UNIVERSIDAD NACIONAL DE LOJA

Facultad de Energía, las Industrias y los Recursos Naturales No Renovables

Carrera de Ingeniería en Sistemas

"DESARROLLO DE UN PROTOTIPO PARA EL SERVICIO DE AUTENTICACIÓN CENTRAL DE USUARIOS EN APLICACIONES WEB."

TESIS DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS

Autores:

Aguilar Soto Wilmer Antonio
Armijos Ordóñez Manuel Stalin

Director:

Ing. Coronel Romero Edison Leonardo, Mg.Sc

LOJA - ECUADOR
2019

Certificación

Ing. Edison Leonardo Coronel Romero, Mg. Sc.

DOCENTE DE LA CARRERA DE INGENIERÍA EN SISTEMAS DE LA UNIVERSIDAD NACIONAL DE LOJA, DIRECTOR DE TRABAJO DE TESIS

CERTIFICA:

Que los egresados **Wilmer Antonio Aguilar Soto** y **Manuel Stalin Armijos Ordóñez**, realizaron el trabajo de tesis denominado "**Desarrollo de un Prototipo Para el Servicio de Autenticación Central de Usuarios en Aplicaciones Web**" bajo mi dirección y asesoramiento, mismo que fue revisado, enmendado y corregido minuciosamente. En virtud que el trabajo de tesis reúne, a satisfacción las cualidades de fondo y forma exigidas para un trabajo de este nivel, autorizo su presentación, sustentación y defensa ante el tribunal respectivo.

Loja, 31 de mayo del 2019

Ing. Edison Leonardo Coronel Romero, Mg. Sc.
DIRECTOR DEL TRABAJO DE TESIS

Autoría

Yo, **Wilmer Antonio Aguilar Soto** y **Manuel Stalin Armijos Ordóñez**, declaramos ser los autores del presente trabajo de tesis y eximimos expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales por el contenido de esta.

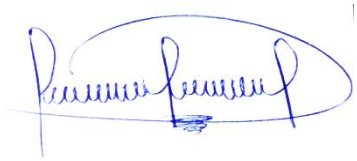
Adicionalmente aceptamos y autorizamos a la Universidad Nacional de Loja, la publicación del Trabajo de Tesis en el Repositorio Institucional – Biblioteca Virtual.

Loja, 31 de mayo del 2019

Firma: 

Autor: Wilmer Antonio Aguilar Soto

Cédula: 1900481878

Firma: 

Autor: Manuel Stalin Armijos Ordóñez

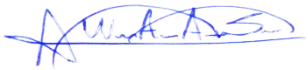
Cédula: 1105593238

CARTA DE AUTORIZACIÓN DE TESIS POR PARTE DE LOS AUTORES, PARA LA CONSULTA, REPRODUCCIÓN PARCIAL O TOTAL Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO.

Yo, **Wilmer Antonio Aguilar Soto** y **Manuel Stalin Armijos Ordóñez**, declaramos ser los autores del trabajo de tesis: "**Desarrollo de un Prototipo para el Servicio de Autenticación Central de Usuarios en Aplicaciones Web**", como requisito para optar al grado de: **INGENIERO EN SISTEMAS**; autorizamos al Sistema Bibliotecario de la Universidad Nacional de Loja para con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el repositorio Digital Institucional:

Los usuarios pueden consultar el contenido de este trabajo en el RDI, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad. La Universidad Nacional de Loja, no se responsabiliza por plagio o copia del trabajo de titulación que realice un tercero.

Para constancia de esta autorización, en la ciudad de Loja, el día veintisiete del mes de junio del dos mil diecinueve.

Firma:  .

Autor: Wilmer Antonio Aguilar Soto

Cédula: 1900481878

Dirección: Loja (Punzara, Calle España y Paraguay)

Correo Electrónico: waaguilars@unl.edu.ec

Celular: 0959733089

Firma: 

Autor: Manuel Stalin Armijos Ordóñez

Cédula: 1105593238

Dirección: Loja (Carigan, Av. Interbarrial y 5ta San Luis)

Correo Electrónico: manuel.s.armijos@unl.edu.ec

Celular: 0991189175

DATOS COMPLEMENTARIOS

Director de Trabajo de Titulación: Ing. Edison Leonardo Coronel Romero, Mg. Sc.

Tribunal de Grado: Ing. Hernán Leonardo Torres Carrión Mg. Sc.
Ing. José Oswaldo Guamán Quinche Mg. Sc.
Ing. Oscar Miguel Cumbicos Pineda Mg. Sc.

Dedicatoría

Antonio Aguilar

El presente trabajo de titulación lo dedico con infinito amor, afecto y humildad, primeramente, a mi madre que con su cariño y bondad supo inculcarme buenos valores y guiarme por el camino de la sabiduría, la educación y enseñarme lo bueno de la vida, a mi padre que supo cuidarme y protegerme de los males, a mi abuelita que para mí es mi segunda madre, que con todo el amor del mundo supo cuidarme. También quiero dedicar este trabajo, a mis hermanos (as) y a toda mi familia que con su apoyo moral e incondicional hicieron posible el poder alcanzar tan prestigiada meta de mi vida.

Manuel Armijos

El presente trabajo de titulación lo dedico con mucho amor y cariño primeramente a mi madre ANA que me inculco buenos valores y me guió por el buen camino de la sabiduría, educación y enseñarme lo valioso de la vida, a mi padre biológico MANUEL quién supo cuidarme y protegerme de las adversidades de la vida, a la pareja de mi madre WILLAN quien lo considero mi segundo padre quien supo apoyarme en mis estudios, a mi abuelita que para mí es mi segunda madre, que con todo el amor del mundo supo cuidarme. También quiero dedicar este trabajo, a mis hermanos (as) y a toda mi familia que con su apoyo moral e incondicional hicieron posible el poder alcanzar tan prestigiada meta de mi vida.

Agradecimiento

Queremos dejar constancia de nuestro agradecimiento sincero a nuestros padres, familiares y amigos ya que siempre nos han brindado todo su apoyo y comprensión, que hoy se ven reflejados en este logro profesional.

Le agradecemos infinitamente a Dios por regalarnos salud y bienestar, y sobre todo por brindarnos la capacidad de adquirir los conocimientos necesarios para desarrollar el presente proyecto.

De manera muy especial, un agradecimiento sincero y consideración profunda a la Universidad Nacional de Loja, en particular a la Carrera de Ingeniería en Sistemas junto a su personal Administrativo y Docente y a la Unidad de Telecomunicaciones e Información, que gracias a su experiencia supieron guiarnos para cumplir con nuestras metas.

Finalmente, un agradecimiento especial a nuestro director del Trabajo de Tesis, Ing. Edison Leonardo Coronel, Mg. Sc. por su sabia dirección y excelente asesoría durante todo el desarrollo del presente Trabajo de Tesis.

Índice de Contenidos

Índice de Tablas

Índice de Figuras

GLOSARIO

TT: Trabajo de Titulación

UTI: Unidad de Telecomunicaciones e Información

UNL: Universidad Nacional de Loja

SLR: Systematic Literature Review (en español, Revisión Sistemática de Literatura).

CAS: Central Authentication Service (en español, Servicio de Autenticación Central).

DevOps: Development & Operations (en español, Desarrollo y Operaciones).

API: Application Programming Interface (en español, Interfaz de Programación de Aplicaciones).

