vacíos para ingresar el usuario y la contraseña. El actor pulsa sobre la opción *Buscar*. El sistema muestra una tabla con los grupos de domino a los que pertenece el usuario. El actor pulsa sobre la opción *Verificar*. El sistema muestra un mensaje indicando si el usuario existe o no y si la contraseña es correcta.
Flujo Alternativo	No existe flujo alternativo.

Caso de Uso: Comprobar los Usuarios de Domino del servidor LDAP	
Post Condiciones	No existen post condiciones.

B28. Caso de Uso: Visualizar y Manipular los Repositorios asociados a una Regla de Autenticación

Caso de Uso: Visualizar y Manipular los Repositorios asociados a una Regla de Autenticación	
Descripción	Permite visualizar, modificar y eliminar los repositorios asociados a una regla de autenticación.
Actores	Administrador del Sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Reglas de Autenticación* pestaña *Repositorios* dentro de la sección de *Políticas de Acceso.*
Flujo Normal	El sistema muestra una tabla listando las reglas de autenticación y los repositorios correspondientes a la regla seleccionada. El actor pulsa sobre el nombre de una regla. El sistema muestra debajo de ella dos cuadros en los que se indica los repositorios asociados y los disponibles en los cuales el actor puede elegir que repositorios desea utilizar. El actor pulsa sobre la opción *Aplicar*. El sistema graba la configuración realizada y muestra la pantalla actualizada del estado de las reglas.
Flujo Alternativo	No existe flujo alternativo.
Post Condiciones	No existen post condiciones.

B29. Caso de Uso: Visualizar y Manipular las condiciones asociadas a una Regla de Autenticación

Caso de Uso: Visualizar y Manipular las condiciones asociadas a una Regla de Autenticación	
Descripción	Permite crear, visualizar, modificar y eliminar las condiciones asociadas a una regla de autenticación.
Actores	Administrador del sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Reglas de Autenticación* pestaña *Edición de Reglas* dentro de la sección de *Políticas de Acceso*.
Flujo Normal	El sistema muestra una tabla listando las reglas de autenticación adicionalmente las condiciones correspondientes a la regla seleccionada. El actor pulsa sobre el nombre de una regla. El sistema muestra debajo todas las condiciones asociadas en las cuales el actor puede elegir qué condiciones modificar, eliminar o agregar. El actor pulsa sobre la opción *Nueva Regla*. El sistema muestra los campos vacíos para el ingreso de la primera condición de esta regla. El actor pulsa sobre la opción *Eliminar*. El sistema muestra una alerta de confirmación para eliminar por completo la regla.
Flujo Alternativo	No existe flujo alternativo.
Post Condiciones	No existen post condiciones.

B30. Caso de Uso: Visualizar y Manipular los Perfiles y sus Atributos asociados

Caso de Uso: Visualizar y Manipular los Perfiles y sus Atributos asociados	
Descripción	Permite visualizar, modificar y eliminar los perfiles y sus atributos asociados.
Actores	Administrador del sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Perfiles de Red* pestaña *Perfiles* dentro de la sección de *Políticas de Acceso*.
Flujo Normal	El sistema muestra una tabla listando los Perfiles de Red adicionalmente los atributos asociados al perfil seleccionado. El actor pulsa sobre el nombre de un perfil. El sistema muestra debajo todos los atributos asociadas en los cuales el actor puede elegir qué condiciones modificar, eliminar o agregar. El actor pulsa sobre la opción *Nueva Regla*. El sistema muestra los campos vacíos para el ingreso de la primera condición de esta regla. El actor pulsa sobre la opción *Eliminar*. El sistema muestra una alerta de confirmación para eliminar por completo la regla.
Flujo Alternativo	No existe flujo alternativo.
Post Condiciones	No existen post condiciones.

B31. Caso de Uso: Visualizar el diccionario de atributos de FreeRADIUS

Caso de Uso: Visualizar el diccionario de atributos de FreeRADIUS	
Descripción	Permite visualizar el diccionario de atributos de FreeRADIUS.
Actores	Administrador del sistema.
Precondiciones	El usuario puede debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Perfiles de Red* pestaña *Diccionario de Atributos* dentro de la sección de *Políticas de Acceso*.
Flujo Normal	El sistema muestra un combo box en donde se elige el diccionario a visualizar de acuerdo al valor seleccionado el sistema procede a recuperar el archivo correspondiente.
Flujo Alternativo	No existe flujo alternativo.
Post Condiciones	No existen post condiciones.

B32. Caso de Uso: Visualizar y Manipular los Perfiles asociados a una Regla de Autorización

Caso de Uso: Visualizar y Manipular los Perfiles asociados a una Regla de Autorización	
Descripción	Permite visualizar, modificar y eliminar los perfiles asociados a una regla de autorización.
Actores	Administrador del sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Reglas de Autorización* pestaña *Perfiles* dentro de la sección de *Políticas de Acceso*.
Flujo Normal	El sistema muestra una tabla listando las reglas de autorización adicionalmente los perfiles correspondientes a la regla seleccionada. El actor pulsa sobre el nombre de una regla. El sistema muestra dos cuadros en los que se indica los perfiles asociados y los disponibles en los cuales el actor puede elegir que perfiles desean utilizar. El actor pulsa sobre la opción *Aplicar*. El sistema guarda la configuración realizada y muestra la pantalla actualizada del estado de las reglas.
Flujo Alternativo	No existe flujo alternativo.
Post Condiciones	No existen post condiciones.

B33. Caso de Uso: Visualizar y Manipular las condiciones asociadas a una Regla de Autorización

Caso de Uso: Visualizar y Manipular las condiciones asociadas a una Regla de Autorización	
Descripción	Permite crear, visualizar, modificar y eliminar las condiciones asociadas a una regla de autorización.
Actores	Administrador del sistema.
Precondiciones	El usuario debe haberse validado en el sistema con perfil de Administrador y haber accedido a la página *Reglas de Autorización* pestaña *Edición de Reglas* dentro de la sección de *Políticas de Acceso*.
Flujo Normal	El sistema muestra una tabla listando las reglas de autorización y las condiciones correspondientes a la regla seleccionada. El actor pulsa sobre el nombre de una regla. El sistema muestra todas las condiciones asociadas en las cuales el actor puede elegir qué condiciones modificar, eliminar o agregar. El actor pulsa sobre la opción *Nueva Regla*. El sistema muestra los campos vacíos para el ingreso de la primera condición de esta regla. El actor pulsa sobre la opción *Eliminar*. El sistema muestra una alerta de confirmación para eliminar por completo la regla.
Flujo Alternativo	No existe flujo alternativo.
Post Condiciones	No existen post condiciones.

B34. Caso de Uso: Generar Reporte de Eventos del Sistema

Caso de Uso: Generar Reporte de Eventos del Sistema	
Descripción	Permite visualizar un reporte del estado y funcionamiento del servidor FreeRADIUS.
Actores	Usuario.
Precondiciones	El usuario debe haberse iniciado sesión (con cualquiera de los perfiles existentes) y accedido a la página *Eventos del Sistema* dentro de la sección *Visor de Eventos*.
Flujo Normal	Por defecto, el sistema muestra la fecha actual y muestra un reporte de los eventos del sistema de las últimas 24 horas. El usuario modifica la Fecha de Inicio del Reporte. El usuario modifica la Fecha de Fin del Reporte. El usuario modifica el tipo de mensajes que desea que aparezcan en el Reporte (Info/Error/Todos los Mensajes). El usuario presiona la tecla Actualizar. El reporte de Eventos del Sistema personalizado se muestra en pantalla.
Flujo Alternativo	Si los criterios de búsqueda no coinciden con la información disponible en el servidor, se muestra el mensaje: “La búsqueda no produjo resultados”
Post Condiciones	No aplica.

