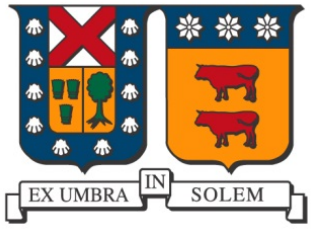
2022-06

“PLAN INTEGRADO DE AUTENTICACIÓN PARA EL DEPARTAMENTO DE INFORMÁTICA EN LA UTFSM

PALACIOS BARAHONA, ADBIAS MANASES

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE



“PLAN INTEGRADO DE AUTENTICACIÓN PARA EL DEPARTAMENTO DE INFORMÁTICA EN LA UTFSM”

ADBIAS MANASES PALACIOS BARAHONA

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Horst Von Brand
Profesor Correferente: Yonathan Dossow

Junio - 2022

DEDICATORIA

Dedico este logro a mi familia, por la persistencia y esfuerzo que logró ayudarme a ser quien soy.

AGRADECIMIENTOS

Quiero agradecer a todo el equipo LabComp que hizo posible este trabajo junto con ayudarme a ser un mejor profesional.

También agradecer al equipo del Datacenter o uSCI, el cual fue muy importante para mi desarrollo tanto personal como profesional.

Muchas gracias a esas personas de la universidad que apoyaron incondicionalmente el desarrollo de este trabajo y también mi aprendizaje: Yonathan, Eduardo, Geordy, entre otros.

Gracias a mis amigos que tienen un valor muy especial para mí más allá del ámbito profesional.

Y muchas gracias al profesor Horst que me apoyó en este trabajo, y me acompañó en este proceso de aprendizaje.

RESUMEN

Resumen— El sistema de autenticación del Departamento de Informática actualmente está basado en el protocolo LDAP, lo cual pueden encontrarse en este, vulnerabilidades de distinto tipo junto a no tener características adicionales necesarias. Si bien LDAP es una poderosa herramienta, existen en el mercado otras opciones que combinan este mismo con protocolos de autenticación más poderosos como Kerberos, agregando robustez al sistema de autenticación y brindando características adicionales a lo que se puede solo lograr con LDAP. Sin embargo, existen múltiples obstáculos que impiden una transición limpia, por lo que a continuación se propondrá un plan de trabajo para mejorar el sistema de autenticación, disminuyendo la mayor cantidad del impacto negativo que podría existir.

Palabras Clave— Autenticación; Seguridad; LDAP; Kerberos

ABSTRACT

Abstract— The Informatics department's authentication system is based in the LDAP protocol, meaning we could find different vulnerability types and it lacks certain needed additional features. Even though LDAP is a powerful protocol, there are many options that use different protocols like Kerberos, adding hardiness to the authentication and multiple additional features in contrast to what is only achievable with LDAP. However, there are multiple obstacles that prevent a clean transition, and thus up next, we will propose a work plan to improve the authentication system, diminishing the negative impact that could exist.

Keywords— Authentication; Security; LDAP; Kerberos

GLOSARIO

2FA: Two Factor Authentication
AD: Active Directory
DI: Departamento de Informática
DC: Domain Controller
GSSAPI: Generic Security Service Application Program Interface
LDAP: Lightweight Directory Access Protocol
MFA: Multi Factor Authentication
NFS: Network File System
OTP: One Time Password
RADIUS: Remote Authentication Dial-In User Service
SPNEGO: Simple and Protected GSSAPI Negotiation Mechanism
SSO: Single Sign On
SASL: Simple Authentication and Security Layer
SAML: Security Assertion Markup Language
TLS: Transport Layer Security
UTFSM: Universidad Técnica Federico Santa María

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE CUADROS

ÍNDICE DE CUADROS DE CÓDIGO

INTRODUCCIÓN

Dentro del presente documento, se analizará el actual estado del sistema de autenticación del Departamento de Informática de la Universidad Técnica Federico Santa María y se buscará proponer un plan de trabajo con mejoras. Se realizará una revisión y análisis de la situación actual, identificando desventajas, buscando alternativas, y aplicando alguna de éstas para finalmente comparar y analizar si fue posible resolver los problemas planteados al principio. Se tomarán en consideración parámetros como la seguridad, disponibilidad e integridad de la información y en que aristas se puede complementar y ayudar con el plan de trabajo. Por último, un punto importante que buscará resolver el plan de trabajo es la migración de datos, lo cual ayudaría en caso de que se proponga utilizar los sistemas propuestos en esta memoria y posiblemente adaptarse para ambientes similares que tengan los sistemas documentados.

CAPÍTULO 1
DEFINICIÓN DEL PROBLEMA

Actualmente el Departamento de Informática ofrece distintos servicios que son necesarios para estudiantes y profesores. Todas estas aplicaciones requieren un sistema unificado de autenticación que sea fácilmente accesible por los usuarios y que sea compatible con sistemas de distintas plataformas. Hoy en día, es utilizado el protocolo LDAP (Subsección 2.1), el cual se encarga de autenticación y base de datos de usuarios.

Para el caso de los clientes del servicio de autenticación, lo ideal es que la aplicación a implementar tenga de alguna forma compatibilidad con LDAP (lo cual en gran parte las aplicaciones cumplen debido a su uso masivo). En este caso, si se es compatible, solo se necesita agregar los datos del servicio de autenticación, o en el caso de terminales, es necesario instalar algún cliente LDAP, lo cual agregará la funcionalidad requerida.

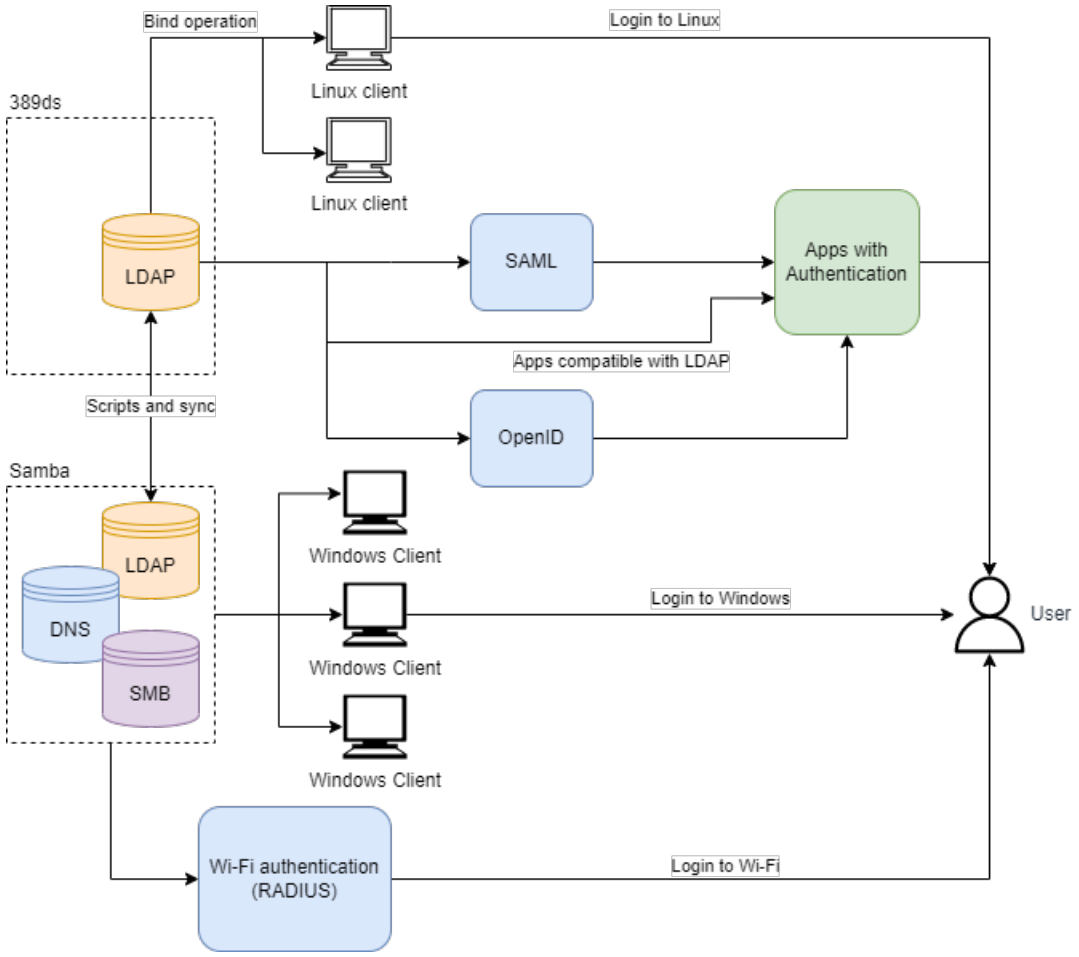


Figura 1: Esquema del sistema de autenticación del Departamento de Informática.
Fuente: Elaboración propia.

Junto a esto, también se utilizan los protocolos de autenticación SAML (Subsubsección 2.5.1) y OpenID Connect (Subsubsección 2.5.2) para consumir LDAP, en caso de que las aplicaciones no ofrezcan compatibilidad directa con el mismo. Todo lo anterior genera un ecosistema bastante completo que entrega a los usuarios una funcionalidad bastante importante: junto con un nombre único y una contraseña, ser capaz de poder ingresar a cualquier servicio del Departamento de Informática, sin necesidad de crear nuevas credenciales por cada aplicación o servicio.

Además, se agrega otra capa de compatibilidad para terminales con sistema operativo Microsoft Windows, para lo cual se hace uso de Samba AD DC, aplicación que provee los servicios de autenticación y todo lo necesario para que cada usuario tenga credenciales e intercambio de archivos en dicho sistema.

Finalmente, existen aplicaciones que proveen distintos servicios a los usuarios que de alguna forma usan los servicios de autenticación y que son esenciales para el ecosistema.

Uno de los problemas que ha surgido a lo largo de los años es tomar en cuenta la disponibilidad, seguridad y accesibilidad de este servicio. Es importante recalcar que es necesario asegurar la confianza del servicio de autenticación, debido a que contiene datos necesarios para el inicio de sesión en terminales, así como contraseñas y dirección del almacenamiento para cada usuario.

Como solución, se han levantado más instancias del servicio LDAP (debido a lo vital que es), cosa que ayuda a la disponibilidad del servicio, pero agrega complejidad a la administración. La disponibilidad, seguridad y accesibilidad del servicio de autenticación junto al cumplimiento de casos de uso o requerimientos posibles, serán métricas donde se centrará nuestra propuesta de memoria, la cual buscará evaluar la implementación de un plan integrado con respecto al servicio de autenticación.

Otro problema que deriva de la complejidad del sistema es no poder reiniciar la contraseña de una cuenta debido al intrincado proceso que requiere la creación del usuario. Como solución de esto, se disponen alternativas: los ayudantes de distintos laboratorios tienen accesos privilegiados a los sistemas de autenticación que permiten reiniciar o propiamente cambiar las contraseñas de acuerdo con lo que se necesite, y concretamente existen en desarrollo algunos proyectos que tratan de suplir la necesidad de hacerlo automático.

También es posible encontrar distintos errores dentro de la información provista por LDAP, dado que estos datos son manejados semi automáticamente por el administrador del Departamento de Informática. Esto puede dar lugar a problemas para la administración por otras personas y una gran barrera de entrada en caso de que el administrador no pueda asistir más en el sistema.

Luego también, otro proceso que existe dentro del Departamento de Informática es la posibilidad del apoyo de estudiantes como ayudantes del área de administración en forma de laboratorios dentro de éste, lo cual facilita tareas que eventualmente terminarían siendo

tareas para el administrador del Departamento de Informática. Esto trae su parte negativa también ya que si bien se ha visto una gran participación de parte de los estudiantes debido a nuevos proyectos que se entregan en el área, es posible que también se conlleve un riesgo de seguridad en el caso de que un estudiante pudiera entregar información a terceros.

Algunos ejemplos de laboratorios manejados por estudiantes serian el Laboratorio de Computación, Laboratorio de Proyectos Informáticos y el Laboratorio de Integración Tecnológica.

Servicios como el sistema de correo y el sistema de directorios para el Laboratorio de Computación deben poseer acceso al sistema de autenticación debido a ser servicios provistos por el laboratorio a los estudiantes, pero deben crear manualmente los registros para que sea posible su uso por los usuarios.

A continuación, se realizará un resumen de los problemas que provienen del sistema actual de autenticación:

- No existe forma de reiniciar la contraseña de forma automática y transparente: Se debe realizar por algún ayudante de alguno de los laboratorios, administradores, o gente involucrada en el sistema de autenticación (actualmente existen proyectos dentro del Departamento de Informática que se encargan de resolver este problema).
- No existe la posibilidad de usar 2FA (*two-factor authentication*) o MFA (*multi-factor authentication*).
- El proceso de SSO comienza cuando se inicia sesión a través de un navegador: Si bien esto es normal para el proceso usuario aplicación, sería una buena característica que el proceso Single Sign-On se haga también en el inicio de sesión para no exigirle al usuario nuevamente sus credenciales ya ingresadas.
- Datos personales solo pueden ser modificados por administradores: Esto puede ser dependiendo de la política del Departamento de Informática, pero últimamente es muy requerido los cambios de dirección, teléfono, entre otros que necesariamente deben hacerse solicitudes al administrador.
- El proceso de creación de cuentas es semiautomático, lo que puede traer errores a nivel de información del usuario.
- Se necesita que los ayudantes de los laboratorios sean administradores del sistema de autenticación en caso de que ocurra alguno de los problemas anteriores (usuarios que necesiten reiniciar contraseñas, problemas al intentar entrar en los terminales), pero esto a su vez representa un posible riesgo de seguridad dado que los administradores, en caso de tener comprometidas sus cuentas, pueden representar una vulnerabilidad grave.

- Problemas de distribución de permisos: Dado lo anterior, los administradores a veces no obtienen los suficientes permisos debido al cambio periódico de ayudantes durante distintos semestres.

1.1. Objetivos

- Recoger requerimientos y definir marco en el cual se aplicaría el plan integrado de autenticación.
- Identificar posibles candidatos para mejorar el sistema.
- Evaluar los posibles candidatos y elegir uno para posible candidato de implementación.
- Aplicar el plan integrado en un ambiente controlado, pero similar al real.
- Contrastar el plan integrado al actual sistema utilizado para encontrar fortalezas y debilidades que puedan surgir a través del proceso.

1.2. Requerimientos

Para complementar a los objetivos a cumplir, definiremos una serie de requerimientos del sistema de autenticación, de forma que podamos validar el plan integrado y buscar que podamos abarcar la mayor cantidad posible.

- Un usuario puede autenticarse de acuerdo a un nombre de usuario y una contraseña.
- El usuario puede autenticarse en un computador, habilitado para su uso, dentro de la red del Departamento de Informática.
- El usuario puede usar aplicaciones del Departamento de Informática, dentro y fuera de la universidad, autenticándose con sus credenciales de usuario.
- El usuario debe ser capaz de ingresar en un computador habilitado sin importar el sistema operativo.
- El usuario debe ser capaz de aprovechar las ventajas del Single Sign On.
- El usuario debe ser capaz de obtener permisos en base a su categoría dentro de la universidad (ayudante, estudiante, ayudante administrativo, profesor, administrador, etc.).
- Debe existir una (o más) política(s) de contraseñas.
- El usuario debe obtener la posibilidad de usar 2FA y/o MFA.

- Debe haber un sistema que permita administrar en forma integrada permisos, privilegios y accesos de ayudantes y demás personal.
- Debe haber un sistema que permita reiniciar la contraseña de acuerdo a la necesidad de un administrador o un ayudante.
- Usuarios deben poder cambiar su contraseña.
- Usuarios deben poder cambiar sus datos personales o agregar nuevos datos (siempre y cuando el Departamento de informática lo permita).
- Debe haber un sistema que permita administrar permisos a aplicaciones o servicios, en caso de que lo requieran.
- Los sistemas propuestos deben estar disponibles en los repositorios oficiales de algún sistema operativo Linux, en el caso de servidores Linux, debido al tiempo limitado de los administradores y la complejidad de mantener e integrar código sin empaquetar.
- Las alternativas deben garantizar que van a ser compatibles de un modo u otro con las aplicaciones y servicios existentes que lo requieran.

CAPÍTULO 2
MARCO CONCEPTUAL

2.1. LDAP

Lightweight Directory Access Protocol (actualmente en su versión 3), es un protocolo de acceso para proveer servicios de directorios basado en el estándar X.500, el cual es descrito en el RFC 4511 (Sermersheim, 2006). Los directorios son bases de datos, pero la estructura es jerárquica, lo cual significa que es mucho más descriptiva basada en atributos, en comparación a una base de datos relacional. Generalmente dentro de los directorios de información es muy común leer mucho más que escribir datos, lo cual se vuelve vital que estos entreguen respuestas rápidas en gran cantidad.

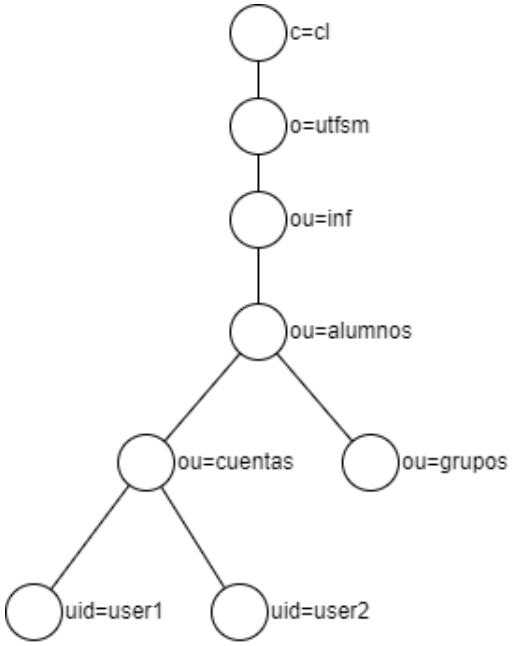


Figura 2: Estructura de Directorios LDAP.
Fuente: Elaboración propia.

Junto a esto podemos también consultar la información en la documentación de Red Hat sobre RHEL (Red Hat Inc., s.f.-b):

Es un conjunto de protocolos abiertos usados para acceder a información centralizada guardada sobre la red. Está basada en el estándar X.500 para intercambio de directorios, pero es menos complejo y usa menos recursos. Por esta razón, algunas veces es descrito como "X.500 Lite".

Como X.500, LDAP organiza la información de manera jerárquica usando directorios. Estos

directorios pueden guardar una variedad de información, incluyendo, pero no limitándose a: nombres, direcciones y números de teléfonos. Puede ser utilizado de forma similar al ya obsoleto NIS (Network Information Service), habilitando a cualquiera para acceder a su cuenta por medio de cualquier máquina que esté dentro de la red de LDAP.

LDAP es comúnmente utilizado para manejar de forma centralizada, usuarios y grupos, autenticación de usuarios, o configuraciones de sistema. También puede ser utilizado como un directorio de teléfonos virtual, permitiendo a los usuarios un fácil acceso a la información de contacto de otros usuarios. Adicionalmente, puede referir un usuario a otros servidores LDAP dentro del mundo, y así proveer un repositorio de información global. Sin embargo, es frecuentemente usado en organizaciones individuales como universidades, departamentos de gobierno y compañías privadas.

Si bien es posible utilizar LDAP para guardar cualquier tipo de información, su uso más común es para guardar nombres de usuario, contraseñas y atributos asociados a éstos, es decir, información de usuarios.

2.1.1. LDAP en el Departamento de Informática

```
dn: uid=apalacio,ou=cuentas,ou=valparaiso,ou=alumnos,ou=inf,o=utfsm,c=cl
memberOfList: ...
shadowLastChange: ...
uidNumber: ...
uid: apalacio
sn: Palacios Barahona
...
cn: adbias manases palacios barahona
acceso: ...
```

Listado 1: Ejemplo de consulta a la base de datos LDAP

Como podemos ver en el cuadro anterior [1], una entrada en la base de datos consiste en un conjunto de atributos que pueden tener uno o más valores. Cada atributo esta previamente definido en la configuración de la base de datos llamado “esquema” o en inglés “schema”. Cada entrada posee un identificador único para su consulta.

El identificador único correspondería a cada usuario, grupo u objeto que se pueda ingresar, lo que en la base de datos se considera como *Distinguished Name* (dn), el cual consiste en un *Relative Distinguished Name*, conjunto de atributos que suceden al primer registro, y luego se une con el directorio raíz.

En este caso, los directorios que preceden al nombre del usuario consisten en una descripción de la entrada, es decir, atributos adicionales donde el usuario puede ser agrupado. Siguiendo el ejemplo anterior, el usuario `apalacio` pertenece a los *Organization Unit* (ou) “grupos”, “valparaiso”, “alumnos”, “inf”, junto con *Organization* (o) “utfsm” y *Country* (c) “cl”.

Esto nos ayuda a saber que el usuario pertenece a los grupos indicados anteriormente donde nos indican su región, su posición en la organización, información genérica (ou=grupos), la institución (UTFSM), el departamento y el país.

En algunos casos, se utiliza el atributo `dc` que proviene de *Domain Component*, para hacer el registro de LDAP. Los atributos que corresponden al usuario en el ejemplo anterior serían las siguientes líneas aparte del DN, como *objectClass*, *gidNumber*, entre otros.

Junto a todo esto, información que puede cambiar debe declararse en atributos, los cuales, por ejemplo, dentro del Departamento de Informática pueden ser (y no están limitados a): rol, RUT, acceso, etc.