SSO: Single Sign-On (en español, Inicio de Sesión Único).

XP: Extreme Programming (en español, Programación Extrema).

SAC: Servicio de Administración Central

Jasig: Java in Administration Special Interest Group (en español, Java en el Grupo de Interés especial de Administración).

LDAP: Lightweight Directory Access Protocol (en español, Protocolo Ligero de Acceso a Diretorios).

OpenLDAP: Implementación libre y de código abierto de LDAP.

BBDD: Base de Datos

TOKEN: Componente léxico de una cadena de caracteres.

Ticket o Tique: comprobante o recibo.

OpenSource: Código abierto

SiGUCAS: Sistema de Gestión Único CAS

HTTPS: Hypertext Transfer Protocol Secure (en español, Protocolo Seguro de Transferencia de Hipertexto).

SHA: Secure Hash Algorithm (en español, Algoritmo de Hash Seguro).

LOSEP: Ley Orgánica de Servicio Público

LOES: Ley Orgánica de Eduación Superior

XML: Extensible Markup Language (en español, Lenguaje de Marcado Extensible).

SOAP: Simple Object Access Protocol (en español, Protocolo Simple de Acceso a Objetos).

WSDL: Web Services Description Language (en español, Lenguaje de Descripción de Servicios Web).

1. Título

“DESARROLLO DE UN PROTOTIPO PARA EL SERVICIO DE AUTENTICACIÓN CENTRAL DE USUARIOS EN APLICACIONES WEB."

2. Resumen

En los últimos años el uso de las aplicaciones Web, ha evolucionado constantemente, teniendo su impacto directamente en la autenticación de usuarios para múltiples aplicaciones; por lo cual el usuario debe manejar varios identificadores de usuario y clave asociada, lo que resulta incómodo para los mismos. Con este fin se planteó la siguiente interrogante ¿A través de una autenticación única, utilizando el servicio de autenticación central para múltiples aplicaciones, se realizará una mejor administración de los roles de cada usuario, dando una mejor seguridad y optimización de recursos?.

El presente TT se lo desarrollo en un ambiente simulado y como caso de estudio se lo implantó en la Unidad de Telecomunicaciones e Información (UTI) de la Universidad Nacional de Loja (UNL), para ello se desarrollaron dos Systematic Literature Review (SLR - en español, Revisión Sistemática de Literatura), en base al protocolo de Bárbara Kitchenham, resultando en la selección de 30 estudios primarios, cuyos principales hallazgos permitieron establecer las características que deben cumplir las aplicaciones Web para ser integradas con el protocolo Central Authentication Service (CAS - en español, Servicio de Autenticación Central). Para el almacenamiento de la información de los usuarios se diseñó y desarrolló una estructura jerárquica en el servidor OpenLDAP, junto con un Web Service (en español, Servicio Web), que contiene funciones de autenticación y administración. Conjuntamente se utilizó el enfoque y ciclo Development & Operations (DevOps - en español, Desarrollo y Operaciones) para la integración de las aplicaciones Web previamente seleccionadas y evaluación del prototipo propuesto, por parte del personal de la UTI de la UNL.

De esta manera se pudo determinar que el prototipo propuesto obtuvo como resultado una calificación positiva del 99,42% en las pruebas de funcionalidad y un 100% en las pruebas de aceptación por parte del personal de UTI, corroborando su funcionamiento y concluyendo en su implantación para esta entidad de educación superior, como trabajo futuro se recomienda el uso de una Application Programming Interface (API - en español, Interfaz de Programación de Aplicaciones) que ofrece el protocolo CAS, para aquellas aplicaciones desarrolladas en los distintos lenguajes de programación y que no tienen soporte directo con el protocolo antes mencionado.

2.1. Summary

In recent years the use of Web applications, has evolved constantly, having its impact directly on authentication of users for multiple applications; so the user must manage several IDs username and associated password, which is uncomfortable for them. To this end, the following question was raised would ¿ Through a single authentication, using the central authentication service for multiple applications, will a better management of the roles of each user be carried out, giving a better security and optimization of resources?.

The present TT was developed in a simulated environment and, as a case study, it was implemented in the Telecommunications and Information Unit (UTI) of the National University of Loja (UNL). For this purpose, two Systematic Literature Review (SLR) were developed, based on Bárbara Kitchenham's protocol, resulting in the selection of 30 primary studies, whose main findings allowed to establish the characteristics that Web applications must comply to be integrated with the Central Authentication Service (CAS) protocol. For the storage of user information, a hierarchical structure was designed and developed on the OpenLDAP server, along with a Web Service, which contains authentication and administration functions. The Development & Operations (DevOps) approach and cycle were used jointly for the integration of the previously selected Web applications and evaluation of the proposed prototype by the staff of the UTI of the UNL.

In this way we could determine that the proposed prototype resulted in a positive score of 99,42% in testing functionality and 100% in tests for acceptance by staff of UTI, corroborating its operation and concluding in its implementation in this institution of higher education, as work future recommends the use of an Application Programming Interface (API) using the CAS Protocol, for those applications developed in different programming languages and have no direct support with external authentication services.

3. Introducción

La constante evolución de las TIC'S en el acceso a diferentes aplicaciones Web, generan un gran impacto directamente en la autenticación de usuarios y acceso a sus recursos, lo que los hace cada vez más importante; por lo que deben manejar varios identificadores de usuario y clave asociada, resultando incómodo para los mismos. Los problemas en seguridad, almacenamiento de la información y autenticación a múltiples aplicaciones Web, requieren que el usuario demuestre ser quien dice ser, esto lo realiza mediante credenciales de acceso, generando inconvenientes a los usuarios, ya que deben manejar varios identificadores de usuario y una clave asociada, lo que resulta incómodo para los mismos.

El objetivo del presente TT es mostrar el resultado de la implantación de un Sistema de Autenticación Única, mediante el uso de Single Sign-On (SSO - en español, Inicio de Sesión Único). Así mismo se destaca el uso del protocolo CAS que tiene como objetivo otorgar a los usuarios el acceso a múltiples aplicaciones, al tiempo que proporciona una credencial de usuario (como un identificador de usuario y clave asociada) es decir siendo autenticado solo una vez. Beneficiando a los usuarios pertenecientes a una entidad y que consumen los recursos de las distintas aplicaciones.

Para ello se planteó como objetivo principal "Desarrollar un prototipo para la autenticación única y centralizada de usuarios en múltiples aplicaciones Web." y para cumplir con este, se definió 3 objetivos específicos los cuales son: "Analizar la autenticación del protocolo CAS, como mecanismo centralizado.", "Diseñar y desarrollar un prototipo de servicio de autenticación central, para múltiples aplicaciones Web." y "Evaluar el Servicio de Autenticación Central desarrollado a través de los DevOps.".