B35. Caso de Uso: Personalizar Gráfico Solicitudes de Autenticación VS NAS/Solicitudes de Autenticación VS Usuario

Caso de Uso: Personalizar Gráfico Solicitudes de Autenticación VS NAS/Solicitudes de Autenticación VS Usuario	
Descripción	Permite visualizar un gráfico de barras de los 10 NAS/Usuarios que han realizado el mayor número de solicitudes de autenticación en un rango de tiempo personalizado.
Actores	Usuario.
Precondiciones	El usuario debe haber iniciado sesión en el sistema (con cualquiera de los perfiles existentes) y haber accedido a la página *Registro* dentro de la sección *Visor de Eventos*.
Flujo Normal	Por defecto, el sistema calcula la fecha actual y muestra gráficos con resultados basados en las últimas 24 horas. El usuario modifica la Fecha de Inicio del Reporte. El usuario modifica la Fecha de Fin del Reporte. El usuario presiona la tecla Actualizar. El gráfico de barras se muestra en pantalla.
Flujo Alternativo	No aplica
Post Condiciones	No aplica.

B36. Caso de Uso: Personalizar y Descargar Reportes del Registro AAA

Caso de Uso: Personalizar y Descargar Reportes del Registro AAA	
Descripción	Permite visualizar reportes personalizados de: Solicitudes de Autenticación, Sesiones por Tiempo y Sesiones por Tráfico
Actores	Usuario.
Precondiciones	El usuario debe haber iniciado sesión en el sistema (con cualquiera de los perfiles existentes) y haber accedido a la página *Registro AAA* a través de la página *Registro* dentro de la sección *Visor de Eventos*.
Flujo Normal	Por defecto, el sistema muestra la fecha actual y muestra un reporte con datos de las últimas 24 horas. El usuario modifica la Fecha de Inicio del Reporte. El usuario modifica la Fecha de Fin del Reporte. El usuario modifica el campo Nombre de Usuario. El usuario modifica el campo IP del NAS. El usuario presiona el botón Actualizar. El reporte personalizado con los filtros indicados por el usuario se muestra en pantalla.
Flujo Alternativo	El usuario presiona el botón Descargar; el reporte se genera en un archivo .xls
Post Condiciones	No aplica.

B37. Caso de Uso: Fijar Hora de Fin de Sesiones Abiertas

Caso de Uso: Fijar Hora de Fin de Sesiones Abiertas.	
Descripción	Permite fijar la hora actual en el campo de hora de finalización de la sesión en las sesiones abiertas.
Actores	Usuario de la Interfaz Web.
Precondiciones	El usuario debe haber iniciado sesión en el sistema (con cualquiera de los perfiles existentes) y haber accedido a la página *Registro AAA* a través de la página *Registro* dentro de la sección *Visor de Eventos*.
Flujo Normal	Por defecto, el sistema muestra una lista de las sesiones con hora de finalización nula. El usuario presiona el botón *Cerrar Sesión* y la hora de finalización de la sesión se cambia a la hora actual. El sistema muestra una lista actualizada de las sesiones con hora de finalización nula.
Flujo Alternativo	No aplica
Post Condiciones	No aplica

B38. Caso de Uso: Crear/Usar/Eliminar/Descargar respaldos de la Configuración del Servidor

Caso de Uso: Crear/Usar/Eliminar/Descargar respaldos de la Configuración del Servidor	
Descripción	Permite crear respaldos de la configuración del Servidor.
Actores	Usuario.
Precondiciones	El usuario debe haber iniciado sesión en el sistema (con cualquiera de los perfiles existentes) y haber accedido a la página *Recuperación y Respaldos* a través de la página *Registro* dentro de la sección *Visor de Eventos*.
Flujo Normal	El sistema muestra la lista de los archivos de respaldo existentes. El usuario presiona el botón *Nuevo Respaldo* y se crea un nuevo archivo .tgz con los archivos de configuración del servidor. El usuario presiona el botón *Eliminar* y se borra del sistema al archivo .tgz seleccionado. El usuario presiona el botón *Usar* y se borra la configuración actual del servidor FreeRADIUS para reemplazarla con el archivo .tgz seleccionado después de descomprimirlo. El usuario presiona el botón *Descargar* y el archivo .tgz seleccionado se transfiere al equipo cliente.
Flujo Alternativo	No aplica.
Post Condiciones	No aplica.

ANEXO C

Guía de Implementación del Servidor AAA Simplificado "SAAAS!"

Guía de Implementación del Servidor AAA Simplificado: SAAAS!

1. Presentando a SAAAS!

El Servidor AAA Simplificado, SAAAS!, es un prototipo de Meta Distribución basada en Debian GNU/Linux 6.0 (nombre clave Squeeze), cuyo objetivo es facilitar la implementación de servicios AAA en redes pequeñas a través de una consola de administración Web para FreeRADIUS.

2. Arranque de la Meta Distribución SAAAS!

El presente documento contiene las instrucciones de instalación de SAAAS!, para PCs de 32 bits, arquitectura i386.

SAAAS! se distribuye a través de un CD arrancable en modo Live y puede instalarse en disco ejecutando el instalador de la Meta Distribución cuando está corriendo en modo Live.

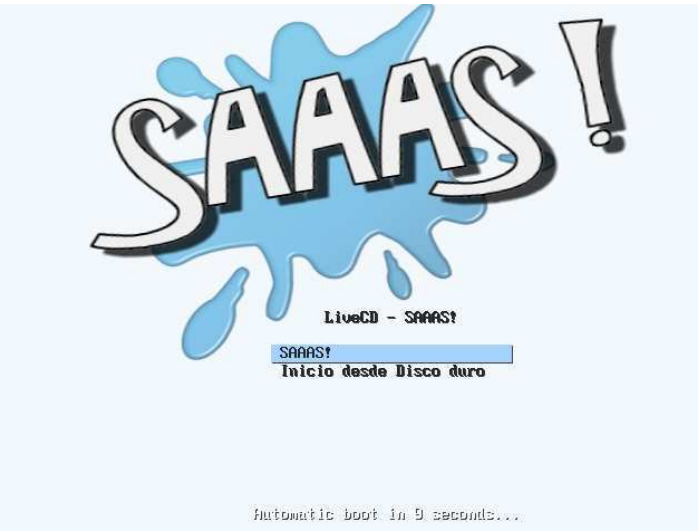


Ilustración 1 LiveCD de SAAAS!

La Meta Distribución “SAAAS!” viene con parámetros de usuario y contraseña configurados por defecto. Estos parámetros se detallan en la Tabla 1. y deberán ser cambiados inmediatamente después de la instalación en disco de la Meta Distribución.

Contraseña del usuario “root”	saaas
Usuario Web	admin
Contraseña del Usuario Web	saaas!

Tabla 1. Usuarios por defecto de “SAAAS!”

3. Usando el instalador de SAAAS!

Para instalar la Meta Distribución SAAAS! en el disco duro, el usuario debe loguearse como root en el LiveCD y ejecutar el comando “setup” en el prompt. El diálogo de instalación iniciará después de eso.

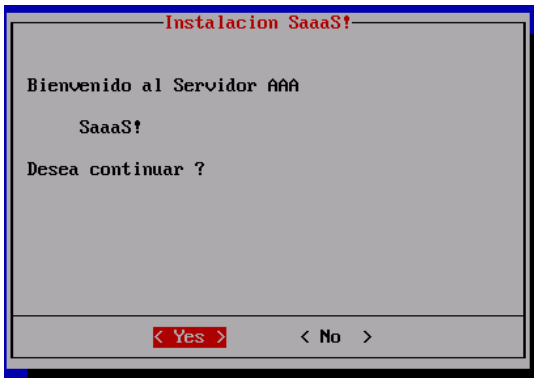


Ilustración 2 Diálogo de Instalación

El instalador de “SAAAS!” ofrece dos alternativas para el particionamiento del disco, para usuarios poco familiarizados con Linux se recomienda seleccionar el particionamiento automático, que sin intervención del usuario creará en el disco duro la partición SWAP y la partición destinada para el sistema de archivos LINUX.