Actualmente se utiliza 389 Directory Server como servidor LDAP dentro del Departamento de Informática, lo cual junto con un certificado provisto por el Departamento de Informática se puede utilizar a través de TLS y, por lo tanto, un canal cifrado. Sin embargo, esta configuración es opcional: es posible configurar clientes sin TLS y para acceder a LDAP solo es necesario estar dentro de la red autorizada. Es vital, para evitar posibles problemas de seguridad, que los sistemas estén obligados a utilizar TLS siempre y cuando se pueda.

Podemos encontrar dentro del universo de posibles proveedores LDAP a aplicaciones tales como:

- 389 Directory Server
- Apache Directory Server
- Apple Open Directory
- FreeIPA
- OpenLDAP
- Oracle Internet Directory
- RedHat Directory Server
- Samba
- Active Directory

2.1.2. 389 Directory Server

Actualmente dentro del Departamento de Informática, se utiliza la aplicación llamada 389 Directory Server (389ds) el cual implementa el protocolo LDAP. Es un derivado del proyecto *slapd* de la Universidad de Michigan (University of Michigan, 1996), que a lo largo de los años fue introducido en varias compañías informáticas.

Anteriormente denominado Fedora Directory Server, es la solución de Red Hat Open Source para un servidor LDAP, donde existe la versión con soporte dedicado llamada Red Hat Directory Server.

Es un proyecto directamente desarrollado del antiguo Netscape Directory Server (NDS), el cual fue vendido a Sun Microsystems para distribuirse como JES/SunOne Directory Server, el cual finalmente fue obtenido por Red Hat y liberado al público como proyecto Open Source.

Con su primera versión en el 2005, 389ds obtiene su nombre debido al puerto que es utilizado para el protocolo LDAP. Actualmente tiene soporte en Linux y algunos sistemas UNIX. Es código abierto (open-source) con excepción de algunos plugins que son incluidos.

Existe su contraparte llamada Red Hat Directory Server, el cual es comercializado por Red Hat entregando soporte adicional a los servicios.

En el Departamento de Informática, el servicio se utiliza tanto para recursos de información como autenticación del usuario. En esto último, se utiliza la operación *bind*, que es la opción incluida dentro de la suite de LDAP para autenticar. Podemos agregar que, la idea de LDAP dentro del Departamento de Informática es proveer información de los distintos usuarios dentro de la universidad, por lo que implica que no solo se tienen los parámetros comunes, sino que también podemos encontrar atributos tales como rol institucional, RUT, sede del usuario, email, y en algunos casos se puede agregar teléfono, certificados y direcciones de páginas web.

2.1.3. Proceso de entrega de cuentas del Departamento de Informática

Debido a la importancia de las cuentas de informática, para acceder a los servicios estudiantes del Departamento de Informática es necesario pasar por un proceso que describiremos a continuación:

1. El usuario debe inscribirse a la carrera de informática o entra a clases relacionadas con el Departamento de Informática que requieran el uso de la plataforma Moodle exclusiva de informática (reemplazado debido al cambio de políticas de la universidad, donde se les solicita a todos los profesores que utilicen necesariamente Aula USM). También pueden ser factores de creación: Acceso al servicio de correo, acceso al sistema GitLab del Departamento de Informática o acceso a los terminales de algún laboratorio.

2. Administrador crea los usuarios en el sistema LDAP, ya sea vía manual o automatizada a través de scripts.

3. Administrador crea usuarios en el sistema de correo, el directorio de archivos, junto con todo lo necesario para que pueda ser utilizado en los servicios del Departamento de Informática.

4. Administrador configura la redirección de correos del correo de informática al correo entregado/solicitado.
5. Administrador crea la contraseña con la política de seguridad correspondiente e insta al usuario a que cambie la contraseña.

En el punto 5, el administrador pierde el control de la cuenta, cosa que si bien depende del usuario, representa un problema futuro para él debido a que no recordará la contraseña y el documento entregado no cumplirá su función. Esto provoca que existan distintas instancias en donde el usuario se pueda acercar para reiniciar la contraseña. Sin embargo, estas plataformas se fueron dando de baja debido a la mantención y posibles problemas de seguridad que representaban. Últimamente se ha dado impulso a lograr la mantención de un portal para que cada usuario pueda reiniciar la contraseña con las credenciales de la universidad (cuenta de autenticación USM es distinto a la autenticación del Departamento de Informática) debido a los problemas de presencialidad.

2.1.4. Seguridad de LDAP

En el RFC 4513 (R. Harrison, 2006) se describe los procesos de seguridad que se encuentran en LDAP para mitigar los problemas posibles del acceso no autorizado y sobre todo por el hecho de que LDAP es una plataforma que podría potencialmente contener contraseñas.

Dentro de los posibles problemas que se describen en el RFC, se encuentran:

1. Acceso no autorizado de la información del servidor a través de operaciones de extracción de datos.
2. Acceso no autorizado de la información del servidor por medio de monitoreo de accesos.
3. Acceso no autorizado, reutilizando la información del cliente por medio de monitoreo de acceso.
4. Modificación no autorizada de la información del servidor.
5. Modificación no autorizada de la configuración del servidor.
6. DoS (Denegación de Servicio): Uso de recursos en exceso con la finalidad de negarle el acceso a los usuarios.
7. Spoofing: Confundir al usuario o cliente, de modo que crea que la información proviene del servidor por medio de modificando la información en tránsito o dirigiendo la conexión del cliente erróneamente.
8. Hijacking: El atacante toma control de la sesión.

La mayoría de estos problemas pueden ser solucionados cifrando la comunicación con TLS, por lo que se hace imprescindible para utilizar LDAP como sistema de autenticación.

2.1.5. Estado actual de LDAP

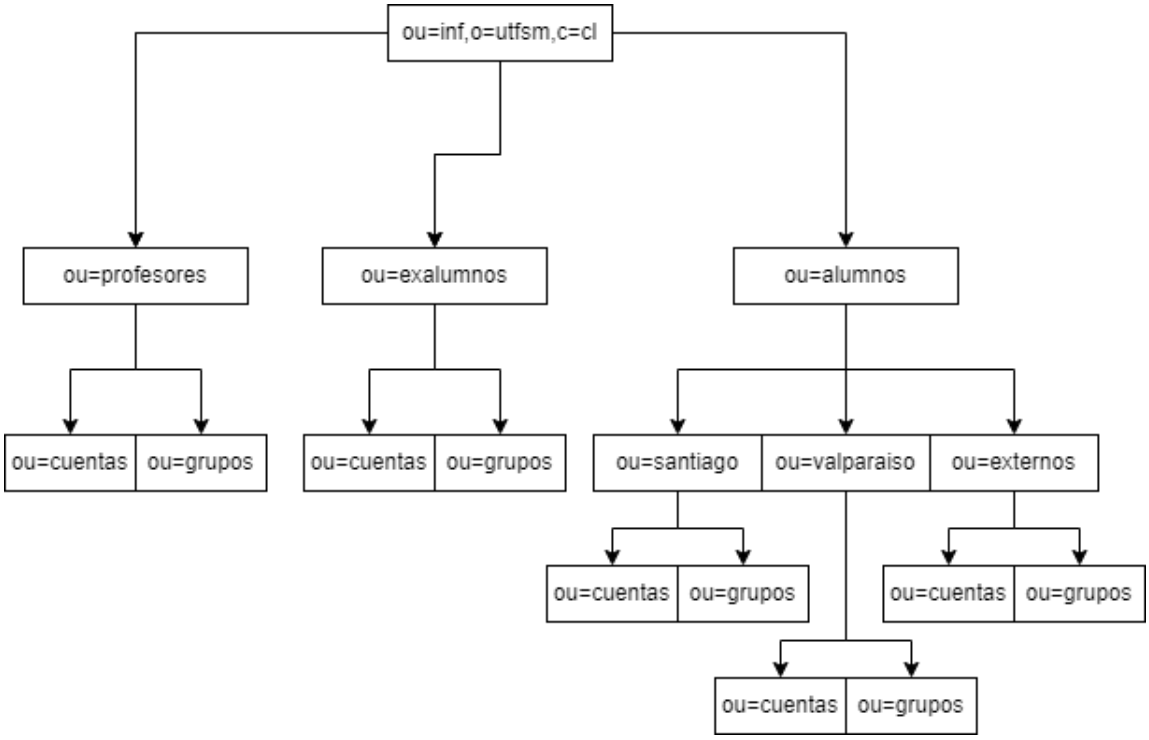


Figura 3: Diagrama del estado actual de parte de la estructura de LDAP.
Fuente: Elaboración propia.

Como describimos en los puntos anteriores, existe actualmente el servicio de autenticación dentro del Departamento de Informática que puede ser descrito en su núcleo como LDAP, implementado con 389 Directory Server. Los demás servicios que detallaremos en los siguientes puntos son capas de compatibilidad o servicios adicionales para darle finalmente a LDAP más funcionalidad y abarcar la mayor cantidad posible de casos de uso de autenticación posible.

En la figura anterior 3, podemos ver un pequeño extracto de cómo está compuesta la estructura jerárquica de LDAP. Esto es con la finalidad de que los usuarios sean definidos en grupos totalmente separados, identificando o pudiendo identificar, de acuerdo con el dn base, que usuarios pueden utilizar un servicio determinado. Agregar que esto sería una limitante suave, es decir, no hay nada que impida a un administrador conectarse a un grupo raíz más grande. También se pueden utilizar atributos como roles dentro de LDAP, pero la desventaja es que dependería de cada aplicación el uso que se le daría.

2.2. Active Directory

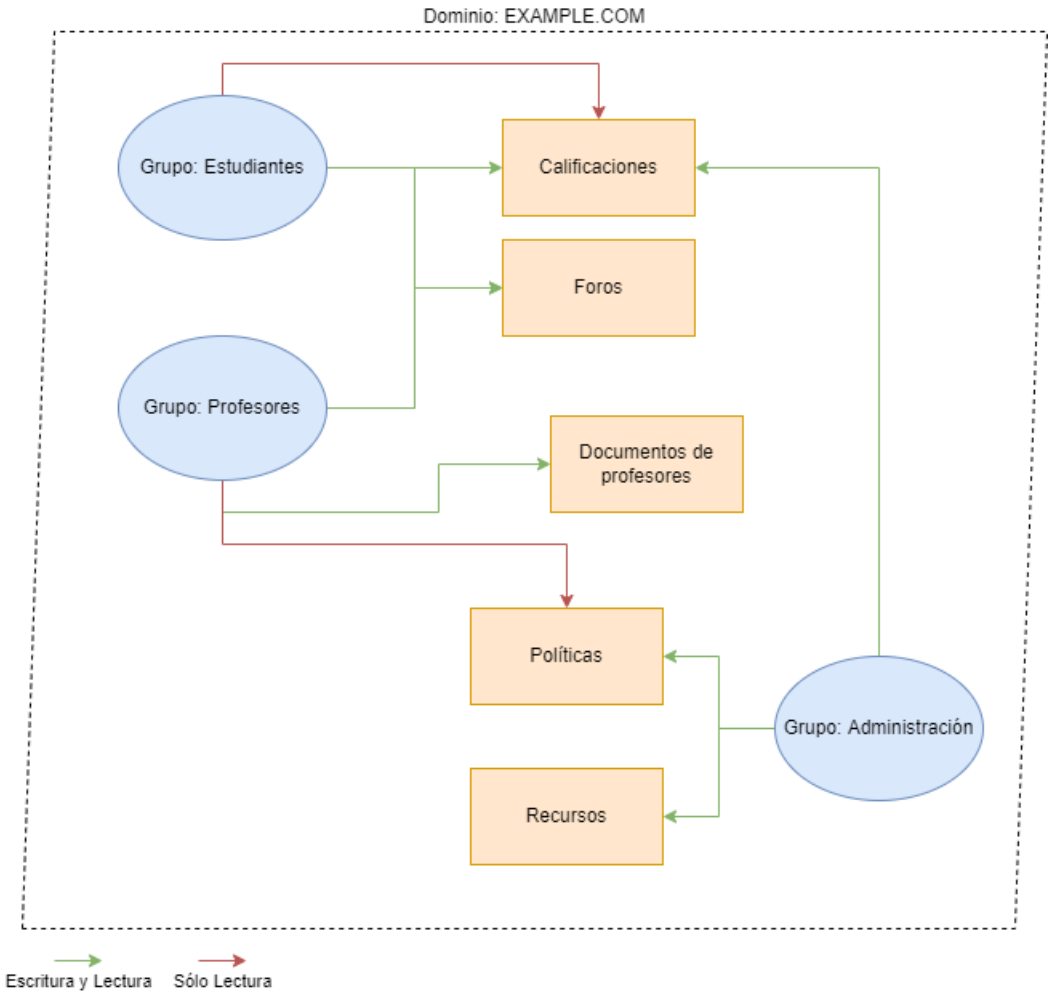


Figura 4: Jerarquía de una compañía con AD.
Fuente: Elaboración propia.

Active Directory es una interpretación e implementación de X.500 de Microsoft (incompatible con otros sistemas similares), la cual provee sistemas de autenticación para terminales con el sistema operativo Microsoft Windows y otros servicios que sean compatibles con éste. Hay que destacar que, al ser un sistema propietario, existen varias dificultades a la hora de considerarlo como una solución, especialmente en ambientes como el presentado en el Departamento de Informática.

Active Directory se describe como la solución de autenticación y autorización de Microsoft, implementando protocolos como LDAP, Kerberos y DNS bajo los propios productos de este.

A continuación, entraremos en más detalle con este sistema para considerar las posibles

ventajas y desventajas que podría tener considerarlo dentro del plan integrado, además de que es la alternativa recomendada por Microsoft para los servicios de autenticación y autorización centralizado para terminales Windows.

Dentro del Departamento de Informática, se utiliza Samba AD DC (implementación de protocolos que se utilizan en Active Directory en Linux) como capa de compatibilidad para equipos Windows junto a otros servicios que requieran AD y no sean compatibles con solo LDAP.

Es posible encontrar más información dentro de la documentación entregada por Microsoft (Microsoft, s.f.).

2.2.1. Seguridad

También es posible rescatar algunas medidas de seguridad que son recomendadas en la documentación de Microsoft:

- Restricciones RDP (Remote Desktop): Windows Server permite el manejo remoto del sistema operativo para temas de administración, en caso de que no se tenga acceso físico. Esto es conveniente para administradores, sin embargo, la incorrecta configuración de este servicio puede comprometer fácilmente el servidor. Se recomienda sólo abrir el servicio de RDP a rangos de direcciones IP que permitan el acceso a administradores y bloquear todo el resto, junto a utilizar contraseñas complejas.
- Parches y administración de configuración de los DC: Microsoft lanza continuamente parches de seguridad y actualizaciones al sistema operativo Windows Server, y también a los servicios asociados para solucionar vulnerabilidades que van siendo conocidas día a día. Se recomienda una política de actualización dentro de la empresa que permita a los servidores actualizarse cada vez que haya una actualización de seguridad. Además, se recomienda una configuración adecuada para el servidor Active Directory DC, enfocada en la seguridad.
- Bloqueo de internet para los DC: Una vez configurado el servicio de Active Directory, no es necesario el acceso de internet, a excepción de las actualizaciones lo cual incluso puede ser gestionado para que sea posible hacerlo offline. Se recomienda bloquear el acceso a internet para evitar ataques.
- Restricciones de Firewall perimetral: Se recomienda cerrar el firewall completamente para puertos que no se utilicen dentro del servidor y solo admitir la subred correspondiente al Active Directory.
- Prevención de navegación web dentro de los DC: Siendo un sistema operativo completo, Windows Server permite a los usuarios utilizarlo como un sistema normal con la adición de servicios. Sin embargo, la utilización de navegadores y otras aplicaciones externas en estos servidores puede llevar a vulnerabilidades no deseadas e incluso aperturas que pueden comprometer la seguridad del servidor.

2.2.2. Implementación Azure AD

Una de las posibles soluciones que entrega Microsoft es la posibilidad de dar disponibilidad a un servidor Active Directory en la nube. A continuación, se adjunta una comparativa de Azure AD DS (solución en la nube) y un Active Directory común.

Características	**Azure AD DS**	**AD DS on-premise**
Servicio entregado	✓	✗
Despliegue seguro	✓	✓[1]
Servicio DNS	✓[2]	✓
Privilegios de administrador de dominio o empresa	✗	✓
Unión a dominios	✓	✓
Autenticación con NTLM y Kerberos	✓	✓
Delegación acotada de Kerberos	✓[3]	✓[4]
Estructura OU a medida	✓	✓
Políticas de grupo	✓	✓
Extensiones del *schema*	✗	✓
Dominio AD / Forest Trusts	✓[5]	✓
LDAP Seguro (LDAPS)	✓	✓
Lectura LDAP	✓	✓
Escritura LDAP	✓[6]	✓
Despliegue distribuido geográficamente	✓	✓

Tabla 1: Comparación entre Azure AD DS & AD DS on-premise

Para agregar nuevos dispositivos se necesita usar herramientas como Microsoft Endpoint Manager.

A pesar de que Microsoft entregue una buena alternativa en la nube, no es muy útil en nuestro caso debido a que el ecosistema del Departamento de Informática es orientado a sistemas Linux y finalmente se agrega compatibilidad a sistemas Windows.

[1] El administrador asegura el despliegue.
[2] Servicio entregado por Microsoft.
[3] Basado en recursos.
[4] Basado en recursos y cuentas.
[5] Solo Forest trusts salientes.
[6] Dentro del dominio controlado.

2.3. AD en Linux: Samba

Samba (Carter y col., 2007) es la suite open source de compatibilidad de Microsoft Windows con Linux/Unix, dando provisión a los sistemas de autenticación, así como entregando los servicios de SMB/CIFS y Active Directory. Es importante este servicio, ya que permite al administrador con sistemas Linux/Unix integrar clientes Microsoft Windows en el ecosistema sin necesidad de tener un servidor Windows en el mismo, y con los problemas que conllevan el software propietario. Samba es open source, por lo que también representa una ventaja significativa por sobre AD en temas de comunidad y precio por los servicios.

Según la documentación de Samba de Tranquil IT (empresa que ofrece servicios de Samba AD) (Tranquil IT, 2021):

Samba-AD permite el provisionamiento y control de un dominio Active Directory que provee los siguientes servicios:

- Directorio LDAP
- DNS
- NTP (Servicio de sincronización de tiempo)
- Kerberos
- Servicio de distribución de tokens KDC
- Replicación multi-AD
- Configuraciones de seguridad por medio de GPO (Group Policies)

La tecnología en su núcleo de ser compatible es la comunicación y el protocolo para compartir archivos SMB, de donde proviene el nombre del software. La gran presencia que tiene Microsoft en las empresas en el mundo, tanto en el sector de empresas públicas como en la educación, junto con la implementación de Samba al ser software libre y gratis permitiría al protocolo SMB imponerse como un estándar de intercambio de archivos en redes con clientes Windows, Linux y Mac.

Esto incluiría:

- Identificación y autenticación centralizada junto con la administración de ésta en los modos AD y NT4.
- Administración de grupos centralizada.
- Intercambio de archivos de acuerdo con el protocolo SMB que está siendo utilizado.

- Administración centralizada de accesos y permisos a archivos y directorios.
- Uso centralizado de impresoras.

2.3.1. Samba 4

El proyecto Samba 4 fue lanzado en 2005 con el objetivo principal de reescribir completamente Samba de acuerdo con las especificaciones reveladas por Microsoft siguiendo una demanda de la Unión Europea en contra de este, por el abuso de su posición dominante.

Al mismo tiempo, el código base de Samba 3 estaba siendo difícil de mantener. Hasta ese momento Samba había sido desarrollado gradualmente por medio de ingeniería inversa de los protocolos usados por Microsoft; los desarrolladores de Samba habían estudiado como los clientes y servidores Windows se comunicaban unos con otros y de acuerdo con los datos obtenidos habían desarrollado Samba empíricamente para reproducir el comportamiento de los protocolos.

Desde esta fecha, el proyecto había obtenido mayor acceso libre a las especificaciones oficiales de Microsoft, lo que facilitó enormemente el desarrollo. Para consolidar el objetivo de la interoperabilidad, los distintos actores e interesados de los protocolos se reunían en un evento llamado PlugFest para revisar y comprobar las diferentes implementaciones del protocolo SMB.

En 2012, a medida que el interés en Samba 4 crecía, se volvió aparente que la implementación que se basaba exclusivamente en las especificaciones entregadas por Microsoft no era funcional y que el protocolo SMB implementado por Microsoft era mucho más complejo y muy mal documentado.

La versión 3 de Samba había sido desarrollado empíricamente y no desde la documentación oficial. Por lo tanto, Samba 3 solo funcionaba correctamente implementando los servicios de intercambio de archivos e impresoras. Históricamente, Microsoft ha sido muy cuidadoso sobre la compatibilidad de versiones anteriores y el comportamiento de las versiones anteriores han sido replicadas en las nuevas versiones.

La reescritura de Samba 4 involucró 3 grandes componentes:

- El componente Active Directory
- El componente de intercambio de archivos smbd
- El componente de mapeo de usuarios winbindd

2.4. Métodos de autenticación

Como explicamos anteriormente, LDAP se utiliza tanto para consulta de información, así como la misma autenticación. Sin embargo, existen otros métodos de autenticación dentro de la literatura.