Para abordar lo antes mencionado, se realizó dos SLR, la primera para el análisis del protocolo CAS para una autenticación única y centralizada y la segunda para analizar el protocolo LDAP como mecanismo centralizado, permitiéndonos cumplir con el primer objetivo. Para dar cumplimiento al segundo objetivo se diseño una estructura jerárquica en el servidor OpenLDAP, se usó la Metodología Extreme Programming (XP - en español, Programación Extrema) que nos permite comprobar el funcionamiento del Servicio Web que se desarrollo para el Servicio de Administración Central (SAC) mediante un sistema Web desarrollado en el lenguaje de programación

PHP con la ejecución de sus 4 subfases y se configuró el Sistema Jasig CAS con sus métodos de autenticación única.
Para llevar a cabo el tercer objetivo, su validación y calificación se lo ejecutó en un entorno simulado en la Unidad de Telecomunicaciones e Información (UTI) de la Universidad Nacional de Loja (UNL), haciendo uso de las 6 fases del ciclo DevOps, estabilizando el sistema desarrollado y concluyendo que los 7 profesionales de dicha entidad calificaron el presente TT en un 99,42% las pruebas de funcionalidad y en un 100% las pruebas de aceptación, por lo caul derivó en su implantación para el uso de toda la comunidad universitaria de la UNL.

El sistema proveniente del TT, se encuentra desarrollado únicamente para aplicaciones Web que permitan la conexión e integración del protocolo CAS y clientes PHP, proyectando para trabajos futuros el uso del API que ofrece el Protocolo CAS y así pueden hacer uso de la misma las distintas aplicaciones o sistemas Web desarrollados en otros lenguajes de programación.

Por otro parte, el presente documento se encuentra estructurado de la siguiente manera:
En la sección Revisión de Literatura, se elabora nueve capítulos del área de estudio que ayudaron a sustentar los conocimientos aplicados en la ejecución del presente TT. La sección de Materiales y Métodos, permitie detallar el contexto y procedimiento para el desarrollo del presente TT, los recursos utilizados y los participantes que intervinieron en todo su desarrollo. La sección Resultados, sirve para presentar todos los datos relevantes obtenidos en la ejecución del TT, para el primer objetivo se utiliza dos SLR basadas en el protocolo propuesto por Kitchenham [1], para el segundo objetivo se desarrolla una estructura jerárquica para almacenar las credenciales de usuarios, se usa la metodología XP para la creación de un sistema de administración central y se implementa Java in Administration Special Interest Group (Jasig - en español, Java en el Grupo de Interés especial de Administración), denominado Sistema Jasig CAS, para el tercer objetivo se usa el ciclo de vida de los DevOps en el desarrollo e implementación del prototipo propuesto. En la sección Discusión, se realiza un análisis de los resultados obtenidos y como se cumple con los objetivos. La sección Conclusiones, permitie expresar las ideas más relevantes rescatadas luego de haber culminado el presente TT. Finalmente, en la sección Recomendaciones, se plantea aspectos a considerar para el desarrollo de trabajos futuros.

4. Revisión de Literatura

En esta sección se realiza la recolección de información bibliográfica más relevante sobre el TT que permitió dar a conocer de una forma general las definiciones.

Se consideran los temas más importantes para el desarrollo del presente TT como son: SSO y el protocolo CAS para la autenticación única de usuarios, además la información de un Directorio Activo y Lightweight Directory Access Protocol (LDAP - en español, Protocolo Ligero de Acceso a Diretorios) en el acceso a la información central, la cual deriva en el uso del servidor OpenLDAP (Implementación libre y de código abierto de LDAP) para el almacenamiento de la información. También se describe la definición de los Servicios Web como la tecnología para intercambiar datos entre los diferentes sistemas implementados en el prototipo propuesto, se menciona la Metodología XP, el ciclo DevOps y por último se describen los trabajos relacionados más relevantes.

4.1. Inicio de sesión único

El inicio de sesión único es un proceso de autenticación de sesión de usuario que le permite proporcionar sus credenciales una vez para acceder a varias aplicaciones. El inicio de sesión único autentica al usuario para acceder a todas las aplicaciones a las que ha sido autorizado para acceder. Elimina futuras solicitudes de autenticación cuando el usuario cambia de aplicaciones durante esa sesión en particular [2].

El inicio de sesión único en la Web funciona estrictamente con las aplicaciones a las que se accede con un navegador Web. La solicitud para acceder a un recurso Web es interceptada por un componente en el servidor Web o por la propia aplicación. Los usuarios no autenticados se desvían a un servicio de autenticación y se devuelven solo después de una autenticación exitosa.

4.1.1. Single Sign-On (SSO)

Es un sistema que permite a un usuario acceder a múltiples servicios, o sistemas de aplicación después de ser autenticado solo una vez. El proceso de un SSO requiere que el usuario inicie sesión por medio de un portal solo una vez al comienzo, y luego durante la sesión el sistema SSO le proporciona de modo transparente al usuario el acceso a los diferentes servicios, recurso o aplicaciones del sistema que se encuentran asignados [2].

El proceso de un SSO, como se indica en la Figura 1 en donde el usuario inicia sesión por medio de un portal solo una vez al comienzo, y luego durante la sesión el sistema SSO le proporciona de modo transparente al usuario el acceso a los diferentes servicios, recurso o aplicaciones del sistema que se encuentran asignados [3].

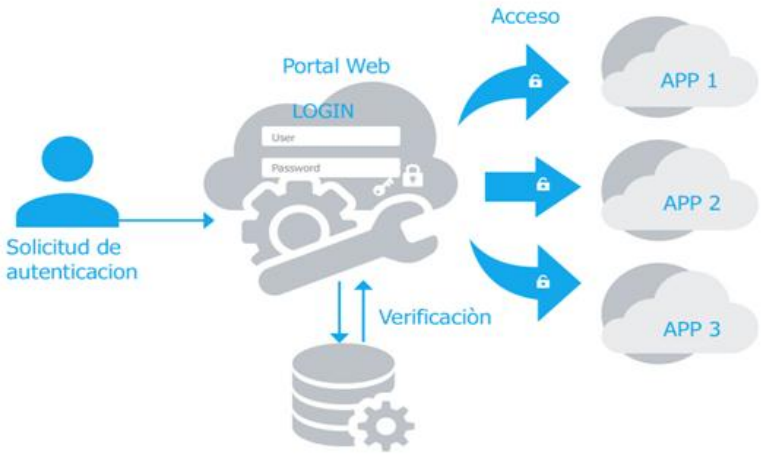


Figura 1. Sistema de Autenticación Única SSO [3].

4.1.1.1. Ventajas y desventajas de un SSO.

La TABLA **I** indica las ventajas y desventajas de un SSO [4].

TABLA I.
VENTAJAS Y DESVENTAJAS DEL MECANISMO SSO

VENTAJAS	DESVENTAJAS
La principal ventaja es que desaparece la comentada fatiga de autenticación. Aumenta la velocidad de utilización al no necesitar recurrentes peticiones de identidad.	Mayor impacto en el caso de pérdida de credenciales. El usuario pierde el control de todos los sistemas relacionados.
No hay que hacer esfuerzo memorístico ya que sólo se precisa una contraseña.	Mayor riesgo de producirse una suplantación de identidades en los accesos externos de los usuarios.
Implementación muy sencilla.	Si el sistema donde reside este mecanismo se cae, se puede producir una denegación de acceso a todos los sistemas relacionados. Por ello el sistema SSO tiene carácter crítico y tendrá que ser un sistema de High Availability (HA)
Numerosas fuentes de datos para las credenciales. Podemos utilizar BBDD, Servicios de directorio, etc	

Según lo expresa [5] la mayor ventaja que tiene el uso de un mecanismo SSO es que el usuario no tiene que recordar todas las credenciales de las diferentes aplicaciones.