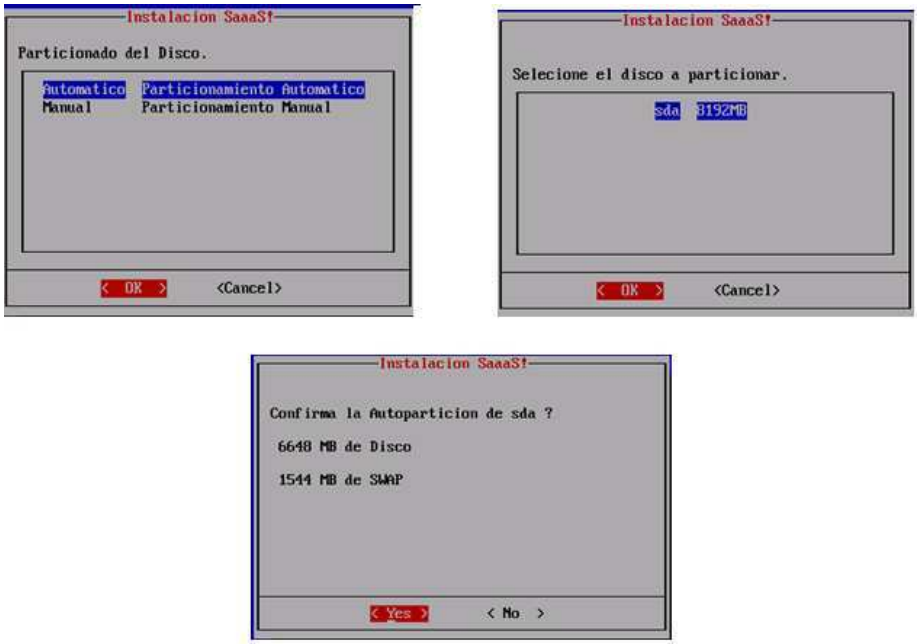


Ilustración 3 Partición automática del Disco

En caso de escoger el particionamiento manual se abrirá la herramienta *cfdisk* de Linux, el usuario debe crear por lo menos una partición SWAP y una partición para el sistema operativo.

Ilustración 4 Particionamiento Manual del Disco.

Una vez creadas las particiones, el diálogo de instalación solicita al usuario la elección de las particiones disponibles para SWAP y para el sistema de archivos.

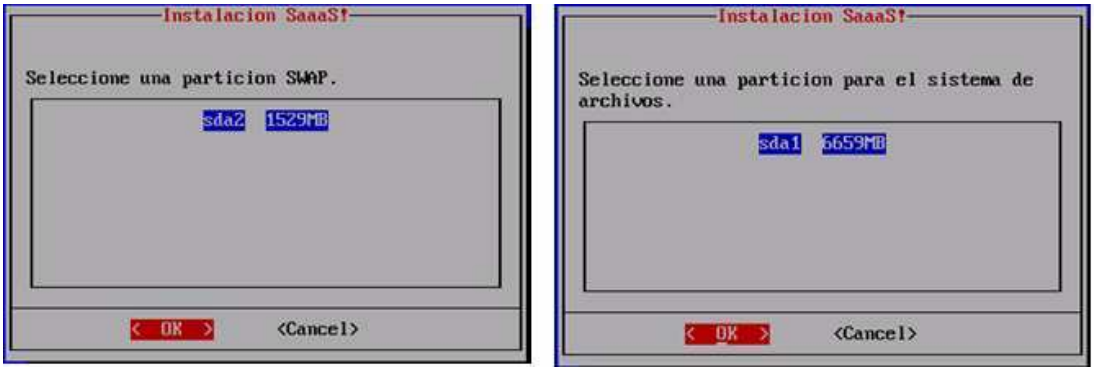


Ilustración 5 Selección de particiones durante la instalación.

A partir de este momento, se le solicitará los datos para el login del root y un usuario local, además del nombre del equipo. Una vez que haya completado todos los campos solicitados, el usuario debe confirmar que desea proceder con la instalación y el proceso de copiado del sistema de archivos en el disco duro iniciará. Una vez finalizada la instalación se le solicitará al usuario que reinicie el equipo.

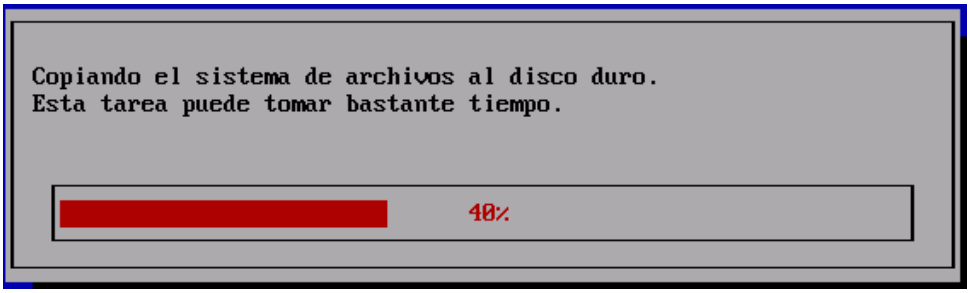


Ilustración 6 Diálogo de Información del proceso de instalación de “SAAAS!”

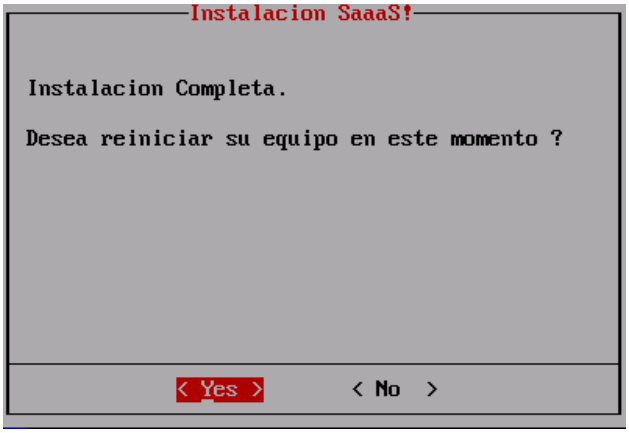


Ilustración 7 Mensaje de éxito de la instalación de “SAAAS!”

4. Empezando a usar “SAAAS!”

Una vez instalada la meta distribución, el ingreso al módulo de Administración Web se lo realizará apuntando al url http://192.168.1.5 en el explorador. Se le solicitará al usuario sus credenciales para poder hacer uso de la aplicación. El usuario y password por defecto son: *admin* y *saaas!*

Una vez autenticado se muestra la pantalla principal y el menú donde se encuentran todas las secciones.

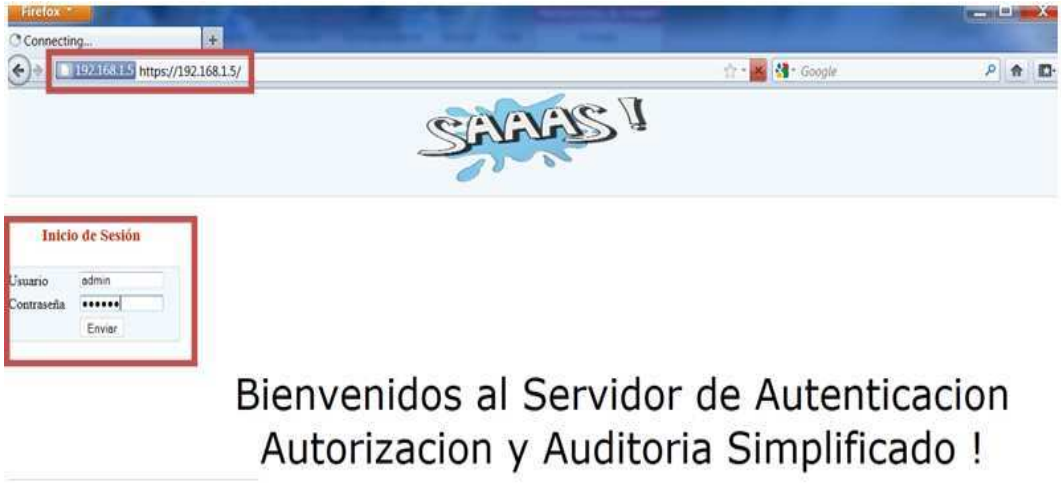


Ilustración 8 Mensaje de de autenticación de usuarios Web de “SAAAS!”