2.4.1. LDAP Bind Operation

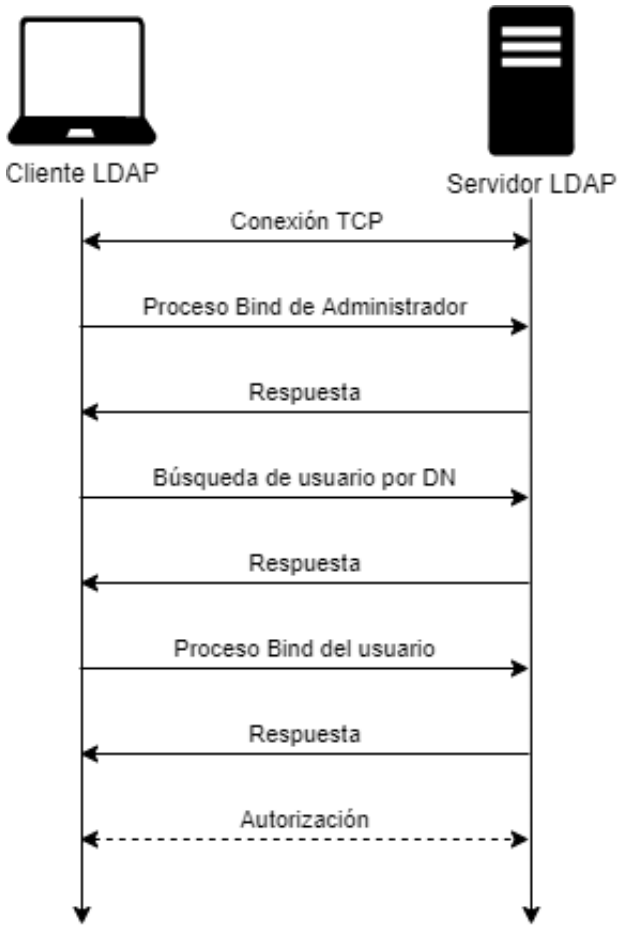


Figura 5: Ejemplo LDAP Bind.
Fuente: Elaboración propia.

La operación bind es una característica integrada de LDAP que es capaz de autenticar clientes con la base de datos LDAP. Consiste en tres partes, donde se necesita la versión de LDAP que se esta utilizando, que o quien está siendo autenticado y alguna prueba de identidad, que puede ser una contraseña o un certificado, entre otros. A este paso se pueden agregar verificaciones adicionales para que satisfagan las necesidades de la autenticación en cuestión.

Es posible usar dos tipos de autenticaciones con la operación bind: simple o SASL. En la autenticación simple, una contraseña es provista como prueba de identidad del usuario junto con el nombre del usuario o el DN (generalmente los servicios que utilizan LDAP tienen adicionalmente una configuración para saber donde va a estar el usuario en la base de datos por lo que son capaces de encontrar el DN). La contraseña es transmitida sin ningún tipo de seguridad, por lo que se hace necesario usar TLS o StartTLS para evitar problemas de seguridad.

Además, la operación bind permite autenticar usuarios sin ningún tipo de autenticación, lo que es denominado una *autenticación anónima*.

2.4.2. Kerberos

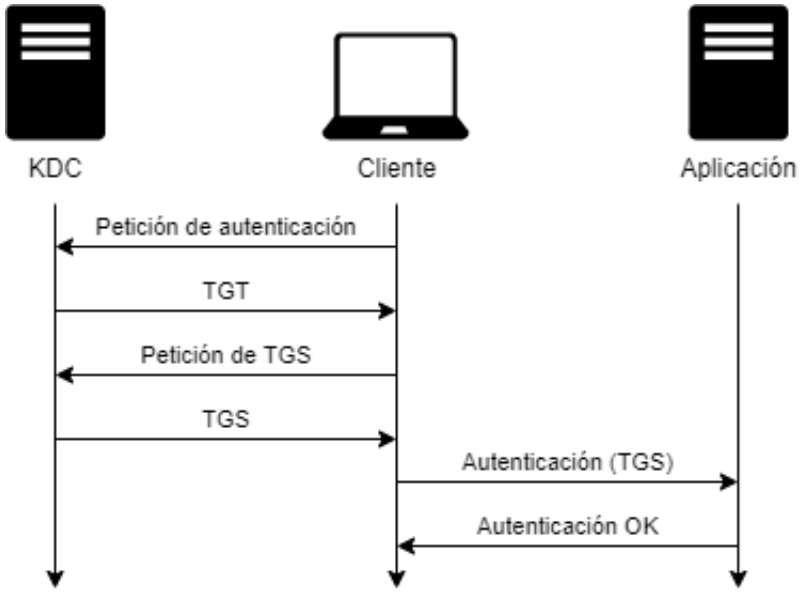


Figura 6: Autenticación de Kerberos.
Fuente: Elaboración propia.

Puede ser mejor descrito como un sistema de autenticación que permite identificación de manera segura sin necesidad de usar TLS. Es definido en el RFC 4120 (Neuman y col., 2005), y provee de llaves de autenticación llamadas “tickets”, los cuales utilizan cifrado simétrico, de tal forma que los tickets son comprobados por un tercero de confianza.

Kerberos funciona como encargado de la autenticación, entregando tickets: estructuras de Kerberos que permiten autenticar a un usuario en un servicio. Estos tickets vienen cifrados con algoritmos de cifrado modernos, de tal forma de evitar la obtención de credenciales por fuerza bruta.

Como podemos ver en la imagen, el cliente pide una petición de autenticación a Kerberos. Esto es asociado con el nombre de usuario, la fecha, entre otros. Este mensaje está cifrado y solo puede ser decodificado por el servidor de Kerberos (KDC: Key Distribution Center). Si el usuario se verifica correctamente, el KDC entrega un TGT (Ticket Granting Ticket) que corresponde a un ticket que puede solicitar tickets de acuerdo con las necesidades del servicio. Los datos vienen cifrados.

Para pedir un ticket de servicio, el cliente debe una petición de TGS (Ticket Granting Service) con el TGT necesario, y la información de usuario junto con la fecha. Finalmente, el KDC verifica que los datos sean correctos junto con el TGT, y si es que son correctos, se entrega un TGS que viene cifrado con un hash asociado al servicio.

Si todo está correcto, el cliente puede usar dicho TGS para entregarlo a un servicio y así efectivamente autenticarse en la aplicación.

A pesar de ser una buena opción de autenticación, por sí solo no es recomendable utilizarlo debido a que no puede almacenar información que podría ser necesaria para identificar y autorizar al usuario en sistemas (así como LDAP si puede). Sin embargo, es posible utilizar Kerberos y LDAP para asegurar seguridad e independencia de autenticación y autorización, así como hay servicios completos como los llamados Identity Management (*Idm*) que son un conjunto de herramientas y aplicaciones que aseguran la identidad de un cliente y autenticación segura (ej. RedHat Idm, FreeIPA, Samba AD DC, Active Directory), sin la necesidad de montar el sistema manualmente (LDAP y Kerberos). También estas aplicaciones buscan suplir necesidades adyacentes al proceso de autenticación (DNS, Intercambio de archivos, ACLs).

2.4.3. SASL

Definido en el RFC 4422 (Melnikov & Zeilenga, 2006), el sistema *Simple Authentication and Security Layer* (SASL por las siglas) se encarga de agregar compatibilidad con cualquier mecanismo que lo soporte. Además, se encarga de separar la autenticación del sistema de información, junto con proveer una capa de seguridad de los datos que ofrece integridad y confidencialidad.

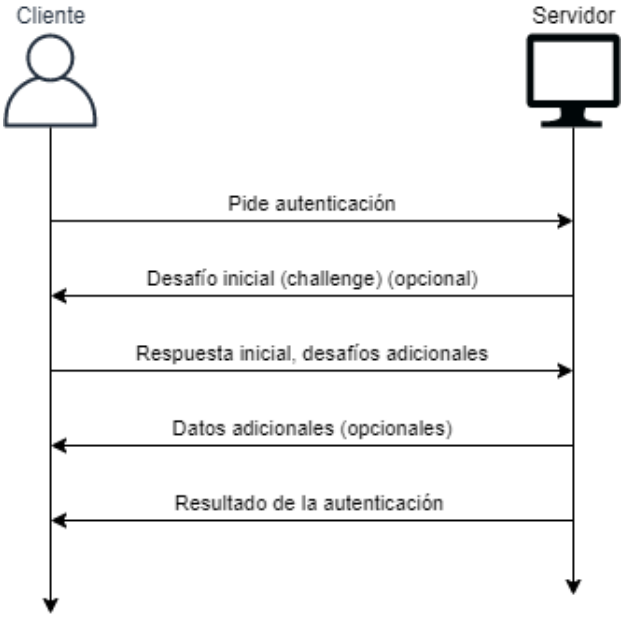


Figura 7: Autenticación a través de SASL.
Fuente: Elaboración propia.

Algunos de los mecanismos disponibles para SASL y que son comúnmente usados son:

- PLAIN
- KERBEROS_V5
- GSSAPI
- EXTERNAL
- ANONYMOUS
- NTLM
- GSS-SPNEGO
- OAUTHBEARER
- OPENID20

2.4.4. GSSAPI

Descrito en el RFC 4178 (Linn, 2000), el sistema *Generic Security Service Application Program Interface* (GSSAPI) es un mecanismo de autenticación SASL que define que el sistema de

autenticación debe funcionar con una API estándar de seguridad. La ventaja de éste es el uso con Kerberos lo que beneficia potencialmente a todos los servicios que implementen GSSAPI para traer compatibilidad con autenticación.

Una de las características de las aplicaciones con GSSAPI es el intercambio de *tokens* o códigos de seguridad que permiten el establecimiento de un contexto de seguridad a través de una red no segura. Éstos no son fácilmente visibles en el alto nivel, ofuscando los mensajes de autenticación y asegurando confidencialidad y autenticidad a la comunicación entre cliente y servidor.

2.4.5. SPNEGO

Dentro de GSSAPI existe SPNEGO (Simple and Protected GSS-API Negotiation Mechanism), un mecanismo encargado de negociar el sistema de seguridad a utilizar dentro de la autenticación. Debido a que Kerberos no puede funcionar por sí solo a través de HTTP, SPNEGO entra de forma que, la primera petición es donde se generan tokens para configurar el sistema que se utilizará para la autenticación, donde el navegador web se comunica con Kerberos (o NTLM) para así definir la identidad del usuario.

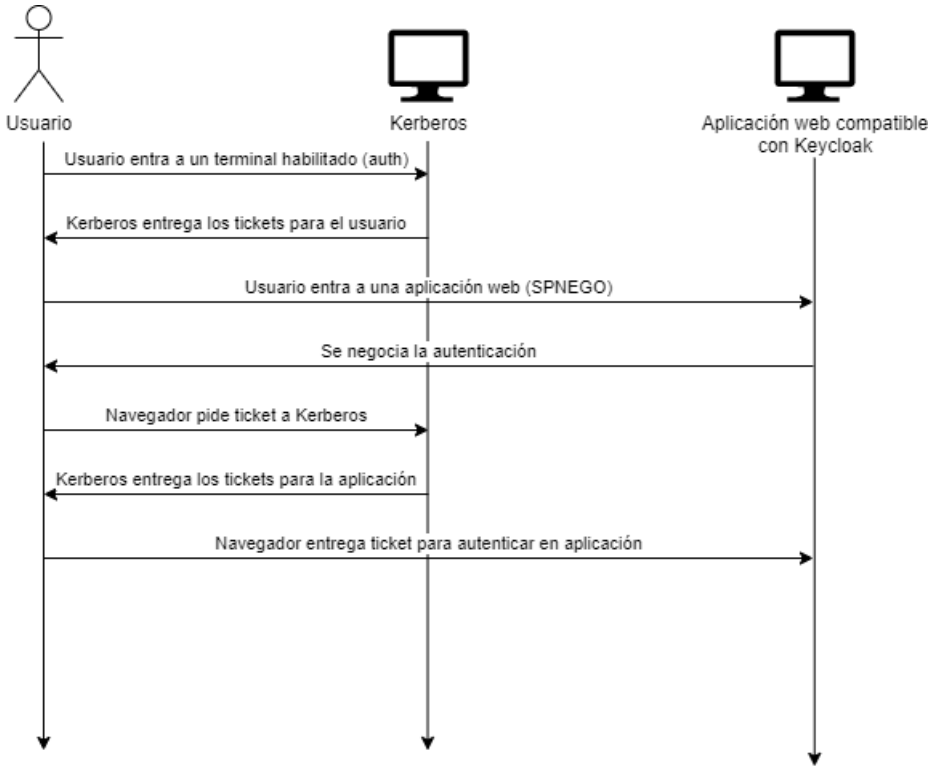


Figura 8: Flujo de interacción de aplicaciones web con SPNEGO.
Fuente: Elaboración propia.

2.4.6. Single Sign On (SSO)

Es un esquema de autenticación que permite a un usuario autenticarse frente a una aplicación con una única identificación a cualquier sistema que esté relacionado de alguna manera. Se considera como un sistema Single Sign On, cuando es posible autenticarse una única vez, y utilizar todos los sistemas habilitados que estén en el entorno correspondiente, sin tener que ingresar las mismas credenciales nuevamente.

Las ventajas de tener este esquema aplicado, del punto de vista de administración, es que los usuarios no necesitan distintos identificadores y/o contraseñas para entrar a distintos sistemas, sino que se logra la centralización de los datos de usuarios.

Para los usuarios es bastante más cómodo debido a que no necesita cambiar la clave en múltiples sistemas y solo necesita autenticarse una vez en el sistema para acceder a todas las aplicaciones protegidas.

Además, al estar los datos centralizados, podemos definir varias facilidades a la hora de los cambios de datos de algún usuario.

2.5. Otros protocolos actuales en uso

Para plantear nuestro plan integrado, debemos tomar en consideración todos los sistemas que se encuentran en uso y que dependen del sistema de autenticación actual. Dentro de los protocolos de autenticación es posible encontrar todos estos, debido a que las aplicaciones desarrolladas dentro del Departamento de informática hacían uso de distintos protocolos de autenticación que en su momento permitían la compatibilidad con LDAP. No hubo estandarización de estos protocolos hasta hace un tiempo, donde se dio el impulso para que todas las aplicaciones del Departamento de Informática utilizaran OpenID Connect.

A continuación, los detallaremos:

2.5.1. SAML

SAML (Security Assertion Markup Language) (Hughes & Maler, 2005) corresponde a un estándar que define un *framework* XML común para intercambio de información de seguridad entre distintas aplicaciones. Dentro de SAML existe un IdP (Identity Provider) que se encarga de autorizar la información que es provista a las aplicaciones, que en este caso se llamarían SPs (Service Providers). Junto a esto, también se proveen los atributos correspondientes para el acceso y control de los servicios de tal forma que todo el ecosistema de aplicaciones sepa concretamente que servicios puede acceder el usuario. Se logra, además, asegurar el Single Sign On (SSO) ya que el proceso una vez completado de autenticación y debido a que los servicios compartirían el mismo IdP, se hace conveniente la transferencia de la sesión, aumentando la seguridad y la experiencia de usuario.

Esto agrega una capa de compatibilidad sobre ciertas aplicaciones que solamente pueden utilizar este método de autenticación y no pueden moverse a protocolos más nuevos como OAuth. SAML es compatible con LDAP para obtener los datos de la sesión y luego SAML maneja las sesiones entre distintos sistemas por las capacidades de tener Single Sign-On.

2.5.2. OpenID Connect

OpenID Connect (Sakimura y col., 2014) es un protocolo que permite la autenticación de usuarios y aplicaciones, ya que es una capa de identidad por sobre el protocolo OAuth 2.0 (RFC 6749). Permite a los clientes autenticarse y obtener información de usuarios, con la particularidad que es similar a la arquitectura REST. OpenID Connect es un protocolo que permite la autenticación cubriendo los casos de uso de los anteriores. Si bien es difícil hacer una comparativa a los protocolos anteriores, podemos destacar que es uno de los más utilizados debido a que es una capa por sobre el ampliamente utilizado OAuth.

2.5.3. Redes Wi-fi

En el Departamento de Informática se utilizan Access Points (APs) de la marca Ruckus. En resumen, se utiliza el sistema 802.1X dentro de las redes inalámbricas lo cual permite hacer uso de la autenticación basada en Active Directory. Los APs permiten configurar el sistema de autenticación con Active Directory y RADIUS.

RADIUS (Rigney y col., 2000) consiste en el sistema de autenticación que se utiliza en el sistema 802.1X y se encarga de resolver las necesidades de autenticación por red. Un ejemplo de servicio que implementan este protocolo es FreeRADIUS que tiene soporte para LDAP, Kerberos, Active Directory, entre otros, para el desafío y autenticación externa.

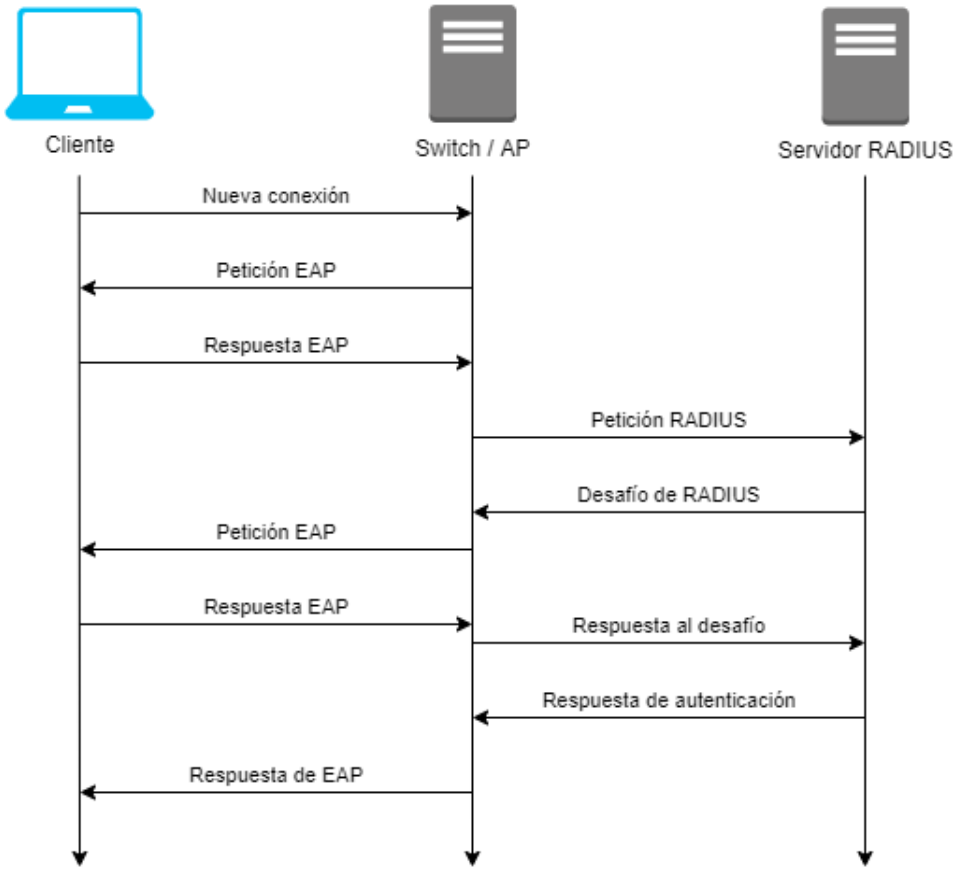


Figura 9: Diagrama del sistema 802.1X
Fuente: Elaboración propia.

2.6. Keycloak/Red Hat SSO

El sistema que nos permite fácilmente administrar los protocolos de autenticación es Keycloak/Red Hat SSO. Este sistema permite la adición de sistemas de autenticación como User Federation, lo que permite centralizar todo el sistema de autenticación y ver el protocolo con la adición de plugins. El sistema permite agregar LDAP y AD para su uso como base de datos de usuario y autenticación.

2.7. Ejemplos de aplicaciones que usan autenticación en el DI

2.7.1. Zimbra

Zimbra es la aplicación de correos que se utiliza en el Departamento de informática. Es utilizado, a diferencia de sus competidores, por la facilidad de uso y debido a que tiene un conjunto de herramientas muy completo para proveer servicios de correo electrónico.

El proceso de creación de cuentas de usuario debe ir acompañado por la creación de la casilla de correos electrónicos, junto con otras tareas que son necesarias para entregarle los servicios correspondientes al usuario.

Zimbra permite autenticación por Active Directory, LDAP o Kerberos. También es posible configurar la autenticación por SPNEGO, lo cual debe probarse en los sistemas propuestos.

De todos modos, la solución a este problema sería implementar la opción de autenticación por LDAP que FreeIPA también provee. Esto hay que evitarlo, ya que la idea es utilizar Kerberos para la autenticación ya que esto permite que el SSO funcione.

2.7.2. Intranet

Sistema del Departamento de Informática que sirve para actividades generales que pueden utilizar profesores y estudiantes. Está basada en Ruby on Rails y su autenticación en un principio estaba basada en SAML (debido a esto la búsqueda de sistemas que puedan proveer y mantener este protocolo). Últimamente se han realizado esfuerzos para cambiar la autenticación a OpenID Connect.

El sistema está cubierto por los protocolos que soporta Keycloak.

2.7.3. Sistema de prácticas

Sistema del Departamento de Informática que realiza lo que dice su nombre: ayuda a la recolección de datos de los estudiantes que están realizando prácticas, y actividades similares a estas. También ayuda a los profesores a cargo que pueden revisar el sistema, con un formato estandarizado para realizar las gestiones que requiere la aprobación de la práctica.