4.1.1.2. Tipos de SSO

Existen 5 tipos principales de SSO también se les llama Reduced Sign on Systems (en español, sistemas de autenticación reducida). La TABLA II detalla cada tipo de SSO [4]:

TABLA II.
TIPOS DE SSO.

Nombre	Descripción
Enterprise single sign-on (E-SSO)	Los sistemas E-SSO permiten interactuar con sistemas que pueden deshabilitar la presentación de la pantalla de login.
Web single sign-on (Web-SSO)	Trabaja solo con aplicaciones y recursos accedidos vía Web. Los accesos son interceptados (por ejemplo, mediante un proxy). Los usuarios no autenticados que tratan de acceder, son redirigidos a un servidor o servicio Web de autenticación y regresan solo después de haber logrado un acceso exitoso o con un TOKEN de autenticación para la aplicación destino. Se utilizan cookies, parámetros por GET (más inseguro) o POST para reconocer aquellos usuarios que acceden y su estado de autenticación.
Kerberos	Los usuarios se registran en el servidor Kerberos y reciben un "ticket", luego las aplicaciones cliente lo presentan para obtener acceso del recurso solicitado.
Identidad federada	Utiliza protocolos basados en estándares para habilitar que las aplicaciones puedan identificar los clientes sin necesidad de autenticación redundante.
OpenID	Es un proceso de SSO distribuido y descentralizado donde la identidad se compila en una url que cualquier aplicación o servidor puede verificar.

4.1.1.3. Implementación de un SSO

Son numerosas las implementaciones que proporcionan SSO, la siguiente relación son implementaciones muy utilizadas con las características más relevantes y basadas en licencias OpenSource/FreeSoftware [4].

- **CAS**. Implementación con amplio soporte. Posee licencia Apache 2.0. El desarrollo de CAS de Apereo es bastante conocido. Soporte para CAS, SAML1, SAML2, OAuth2, SCIM, OpenID Connect y WS-Fed protocolos.
- **JBoss SSO**. Desarrollado por Red Hat. SSO federado. Free Software.
- **JOSSO.** Desarrollado por JOSSO. SSO Server. Free Software.
- **Keycloak**. SSO federado desarrollado por Red Hat. Soporta los protocolos normalizados: OpenID Connect, OAuth 2.0 and SAML 2.0 para la Web, clustering y single sign on.
- **OpenAM**. Solución de la empresa Forge Rock. Tiene su origen en OpenSSO desarrollado por SUN. LLeva a cabo la gestión de accesos, derechos y la plataforma del servidor de federación. Arquitectura basada en Java con soporte para los protocolos: SAML, WS- Federation, OpenID y XACML.
- **Shibboleth**. Proyecto de identidad federada, Con licencia Apache. Se apoya en el estándar SAML.
- **WSO2 Identity Server**. Servidor de identidades creado por WSO2, creador del Enterprise Service Bus WSO2, con soporte para: SAML 2.0, OpenID, OpenID Connect, OAuth 2.0, SCIM, XACML, Federación pasiva.

4.2. Central Authentication Service (CAS)

CAS fue concebido y desarrollado por Shawn Bayern de Tecnología y Planificación de la Universidad de Yale. Más tarde fue mantenido por Drew Mazurek en Yale. CAS 1.0 implementó el inicio de sesión único. CAS 2.0 introdujo la autenticación de proxy de múltiples niveles. Varias otras distribuciones CAS se han desarrollado con nuevas características.

CAS, es un sistema de autenticación creado para proporcionar una forma confiable para que una aplicación autentique a un usuario. Numerosas solicitudes de contraseña y diferentes credenciales requeridas para cada sistema han creado la necesidad de que las instituciones y organizaciones adopten un proceso de autenticación de inicio de sesión único seguro en la Web. CAS incluye un inicio de sesión único que proporciona comodidad al usuario, ya que lo protege contra la proliferación de credenciales y la exposición de contraseñas, y centraliza la experiencia de inicio de sesión [6][7]**.**

CAS es un protocolo de inicio de sesión único para la Web. Su propósito es permitir que un usuario acceda a múltiples aplicaciones mientras proporciona sus credenciales (como identificación de usuario y contraseña) solo una vez. El protocolo CAS involucra al menos tres partes [8]:

- Un navegador Web cliente.
- La aplicación Web solicita autenticación (llamada al servicio CAS).
- El servidor CAS.

4.2.1. Características CAS

Las características del protocolo CAS se describen a continuación [4][3][6]:

- Esta implementación es Open Source.
- Proporciona el protocolo Open Source CAS muy bien documentado.
- Servidor Java.
- Integración con un amplio abanico de clientes: Java, .Net, PHP, Perl, Apache, uPortal y otros.
- Proporciona numerosos módulos que permiten utilizar varios métodos de autenticación. Entre otros tenemos: LDAP, base de datos, X.509, JASS, RADIUS, SPINEGO, Apache Cassandra, Remote Adress, JWT, Rest.
- Soporte para múltiples protocolos. Además de soportar el protocolo propio, CAS, soporta los siguientes:
    - OpenID es un estándar de identificación digital descentralizado.

- Open Authorization (OAuth) es un estándar abierto que permite flujos simples de autorización para sitios Web o aplicaciones informáticas.
- OpenID Connect (OIDC) es un protocolo de autenticación implementada utilizando OAuth 2.0, un framework de autorización.
- WS-Federation (Web Services Federation) es una especificación de Identity Federation.
- Security Assertion Markup Language (SAML1) lenguaje de marcado de aserción de seguridad, es un estándar XML para intercambiar datos de autenticación y autorización entre dominios de seguridad.
- SAML2 es una versión del estándar SAML basado en XML que utiliza tokens de seguridad que contiene aserciones.
- REST Protocol Representational State Transfe (en español, Transferencia de Estado Representacional) es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web - WWW (red informática mundial).

- Se integra con uPortal, BlueSocket, TikiWiki, Mule, Liferay, Moodle y otra documentación de la comunidad y soporte de implementación.
- Tiene una amplia base de desarrolladores. Más de cuarenta universidades, entre otras entidades, participan en el desarrollo.
- Numerosos casos de éxito. Es de reseñar la valoración que hizo la universidad de Murcia para proveerse de un mecanismo de SSO.
- Abundante software de terceros da soporte a CAS como mecanismo de autenticación. Excelentes características de escalabilidad.
- Documentación comunitaria y soporte a la implementación.
- Una extensa comunidad de adoptantes.

4.2.2. Jasig CAS

En diciembre de 2004, CAS se convirtió en un proyecto de Java in Administration Special Interest Group – Jasig (en español, Java en el Grupo de Interés especial de Administración), que a partir de 2008 es responsable de su mantenimiento y desarrollo. Anteriormente llamado "Yale CAS", CAS ahora también se conoce como "Jasig CAS". En 2010, Jasig entró en conversaciones con la Fundación Sakai para fusionar las dos organizaciones. Ambas organizaciones se consolidaron como Apereo Foundation en diciembre de 2012. En diciembre de 2006, la Fundación Andrew W. Mellon otorgó a Yale su Primer Premio Anual Mellon de Colaboración Tecnológica, por un monto de $ 50,000, para el desarrollo de CAS de Yale. En el momento de ese

premio CAS se usaba en "cientos de campus universitarios (entre otros beneficiarios)". En mayo de 2014, se publicó la especificación 3.0 del protocolo CAS [4].