4.1. Configuración Básica

4.1.1. *Información General*

La página Información General consta de 4 pestañas informativas en las que al presionar sobre ellas muestra la información del estado de salud del servidor.

Ilustración 9 Información General “SAAAS!”

4.1.2. *Configuración Básica*

La página Configuración Básica consta de cuatro pestañas. En la pestaña General se puede editar la información referente al nombre del equipo, el sufijo de dominio, y el servidor DNS primario y secundario que se utilizará para realizar las consultas DNS (Si se va a realizar la integración del servidor con un Active Directory el servidor DNS primario deberá ser la dirección IP del controlador de dominio).

Para realizar la edición se da click en el icono , automáticamente cambia la presentación de la información a un cuadro de texto editable. Para guardar esta información se da click en el icono , donde se presentara una ventana emergente pidiendo la confirmación de esta accion.

Ilustración 10 Configuración Básica "SAAAS!"

En la pestaña Interfaces de Red se puede editar la configuración de las tarjetas de red del servidor, al presionar sobre el icono el campo Tipo cambia a un cuadro de selección con las opciones: Dirección IP Estática o Dinámica. Si la opción escogida es Estática se muestran como campos editables la Dirección IP, Máscara de Red y Gateway. Para guardar esta información se da click en el icono donde se presenta una ventana emergente pidiendo la confirmación de esta acción.

Ilustración 11 Configuración Básica (Interfaces de Red) "SAAAS!"

En la pestaña Rutas se visualiza la información referente a la tabla de enrutamiento del servidor.

Ilustración 12 Configuración Básica (Rutas) “SAAAS!”

En la pestaña Nueva Interfaz Virtual se puede agregar interfaces virtuales al servidor, la pestaña muestra un cuadro de selección con el nombre de las interfaces reales, en este cuadro se escoge la interfaz a la que hara referencia la interfaz virtual, adicional se muestra 2 cuadros editables para la Dirección IP y la Máscara de Red. Para guardar esta información se da click en el icono donde se presenta una ventana emergente pidiendo la confirmación de esta acción.

Ilustración 13 Configuración Básica (Nueva Interfaz Virtual) “SAAAS!”

4.2. Administración de Certificados

4.2.1. *Configuración Básica*

La página Configuración Básica permite la generación del certificado de la Autoridad Certificadora por lo que muestra una tabla con los siguientes campos editables:

- Organización: Nombre de la organización en la que esta instalado el servidor ¨SAAAS!¨.
- Departamento: Nombre del departamento administrativo encargado del servidor ¨SAAAS!¨.
- Alias: Nombre del servidor ¨SAAAS!¨.
- E-mail: Dirección de correo del personal encargado del servidor ¨SAAAS!¨.
- Ciudad: Nombre de la ciudad en la que se instalo el servidor ¨SAAAS!¨.
- Estado/Provincia: Nombre del estado o provincia en la que se instalo el servidor ¨SAAAS!¨.
- País: Dos letras que indiquen el país en el que se instalo el servidor ¨SAAAS!¨.
- Contraseña: Valor mínimo 8 caracteres.
- Duracion: Duracion del certificado en años.
- Tamaño de llave: nivel de encripción que maneja el certificado.

Para generar el certificado se da click en el botón Enviar.

Ilustración 14 Administración de Certificados (Configuración Básica) “SAAAS!”

4.2.2. *Certificados*

La página Certificados permite la generación de un certificado firmado por la Autoridad Certificadora, adicionalmente permite una vista previa del certificado, la posibilidad de descargarlo, revocarlo o renovarlo, se da un click sobre el nombre del certificado, la fila seleccionada cambiara de color, mostrando las opciones en la parte inferior de la página.

Ilustración 15 Administración de Certificados (Certificados) "SAAAS!"

Si se presiona sobre la opción Nuevo se muestra una tabla con los siguientes campos editables:

- ✓ Nombre: Nombre del servidor ¨SAAAS!¨.
- ✓ Organización: Nombre de la organización en la que esta instalado el servidor ¨SAAAS!¨.
- ✓ Departamento: Nombre del departamento administrativo encargado del servidor ¨SAAAS!¨.
- ✓ E-mail: Dirección de correo del personal encargado del servidor ¨SAAAS!¨.
- ✓ Ciudad: Nombre de la ciudad en la que se instalo el servidor ¨SAAAS!¨.
- ✓ Estado/Provincia: Nombre del estado o provincia en la que se instalo el servidor ¨SAAAS!¨.
- ✓ País: Dos letras que indiquen el país en el que se instalo el servidor ¨SAAAS!¨.
- ✓ Contraseña: Valor mínimo 8 caracteres.
- ✓ Duracion: Duracion del certificado en años.
- ✓ Tamaño de llave: Nivel de encripción que maneja el certificado.
- ✓ Tipo de certificado: El propósito del certificado (Cliente o Servidor).

Para generar el certificado se da click en el botón Enviar.

Ilustración 16 Administración de Certificados (Certificado Nuevo) “SAAAS!”

Si se presiona sobre la opción Ver se muestra una tabla con los datos del certificado.

Si se presiona sobre la opción Dercargar se muestra una ventana que permite descargar el certificado.

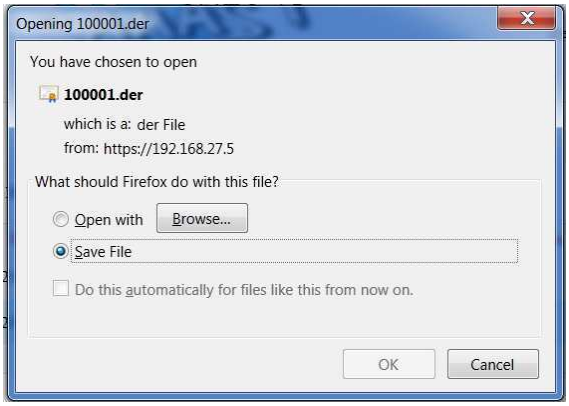


Ilustración 17 Administración de Certificados (Descargar Certificado) “SAAAS!”

Si se presiona sobre la opción Revocar se muestra una tabla con la información del certificado y un mensaje de confirmación de la revocación.

Ilustración 18 Administración de Certificados (Revocar Certificado) "SAAAS!"

Si se presiona sobre la opción Renovar se muestra una tabla con la información del certificado y un mensaje de confirmación de la renovación.

Ilustración 19 Administración de Certificados (Renovar Certificado) "SAAAS!"

4.2.3. *Protocolos de Autenticación*

La página Protocolos de Autenticación permite la configuración de los protocolos de autenticación que se va a utilizar a nivel de la plataforma de EAP, basta con marcar los cuadros de chequeo correspondientes.

Ilustración 20 Administración de Certificados (Protocolos de Autenticación) "SAAAS!"

Adicionalmente permite visualizar y configurar los certificados utilizados para los tuneles TLS, tanto el certificado del servidor con su clave compartida, su clave privada y el certificado de la Autoridad Certificadora.