La autenticación está basada en OpenID Connect, lo que está cubierto por los protocolos que soporta Keycloak.

2.7.4. Sistema de ayudantías

Sistema del Departamento de Informática que recoge las postulaciones y la aprobación de éstas mismas, para agilizar el proceso de ayudantías dentro de las asignaturas dentro del mismo. La autenticación está basada en OpenID Connect, lo que está cubierto por los protocolos que permite Keycloak.

2.7.5. GitLab

Sistema del Departamento de Informática que guarda repositorios de código para los usuarios que lo necesiten, tanto en asignaturas como proyectos personales.

GitLab permite el uso de OpenID Connect para la autenticación[1], así como también LDAP.

2.7.6. Moodle

Sistema del Departamento de Informática que permite la utilización de foros, así como intercambio de archivos e información relacionada a asignaturas y todo relacionado a estudios.

Moodle permite OpenID Connect para la autenticación[2].

2.7.7. GLPI

Sistema de tickets o tiquetes, que permite a los funcionarios resolver problemas que puedan surgir dentro del Departamento de Informática. En vez de llamar a través del teléfono a los

[1] https://docs.gitlab.com/ee/administration/auth/oidc.html
[2] https://moodle.org/plugins/auth_oidc

funcionarios, se pueden agregar avisos a este sistema que permite agregar recordatorios, inventarios, enviar correos de soporte, entre otros.

Utiliza el protocolo OpenID Connect.

2.8. Otros sistemas adyacentes relevantes para la autenticación

2.8.1. NFS

Cada credencial de usuario debe ir acompañado de un proceso de creación dentro del punto de montaje NFS (Network File System). Esto es debido a que, para tener un sistema centralizado con intercambio de archivos, donde no importase cual terminal se utilice, los mismos documentos estarán ahí, es necesario utilizar el sistema AutoFS para montaje de carpetas de usuario, donde se utilice un sistema que indique a los terminales de que carpeta montar.

Esto solo se toma en consideración para terminales Linux debido a que en Windows ya tiene su propio sistema centralizado de archivos (SMB).

CAPÍTULO 3
PROPUESTA DE SOLUCIÓN

Dentro de nuestra propuesta tenemos que cumplir la mayor cantidad de casos de uso, por lo que a continuación describiremos las posibles vías de mejora del actual sistema de autenticación del DI, y luego describiremos cada uno de los componentes del sistema con sus ventajas y desventajas.

3.1. Plan 1: Procesos diferentes con comunicación por sincronización

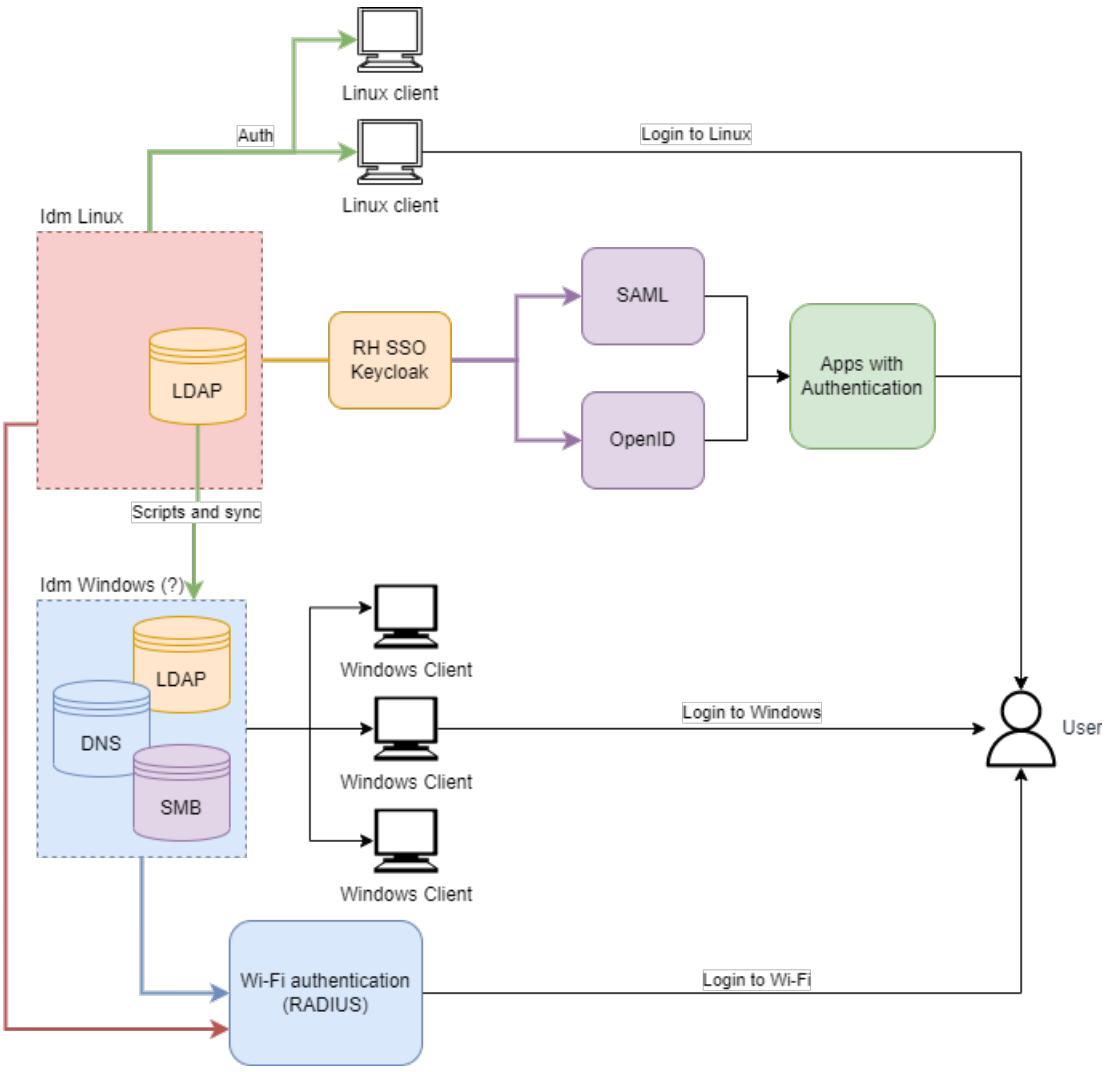


Figura 10: Plan 1.
Fuente: Elaboración propia

Dentro de este plan, proponemos un sistema similar al que existe actualmente: reemplazar el servicio LDAP por un Idm para subsistemas Linux (como podría ser FreeIPA o RH Idm) asi como conservar el sistema de AD que se encuentra actualmente en el DI, implementado con Samba AD DC.

Esto ayudaría en varios sentidos: se conservaría un sistema completo para la compatibilidad de terminales Windows y otros sistemas que no fueran compatibles con FreeIPA, pero agregaríamos las mejoras de tener un sistema centralizado como FreeIPA en vez de solo utilizar LDAP.

Esto traería sus problemas: necesitamos considerar que necesitamos un método de sincronización entre el Idm Linux y el Idm Windows a elegir, lo cual podría resolverse de acuerdo a las alternativas que tengan estos dos sistemas.

3.2. Plan 2: Sistema único con enfoque en Windows

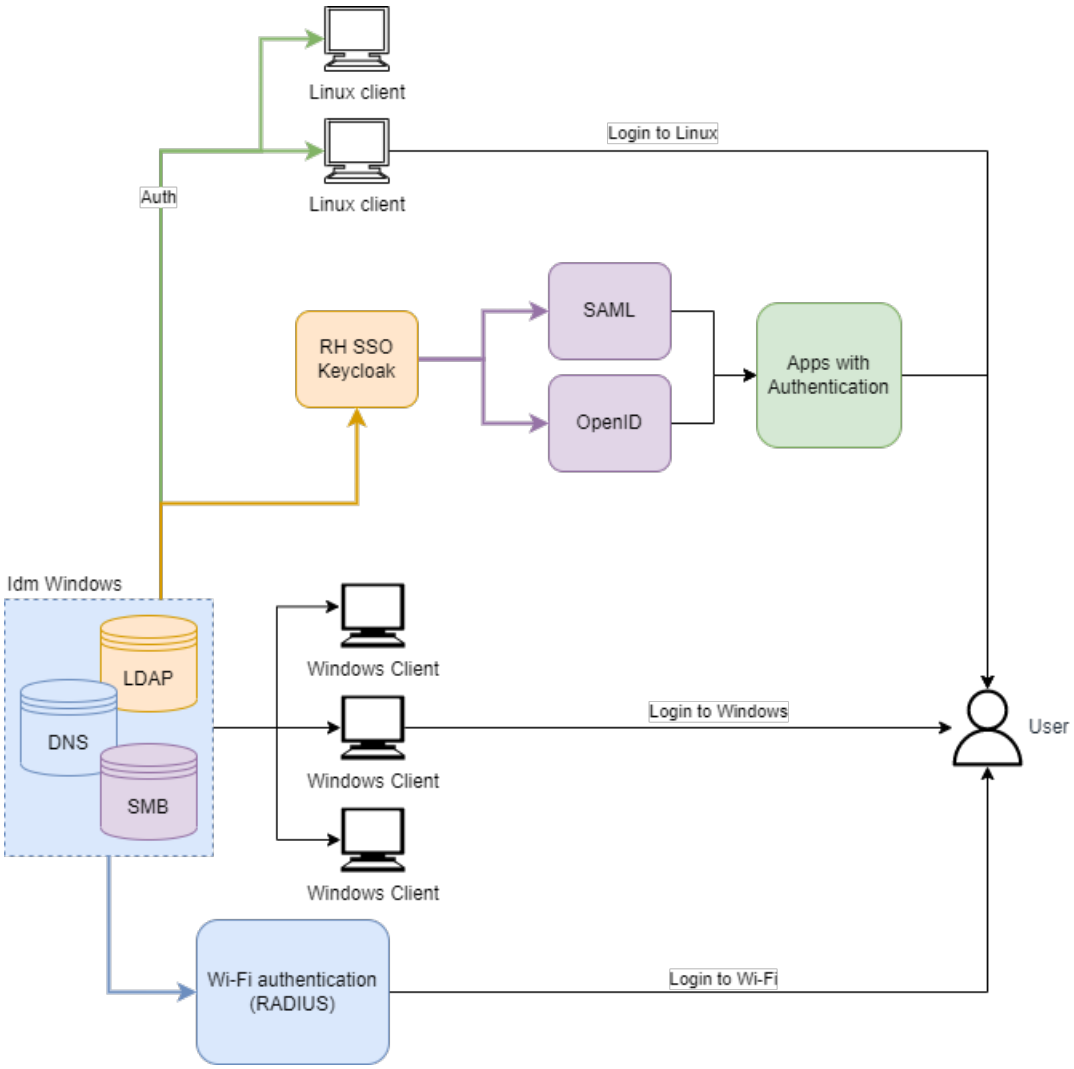


Figura 11: Plan 2.
Fuente: Elaboración propia

El siguiente plan sería resolver el problema de autenticación manteniendo solo un sistema de autenticación para el Departamento de Informática. Sería una vía de resolver adecuadamente el problema, pero entrarían factores a considerar dentro de éste que no serían problemas técnicos: licenciamiento, actualizaciones, mantenimiento, entre otros.

Por su parte traería las ventajas de mantener un servicio unificado y de acuerdo con el Idm que se pueda escoger, traería todas las ventajas que incluya la herramienta: es decir, si es que se escoge AD de Microsoft es muy probable encontrar soporte para el sistema mientras que es más complicado para Samba AD DC (SerNet ofrece el servicio Samba+).

En resumen, probablemente tendría más problemas para el futuro más que resolver posibles problemas actuales.

3.3. Plan 3: Sistema único con enfoque en Linux

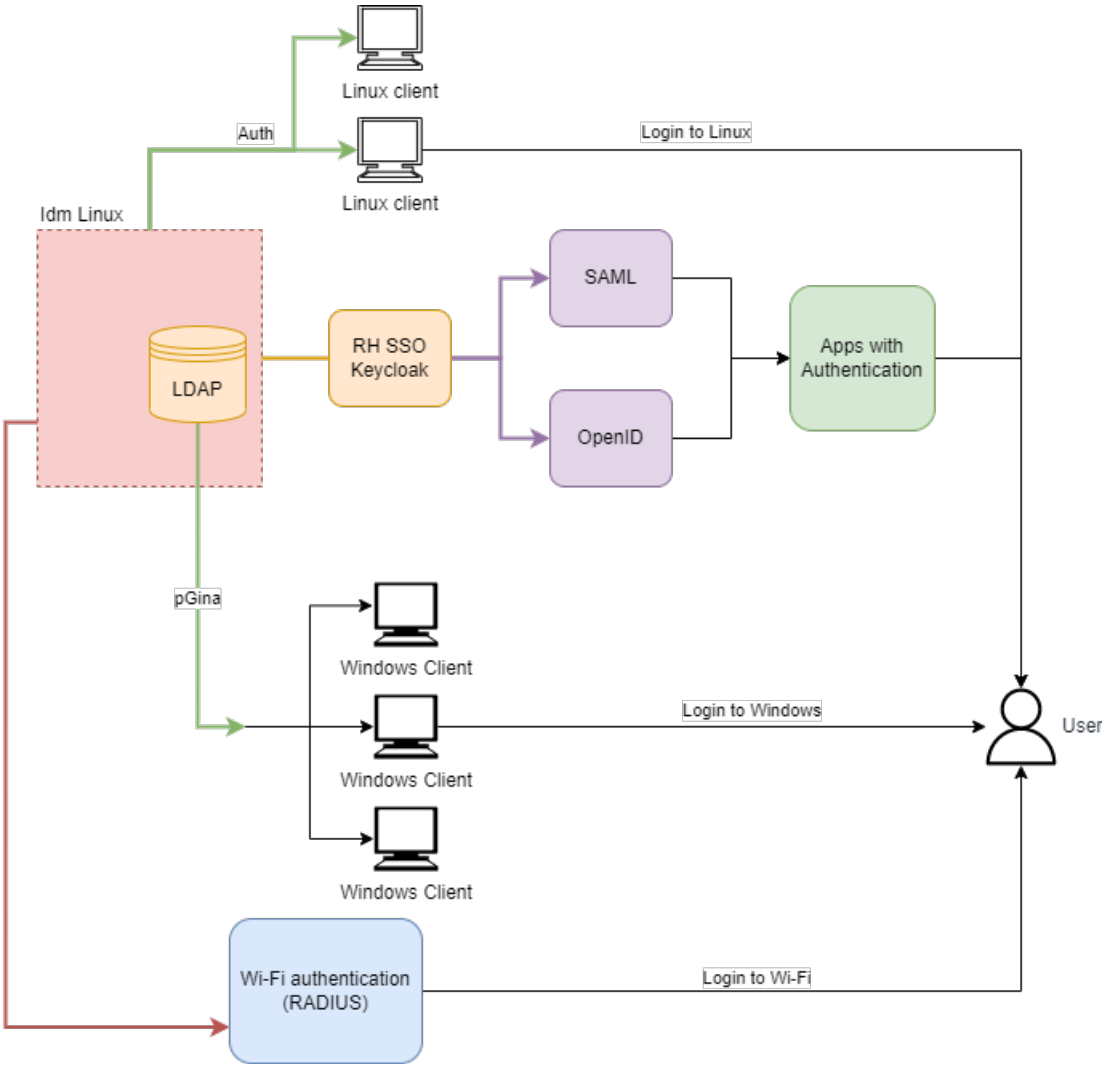


Figura 12: Plan 3.
Fuente: Elaboración propia

Por último, también se presenta la alternativa de utilizar sólo, como sistema centralizado al Idm Linux. En este caso se podrían resolver algunos problemas de compatibilidad para los terminales Windows con algunas herramientas, pero no sería una experiencia al 100 % adecuada comparable con la autenticación AD.

3.4. Tabla comparativa de planes

Características	Plan 1	Plan 2	Plan 3
Compatibilidad con equipos Windows out-of-the-box	✓	✓	✗
Compatibilidad con equipos Linux out-of-the-box	✓	✗	✓
2FA/MFA	✓	✗	✓
Crear grupos	✓	✓	✓
Agregar aplicaciones	✓	✓	✓
Disponibilidad en repositorios para servidores	✓	✓[1]	✓
Disponibilidad en repositorios para clientes	✓	✓	✓[2]
Compatibilidad con Keycloak	✓	✓	✓
Compatibilidad con SPNEGO	✓	✓	✓
Compatibilidad RADIUS	✓	✓	✓

Tabla 2: Características de cada uno de los planes

3.5. Descripción de componentes

3.5.1. FreeIPA

De acuerdo con lo analizado anteriormente, y siguiendo la misma línea anterior, Red Hat y la comunidad provee de un servicio llamado FreeIPA, orientado a ser una suite de servicios que apoyan todo el sistema de autenticación. También existe su contra parte de pago llamada Red Hat IdM que aplica el mismo sistema, pero con soporte dedicado y con diferente diseño.

Atributos	FreeIPA	Red Hat IdM
Soporte	✗	✓
Código Abierto	✓	✓

Tabla 3: Comparación entre FreeIPA & Red Hat IdM

La idea de esta solución es, según su documentación (Red Hat Inc., s.f.-c), *una solución integrada para identidad y autenticación para entornos de redes Linux/UNIX. Un servidor FreeIPA provee autenticación centralizada, autenticación e información de cuentas por medio de guardado de datos del usuario, grupos, dispositivos y otros objetos necesarios para administrar los aspectos de seguridad de una red de computadores*.

[1] Samba AD está disponible en repositorios de Fedora, CentOS y RHEL. Active Directory se encuentra incluido en Windows Server.

[2] pGina está disponible en directorio de GitHub y como paquete en Chocolatey.

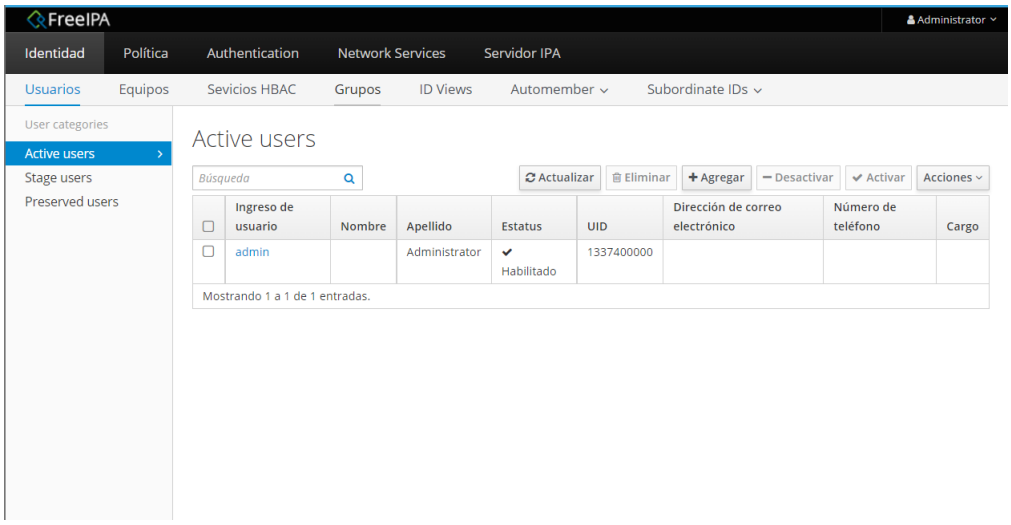


Figura 13: FreeIPA usuarios.
Fuente: Elaboración propia.

Debido a que los componentes principales serían parecidos a los que tenemos actualmente, no sería tan drástico el cambio necesario para implementar el nuevo sistema a excepción de algunos sistemas en concreto:

- LDAP si puede migrar credenciales cifradas a FreeIPA, pero debe pasar por un proceso de migración para transportar las credenciales de LDAP a Kerberos.
- FreeIPA es una caja negra en general, pero es posible interactuar con los componentes por separado. La idea es ocupar el sistema Idm en su totalidad y no solamente usar sistemas por separado (usar LDAP o Kerberos para la autenticación).
- Romperíamos un requerimiento de compatibilidad con Active Directory ya que actualmente se utiliza un script de sincronización de LDAP con AD. Sin embargo, como FreeIPA está basado en 389ds es posible que el sistema de sincronización actual funcione de la misma forma. Además, se podría tener en consideración recurrir a otras alternativas en compatibilidad que incluye FreeIPA (cross-forest trusts o sincronización alternativa).
- Cambios al *schema* de LDAP implicarían desarrollar módulos de FreeIPA para que luego puedan ser fácilmente modificables en la página web.

FreeIPA permite la configuración de servidores RADIUS[3], protocolo el cual son compatibles los dispositivos Ruckus.

[3] https://www.freeipa.org/page/Using_FreeIPA_and_FreeRadius_as_a_RADIUS_based_software_token_OTP_system_with_CentOS/RedHat_7

FreeIPA permite la configuración con Zimbra a través de su compatibilidad con autenticación LDAP[4] y usar Kerberos con SPNEGO[5], en caso de que los usuarios ya tengan su ticket de Kerberos en el terminal a usar.

Por último, Keycloak permite la configuración de LDAP y Kerberos para la autenticación en conjunto, por lo que tendríamos además compatibilidad para las aplicaciones con protocolos de autenticación antes vistos.