El servidor Jasig CAS es un servlet de Java creado en Spring Framework, cuya responsabilidad principal es autenticar a los usuarios y otorgar acceso a los servicios habilitados para CAS, comúnmente llamados clientes CAS, mediante la emisión y validación de tickets. Una sesión de SSO se crea cuando el servidor emite un Ticket Granting Ticket TGT (en español, Ticket de Otorgamiento de Tickets) al usuario al iniciar sesión correctamente. Se emite un Service Ticket ST (en español, Ticket de Servicio) a un servicio a petición del usuario a través de redirecciones del navegador utilizando el TGT como un token. El ST se valida posteriormente en el servidor CAS a través de la comunicación del canal de retorno. Estas interacciones se describen con gran detalle en el documento del Protocolo CAS [9].

El protocolo CAS es un protocolo simple y poderoso basado en tickets. Se trata de uno o muchos clientes y un servidor. Los clientes están integrados en diferentes aplicaciones CASified (llamadas "servicios CAS"), mientras que el servidor CAS es un componente independiente [8]:

- El servidor CAS es responsable de autenticar a los usuarios y otorgar accesos a las aplicaciones.
- Los clientes CAS protegen las aplicaciones CAS y recuperan la identidad de los usuarios otorgados desde el servidor CAS.

Los conceptos clave son [8]:

- El TGT, almacenado en la TG-Cookie, representa una sesión de SSO para un usuario.
- El ST, transmitido como un GET parámetro en las urls, representa el acceso otorgado por el servidor CAS a la aplicación CASified para un usuario específico.

4.2.3. Arquitectura

El servidor CAS y los clientes CAS comprenden los dos componentes físicos de la arquitectura del sistema CAS que se comunican mediante varios protocolos como se puede observar en la Figura 2.

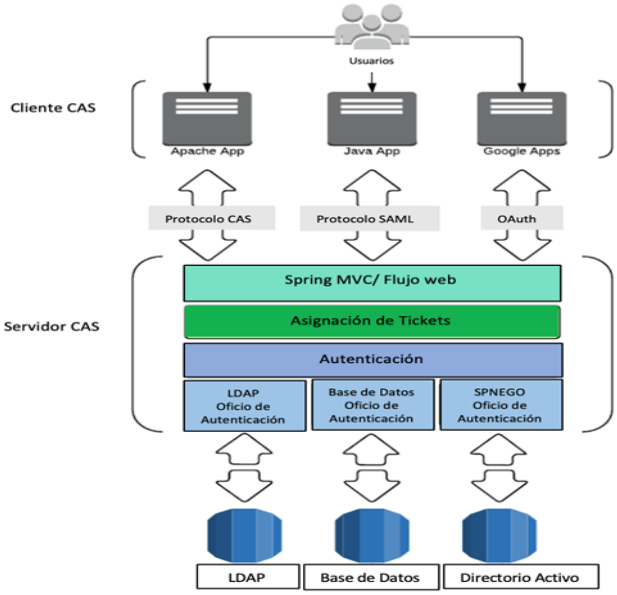


Figura 2. Arquitectura del sistema Jasig CAS [9]**.**

La Figura 2, describe al cliente CAS que es cualquier aplicación habilitada para CAS que puede comunicarse con el servidor a través de un protocolo compatible y el servidor CAS, cuya responsabilidad principal es autenticar a los usuarios y otorgar acceso a los servicios habilitados para CAS, comúnmente llamados clientes CAS, mediante la emisión y validación de tickets [9].

4.2.3.1. Flujograma del funcionamiento de CAS

Jasig CAS, tiene un flujo de autenticación básica SSO, en la Figura 3, se indica el procedimiento para realizar la autenticación única, basada en tickets.

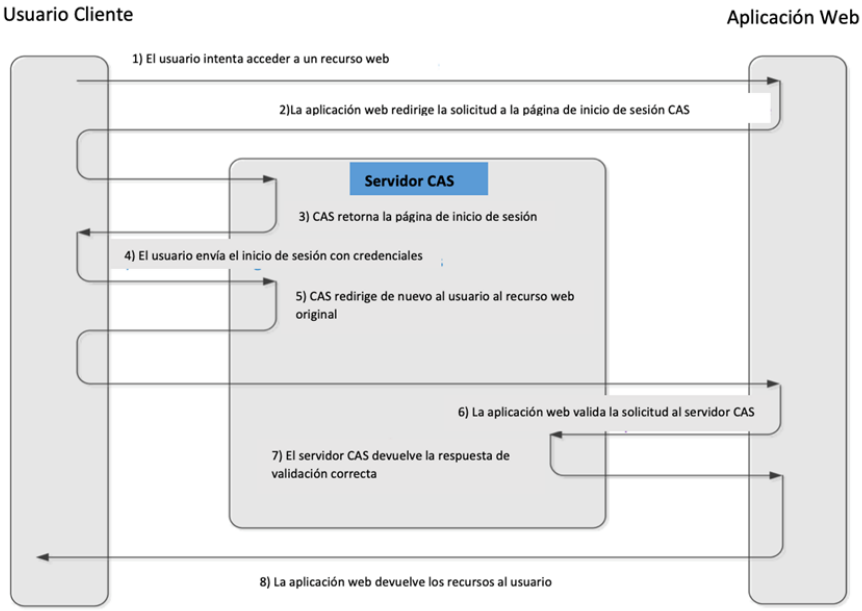


Figura 3. Flujograma básico de la autenticación CAS [8]**.**

1. Un usuario solicita un recurso Web específico URI en el servidor Web por ejemplo http://moodle/login.php. La aplicación Web aprovecha un servidor CAS para proporcionar el servicio de autenticación.
2. Si la aplicación Web comprueba la solicitud y no encuentra tickets de CAS junto con la solicitud, redirigirá la solicitud a un URI de inicio de sesión de CAS. El agente de usuario CAS redirige la solicitud del usuario al inicio de sesión del servidor CAS. Esta redirección también se produce cuando un ticket CAS no es válido. El URI de redireccionamiento debe contener un parámetro de "servicio" cuyo valor es el URI de recurso Web original. Esto es importante por dos razones:
    2.1. El servidor CAS producirá un ST para el acceso de recursos. Dado que cada ST es específico para un recurso único y un uso único, el servidor CAS debe saber a qué servicio URI está intentando acceder el usuario.
    2.2. El servidor CAS necesita conocer el URI original del servicio para redirigir al usuario después de una autenticación exitosa.
    2.3. Un URI redirigido sigue este patrón https://cas-server/login?service=http://moodle/login.php
3. El servidor CAS devuelve una página HTML que contiene un formulario de inicio de sesión.
4. El usuario publica el ID de usuario y la contraseña en el servidor CAS.
5. El servidor CAS debe validar la credencial del usuario contra un servidor de directorio de usuario o un servidor LDAP. El servidor de CAS también debe verificar si el URI del servicio dentro del parámetro "servicio" (http://moodle/login.php) se ha registrado como un servicio de CAS. Si no, la autenticación falla incluso si la credencial del usuario es correcta.