Al dar click sobre el botón Aplicar guardaremos la configuración.

Ilustración 21 Administración de Certificados (Protocolos de Autenticación - Certificados) "SAAAS!"

4.3. Configuración del Servidor

4.3.1. *Clientes Radius (NAS)*

La página Clientes Radius consta de 2 pestañas, la pestaña Ingreso de Cilentes permite la manipulación de los datos de los clientes (Switches, Routers, Access points, etc), al dar un click en Nuevo se muestra una fila con campos vacios donde ingresaremos los siguientes datos:

- ✓ Ip/DNS: Direccion IP o nombre de dominio del cliente.
- ✓ Nombre: Nombre del equipo.
- ✓ Secreto: Secreto compartido entre el servidor “SAAAS!” y el cliente.
- ✓ Grupo: Grupo al que pertenece el equipo.
- ✓ Descripción: Una breve descripción del equipo.

Adicionalmente al dar un click sobre el campo Ip/DNS correspondiente a cada registro, la fila cambia de color, los campos se vuelven editables y se muestran las opciones actualizar y eliminar en la parte inferior.

Ilustración 22 Configuración del Servidor (Clientes Radius) “SAAAS!”

La pestaña Grupo de Cilentes permite la manipulación de los datos de los grupos de clientes, al dar un click en Nuevo se muestra una fila con campos vacios donde ingresaremos los siguientes datos:

- ✓ Nombre: Nombre del grupo.
- ✓ Descripción: Una breve descripción del grupo.

Adicionalmente al dar un click sobre el campo nombre correspondiente a cada registro, la fila cambia de color, los campos se vuelven editables y se muestran las opciones actualizar y eliminar en la parte inferior.

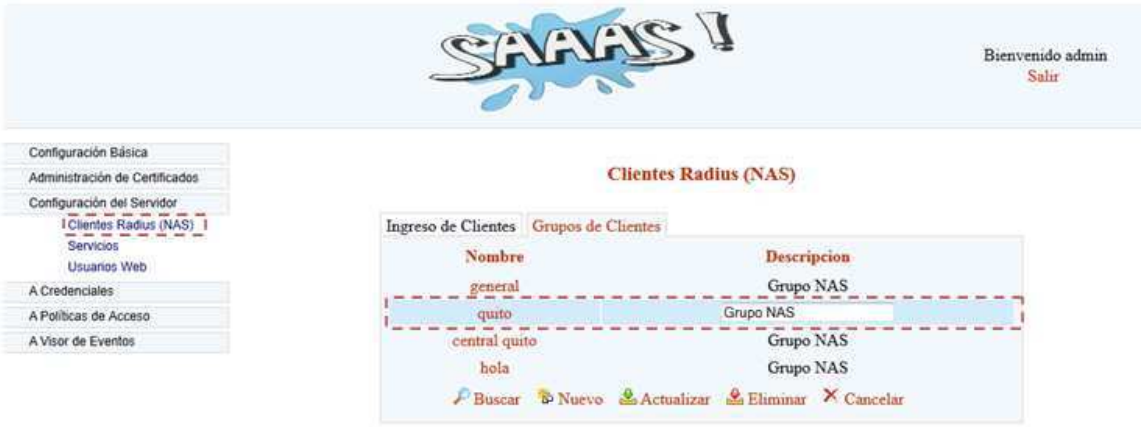


Ilustración 23 Configuración del Servidor (Grupos de Clientes Radius) "SAAAS!"

4.3.2. *Servicios*

La página Servicios permite manipular el estado de los servicios que utiliza el servidor "SAAAS!", si el servicio esta Corriendo se muestran las opciones de Reiniciar y Parar, si el servicio esta Detenido se muestra la opción de Iniciar, para todas las opciones se muestra una ventana de confirmación.

Ilustración 24 Configuración del Servidor (Servicios) "SAAAS!"

4.3.3. *Usuarios Web*

La página Usuarios Web muestra una tabla con el nombre del usuario y el perfil de administración que posee, adicional muestra el icono , que permite modificar la contraseña y el perfil del usuario, el icono para eliminar al

usuario y el icono Nuevo Usuario que al presionarlo muestra una tabla con los siguientes campos:

- Nombre de Usuario: Usuario con el que se ingresara a la aplicación.
- Perfil: Tipo de Perfil, este puede ser Administrador, Monitoreo y Operador
- Contraseña: La contraseña que se le pedirá al usuario cada ves que ingrese a la aplicación.

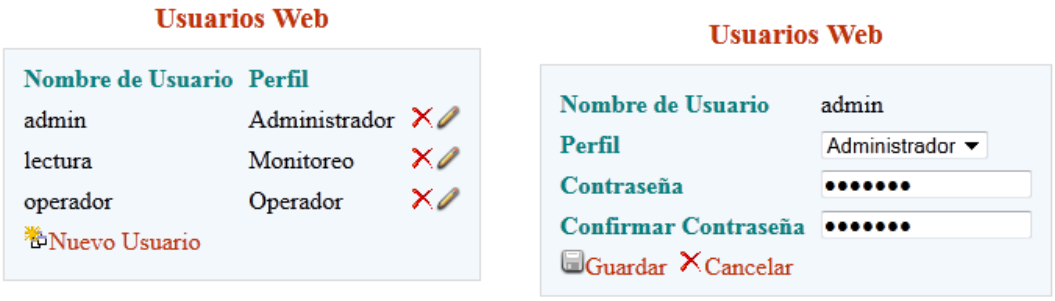


Ilustración 25 Configuración del Servidor (Usuarios Web) “SAAAS!”

4.4. **Credenciales**

4.4.1. *Usuarios Internos*

La página Usuarios Internos consta de 2 pestañas, la pestaña Ingreso de Usuarios permite la manipulación de los datos de los usuarios internos, al dar un click en Nuevo se muestra una fila con campos vacios donde ingresaremos los siguientes datos:

- ✓ Nombre: Nombre del usuario que será requerida en el proceso de autenticación.
- ✓ Contraseña: Contraseña que será requerida en el proceso de autenticación.
- ✓ Grupo: Grupo al que pertenece el usuario.
- ✓ Descripción: Una breve descripción del usuario.

Adicionalmente al dar un click sobre el nombre del usuario, la fila cambia de color, los campos se vuelven editables y se muestran las opciones actualizar y eliminar en la parte inferior.

Ilustración 26 Credenciales (Usuarios Internos) “SAAAS!”

La pestaña Grupos de Usuarios permite la manipulación de los datos de los grupos de usuarios, al dar un click en Nuevo se muestra una fila con campos vacios donde ingresaremos los siguientes datos:

- ✓ Nombre: Nombre del grupo.
- ✓ Descripción: Una breve descripción del grupo.

Adicionalmente al dar un click sobre el campo nombre correspondiente a cada registro, la fila cambia de color, los campos se vuelven editables y se muestran las opciones actualizar y eliminar en la parte inferior.

Ilustración 27 Credenciales (Grupos de Usuarios) “SAAAS!”

4.4.2. *Equipos Internos*

La página Equipos Internos consta de 2 pestañas, la pestaña Ingreso de Equipos permite la manipulación de los datos de los equipos que se van a autenticar con su dirección MAC (Impresoras, Telefonos IP, etc.), al dar un

click en Nuevo se muestra una fila con campos vacios donde ingresaremos los siguientes datos:

- ✓ Dirección MAC: Dirección física expresada en pares hexadecimales separados por un guión medio.
- ✓ Grupo: Grupo al que pertenece el equipo.
- ✓ Descripción: Una breve descripción del equipo.

Adicionalmente al dar un click sobre la dirección MAC, la fila cambia de color, los campos se vuelven editables y se muestran las opciones actualizar y eliminar en la parte inferior.

Ilustración 28 Credenciales (Equipos Internos) “SAAAS!”