3.5.2. pGina

Para el caso de uso de utilizar terminales Windows, se pudo encontrar alternativas de autenticación, sin necesidad de tener un sistema basado en AD. Como posibles alternativas, existe pGina: una implementación del proveedor de credenciales de Windows open source. La ventaja de usar este sistema es que al ser open source, está abierto a posibles desarrollos de nuevos plugins.

Otra ventaja de usar este sistema es que no sería necesario usar un Active Directory o Samba para lograr la compatibilidad con Windows, asumiendo que no nos interesa guardar la información del usuario en un sistema centralizado.

Este sistema se implementó en un cliente Windows, utilizando el backend de LDAP con FreeIPA y es posible entrar el terminal. Esto cumpliría con un caso de uso, pero es posible que no sea suficiente.

Entre las desventajas se encuentran las posibles fallas de seguridad que puedan traer el sistema, debido a que no posee mantenimiento desde el 2013 (obtenida la fecha desde GitHub). Además, agregar que, al ser solo autenticación, no posee ninguna ventaja en comparación a un Active Directory o similares. De acuerdo con nuestros requerimientos, se pudo concluir que el plugin de LDAP sería útil para utilizarlo en conjunto con FreeIPA, pero no daría la ventaja de SSO ni la seguridad que traería Kerberos, ni poder entregar los mismos documentos de los usuarios en distintos terminales Windows.

Además, agregar que otra desventaja de implementar esta solución sería el hecho que tendríamos que descargar la aplicación de su repositorio de GitHub y habilitar de alguna manera la aplicación para su instalación centralizada (para Linux se usan scripts de instalación y repositorios).

[4] https://wiki.zimbra.com/wiki/King0770-Notes-External-Authentication-with-LDAP
[5] https://wiki.zimbra.com/wiki/Configuring_SPNEGO_Single_Sign-On

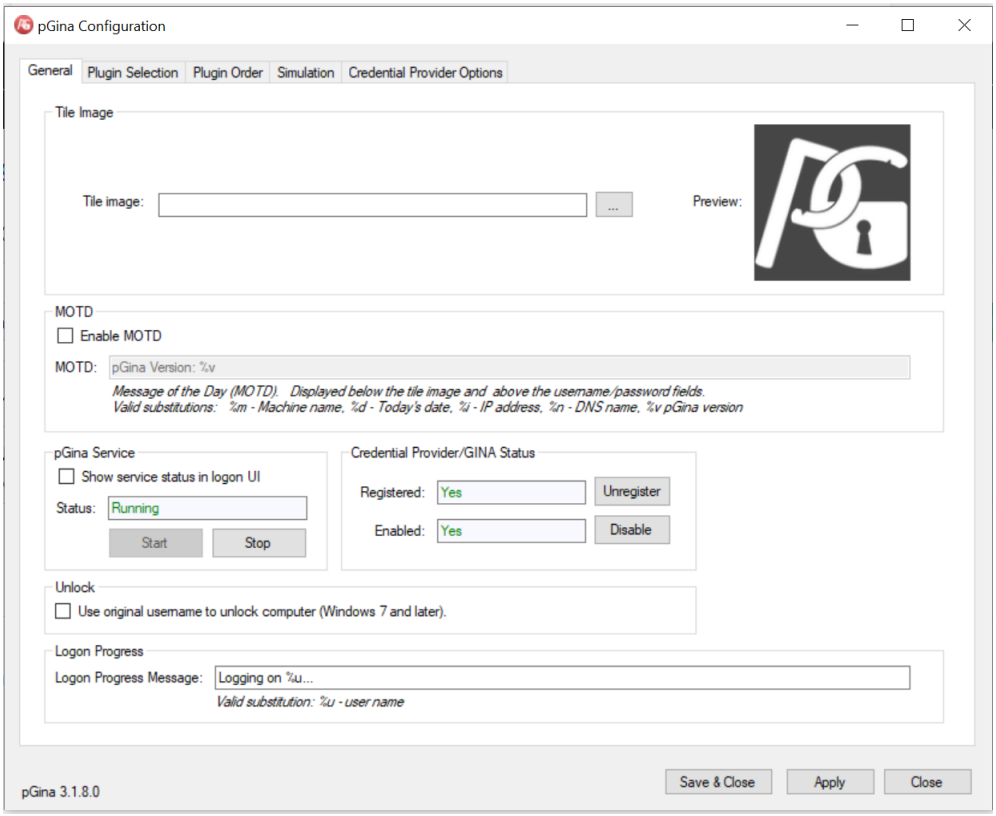


Figura 14: Interfaz de pGina.
Fuente: Elaboración propia.

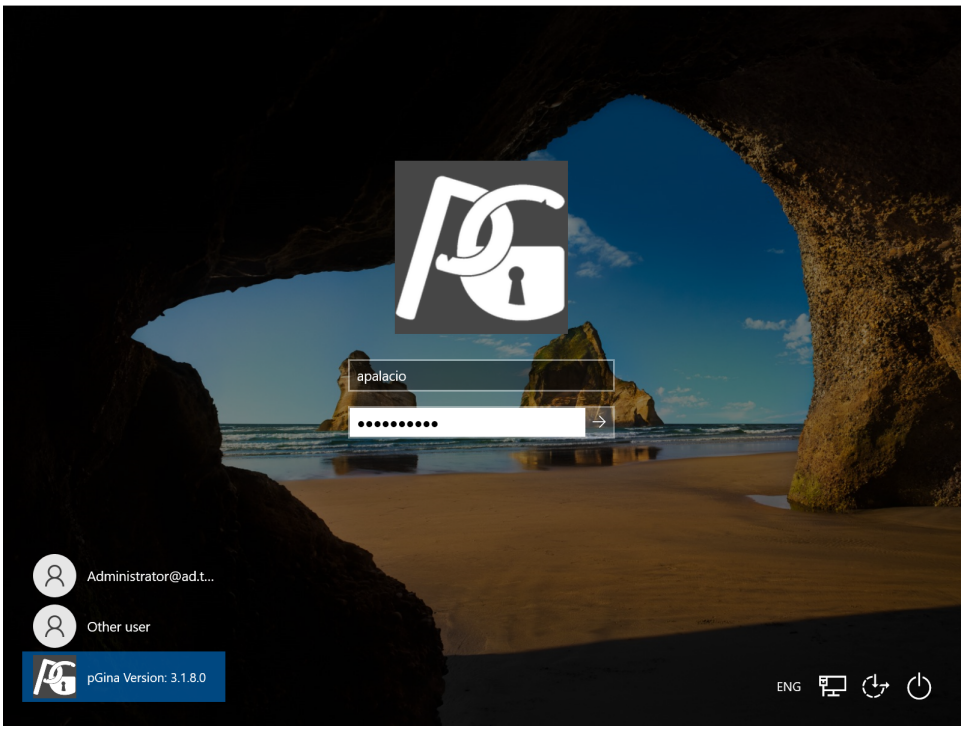


Figura 15: pGina en pantalla de inicio de Windows.
Fuente: Elaboración propia.

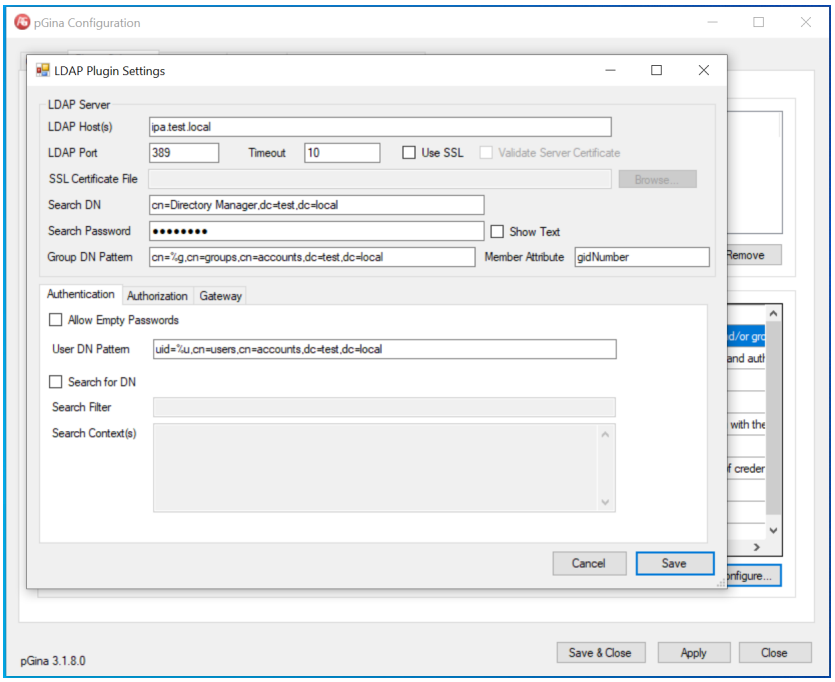


Figura 16: Configuración de LDAP en pGina.
Fuente: Elaboración propia.

3.5.3. Active Directory

Como ya vimos anteriormente, en teoría es posible tomar el control de la autenticación con Active Directory. Sin embargo, esto tiene varias ventajas y desventajas:

Ventajas:

- Se podrían unificar todos los servicios: debido a que varios sistemas implican el uso de un Active Directory en específico, se podría omitir el paso de considerar un Idm para usuarios Linux y simplemente optar por utilizar Windows Server y la solución de AD. Esto podría facilitar la creación de usuarios y asignación de recursos ya que el servicio estaría en un sistema centralizado.
- Existe un módulo de PAM para AD: para la solución de máquinas o terminales Linux, se podría implementar un módulo de autenticación que tiene PAM para permitir el uso de credenciales AD en Linux. Sería un proceso similar al que se tiene para habilitar la configuración por LDAP.

Desventajas:

- El servidor se tiene que implementar con Windows Server, lo que requeriría la posesión de licencias del sistema antes mencionado. La universidad posee dichos accesos a licencias de software Microsoft, pero como el Departamento de Informática generalmente promueve el uso de sistemas Linux e incluso en los laboratorios se utilizan dichos sistemas (a excepción del Laboratorio de Proyectos Informáticos que usa terminales Windows), no tendría mucho sentido construir infraestructura con Windows hoy en día.
- La administración de dichos servidores es mucho más compleja que el actual sistema de administración de servidores RHEL: esto es debido a que Windows es raramente compatible con aplicaciones construidas originalmente en Linux/Unix y viceversa. Últimamente se ha visto la implementación de Windows Subsystem for Linux (WSL) pero en general no permite una implementación de aplicaciones Linux para manejar completamente el sistema operativo base: es conocido más como una herramienta para desarrolladores. Herramientas como Ansible facilitarían el manejo y configuración, pero requerirían un proceso largo de estudio y testeo.

3.5.4. Samba AD DC

Samba es el subsistema de Linux para compatibilizar los sistemas de Windows con un servidor Linux. Implementa muchas características de éste, pero tiene ciertas desventajas:

Ventajas:

- Capa de compatibilidad con Windows: La ventaja de utilizar Samba es la posibilidad de utilizar el modo AD DC para que sea compatible con la autenticación de Windows y administrarlo desde allí, sin necesidad de habilitar un servidor Windows lo cual agregaría mucho valor al sistema de administración.
- Módulo para PAM para autenticación (*pam_winbind*): Al igual que en Active Directory, es posible compatibilizar la autenticación AD con un módulo especial de PAM y terminales Linux.
- RH-SSO es compatible con AD al incorporarlo como base de datos de usuarios: RH SSO/Keycloak es compatible con AD en general para obtener usuarios y usarlos dentro del sistema de autenticación.

Desventajas:

- Falta de características esenciales dentro de Samba que si posee AD: Hasta el momento no es posible tener un cross forest trust entre FreeIPA y Samba AD DC, cosa que si existe con Active Directory. Esta es una característica que no se encuentra junto a muchas otras que todavía faltan por implementar. SerNet ha entregado la información en donde junto con Red Hat han avanzado en este sistema, pero todavía a fecha de hoy (versión 4.15.5) falta desarrollo.

3.5.5. Elección de plan a recomendar y a validar

Para el plan integrado se propone utilizar una combinación entre FreeIPA y Samba (Plan 1), lo que fortalecería la seguridad, agregaría la compatibilidad con la sincronización y tendría compatibilidad con sistemas de uso actuales. Además, debemos agregar que se tiene que diseñar un proceso de migración del antiguo sistema hacia el nuevo.

Todo esto será testeado en un entorno similar al real con datos reales para que, en caso de implementación, todo lo diseñado sea lo más cercano posible para el administrador de infraestructura.

3.6. Sistemas nuevos dentro del proceso de autenticación

3.6.1. 2FA y MFA

2FA y MFA viene de la necesidad de que las contraseñas no son el único medio para que un usuario acredite su identidad. El creciente problema de que las contraseñas simples son relativamente fáciles de obtener por fuerza bruta, se planeó la adición de nuevas capas de

seguridad que permitan al usuario fácilmente acceder con respecto a algo que solamente el usuario posea o sepa.

Entre posibles soluciones de MFA existen:

- PIN de autenticación por medio de una aplicación en otro dispositivo.
- PIN recibido por SMS.
- Preguntas de seguridad.
- Hardware especializado que tenga dichas llaves (Yubikey).
- Escaneo de huella digital o iris.

Uno de los métodos más comunes que existen para 2FA es el llamado OTP (one-time password). Si bien existen múltiples implementaciones, el más comúnmente disponible es el que permite que la contraseña sea válida solamente durante un periodo de tiempo, por lo que el tercero no podría saber esta contraseña adicional a menos que pudiera encontrar el hash correspondiente entre el servidor y usuario.

Para obtener la contraseña asociada al OTP existen múltiples dispositivos y formas, sin embargo, el más común es por medio de una aplicación en un smartphone compatible. Otro medio es enviar la contraseña por medio de SMS, lo cual también resulta efectivo.

CAPÍTULO 4
VALIDACIÓN DE LA SOLUCIÓN

4.1. Implementación del plan integrado

A continuación, relataremos en parte, lo aplicado en un laboratorio de pruebas con respecto a la información adquirida del actual sistema de autenticación. Se comenzó adquiriendo parte de la estructura principal del actual LDAP que se utiliza en el Departamento de Informática. Esto era necesario para que pudiéramos replicar el servidor utilizado y todas las variables de este servidor. Esto lo logramos gracias a credenciales facilitadas por el Departamento de Informática, y ciertos comandos dentro de la subred de los equipos del Laboratorio de Computación.

Una vez logrado esto, montamos un servidor 389ds en nuestro laboratorio de pruebas, utilizando parte del esquema obtenido, y creamos un usuario ficticio que cumplía con las condiciones de ser usuario del Departamento de Informática.

En otra máquina, instalamos FreeIPA con sus configuraciones por omisión. Esto está más detallado en la Subsección A.2.

Luego de verificar de que el usuario fuese válido, utilizamos los scripts de migración (véase Subsección A.4) para confirmar que el sistema de migración funcione bien. Se logró esto sin ningún problema, sin embargo, debido a que las contraseñas fueron guardadas en formato LDAP, se debe hacer un paso adicional para migrar completamente. Esto consiste en: ir a una página en específico o entrar a un terminal y realizar el proceso de cambio de contraseña. Este proceso verifica que la contraseña esté correcta, comparando con la antigua contraseña de LDAP, luego pide un cambio de contraseña, y sincroniza LDAP y Kerberos con la nueva contraseña.

Para la capa de compatibilidad, instalamos Samba 4 en el modo Domain Controller, lo cual es descrito en la Subsección A.3.

Además, activamos el plugin de sincronización en FreeIPA, verificamos que podía sincronizar correctamente los usuarios que utilizamos de prueba y habilitamos un cliente Windows para verificar la funcionalidad de login.

Habilitamos otro cliente, esta vez Fedora, para verificar que los usuarios agregados puedan añadirse correctamente. Esto involucró un poco de configuración adicional, sobre todo con el sistema AutoFS (sistema de centralizado de montaje de archivos).

Finalmente, habilitamos otro servidor con el sistema Keycloak que actualmente está siendo utilizado por el Departamento de Informática, para implementar la capa de autenticación dentro de las aplicaciones para estudiantes y profesores, y agregamos la configuración co-

rrespondiente para la autenticación con FreeIPA.

4.2. Medidas de seguridad recomendadas

Al FreeIPA tener componentes LDAP y Kerberos, también es posible que sea vulnerable a ciertos problemas de estos componentes por separado.

4.2.1. LDAP TLS

El asegurar que la comunicación sea exclusivamente realizada por TLS, asegura que la información esté cifrada para la comunicación cliente-servidor en caso de algún compromiso en la red. Para esto, debemos asegurarnos de que LDAP se comunique solo con TLS y que los clientes usen dicho protocolo para pedir información. Una medida también efectiva sería deshabilitar el puerto 389 para la comunicación sin TLS, pero podría tener problemas de compatibilidad, por lo que queda a criterio del administrador.

4.2.2. Enumeración LDAP

Un problema común que se encuentra con LDAP es la enumeración de distintos componentes del directorio (usuarios, equipos, organizaciones, entre otros). Para dicho problema, una mitigación posible es bloquear el acceso anónimo de LDAP de tal forma que no sea accesible para terceros en caso de que la red esté comprometida.

Véase parámetro `nsslapd-allow-anonymous` en 389ds.

4.2.3. Kerberoasting

En este apartado y generalmente los ataques relacionados con Kerberos, son siempre relacionados al compromiso del ticket de Kerberos para autenticación, por lo que hay que asegurar estos tickets dentro de los clientes.

En el caso de Kerberoasting[6], un ataque especializado a Kerberos, se aprovecha el tipo de cifrado RC4 de los tickets de Kerberos en Active Directory los cuales son muy vulnerables a fuerza bruta. Esto nos obliga a que no debemos utilizar ese tipo de cifrado en lo posible.

Tenemos que analizar con cuidado el configurar este tipo de cifrador debido a que Samba 4 lo usa para compatibilidad con Windows, por lo que hay que estudiar bien cuales son las

[6] https://attack.mitre.org/techniques/T1558/003/

consecuencias de esto.

4.2.4. Medidas de red

Se pueden realizar medidas de seguridad que son comunes para todo servicio pero que ayudaría a este servicio en general:

- No habilitar más puertos de los necesarios dentro del servidor.
- Habilitar el servicio FreeIPA en una subred determinada. El servicio que responde a internet va a ser Keycloak/RH SSO y tiene que solo exponer la capa externa de autenticación.
- Revisar y analizar tráfico de red periódicamente, que pueda ser malicioso o inusual.

4.3. Evaluación del plan integrado

Detallaremos ciertos casos de uso para lograr definir el alcance de este plan de mejora:

1. El usuario puede usar aplicaciones del Departamento de Informática, dentro y fuera de la universidad, autenticándose con sus credenciales de usuario: El uso de Keycloak permite esto.
2. El usuario debe ser capaz de ingresar en un computador habilitado sin importar el sistema operativo: Esto se logra debido a la capa de compatibilidad FreeIPA-Samba a través de la sincronización de usuarios.
3. El usuario debe ser capaz de aprovechar las ventajas del Single Sign On: A esto se le agrega la ventaja que Keycloak reconoce el ticket de Kerberos que está en el terminal donde ingresó el usuario. Lamentablemente para usuarios Windows y usuarios que estén desde otros equipos tienen que pasar por la ventana de autenticación. Esto último es debido a que Keycloak no permite la utilización de tickets RC4 de Kerberos. Habría que analizar la posibilidad de que Samba AD pueda cambiar el sistema de cifrado de tickets. Además, se agrega la compatibilidad para SPNEGO, comprobada en Zimbra.
4. Debe existir una (o más) política(s) de contraseñas: FreeIPA posee políticas de contraseñas.
5. El usuario debe obtener la posibilidad de usar 2FA y/o MFA: FreeIPA posee sistema de 2FA y MFA. Queda pendiente de investigación si es que el sistema de OTP puede hacerse compatible con el OTP integrado de Keycloak.

6. El usuario debe ser capaz de obtener permisos en base a su categoría dentro de la universidad (ayudante, estudiante, ayudante administrativo, profesor, administrador, etc.): En este caso, se logra agregar permisos de forma centralizada a través del portal de FreeIPA.

7. Debe haber un sistema que permita cambiar permisos de ayudantes: FreeIPA tiene acceso rápido a permisos de usuarios y grupos. Además de esto tiene SUDO rules y ACLs.

8. Debe haber un sistema que permita reiniciar la contraseña de acuerdo con la necesidad de un administrador o un ayudante: FreeIPA permite el reinicio de la contraseña de un usuario siempre y cuando dicho proceso sea ejecutado por un administrador. Después al usuario se le pide cambiar la contraseña debido a que se entrega con un estado de expirada.

9. Usuarios deben poder cambiar su contraseña: Es posible cambiarla siempre y cuando los usuarios sepan la antigua. Existen proyectos para reinicio de contraseña por SMS que complementan esta característica de FreeIPA o podría adaptarse el sistema de reinicio de contraseña por número de serie de cédula de identidad.

10. Usuarios deben poder cambiar sus datos personales o agregar nuevos datos (siempre y cuando el Departamento de informática o la universidad lo permitan): FreeIPA permite el cambio.

11. Debe haber un sistema que permita entregar permisos a aplicaciones o servicios, en caso de que lo requieran: Es posible hacer cuentas de sistema en LDAP y cuentas de servicios en Kerberos.