    Después de la validación, el servidor CAS redirige al cliente a la URI del servicio original. Además de eso, CAS agrega la información a continuación en la respuesta de redireccionamiento:
    5.1. Se agrega un ST al URI del servicio original, que representa el acceso autorizado al recurso Web y luego será validado por la aplicación Web. Entonces, la redirección de URI de retorno es como: http://moodle/login.php?Ticket=ST1bdgbwHIReBonmaudvxJlcas
    5.2. Una cookie llamada 'CASTGC' se devuelve al cliente que es esencialmente un TGT.

    El ST se usa solo una vez, pero el TGT se puede reutilizar para representar a un usuario autenticado. Cuando el cliente intenta acceder a otro recurso Web, la cookie de CASTGC se reutiliza para solicitar un nuevo ST sin necesidad de

proporcionar la credencial de usuario nuevamente, siempre que el CASTGC no caduque.

6. La aplicación Web valida la solicitud. Específicamente, obtiene el parámetro "ticket" del URI de solicitud y lo valida en el servidor CAS.
7. El servidor CAS devuelve la respuesta de validación correcta. Además, el servidor CAS puede devolver algunos atributos de usuario después de la validación para que la aplicación Web pueda realizar el control de acceso para los atributos.
8. La aplicación Web devuelve el recurso Web al cliente.

4.3. Directorio Activo

Un directorio es una lista de información acerca de objetos acomodados en algún orden que da detalles de los mismos. Ejemplos comunes son un directorio telefónico y las fichas bibliográficas de los libros. Para un directorio telefónico, los objetos listados son personas, los nombres son ordenados alfabéticamente y los detalles de cada persona son los números telefónicos y direcciones. Los libros de una librería son ordenados por autor o por título y la información adicional como la editorial, el año de publicación, edición, etc., serán los detalles [10].

En términos “computacionales”, un directorio es una base de datos especializada, también conocida como un repositorio de datos, que almacena y ordena información acerca de objetos. Un directorio particular podría listar información acerca de impresoras (los objetos), por ejemplo, la ubicación, velocidad en páginas por minuto, tipo de impresión soportada, marcas, etc [10].

4.3.1. Directorio centralizado

Un directorio es una base de datos especializada específicamente diseñada para buscar y navegar, además de apoyar las funciones básicas de búsqueda y actualización. Los directorios tienden a contener información descriptiva basada en atributos y admiten sofisticadas capacidades de filtrado.

Es una forma organizada para el almacenaje de información, un directorio maneja su estructura de forma jerárquica, un ejemplo claro es cómo maneja los sistemas Unix su sistema de directorios [11].

En si el directorio es como un árbol que se va extendiendo con sus ramas, de esta manera se trabaja la jerarquización. Así un directorio tiene una raíz o un inicio, donde todo se desprende. Esta forma de representar un directorio se le llama DIT (Árbol de información de directorio) [11].

4.4. LDAP

EL Lightweight Directory Access Protocol - LDAP (en español, Protocolo Ligero de Acceso a Directorios) es un conjunto de protocolos abiertos usados para acceder información guardada centralmente a través de la red. Está basado en el estándar X.500 para compartir directorios, pero es menos complejo e intensivo en el uso de recursos. Por esta razón, a veces se habla de LDAP como "X.500 Lite." El estándar X.500 es un directorio que contiene información de forma jerárquica y categorizada, que puede incluir nombres, directorios y números telefónicos [12],[13].

Como X.500, LDAP organiza la información en un modo jerárquico usando directorios. Estos directorios pueden almacenar una gran variedad de información y se pueden incluso usar de forma similar al Servicio de información de red (NIS), permitiendo que cualquiera pueda acceder a su cuenta desde cualquier máquina en la red acreditada con LDAP [13], [14].

LDAP es un sistema cliente/servidor. El servidor puede usar una variedad de bases de datos para guardar un directorio, cada uno optimizado para operaciones de lectura rápidas y en gran volumen. Cuando una aplicación cliente LDAP se conecta a un servidor LDAP puede, o bien consultar un directorio, o intentar modificarlo [12].

4.4.1. Estructura Jerárquica

La estructura de LDAP se basa en una estructura jerárquica de árbol como se puede visualizar en la Figura 4, la cual se denomina árbol de información de directorio donde la información denominada entrada está representada por bifurcaciones denominada raíz. Cada entrada está conformada por un conjunto de pares clave/valor denominados atributos [15].

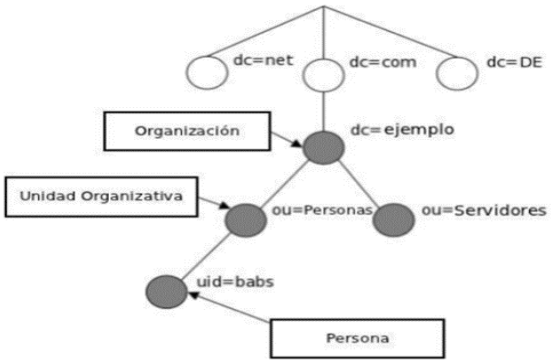


Figura 4. Árbol de directorios LDAP [15]**.**

Casi todos los directorios almacenan información siguiendo una estructura similar a la de una lista telefónica impresa. Las entradas están organizadas en una estructura jerárquica en árbol lo que permite gestionarlas y buscarlas de manera eficaz y versátil. Cada nodo del árbol de datos se lo denomina **entrada**. Cada entrada tiene una denominación, o **DN** (Distinguished Name, nombre distinguido), que se forma de la concatenación de los DNs relativos (o RDNs) de las entradas **padre** hasta llegar a la entrada **raíz** del árbol [16].

4.4.1.1. Como se accede a la Información

LDAP define operaciones para interrogar y actualizar el directorio. Provee operaciones para añadir y borrar entradas del directorio, modificar una entrada existente y cambiar el nombre de una entrada. La mayor parte del tiempo, sin embargo, LDAP se utiliza para buscar información almacenada en el directorio. Las operaciones de búsqueda de LDAP permiten buscar entradas que concuerdan con algún criterio especificado por un filtro de búsqueda. La información puede ser solicitada desde cada entrada que concuerda con dicho criterio [16].

4.4.1.2. Cómo se protege la información de los accesos no autorizados

Algunos servicios de directorio no proveen protección, permitiendo a cualquier persona acceder a la información. LDAP provee un mecanismo de autentificación para los clientes, o la confirmación de identidad en un servidor de directorio, facilitando el camino para un control de acceso que proteja la información que el servidor posee. LDAP también soporta los servicios de privacidad e integridad [12].

4.4.2. Características del Protocolo LDAP

A continuación se describen las principales características del protocolo LDAP [17]:

- **Escalabilidad**

  Los directorios LDAP, particularmente cuando una base de datos relacional hace una copia de seguridad de ellos, como en IBM SecureWay Directory, son muy escalables. El rendimiento de los directorios de gran tamaño con millones de entradas es excelente.

  Debido a la base estándar común, otro factor de escalabilidad es la posibilidad de configurar de manera simple hardware y software de mayores prestaciones. LDAP no se basa en un sistema operativo específico y es independiente del proveedor.

- **Disponibilidad**

  LDAP soporta la réplica y división de espacios de nombres. La réplica permite a varios servidores LDAP almacenar el contenido del mismo directorio. Esto permite a los clientes disponer de estos servidores adicionales cuando uno presenta anomalías.

  La división permite almacenar las secciones de todo el directorio en diferentes ubicaciones de servidores distintos. Esto no sólo aumenta la disponibilidad (ni una sola anomalía) sino que simplifica la gestión distribuida.