La pestaña Grupos de Equipos permite la manipulación de los datos de los grupos de equipos, al dar un click en Nuevo se muestra una fila con campos vacios donde ingresaremos los siguientes datos:

- ✓ Nombre: Nombre del grupo.
- ✓ Descripción: Una breve descripción del grupo.

Adicionalmente al dar un click sobre el campo nombre correspondiente a cada registro, la fila cambia de color, los campos se vuelven editables y se muestran las opciones actualizar y eliminar en la parte inferior.

Ilustración 29 Credenciales (Grupos de Equipos) “SAAAS!”

4.4.3. *Active Directory*

La página Active Diretory permite configurar los siguientes parámetros de integración entre el servidor “SAAAS!” y el controlador de dominio:

- ✓ Dominio: nombre completo del dominio al que se va unir el servidor “SAAAS!”.
- ✓ Controlador de Dominio: Dirección IP del Controador de Dominio esta dirección remplazara al DNS primario.
- ✓ Usuario: Usuario perteneciente al dominio con el que se van a realizar las consultas al Active Directory.
- ✓ Contraseña: Contraseña del usuario con el que se van a realizar las consultas al Active Directory.

Si se da un click en el icono Verificar se realizara una comprobación de la conexión al Active Directory, al presionar el icono Guardar la configuración es guardada y adicionalmente se visualiza 2 pestañas adicionales Grupos del Dominio y Test de Usuarios.

Ilustración 30 Credenciales (Active Directory) “SAAAS!”

En la pestaña Grupos del Dominio se muestran todos los grupos globales del Active Directory.

En la pestaña Test de Usuario, si se da un click en el icono Buscar se realiza una busqueda del usuario y los grupos a los que pertenece en el Active Directory, si se da un click en el icono Verificar se realiza una comprobación tanto del usuario como de su contraseña en el Active Directory.

Ilustración 31 Credenciales (Active Directory Test Usuarios) “SAAAS!”

4.4.4. *LDAP*

La página LDAP permite configurar los siguientes parámetros de integración entre el servidor “SAAAS!” y el servidor LDAP:

- ✓ Dominio: nombre completo del dominio al que se va unir el servidor “SAAAS!”.
- ✓ Servidor LDAP: Dirección IP del Servidor LDAP.
- ✓ Puerto Servidor LDAP: Puerto TCP por el cual se realizaran las consultas al Servidor LDAP.
- ✓ Usuario: Usuario perteneciente al dominio con el que se van a realizar las consultas al servidor LDAP.
- ✓ Contraseña: Contraseña del usuario con el que se van a realizar las consultas al Active Directory.
- ✓ Grupo (objectType): Tipo de objeto del grupo en el servidor LDAP.
- ✓ Grupo (atributo): Atributo que contiene el nombre del grupo en el servidor LDAP.
- ✓ Usuario (atributo): Atributo que se encuentra registrado en el grupo al que pertenece el usuario.
- ✓ Usuario (atributo de grupo): Atributo que contiene los nombres de los grupos a los que pertenece el usuario.
- ✓ Contraseña (atributo): Atributo que contiene la contraseña en texto plano.

- ✓ Samba (LMPassword): Atributo que contiene una de las contraseñas que mantienen la integración con equipos Windows.
- ✓ Samba (NTPassword): Atributo que contiene una de las contraseñas que mantienen la integración con equipos Windows.

Si se da un click en el icono

se realizara una comprobación de la conexión al servidor LDAP, al presionar el icono

la configuración es guardada y adicionalmente se visualiza 2 pestañas adicionales Grupos del Dominio y Test de Usuarios.

Ilustración 32 Credenciales (LDAP) “SAAAS!”

En la pestaña Grupos del Dominio se muestran todos los grupos del servidor LDAP.

En la pestaña Test de Usuario, si se da un click en el icono Buscar se realiza una busqueda del usuario y los grupos a los que pertenece en el servidor LDAP, si se da un click en el icono Verificar se realiza una comprobación tanto del usuario como de su contraseña en el servidor LDAP.

Ilustración 33 Credenciales (LDAP Grupos de Dominio) “SAAAS!”

4.5. Políticas de Acceso

4.5.1. *Reglas de Autenticación*

La página Reglas de Autenticación consta de 2 pestañas, la pestaña de

Repositorios que permite indicar de que repositorios se van a autenticar los usuarios dependiendo de la regla y sus condiciones, para editar estas opciones se da click en el nombre de la regla y nos aparecen 2 cuadros de selección y 2 botones en el medio, el cuadro de la derecha será el que contenga los repositorios a evaluar, el cuadro de la izquierda será el que tenga los repositorios restantes.

Ilustración 34 Políticas de Acceso (Reglas de Autenticación Repositorios) "SAAAS!"

En la pestaña Edicion de Reglas permite crear nuevas reglas de autenticación y modificar las existentes, al dar un click en Nuevo regla se muestra una fila con campos vacios donde ingresaremos una condición a la regla, Para crear una nueva regla es necesario ingresar por lo menos una condición, los atributos de la condición pueden ser:

- NAS-IP-Address: La direccion IP administrativa del cliente.
- Service-Type: El tipo de servicio al que se esta autenticando, por ejemplo: para 802.1x el servicio es Framed, para MAB es Callback-Check, para autenticación con el cliente es Login.

Adicionalmente al dar un click sobre el nombre de una relga se despliegan todas las condiciones que contiene la regla y la opción de añadir una nueva condición a la regla Nueva condición, y al dar click sobre uno de ellas la fila cambiara de color y se muestran un cuadro de selección para el tipo de atributo, un cuadro de texto para el valor del mismo y las opciones Actualizar Eliminar .

Al presionar sobre la opción Aplicar la configuración será aplicada y grabada en el servidor.

Ilustración 35 Políticas de Acceso (Reglas de Autenticación - Edición) "SAAAS!"

Al presionar sobre la opción Aplicar la configuración será aplicada y grabada en el servidor.

4.5.2. *Perfiles de Red*

La página Perfiles de Red consta de 2 pestañas, la pestaña de Diccionario de Atributos muestra documentos del rfc donde se definen los atributos estándares y documentos de los fabricantes, la pestaña Perfiles permite crear nuevos perfiles de autorización y modificar los existentes, al dar un click en Nuevo Perfil se muestra una fila con campos vacios donde ingresaremos el nombre del perfil y un atributo, Para crear un nuevo perfil es necesario ingresar por lo menos un atributo, los atributos pueden ser:

- ✓ VLAN: Atributo que hace referencia al id de la vlan definida en el cliente, puede ser también el nombre de la vlan si el cliente lo soporta.
- ✓ Filter-id: Atributo que hace referencia al id de una lista de acceso definifa en el cliente.
- ✓ Reply-Message: Atributo que hace referencia a un mensaje que será entregado una vez que el usuario se autentique, siempre y cuando el cliente lo soporte.
- ✓ Vendor-Specific: Atributo que hace referencia a atributos propietarios definidos por cada uno de los fabricantes de los clientes. Para este tipo de atributos el ingreso será de la siguiente manera:

Vendor-Specific Cisco-Avpair := "shell:priv-lvl=15"

Adicionalmente al dar un click sobre el nombre del perfil se despliegan todos los atributos que contiene el perfil y la opción de añadir un atributo al perfil Nuevo Atributo, y al dar click sobre uno de ellos la fila cambiara de color y los campos se muestran editables mostrando un cuadro de selección para el tipo de atributo, un cuadro de texto para el valor del mismo y las opciones Actualizar Eliminar .

Al presionar sobre la opción Aplicar la configuración será aplicada y grabada en el servidor.

Ilustración 36 Políticas de Acceso (Perfiles de Red) “SAAAS!”