12. Los sistemas propuestos deben estar disponibles en los repositorios oficiales de algún sistema operativo Linux, en el caso de servidores Linux, debido al tiempo limitado de los administradores y la complejidad de mantener código sin empaquetar: FreeIPA servidor y cliente están disponibles en los repositorios de Fedora y RHEL. Samba AD también está disponible.

13. Las alternativas deben garantizar que van a ser compatibles de un modo u otro con las aplicaciones y servicios existentes que lo requieran: El mínimo de FreeIPA es ser compatible con el sistema LDAP, y se comprueba que puede funcionar completamente sin utilizar Kerberos, por lo que sería compatible con el actual sistema de autenticación sin necesidad de migrar inmediatamente a Kerberos.

También aparte existen ciertos detalles que se requieren para planificar correctamente el plan de mejora:

- La implementación FreeIPA y migración desde LDAP implica que las contraseñas deben migrar desde el sistema de hashes LDAP al sistema de Kerberos. Esto se debe comunicar correctamente al usuario y para facilitar la transición del usuario, se recomienda

desarrollar un plugin que pueda o reiniciar la contraseña desde Keycloak hacia FreeIPA, o avisarle al usuario que ellos deben primero cambiar la contraseña desde un portal web. En este caso, se desarrolló una prueba de concepto que solamente tiene una redirección a la página de FreeIPA, pero en el caso de implementación de este plan, se debe desarrollar algo más robusto dependiendo de las necesidades de los usuarios.

- FreeIPA permite la modificación del esquema de LDAP y permite añadir plugins para su uso en el portal web. Estos plugins están pendientes de desarrollo debido a que escapan del tema de esta memoria.
- Como relatábamos anteriormente, los clientes Windows no son capaces de utilizar las capacidades SSO de Kerberos, debido a que Keycloak no soporta el cifrador RC4 que poseen los tickets de Kerberos emitidos por Samba.

CAPÍTULO 5
CONCLUSIONES

En este trabajo se pudo determinar los alcances de un sistema de autenticación centralizado, y proponer un plan de mejoras analizando fortalezas y debilidades. En nuestro contexto, el Departamento de Informática tiene mucho alcance en distintos sistemas informáticos, y cada día se hace más complejo la administración y rol de cada usuario. Los sistemas propuestos entregan distintas respuestas a los problemas presentados. Si bien ninguno puede cumplir con todos los requerimientos, cada uno tiene sus ventajas por sobre el otro y entrega soluciones de acuerdo con el contexto de los desarrolladores que lo tuvieron en mente.

Además, destacar que últimamente Linux posee bastante influencia en sectores públicos y privados: en servidores el sistema operativo por excelencia es alguna distribución de Linux, debido a la gran capacidad de automatización y por ser código abierto, lo que no requiere costos extras en licencias en comparación a Windows Server.

A pesar de esto, Windows Server no se queda atrás en el despliegue de servicios que son relacionados a Microsoft. Además, como se comentó anteriormente, el Departamento de Informática promueve el uso de sistemas de código abierto, lo que se refleja en los servicios entregados a los estudiantes (laboratorios, aplicaciones, ciertas asignaturas que hacen uso de Linux).

Todo esto, junto a las restricciones del problema nos llevó a concluir que una solución híbrida, Samba AD y FreeIPA, junto con un sistema intermedio de sincronización de usuarios, sería la solución correcta para implementar dentro del Departamento de Informática.

La ventaja de FreeIPA es que junto a la compatibilidad que posee por defecto con servidores LDAP, está basada en 389ds, lo cual nos deja aprovechar todas las características preexistentes que puedan estar dentro del Departamento de Informática, siempre y cuando estos sistemas no permitan el uso del sistema de autenticación Kerberizado (sistemas como Zimbra), por lo que para estos es posible usar la base de datos LDAP hasta que pueda ser solucionado este detalle en los servicios.

5.1. Trabajo futuro

5.1.1. Cross Forest trust FreeIPA Samba AD

Es posible que dentro de las próximas versiones de Samba AD y FreeIPA se implemente la posibilidad de agregar un cross forest entre ambos sistemas. Esto es debido a que ambos sistemas están bastante ligados por temas de compatibilidad con Windows. De hecho, SerNet y Red Hat han estado trabajando de cerca en Samba.

En el estado actual de este, se han podido destacar avances con los cross forest de Samba AD con Windows Server y FreeIPA con Windows Server (que utiliza como capa de compatibilidad a Samba AD).

En Anexos, Subsección A.8 se explica la posible implementación que se requiere para implementar este sistema.

5.1.2. Plugin de compatibilidad y migración Keycloak y FreeIPA

Es posible que el proceso de migración sea demasiado complejo para algunos usuarios del Departamento de Informática. Es por lo que que es posible desarrollar un plugin de compatibilidad FreeIPA/Keycloak que sea capaz de implementar el cambio de contraseña en el mismo portal de Keycloak, para evitar tener que redirigir al usuario al portal de FreeIPA en el proceso de migración.

Dentro del desarrollo de esta memoria, se realizó una prueba de concepto donde se ponía a prueba la API de plugins que provee Keycloak y fue posible agregar una opción a los usuarios donde se podía entregar un link para redirigir al usuario al portal de FreeIPA, por lo que ciertamente es posible desarrollar un plugin nuevo para resolver este aspecto.

5.1.3. Sistema automático de creación de cuentas

Si bien es posible crear cuentas a través de la CLI (Command Line Interface) y a través del portal web, se hace necesario la creación de una herramienta que pueda crear masivamente cuentas de usuario para el proceso de inscripción de estudiantes a inicio de año. El proceso de creación de cuentas conlleva una serie de pasos que detallamos a continuación:

1. El administrador debe calcular el posible nombre de usuario de acuerdo con ciertas reglas.
2. El administrador debe crear el usuario con una contraseña aleatoria.
3. El administrador crea la carpeta en el NFS correspondiente.
4. El administrador crea la casilla de correos en el servidor de correos correspondiente.

Estos puntos se pueden automatizar fácilmente, desarrollando una herramienta a cargo de que pueda generar los puntos vistos anteriormente.

Hay ejemplos en la documentación de una librería escrita en Python para hacer uso de la API de FreeIPA lo que simplificaría bastante el desarrollo de dicha herramienta.[7]

[7]Página web de dicha bibloteca: https://github.com/nordnet/python-freeipa-json.

5.1.4. Reinicio de contraseñas on-demand

Actualmente dentro de FreeIPA, existe el problema de que en caso de que la contraseña se pierda, un administrador debe reiniciarla. Sin embargo, es complicado muchas veces tener la oportunidad de hacerlo, por lo que muchas veces se propone el sistema de reinicio de contraseñas on-demand, lo cual permite que, con una confirmación de identidad externo, se pueda reiniciar la contraseña de un usuario. Este no es un sistema 100 % infalible, y generalmente debe hacerse tomando todas las medidas de seguridad que corresponden.

Dentro del Departamento de informática, existía un proyecto de reinicio de contraseñas que tomaba en consideración los datos de la cédula de identidad, cosa que seguía en desarrollo. El implementar esta alternativa sería muy efectivo para este caso de uso, y debido al punto anterior, se puede cambiar la contraseña haciendo llamadas a la API de FreeIPA.

También existe un proyecto[8] encargado de realizar un reinicio de contraseña a través de un código por SMS, por lo que es posible también utilizar este tipo de sistemas para el caso de uso que hablábamos.

5.1.5. Compatibilidad OTP (2FA) FreeIPA-Keycloak

Dentro de nuestro plan de mejora es posible destacar una gran ventaja de añadir sistemas de autenticación de múltiples factores. Esto es una gran ventaja debido a que agrega un gran factor de seguridad a los usuarios del Departamento de informática en caso de manejar datos de gran confidencialidad. Sin embargo, existe una dualidad dentro de esto sistema de autenticación: Keycloak y FreeIPA implementan su propio sistema de OTP, lo cual podría llevar a que la habilitación del servicio en uno de los 2 no se aplique correctamente en el otro.

Por lo tanto, queda pendiente la investigación de la compatibilidad y un eventual desarrollo de un aplicativo que ayude a la compatibilidad de ambos servicios en este respecto.

[8] https://github.com/larrabee/freeipa-password-reset

ANEXOS

A.1. Instalación de LDAP

Para lograr lo planteado en este documento, se necesita instalar una instancia de pruebas que sea lo más similar posible a lo que se puede encontrar en el Departamento de Informática "LDAP de producción" de ahora en adelante).

La VM que se utilizará para este servidor es la siguiente:

- Sistema Operativo: CentOS 7 (muy similar a RHEL)
- RAM: 512 MB
- vCPUs: 1
- IP: `192.168.122.97`
- Nombre de máquina: `ldap.test.local`
- Paquetes por instalar (`yum install -y`):
  - epel-release
  - idm-console-framework
  - xorg-x11-xauth
  - 389-ds-base
  - 389-admin
  - 389-admin-console
  - 389-ds-console

Iniciar el proceso de instalación que es posible con el script `setup-ds-admin.pl`. Nos preguntará sobre el tipo de instalación, con lo que seleccionaremos el proceso avanzado (opción 3).

Se preguntarán credenciales de administración, junto con la confirmación del nombre del máquina, etc. a lo que deberemos agregar la información correspondiente. El punto importante es definir que el sufijo raíz es `ou=inf,o=utfsm,c=cl` debido a que éste es el sufijo que se encuentra en el servidor LDAP de producción.

Para ingresar a la consola de administración, es necesario el siguiente comando a partir de un cliente autorizado:

```
ssh -X root@192.168.122.97 389-console -a http://192.168.122.97:9830
```

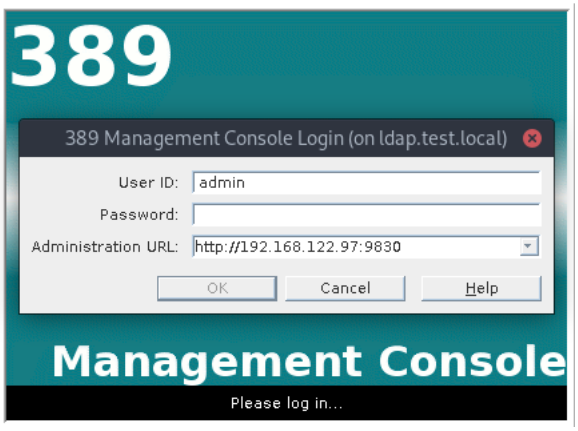


Figura 17: Pantalla de inicio de 389ds.
Fuente: Elaboración propia.

Nota: recordar activar el servicio de LDAP para en caso de apagar la máquina virtual, se pueda volver a usar el servicio sin problemas.

```
systemctl enable dirsrv.target --now
```

```
systemctl enable dirsrv-admin --now
```

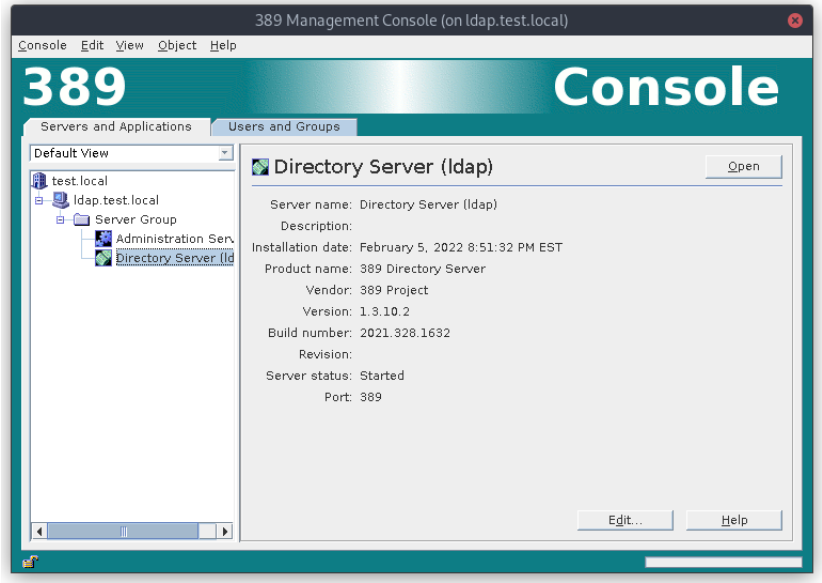


Figura 18: Consola de 389ds luego de ser autenticado.
Fuente: Elaboración propia.

Entrar a Directory Server (con doble clic), y en el tab Directory, y en la opción inf, clic derecho, seleccionar New y finalmente seleccionar Group (en este caso alumnos).

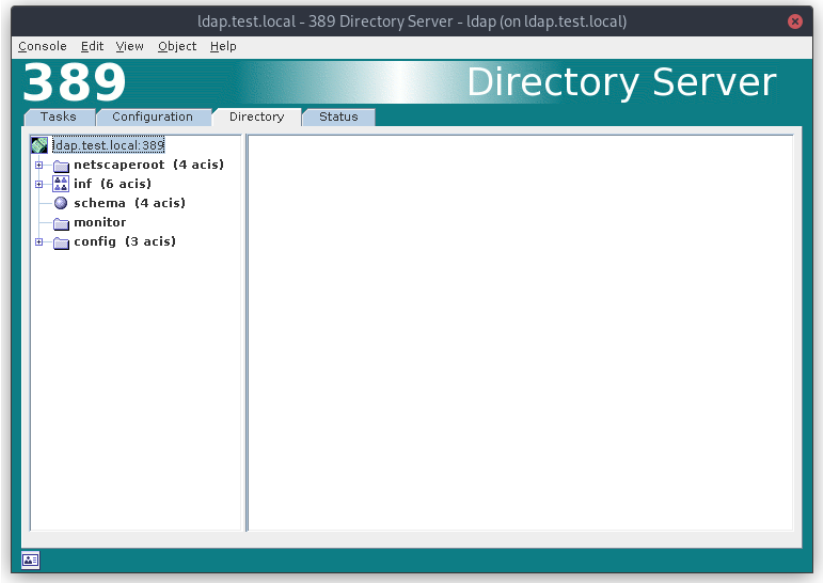


Figura 19: Directorios disponibles en servidor.
Fuente: Elaboración propia.

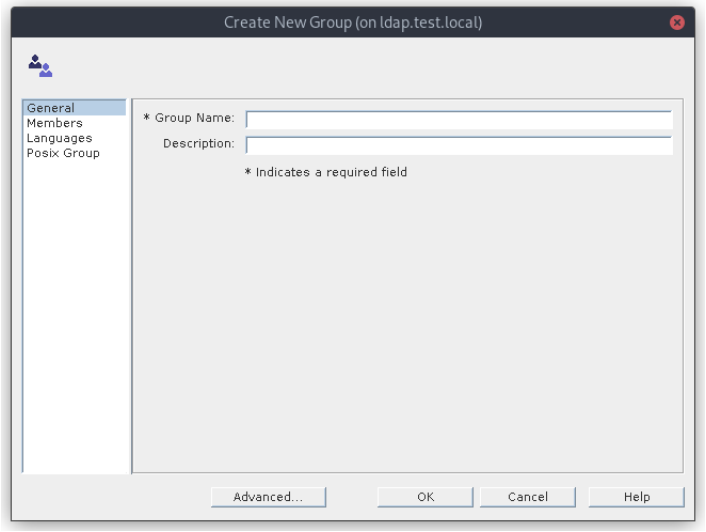


Figura 20: Pantalla para crear grupo.
Fuente: Elaboración propia.

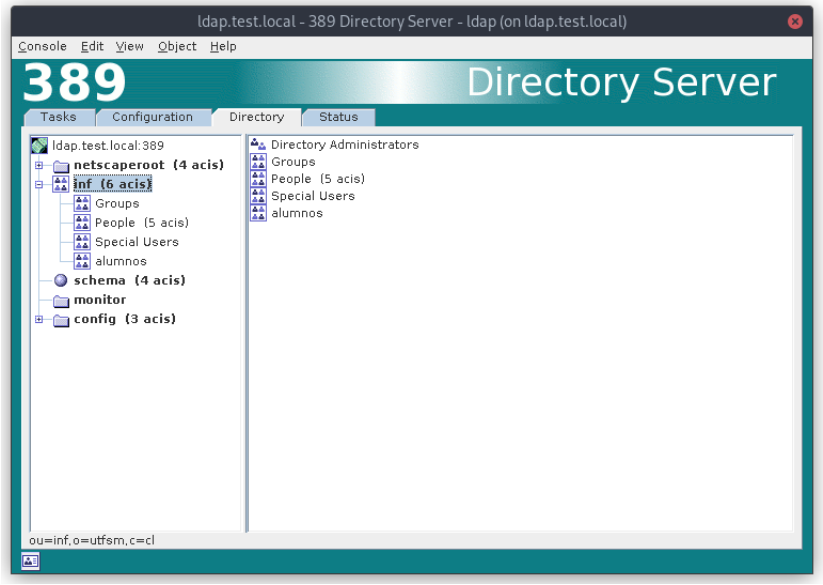


Figura 21: Detalle mostrando grupo recién creado.
Fuente: Elaboración propia.

Luego agregar un nuevo usuario de la misma forma que agregando un grupo solo que en vez de seleccionar Group, seleccionar User.

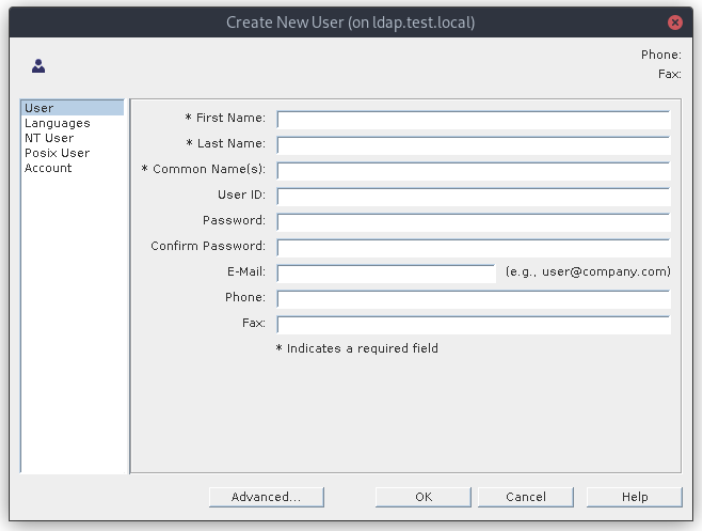


Figura 22: Pantalla para crear usuario
Fuente: Elaboración propia.

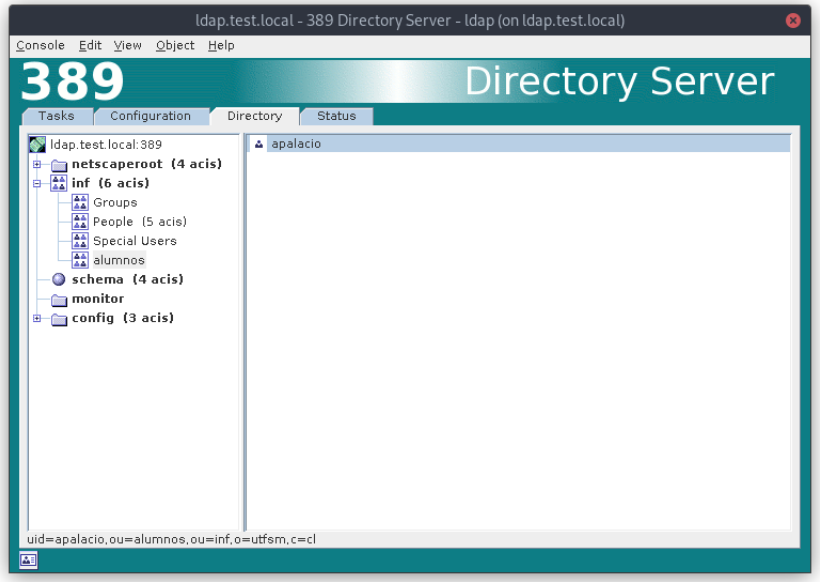


Figura 23: Detalle mostrando usuario recién creado.
Fuente: Elaboración propia.

A.2. Instalación de FreeIPA

La VM que se utilizará para este servidor es la siguiente:

- Sistema Operativo: Fedora 35
- RAM: 2 GB
- vCPUs: 1
- IP: `192.168.100.33`
- Nombre de máquina: `ipa.test.local`
- Paquetes por instalar (`yum install -y`):
  - epel-release
  - freeipa-server-dns
  - freeipa-server
- Versión FreeIPA: 4.9.8

Configurar `/etc/hosts` con la dirección IP y nombre de máquina correspondiente.

Se necesita ejecutar el script `ipa-server-install` para la configuración. En esta se nos pedirá por credenciales de administración y agregar las configuraciones que son necesarias de acuerdo a las necesidades.

Se nos entregará la información necesaria para entrar dentro del panel web y lo necesario para utilizar la CLI.