- **Seguridad**

  LDAP soporta características de seguridad que impiden el acceso no autorizado a datos. Los protocolos de comunicación segura, como SSL y procedimientos de autenticación, junto con las políticas de listas de control de accesos (ACL) para entradas de datos, garantizan el máximo nivel de seguridad.

- **Gestionabilidad**

  Las versiones actuales de LDAP, como IBM SecureWay Directory, proporciona una interfaz gráfica de usuario tanto para la administración de sistemas como para la administración de datos de directorio. Su esquema ampliable dinámicamente le permite ampliar el esquema de directorios sin interrumpir el servicio.

- **Estandarización**

  El protocolo LDAP junto con la mayoría de prestaciones de cliente/servidor relacionadas, las interfaces de programación de aplicaciones (API) y las definiciones de datos están definidos por estándares oficiales o los RFC (solicitud de comentarios) correspondientes.

4.4.3. Ventajas para utilizar LDAP

La mayor ventaja de LDAP es que se puede consolidar información para toda una organización dentro de un repositorio central. Por ejemplo, en vez de administrar listas de usuarios para cada grupo dentro de una organización, puede usar LDAP como directorio central, accesible desde cualquier parte de la red. Puesto que LDAP soporta la Capa de conexión segura (SSL) y la Seguridad de la capa de transporte (TLS), los datos confidenciales se pueden proteger de los curiosos [12], [13].

LDAP también soporta un número de bases de datos back-end en las que se guardan directorios. Esto permite que los administradores tengan la flexibilidad para desplegar la base de datos más indicada para el tipo de información que el servidor tiene que diseminar. También, ya que LDAP tiene una interfaz de programación de aplicaciones (API) bien definida, el número de aplicaciones acreditadas para LDAP son numerosas y están aumentando en cantidad y calidad [12].

4.4.4. Esquema eduPerson

Los esquemas son definiciones que describen qué tipos de información se pueden almacenar como entradas de un directorio LDAP. Es posible que se deba ampliar el esquema del servidor de directorios para admitir clientes del servicio de nombres LDAP.

eduPerson es un esquema de atributos que incluye enlaces a un esquema LDAP y a Security Assertion Markup Language - SAML (en español, Lenguaje de Marcado para Confirmaciones de Seguridad). Está diseñado para incluir y estandarizar atributos de persona y organización ampliamente utilizados en la educación superior y la investigación que no están duplicados en otros objetos ampliamente utilizados como inetOrgPerson [18]**.**

Su objetivo principal es la práctica en las organizaciones en torno a un conjunto común de atributos para la información específica de la educación superior y las mejores prácticas de IAM (gestión de identidad y acceso).

La relación de eduPerson con otros esquemas como inetOrgPerson, amplía y describe los estándares de esquemas existentes para evitar la reinvención al tiempo que agrega atributos específicos y útiles para la educación superior y la investigación [18].

eduPerson mediante sus atributos puede proteger la privacidad de los usuarios, esto mediante sus servicios, los cuales trabajan de la siguiente manera [18]:

- Los servicios pueden basarse en atributos genéricos que no identifican a una persona específica, como eduPersonAffiliation o eduPersonEntitlement.
- Los servicios que necesitan mantener un registro interno que es administrar las preferencias, pueden usar eduPersonTargetedID o eduPersonUniquedID, proporcionando un ID único, pero no correlacionado fácilmente con la actividad en otros servicios.

4.5. OpenLDAP

OpenLDAP es una implementación de código abierto de LDAP desarrollada por el proyecto OpenLDAP. Está liberado bajo su propia licencia OpenLDAP Public License guía [13].

La suite de OpenLDAP, incluye un número de características importantes [13] :

- Slapd: Servidor LDAP standalone.
- Liberías que implementan LDAP.
- Backends: La arquitectura del servidor OpenLDAP está dividida en dos niveles: frontend que maneja las conexiones de redes y el procesamiento del protocolo, y una base de datos backend que se ocupa del almacenamiento de los datos. El servidor slapd puede utilizar arbitrariamente varios backends a la vez y tener arbitrariamente varias instancias de cada backend (por ejemplo, varias bases de datos) activas.
- Soporte LDAPv3: OpenLDAP soporta la Capa de autenticación y seguridad (SASL), la Seguridad de la capa de transporte (TLS) y la Capa de conexión segura (SSL), entre otras mejoras. Muchos de los cambios en el protocolo desde LDAPv2 han sido diseñados para hacer LDAP más seguro.
- Soporte IPv6: OpenLDAP soporta la próxima generación del protocolo de Internet versión 6.
- LDAP sobre IPC: OpenLDAP se puede comunicar dentro de un sistema usando comunicación interproceso (IPC). Esto mejora la seguridad al eliminar la necesidad de comunicarse a través de la red.
- API actualizado: Mejora la forma en que los programadores se conectan para usar servidores de directorio LDAP.
- Soporte LDIFv1: Provee compatibilidad completa con el formato de intercambio de datos, Data Interchange Format (LDIF) versión 1.

4.5.1. Algoritmos de cifrado en el servidor OpenLDAP.

El servidor OpenLDAP, incorpora los siguientes algoritmos de cifrado para la seguridad de la información, como podemos observar en la TABLA III.

TABLA III.
ALGORITMOS DE CIFRADO DEL SERVIDOR OPENLDAP.

Algorítmo	Descripción
Plain	Utiliza el texto en formato original, es decir envía y recibe la información original.
MD5	Message-Digest Algorithm 5 (en español, Algoritmo de Resumen del Mensaje 5) es un algoritmo que produce un código de 128 bits. Fue creado por Ron Rivest en 1991 y se convirtió en el estándar de Internet RFC 1321. Recientemente se han encontrado pequeñas vulnerabilidades en este algoritmo que sugieren un movimiento hacia SHA1 [19].
SMD5	Es un Algoritmo de reducción criptográfico de 128 Bits ampliamente usado.
SHA	Secure Hash Algorithm (en español, Algoritmo de Hash Seguro). Este algoritmo fue declarado estándar Federal Information Processing Standard PUB 180 en 1993, pero en 1995 la Agencia de Seguridad Nacional (NSA) lo sustituyó por una versión mejorada que actualmente se conoce como SHA-1 y que se considera más seguro que MD5. Produce un código hash de 160 bits para mensajes de longitud máxima 264 bits, aunque existen otras variantes poco utilizadas todavía que producen códigos de mayor longitud [19].
SHA-2	Es una familia de algoritmos que incluye a los algoritmos SHA-224, SHA-256, SHA-384 y SHA-512. Se tratan de funciones de hash que reciben un conjunto de datos de cualquier longitud, y devuelve un valor de hash de 224, 256, 384 y 512 bits, respectivamente [20].

4.6. Web Service

Un Servicio Web es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Estas aplicaciones pueden estar desarrolladas en lenguajes de programación diferentes y ser ejecutadas sobre plataformas diferentes. La interoperabilidad se consigue mediante la adopción de estándares abiertos [21].

4.6.1. Arquitectura Orientada a Servicios

Los Servicios Web basados en Simple Object Access Protocol – SOAP (en español, Protocolo Simple de Acceso a Objetos) reciben una gran aceptación por la mayor parte de desarrolladores de software y analistas. Al contrario que los Servicios Web basados en RPC, este estilo es débilmente acoplado, lo cual es preferible ya que se centra en el “contrato” proporcionado por la especificación de las interfaces, más que en los detalles de implementación subyacentes [21].