4.5.3. *Reglas de Autorización*

La página Reglas de Autorización consta de 2 pestañas, la pestaña de Perfiles que permite indicar que perfiles se van a aplicar a los usuarios dependiendo de la regla y sus condiciones, para editar estas opciones se da click en el nombre de la regla y nos aparecen 2 cuadros de selección y 2 botones en el medio, el cuadro de la derecha será el que contenga los perfiles a aplicar, el cuadro de la izquierda será el que tenga los perfiles restantes.

Ilustración 37 Políticas de Acceso (Reglas de Autorización Perfiles) “SAAAS!”

En la pestaña Edicion de Reglas permite crear nuevas reglas de autorización y modificar las existentes, al dar un click en Nuevo regla se muestra una fila con campos vacios donde ingresaremos una condición a la regla, Para crear una nueva regla es necesario ingresar por lo menos una condición, los atributos de la condición pueden ser:

- ✓ NAS-IP-Address: La direccion IP administrativa del cliente.
- ✓ Service-Type: El tipo de servicio al que se esta autenticando, por ejemplo: para 802.1x el servicio es Framed, para MAB es Callback-Check, para autenticación con el cliente es Login.

Adicionalmente al dar un click sobre el nombre de una relga se despliegan todas las condiciones que contiene la regla y la opción de añadir una nueva condición a la regla Nueva condición, y al dar click sobre uno de ellas la fila cambiara de color y se muestran un cuadro de selección para el tipo de atributo, un cuadro de texto para el valor del mismo y las opciones Actualizar Eliminar .

Al presionar sobre la opción Aplicar la configuración será aplicada y grabada en el servidor.

Ilustración 38 Políticas de Acceso (Reglas de Autorización - Edición) "SAAAS!"

Al presionar sobre la opción Aplicar la configuración será aplicada y grabada en el servidor.

4.6. Visor de Eventos

"SAAAS!" permite visualizar los eventos del servidor y los procesos de autenticación en la red de datos. La sección Visor de Eventos de la aplicación permite generar estos informes y también crear y usar respaldos de configuraciones del servidor anteriores. En los siguientes puntos se detallará la manera de realizar estas tareas.

4.6.1. Monitoreo

La página Monitoreo permite generar reportes de los mensajes de logs del servicio FreeRADIUS que se administra mediante "SAAAS!". Por defecto el informe muestra los logs generados el último día.

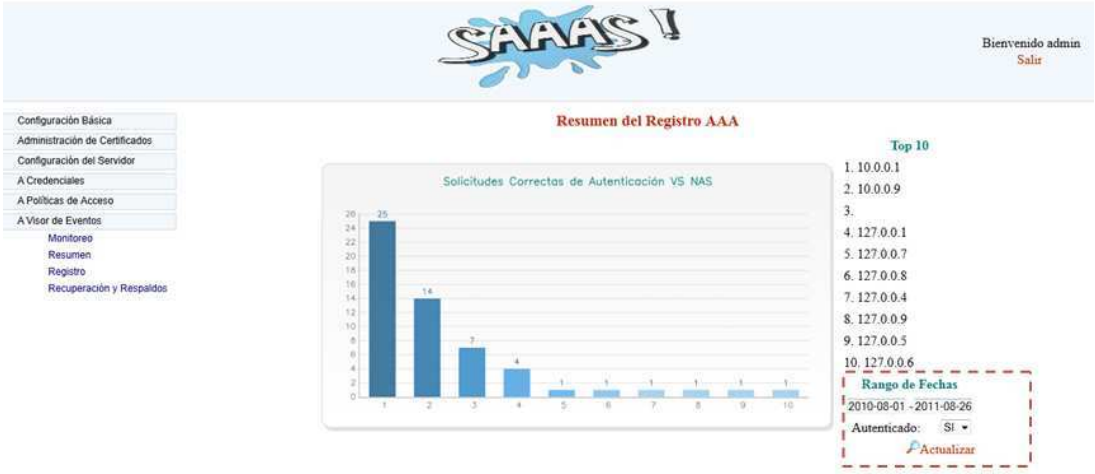
El usuario puede definir las fechas entre las cuales se generaron los logs que desea consultar y el tipo de mensajes que desea ver en el informe. El menú desplegable de tipo de mensaje permite elegir entre TODOS, INFORMACIÓN y ERROR. Para visualizar el informe en el explorador el usuario debe hacer click sobre la opción "Actualizar".

Ilustración 39 Página para generar Informe de Logs del Sistema.

4.6.2. Resumen

La página Resumen permite visualizar en gráficos de barras estadísticas sobre el proceso de autenticación de los usuarios en la red.

El primer gráfico que se muestra es el de “Solicitudes de Autenticación VS NAS” el usuario puede personalizar esta grafica definiendo el rango de fechas en los que se desea hacer el análisis y si quiere ver los registros en los que el usuario se autenticó correctamente o aquellos en los que la autenticación falló. Para visualizar el informe en el explorador el usuario debe hacer click sobre la opción “Actualizar”.



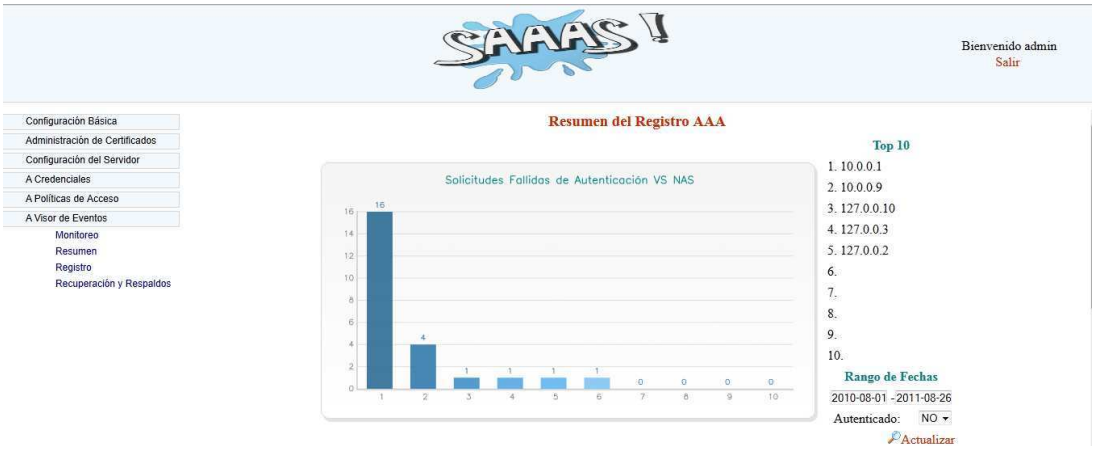


Ilustración 40 Generar Estadísticas de Solicitudes de Autenticación VS. NAS.

El segundo gráfico que se muestra es el de “Sesiones Activas VS. NAS”, el usuario no puede personalizar esta grafica pero puede forzar la actualización de los datos haciendo click sobre la opción “Actualizar”.

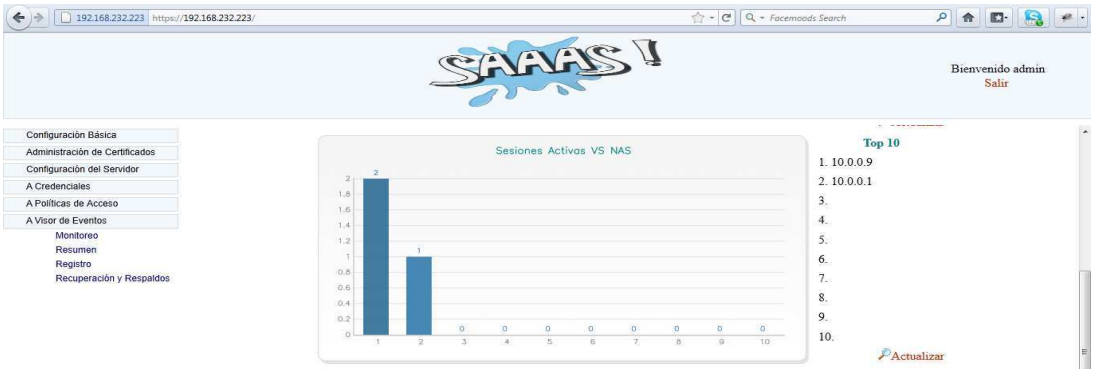


Ilustración 41 Mostrar Gráfico de Estadísticas de Sesiones Activas VS. NAS

El tercer gráfico que se muestra es el de “Solicitudes de Autenticación VS. NAS”, el usuario puede personalizar esta grafica definiendo el rango de fechas en los que se desea hacer el análisis y haciendo click en el botón “Actualizar”.