Configurar el firewall con los puertos correspondientes:

Servicio	Puerto	Protocolo
HTTP	80	TCP
HTTPS	443	TCP
LDAP	389	TCP
LDAPS	636	TCP
Kerberos	88, 464	TCP/UDP
DNS	53	TCP/UDP
NTP	123	UDP
Dogtag Certificate System (OCSP responder)	9180	TCP
Dogtag (agents)	9443	TCP
Dogtag (users, SSL)	9444	TCP
Dogtag (administrators)	9445	TCP
Dogtag (users, client authentication)	9446	TCP
Dogtag (Tomcat)	9701	TCP
Dogtag (internal LDAP database)	7389	TCP

Tabla 4: Puertos para firewall correspondientes a FreeIPA

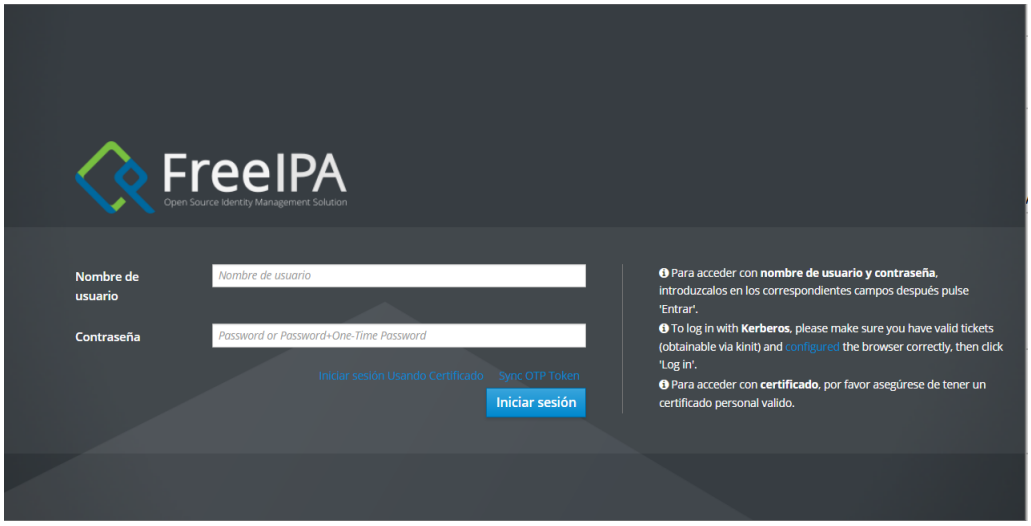


Figura 24: Portal de inicio de FreeIPA.
Fuente: Elaboración propia.

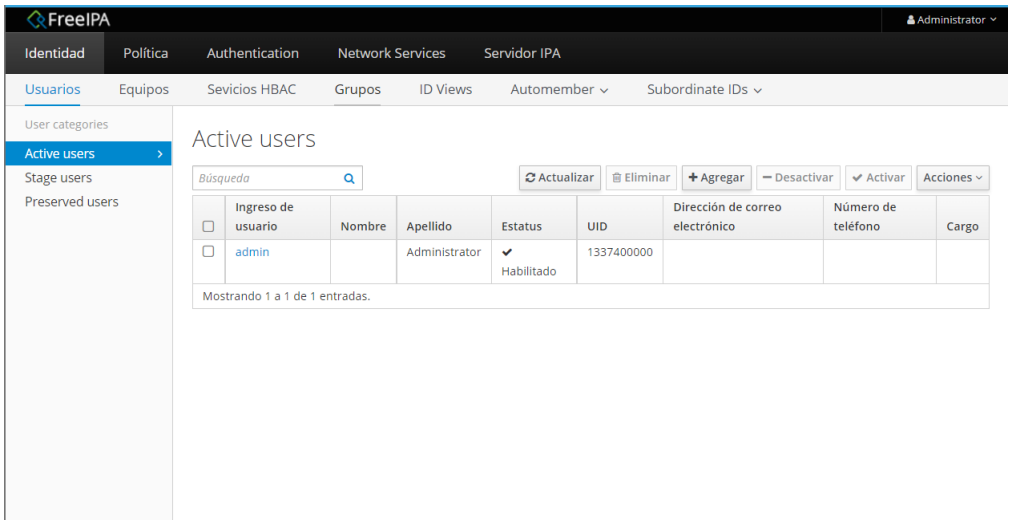


Figura 25: FreeIPA al ingresar con administrador, mostrando la lista de usuarios.
Fuente: Elaboración propia.

A.3. Instalación de Samba

La VM que se utilizará para este servidor es la siguiente:

- Sistema Operativo: Fedora 35 Server
- RAM: 512 MB
- vCPUs: 1
- IP: `192.168.100.34`
- Nombre de máquina: `smb.test.example`
- Paquetes por instalar (`yum install -y`):
  - samba
  - samba-dc
  - samba-dc-bind-dlz
- Versión Samba: 4.15.5

Actualizar `/etc/hosts` con registro correspondiente.

Para actualizar el nombre de la máquina, utilizar el siguiente comando:

```
hostnamectl set-hostname smb.test.example
```

Configurar red con dirección DNS primaria al `127.0.0.1`, y dirección DNS secundaria a la que normalmente se utilizaría. También en caso de que el sistema tenga un sistema de DNS automático (como NetworkManager) desactivarlo.

Mover archivo `smb.conf` en caso de existir a otro lugar para que no interfiera en la instalación.

```
mv /etc/samba/smb.conf /etc/samba/smb.conf.bak
```

Ejecutar el comando de instalación, usando la opción interactiva.

```
samba-tool domain provision --use-rfc2307 --interactive
```

Elegir *realm* y *domain* (AD), seleccionar valores por omisión en *server role* y *backend*, y seleccionar *DNS forwarder* de acuerdo a lo que se necesite. En nuestro caso se usó el *realm* `AD.TEST.EXAMPLE`, el dominio sería `AD`, el *server role* `dc` y el *backend DNS* seria `BIND_DLZ`.

Copiar la configuración de Kerberos a la ruta indicada:

```
cp /var/lib/samba/private/krb5.conf /etc/
```

Copiar la configuración de DNS incluida dentro de la carpeta de Samba, y configurar lo necesario. Recordar que no es necesario añadir zonas que correspondan al dominio del AD ya que esto ya se encarga Samba de agregarlos dinámicamente.

Agregar los puertos de Samba en el firewall, para permitir la comunicación.

Servicio	Puerto	Protocolo
DNS	53	TCP/UDP
Kerberos	88	TCP/UDP
NTP	123	UDP
End Point Mapper (DCE/RPC Locator Service)	135	TCP
NetBIOS Name Service	137	UDP
NetBIOS Datagram	138	UDP
NetBIOS Session	139	TCP
LDAP	389	TCP/UDP
SMB over TCP	445	TCP
Kerberos kpasswd	464	TCP/UDP
LDAPS	636	TCP
Global Catalog	3268	TCP
Global Catalog SSL	3269	TCP
Dynamic RPC Ports	49152-65535	TCP

Tabla 5: Puertos para el firewall correspondientes a Samba

Iniciar y habilitar el servicio de samba y named (BIND):

```
systemctl enable --now samba named
```

A.4. Migración de LDAP a FreeIPA

Para migrar 389ds a FreeIPA se necesitan un par de pasos adicionales antes de utilizar los scripts de ayuda.

- Verificar que no haya más cambios a la base de datos.
- Migrar schema de LDAP a FreeIPA.

El segundo paso es extremadamente importante, debido a que al importar los datos IPA necesita saber que datos recoger dentro de los definidos del usuario para el nuevo LDAP dentro de IPA. Al intentar importar sin haber expandido el schema, el script de migración fallará.

Comando para obtener schema de 389ds.

```
ldapsearch -h <Host> -b 'cn=schema' -s base -D <Directory Manager> -W
'(objectclass=*)' attributeTypes objectClasses
```

```
objectClasses: ( 1.3.6.1.4.1.21532.19.1.0.1 NAME 'sansano' DESC '' SUP
    top STRUCTURAL MAY ( acceso $ accountExpirationDate $ anotacion $
    buildingName $ carreer $ foto $ homepage $ info $ mail $
    mailAlternateAddress $ memberOfList $ promocion $ rol $ rut ) X-ORIGIN
     'user defined' )
attributeTypes: ( 1.3.6.1.4.1.21532.19.1.1.0 NAME 'rut' DESC 'El rol
    unico nacional' SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE X-
    ORIGIN 'user defined' )
attributeTypes: ( 1.3.6.1.4.1.21532.19.1.1.2 NAME 'rol' DESC 'El rol
    Ueseeme' EQUALITY caseIgnoreMatch SUBSTR caseIgnoreSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE X-ORIGIN 'user
    defined' )
```

Listado 2: Ejemplo de atributos del schema actual

```
dn: cn=schema
changetype: modify
add: attributeTypes
attributeTypes: ( 1.3.6.1.4.1.21532.19.1.1.0 NAME 'rut' DESC 'El rol
    unico nacional' SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE X-
    ORIGIN 'user defined' )
-
add: attributeTypes
attributeTypes: ( 1.3.6.1.4.1.21532.19.1.1.2 NAME 'rol' DESC 'El rol
    Ueseeme' EQUALITY caseIgnoreMatch SUBSTR caseIgnoreSubstringsMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15 SINGLE-VALUE X-ORIGIN 'user
    defined' )
-
add: objectclasses
objectClasses: ( 1.3.6.1.4.1.21532.19.1.0.1 NAME 'sansano' DESC '' SUP
    top STRUCTURAL MAY ( rol $ rut ) X-ORIGIN 'user defined' )
```

Listado 3: Ejemplo de archivo LDIF

Preparar archivo LDIF con los datos y modificar el schema en el host de IPA:

```
ldapmodify -D <Directory Manager> -x -W -f <file>
```

Continuar con la migración, entregando las credenciales de administrador:

```
kinit admin
```

Junto a esto modificar el servidor para que esté en modo migración:

```
ipa config-mod --enable-migration=TRUE
```

Migrar la base de datos LDAP, de acuerdo a los parámetros necesitados. En este ejemplo, importamos los de la rama alumnos/valparaiso.

```
ipa migrate-ds --with-compat \
--user-container=ou=cuentas,ou=valparaiso,ou=alumnos \
--group-container=ou=grupos,ou=valparaiso,ou=alumnos
```

Se entregará un mensaje como el siguiente:

"Passwords have been migrated in pre-hashed format. IPA is unable to generate Kerberos keys unless provided with clear text passwords. All migrated users need to login at https://ipa.test.local/ipa/migration/ before they can use their Kerberos accounts."

Para actualizar los datos de los usuarios y que puedan entrar con Kerberos, estos tienen que pasar por un paso adicional que les obliga a entrar por un portal para obtener la contraseña en vez del hash y poder ingresarlas correctamente al sistema Kerberos. Esto tiene 2 opciones, por medio de la página web que entrega el mensaje, o por medio de ingresar por línea de comandos a alguno de los dispositivos clientes de IPA. El portal intenta autenticar con los hashes obtenidos y no es posible ingresar cualquier contraseña. Se hace este paso para lograr la migración de hashes de LDAP a Kerberos.

A.5. Configuración cliente IPA

Para comenzar a configurar los clientes con FreeIPA, es necesario conocer el sistema al que le vamos a instalar. Generalmente es fácilmente compatible con sistemas Fedora y CentOS/RHEL, sin embargo, también es posible instalarlo con sistemas Ubuntu y OpenSUSE. De ahora en adelante asumiremos la instalación en un terminal con el sistema operativo Fedora 35.

Primero debemos asegurarnos que el cliente esté dentro de la red de FreeIPA y que tenga acceso a los servicios de FreeIPA. También antes de comenzar el proceso debemos asegurarnos de tener un usuario con suficientes permisos para enrolar clientes en FreeIPA. Todo esto se asume que se hace con el usuario root o administrador del terminal.

Debemos configurar el dominio y asegurarnos que el DNS apunte correctamente al que contenga los registros de FreeIPA:

```
nmcli con mod enp0s3 ipv4.ignore-auto-dns yes

nmcli con mod enp0s3 ipv4.dns "192.168.100.33 192.168.100.1"

ping ipa.test.local

hostnamectl set-hostname term1.test.local
```

Lo siguiente es instalar el cliente de FreeIPA:

```
yum install -y freeipa-client
```

Luego ejecutaremos el instalador del cliente:

```
ipa-client-install
```

Si todo está configurado correctamente, reconocerá automáticamente el dominio y solo preguntará si se desea cambiar los servidores NTP, junto con preguntar el usuario administrador capaz de registrar clientes FreeIPA. Con esto ya tendremos un cliente capaz de consultar LDAP y Kerberos junto con todos los servicios necesarios. En caso de requerir AutoFS, el script `ipa-client-automount` configura automáticamente toda la información del servidor necesaria en el cliente.

A.6. Configuración sincronización FreeIPA y AD

Para lograr una parte importante de nuestro plan, es lograr que FreeIPA pueda comunicarse con Samba AD para sincronizar los usuarios con sus contraseñas respectivas. A continuación, se mostrarán los pasos necesarios para lograr esta compatibilidad.

La siguiente configuración está basada en la documentación de Red Hat (Red Hat Inc., s.f.-a).

Es necesario primero configurar el servidor AD en el DNS de FreeIPA debido a que todas las llamadas se harán en base al FQDN. También es necesario configurar los certificados de FreeIPA y Samba en el servidor de FreeIPA por lo que en este caso lo que haremos es obtener el certificado de Samba, alojado en `/var/lib/samba/private/tls/ca.pem` y copiarlo en el directorio `/etc/openldap/certs`.

Para el certificado de FreeIPA solo basta visitar el sitio `https://ipa.test.local/ipa/config/ca.crt`, copiar el contenido de la página y luego dejar el certificado en la misma carpeta anterior.

Para finalizar la configuración de certificados, es necesario usar el siguiente comando:

```
openssl rehash /etc/openldap/certs
```

Y finalmente agregar dentro de la configuración de OpenLDAP (`/etc/openldap/ldap.conf`) y comentando la línea de `TLS_CACERT`.:

```
...
TLS_CACERTDIR /etc/openldap/certs
#TLS_CACERT /etc/ipa/ca.crt
...
```

Listado 4: Configuración OpenLDAP

Es posible sincronizar usuarios desde FreeIPA con Active Directory, por medio de los plugins

de compatibilidad que tiene el servidor 389ds contenido en FreeIPA. A continuación, agregaremos la configuración correspondiente:

```
dsconf -D "cn=Directory Manager" ldap://127.0.0.1 replication enable --
    suffix="dc=test,dc=local" --role="supplier" --replica-id=1
```

Listado 5: Habilitación del servicio de replicación en FreeIPA.

```
dsconf -D "cn=Directory Manager" ldap://127.0.0.1 repl-winsync-agmt
    create --suffix="dc=test,dc=local" --host="smb.ad.test.example" --port
    =636 --conn-protocol="LDAPS" --bind-dn="cn=Administrator,cn=Users,dc=
    ad,dc=test,dc=example" --bind-passwd="hola1234!" --win-subtree="cn=
    Users,dc=ad,dc=test,dc=example" --ds-subtree="cn=users,cn=accounts,dc=
    test,dc=local" --win-domain="ad.test.example" --init a1 --sync-users="
    on" --one-way-sync="toWindows"
```

Listado 6: Comando de configuración del plugin de sincronización.

Con el comando anterior habilitamos la sincronización en un sentido de FreeIPA hacia un servidor Active Directory. Esto lo hacemos ya que no queremos la información de Samba AD que sea sincronizada hacia FreeIPA. También el comando debe tener la configuración relacionada a donde están los usuarios de FreeIPA y donde se van a alojar (subtree). Por último, el parámetro `--init` le pide al plugin que comience de inmediato.

En caso de necesitar borrar la configuración, solo basta el comando:

```
dsconf -D "cn=Directory Manager" ldap://127.0.0.1 repl-winsync-agmt
    delete a1 --suffix="dc=test,dc=local"
```

Listado 7: Comando para eliminar la replicación

Un punto muy importante que describe la documentación:

“Una cuenta de usuario del servidor es sincronizada con Active Directory a través de atributos específicos que estén presentes dentro de la información. Toda entrada dentro del servidor debe tener el atributo `ntUserCreateNewAccount` junto con el *objectClass* `ntUser`. Estos atributos avisan al plugin de sincronización que se debe escribir la información del usuario en Active Directory.”

Para lo siguiente, modificaremos las entradas de un usuario ya añadido en FreeIPA con los atributos correspondientes necesarios para ser sincronizados:

```
dn: uid=test,cn=users,cn=accounts,dc=test,dc=local
changetype: modify
add: objectClass
objectClass: ntUser
-
add: ntUserCreateNewAccount
ntUserCreateNewAccount: true
-
add: ntUserDomainId
```

```
ntUserDomainId: test
-
add: ntUserDeleteAccount
ntUserDeleteAccount: true
```

Listado 8: Configuración de LDAP de un usuario para que sea capaz de ser sincronizado a AD.

Advertencia: La configuración de la sincronización debe hacerse antes de agregar los usuarios.

A.7. Configuración cross-forest trust FreeIPA/Windows Server AD

Para la siguiente configuración asumiremos un servidor Windows Server 2012 con las configuraciones de Domain Controller para Active Directory. Después de configurar el AD correctamente en Windows Server, es necesario agregar un registro de DNS forward en ambos, apuntando hacia el otro servidor.

En este caso, se necesita un registro de DNS forward desde Windows Server hacia FreeIPA con la zona y dirección IP, y en FreeIPA se necesita un registro de DNS forward hacia Windows Server con la dirección IP correspondiente.

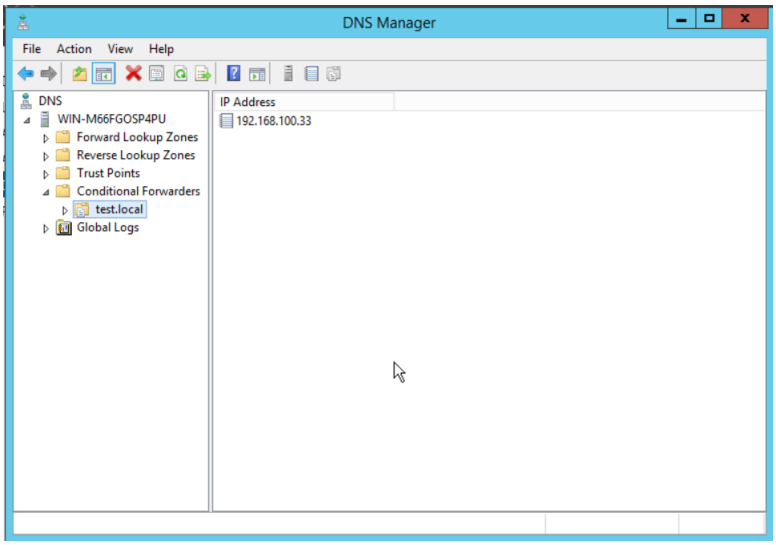


Figura 26: Configuración del registro DNS forward en Windows Server.
Fuente: Elaboración propia.

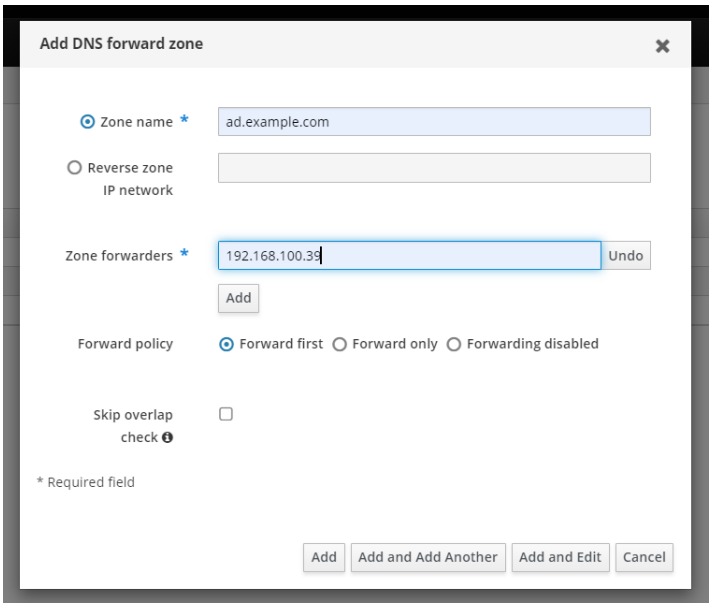


Figura 27: Configuración del registro DNS forward en FreeIPA.
Fuente: Elaboración propia.

Por último, podemos comenzar a agregar el cross-forest trust.

Dentro del portal web de FreeIPA →IPA server →Trusts →Trusts, agregar la configuración correspondiente del AD.

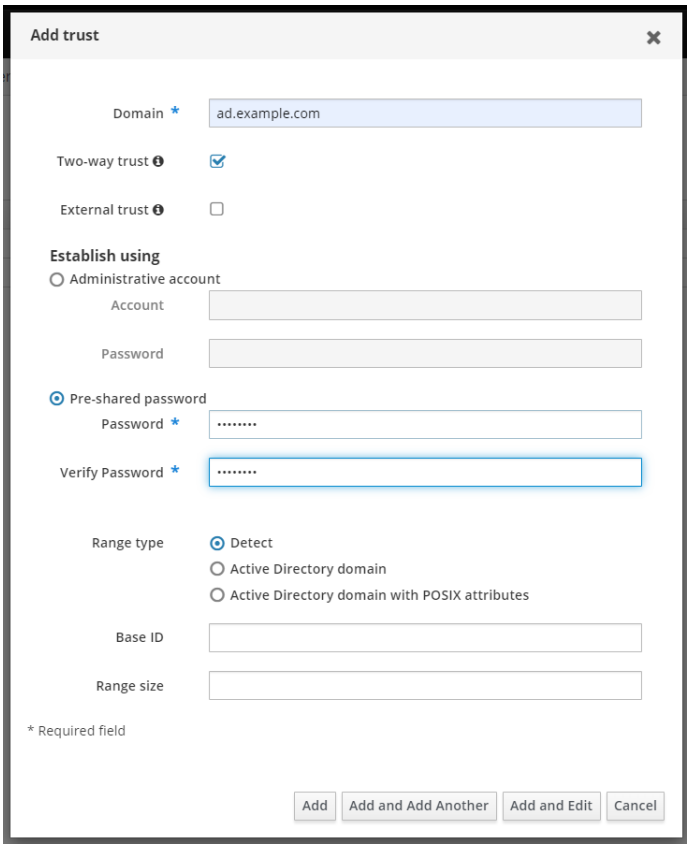


Figura 28: Configuración trust IPA AD
Fuente: Elaboración propia.