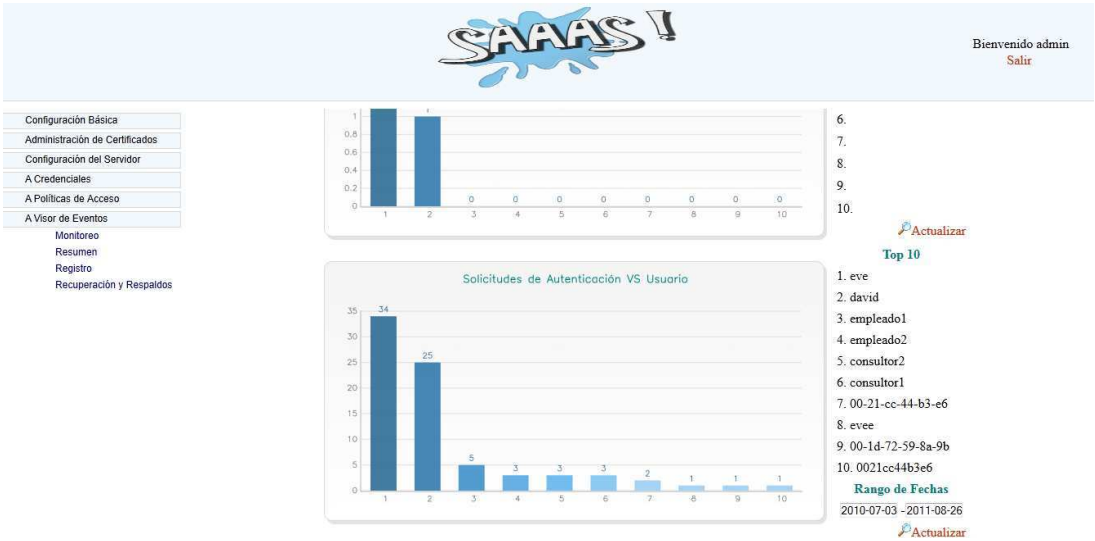


Ilustración 42 Mostrar Gráfico de Estadísticas Solicitudes de Autenticación VS. Usuario.

4.6.3. Registro

La página Registro permite visualizar, personalizar y descargar informes de los poscesos de autenticación y sesiones activas en la red LAN. La página muestra cuatro pestañas: Solicitudes de Autenticación, Sesiones por Tiempo, Sesiones por Tráfico y Sesiones Activas.

El usuario puede personalizar los reportes definiendo el rango de fechas en los que se desea hacer el análisis, y los campos “Nombre de Usuario” o “IP del NAS” que desea filtrar, por defecto estos filtros vienen en blanco y el informe muestra todos los registros existentes sin tomarlos en cuenta. El usuario genera el reporte personalizado haciendo click en el botón “Actualizar”.

Todos los resportes de ésta página pueden descargarse en formato .xls haciendo click sobre el botón “ Descargar”

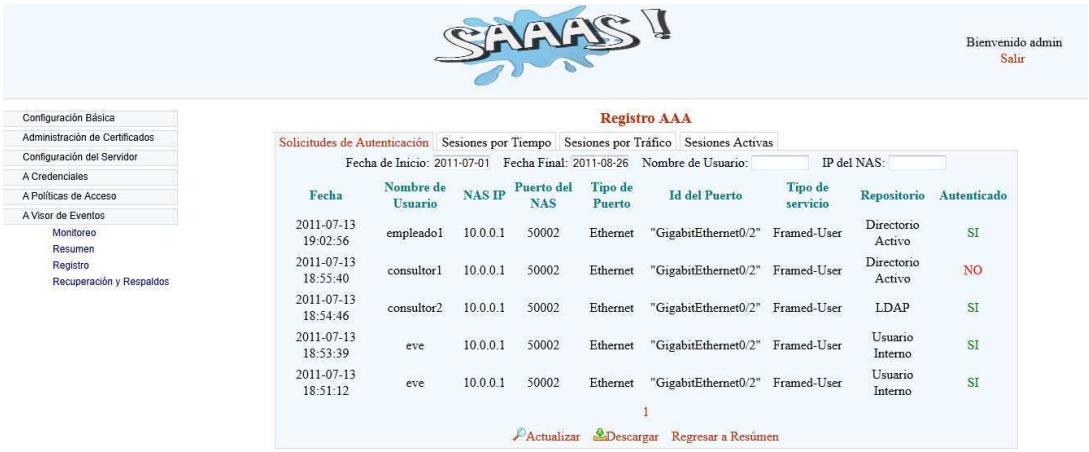


Ilustración 43 Mostrar Reporte "Solicitudes de Autenticación"

El reporte “Solicitudes de Autenticación” muestra el detalle de las solicitudes de autenticación, el usuario que la solicito, el cliente NAS que la reenvió al servidor RADIUS y si tuvo éxito o falló.

Ilustración 44 Mostrar Reporte "Sesiones por Tiempo"

El reporte “Sesiones por Tiempo” muesta el detalle de las sesiones después de la autenticación exitosa ordenadas de mayor a menor según el tiempo que duró la sesión.

Ilustración 45 Mostrar Reporte "Sesiones por Tráfico"

El reporte “Sesiones por Tráfico” muestra el detalle de las sesiones después de la autenticación exitosa ordenadas de mayor a menor según la cantidad de tráfico (bytes) de bajada que generaron en la red.

Ilustración 46 Mostrar Reporte "Sesiones Activas"

El reporte “Sesiones Activas” muestra el detalle de las sesiones que se encuentran activas. A veces el campo que indica que una sesión ha terminado no se actualiza por fallas en el “NAS”, para actualizar manualmente el estado de estas sesiones el usuario debe hacer click sobre la opción “ Cerrar Sesión”.

4.6.4. Recuperación y Respaldos

La página “Recuperación y Respaldos” permite crear, descargar, eliminar y utilizar respaldos de configuraciones anteriores del servidor. Por defecto la página muestra la lista de los respaldos creados. Cada respaldo consiste en una copia de la configuración de FreeRADIUS, la configuración del servicio SAMBA y la copia de la base de datos que maneja el servidor para los procesos de Registro y Auditoría.

Ilustración 47 Mostrar Respaldos del Servidor

Para crear un nuevo respaldo el usuario debe hacer click sobre la opción “Nuevo Respaldo” en la esquina superior derecha de la página.

Para eliminar un respaldo el usuario debe hacer click sobre la opción junto al respaldo que desea eliminar.

Para descargar un respaldo el usuario debe hacer click sobre la opción junto al respaldo que desea descargar.

Ilustración 48 Crear un nuevo Respaldo

Para usar un respaldo el usuario debe hacer click sobre la opción junto al respaldo que desea utilizar. El usuario verá una advertencia indicando que la configuración actual será eliminada. El sistema automáticamente crea un respaldo de la configuración antes de borrarla.

Ilustración 49 Usando un Respaldo de la configuración de SAAAS!

Ilustración 50 Información devuelta después del uso de un respaldo de SAAAS!