Finalmente, se puede configurar el trust dentro de Windows Server.

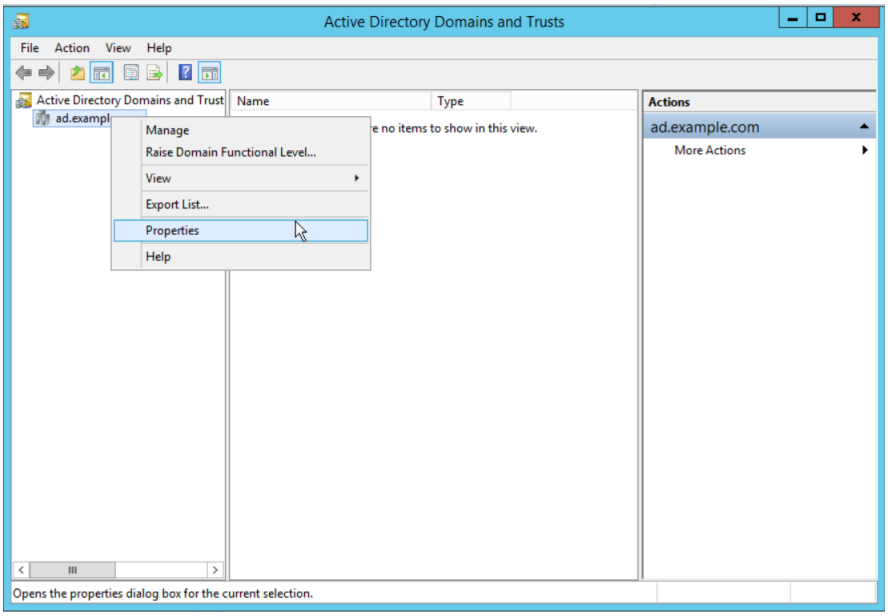


Figura 29: Aplicación de configuración de trusts en Windows Server.
Fuente: Elaboración propia.

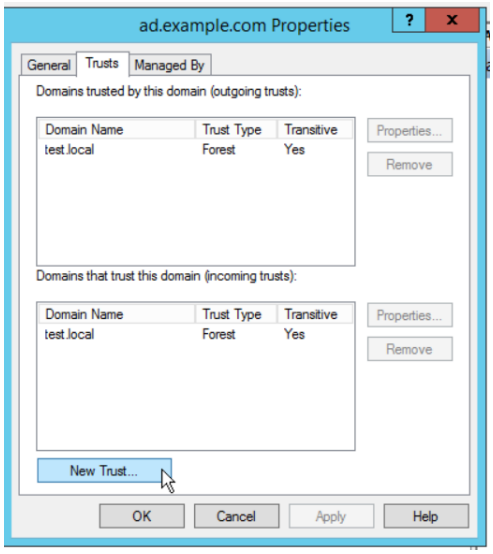


Figura 30: Ventana de configuración de trusts.
Fuente: Elaboración propia.

Se pueden agregar nuevos trusts en la opción “New Trust...”.

Si configurado correctamente, los clientes Windows ahora serán capaces de soportar usuarios de FreeIPA, lo cual tiene un potencial gigante para compatibilidad.

A.8. Configuración cross-forest trust FreeIPA/Samba AD (configuración opcional para futuro)

Lo primero para confirmar que funcione correctamente el script de configuración de trusts de FreeIPA, tenemos que verificar que los DNS de ambos sistemas estén configurados correctamente. Es decir, el DNS del sistema AD debe tener una zona configurada como forward, dirigiendo a la dirección IP de FreeIPA y viceversa.

Se puede verificar con el comando `dig`:

```
dig ipa.test.local @192.168.100.34
```

```
dig ad.test.example @192.168.100.33
```

Verificar que tenemos instalado el paquete de freeipa-ad-trust:

```
dnf install -y ipa-server-trust-ad
```

Ahora se debe ejecutar el script de adtrust:

```
ipa-adtrust-install
```

El script de configuración preguntará por opciones de compatibilidad, los cuales los habilitaremos dependiendo del caso.

Además, se pedirá que se habiliten ciertos puertos en el firewall para la comunicación, los cuales son los siguientes:

Servicio	Puerto	Protocolo
epmap	135	TCP
netbios-dgm	138	TCP/UDP
netbios-ssn	139	TCP/UDP
(C)LDAP	389	UDP
microsoft-ds	445	TCP/UDP
epmap listener range	1024-1300	TCP
msft-gc	3268	TCP

Tabla 6: Lista de puertos para firewall para compatibilidad con trusts

Luego dentro del panel web, es posible configurar el trust en la sección IPA Server →Trusts →Trusts.

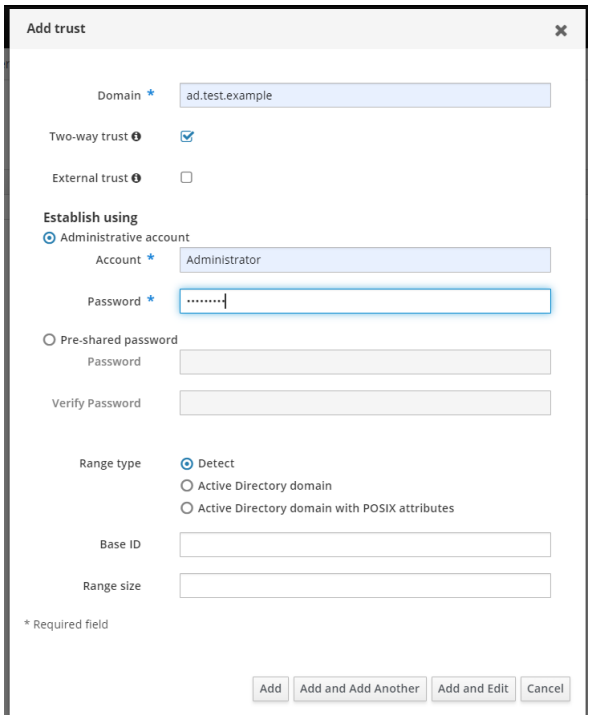


Figura 31: Configuración del trust
Fuente: Elaboración propia.

Para confirmar que el trust esté configurado correctamente, podemos consultar la lista de trusts de Samba AD:

```
[root@smb ~]# samba-tool domain trust list
Type[Forest]    Transitive[Yes] Direction[BOTH]      Name[test.local]
```

Listado 9: Resultado de comando para listar trusts en Samba.

Es importante que el trust sea en 2 direcciones y no externo (transitivo) ya que confía plenamente en el otro servidor.

En la imagen, podemos ver que se puede configurar correctamente un servidor Samba AD como trust, sin embargo, a la hora de intentar entrar con un terminal Windows configurado con el AD correspondiente, no permite entrar debido a “falta de recursos”.

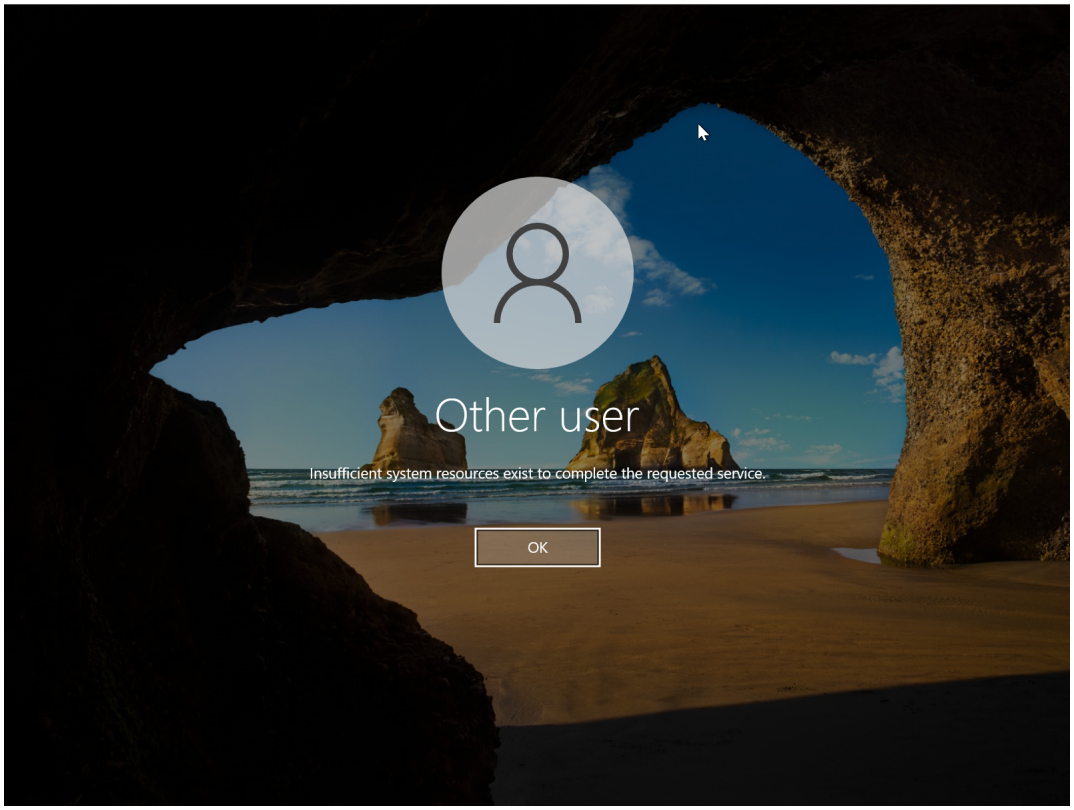


Figura 32: Resultado de intento de login con credenciales de FreeIPA.

Lo interesante es que el usuario al intentar ingresar con credenciales erróneas no permite el acceso y entrega un error de usuario/clave incorrecta lo que indica que si hace la autenticación.

Este error se puede concluir como que FreeIPA no está mapeando correctamente el usuario o Samba no suple la falta de recursos del usuario de FreeIPA, por lo que quedaría como un cross-forest trust no efectivo lo que finalmente no es lo que necesitamos.

Por esto se deben recurrir a otras alternativas o esperar que en el futuro esta característica sea la mejor opción para agregar trusts entre distintos servidores.

A.9. Configuración RH SSO/Keycloak con FreeIPA

Asumiremos Fedora 35 el sistema operativo de la máquina con servidor de Keycloak. Además, asumiremos la versión de Keycloak como 17.0.0.

Seguiremos los pasos de Subsección A.5 en la máquina de Keycloak.

Para que Keycloak sea compatible con los servicios de Kerberos con HTTP, es necesario crear

la entrada DNS en FreeIPA y un nuevo keytab con el host de la máquina:

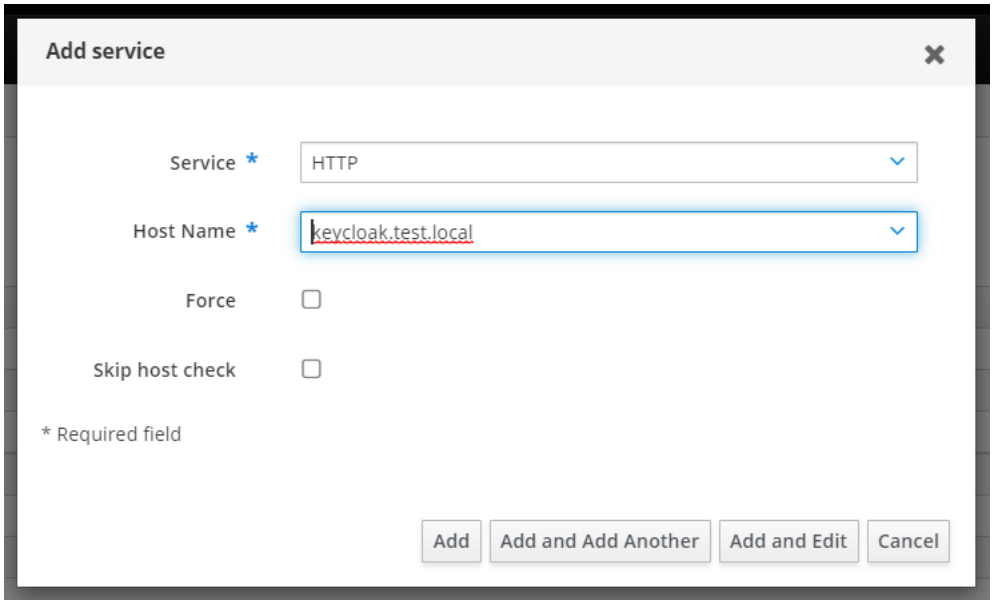


Figura 33: Agregar servicio de HTTP en FreeIPA para Keycloak

Luego para obtener el keytab es necesario ejecutar el siguiente comando:

```
ipa-getkeytab -p HTTP/keycloak.test.local@TEST.LOCAL -k /root/http.keytab
```

Listado 10: Comando para obtener keytab.

Junto a esto habremos configurado correctamente el entorno para aceptar Kerberos en Keycloak.

A continuación, deberemos configurar el servidor Keycloak con los datos correspondientes:

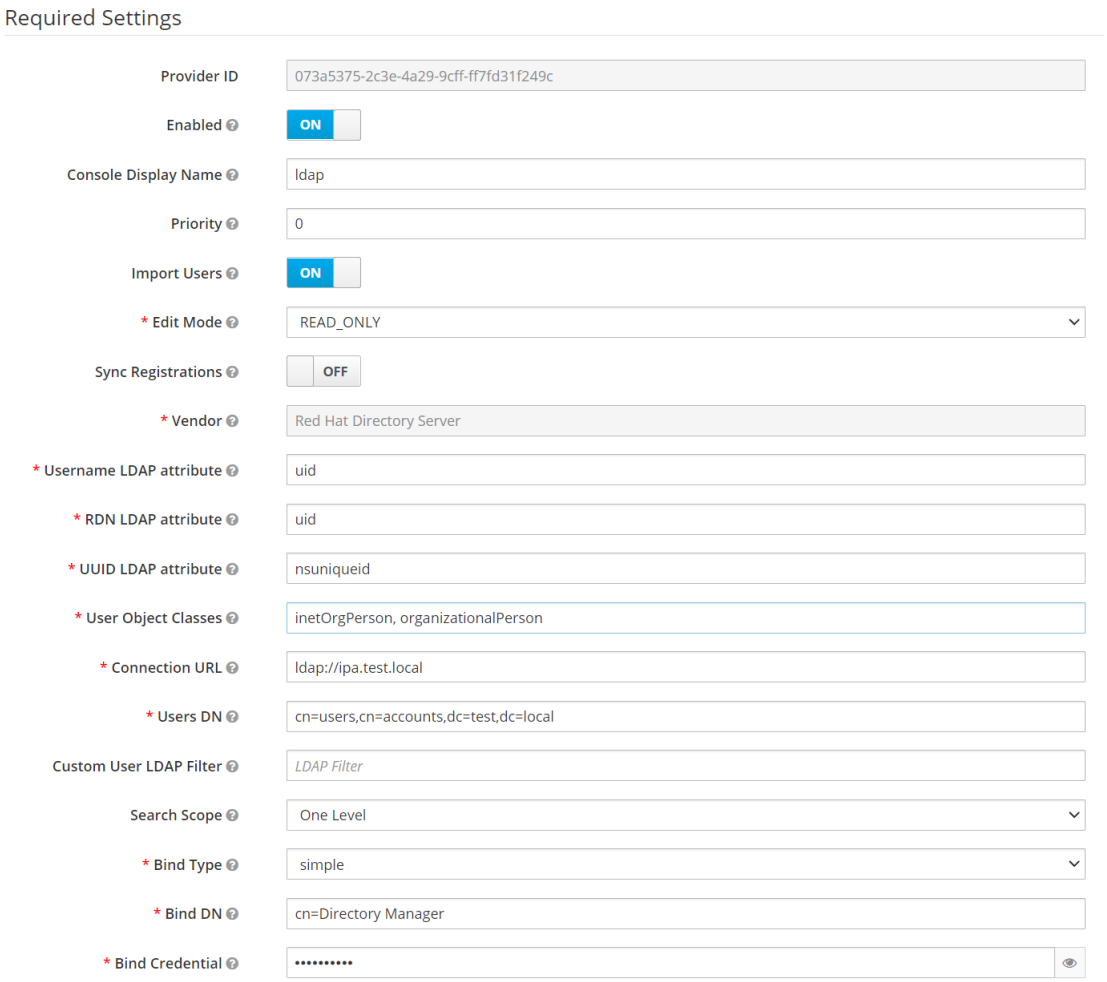


Figura 34: Configuración Keycloak LDAP

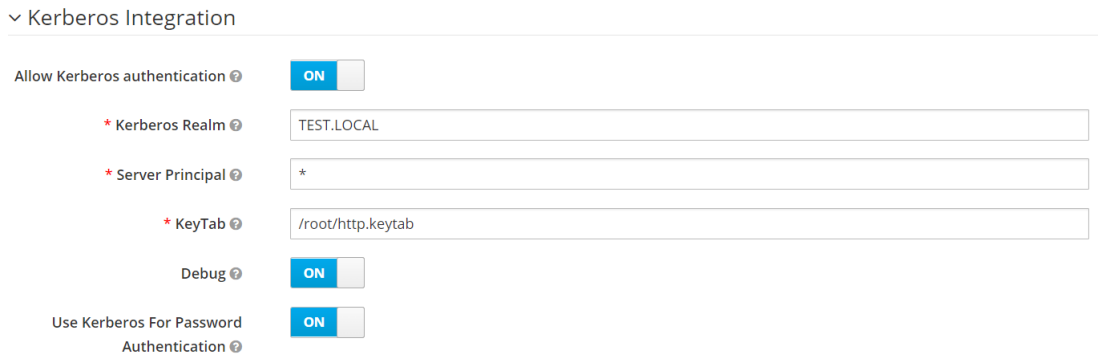


Figura 35: Configuración Keycloak Kerberos

Con esto lograremos compatibilidad de Keycloak con FreeIPA y los usuarios solo tendrán que

utilizar sus credenciales al entrar a la terminal. Agregar que existen configuraciones de los navegadores adicionales para que esto sea compatible.

Configuración Firefox:

- Entrar a Firefox.
- Ingresar about:config en la barra de direcciones.
- En la barra de búsqueda ingresar “navigation”.
- Modificar los atributos `network.negotiate-auth.trusted-uris` y `network.negotiate-auth.delegation-uris` e ingresar el dominio correspondiente.

Referencias

Carter, G., Ts, J. & Eckstein, R. (2007). *Using Samba: A File & Print Server for Linux, Unix & Mac OS X*. O'Reilly Media, Inc.

Hughes, J. & Maler, E. (2005). Security assertion markup language (saml) v2. 0 technical overview. *OASIS SSTC Working Draft sstc-saml-tech-overview-2.0-draft-08*, *13*.

Linn, J. (2000). Generic Security Service Application Program Interface Version 2 Update 1. RFC 2743 (Proposed Standard).

Melnikov, A. & Zeilenga, K. (2006). *Simple authentication and security layer (SASL)* (inf. téc.). RFC 4422, June.

Microsoft. (s.f.). *Windows Server: Identity and Access*. Consultado el 8 de marzo de 2022, desde https://docs.microsoft.com/en-us/windows-server/identity/identity-and-access

Neuman, C., Yu, T., Hartman, S., Raeburn, K. & RFC4120, I. (2005). The Kerberos Network Authentication Service (V5), July 2005, 1-119. ftp://ftp.isi.edu/in-notes/rfc1510.txt

R. Harrison, E. (2006). Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms. *RFC 4513*, *June*.

Red Hat Inc. (s.f.-a). *Red Hat Directory Server 11: Administration Guide*. Consultado el 22 de marzo de 2022, desde https://access.redhat.com/documentation/en-us/red_hat_directory_server/11/html/administration_guide/windows_sync

Red Hat Inc. (s.f.-b). *Red Hat Enterprise Linux 7*, *System-Level Authentication Guide*. Consultado el 1 de marzo de 2022, desde https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/system-level_authentication_guide/ldap_servers

Red Hat Inc. (s.f.-c). *What is FreeIPA?* Consultado el 8 de marzo de 2022, desde https://www.freeipa.org/page/About

Rigney, C., Willens, S., Rubens, A. & Simpson, W. (2000). Remote authentication dial in user service (RADIUS).

Sakimura, N., Bradley, J., Jones, M., De Medeiros, B. & Mortimore, C. (2014). Openid connect core 1.0. *The OpenID Foundation*, S3.

Sermersheim, J. (2006). Rfc 4511: Lightweight directory access protocol (ldap): The protocol. *Internet Engineering Task Force (IETF)*.

Tranquil IT. (2021). *Samba-AD documentation*. Consultado el 13 de marzo de 2022, desde https://samba.tranquil.it/doc/en/

University of Michigan. (1996). *SLAPD and SLURPD Administrators Guide*. Consultado el 23 de febrero de 2022, desde http://websites.umich.edu/~dirsvcs/ldap/doc/guides/slapd/toc.html