Las soluciones SOAP han sido creadas para diseñar y desarrollar sistemas distribuidos que satisfagan objetivos de negocio, como facilidad y flexibilidad de integración con sistemas legados, alineación directa a los procesos de negocio reduciendo costes de implementación, innovación de servicios a clientes y una adaptación ágil ante cambios incluyendo reacción temprana ante la competitividad [21].

Existen diversos estándares relacionados a los Servicios Web, como XML, HTTP, SOAP, REST, WSDL o UDDI. Hay que considerar, sin embargo, que un sistema SOA no necesita utilizar estos estándares para ser "Orientado a Servicios" pero normalmente es altamente recomendable su uso [22].

4.6.2. Desarrollo de Web service con SOAP

SOAP, es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP puede formar la capa base de una pila de protocolos de Web Service (en español, Servicios Web) ofreciendo un framework de mensajería básica en la cual los Servicios Web se puedan construir. Básicamente, este protocolo basado en XML Extensible Markup Language (en español, Lenguaje de Marcado Extensible) consiste de tres partes: (1) un contenedor (envelope), el cual define qué hay en el mensaje y como procesarlo; (2) un conjunto de reglas de codificación para expresar instancias de tipos de datos; y (3)

una convención para representar llamadas a procedimientos y respuestas. Por otro lado, las principales características del protocolo SOAP son [23]:

- Extensibilidad, seguridad y WS-routing son extensiones aplicadas en el desarrollo.
- Neutralidad, SOAP pude ser utilizado sobre cualquier protocolo de transporte como: Hypertext Transfer Protocol - HTTP (en español, Protocolo de Transferencia de Hipertexto), SMTP, TCP o JMS.
- Independencia, SOAP permite cualquier modelo de programación.

XML, es un estándar para la definición de lenguajes de marcas, flexible y extensible, usado en los Servicios Web para especificar lenguajes y protocolos necesarios. Permite definir lenguajes para describir los servicios y representar los mensajes intercambiados [23].

SOAP, Mecanismo de interacción entre extremos que surge a partir de la necesidad de un formato de mensajes neutro, abierto y extensible. La representación de los mensajes de invocación (argumentos) y respuesta como documentos XML. Especifica el modo de interacción, RCP (síncrono) o petición (asíncrono). Los mensajes se mapean en el protocolo de transporte (HTTP, SMTP) [21].

WSDL Web Service Description Language (en español, Lenguaje de Descripción de Servicios Web). La descripción de los servicios y sus interfaces se realiza de forma estándar mediante documentos XML, incluyendo toda la información necesaria para suplir un middleware común centralizado. Especifica las operaciones disponibles, con los parámetros de entrada y de salida. Puede usarse para generar los stubs/skeleton y las capas intermedias necesarias para escribir clientes que invoquen los Servicios Web y servidores que los implementen [21].

UDDI son las siglas del catálogo de negocios de Internet denominado (Universal Description, Discovery and Integration), nos permite la publicación y localización de servicios. La descripción de los servicios (WDSDL) se almacena en un directorio de servicios, UDDI especifica cómo se publican y descubren los servicios y como trabajan los directorios de servicios Web. El servidor da del alta los servicios (WSDL + descripción) y el cliente descubre servicios (WSDL) [23].

4.6.3. Ventajas y Desventajas de los Servicios Web

Un servicio Web es un servicio ofrecido por una aplicación que expone su lógica a clientes de cualquier plataforma. A continuación, en la TABLA IV se indican las ventajas y desventajas de un servicio Web

TABLA IV.
VENTAJAS Y DESVENTAJAS DE UN WEB SERVICE

Ventajas	Desventajas
Aportan interoperabilidad entre aplicaciones de software, independientemente de sus propiedades o de las plataformas sobre las que se instalen [22].	Para realizar transacciones no pueden compararse en su grado de desarrollo con los estándares abiertos de computación distribuida como CORBA (Common Object Request Broker Architecture) [24].
Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido [22].	El problema con HTTP y Hypertext Transfer Protocol Secure – HTTPS (en español, Protocolo Seguro de Transferencia de Hipertexto) cuando se emplean para sustentar Servicios Web es que estos protocolos son stateless, sin estado, la interacción entre el servidor y el cliente es típicamente breve, y cuando no hay datos siendo intercambiados, el servidor y el cliente no tienen conocimiento sobre el otro [24].
Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados [22].	Su rendimiento es bajo si se compara con otros modelos de computación distribuida, tales como RMI (Remote Method Invocation), CORBA, o DCOM (Distributed Component Object Model) [24].
Independencia del lenguaje de programación: El servidor y el cliente no necesitan estar escritos en el mismo lenguaje [22].	Aunque la simplicidad de los Servicios Web es una ventaja, en algunos aspectos puede ser un estorbo. Los Servicios Web usan protocolos basados en texto plano que usan un método demasiado pesado para identificar la información [24].
Los servicios Web permiten centralizar los datos, distribuirlos sobre internet y con las nuevas herramientas pueden ser accedidos a través de una gran variedad de dispositivos [24].	- Tanto HTTP como HTTPs (el núcleo de los protocolos Web) son simples, pero no fueron inicialmente considerados para sesiones largas. Típicamente, un navegador realiza una conexión HTTP, solicita una página Web y después se

	desconecta. En entornos CORBA o RMI, un cliente se conecta al servidor, pudiendo permanecer conectado durante un periodo extenso de tiempo, recibiendo información periódica en el cliente. Esta interacción es difícil de obtener con los Servicios Web, y es necesario un trabajo extra para conseguirlo [24].

4.7. DevOps

DevOps fue acuñado en 2009 por Patrick Debois, que se ha convertido desde entonces en uno de los gurús dentro de la comunidad. El término se conforma de combinar las palabras "desarrollo" y "operaciones, del inglés "Development & Operations", y puede servir como punto de partida para entender qué significa exactamente el término DevOps. Esta nueva cultura no es un proceso, una tecnología concreta o un estándar, sino un conjunto de técnicas, pensamientos, y modelos de trabajo. También se utiliza el término "movimiento DevOps" cuando se habla de temas acerca de la adopción de nuevos ratios e indicadores y tendencias de futuro y "entorno DevOps" para referirse a la estrategia organizativa que sugiere la cultura DevOps [25].

DevOps se describe con frecuencia como una relación más colaborativa y productiva entre los equipos de desarrollo y los equipos de operaciones. Esta relación mejorada y el aumento en la eficiencia en la colaboración reduce el riesgo de producción asociado con cambios o entregas frecuentes desde desarrollo. El concepto DevOps (Development+Operations) postula que en el software empresarial se ha borrado la línea que dividía el desarrollo de las operaciones. Cuando se adoptan nuevas metodologías de desarrollo (como el desarrollo ágil de software) en una organización tradicional con departamentos separados para Desarrollo, Operaciones, Control de calidad y la Implementación, donde antes no había necesidad profunda de integración entre dichos departamentos de TI, ahora requieren cerrar una colaboración multidepartamental [26].

El término DevOps se refiere más que a sólo implementaciones de software: es un conjunto de procesos y métodos para pensar acerca de la comunicación y la colaboración entre los departamentos mencionados anteriormente. Las empresas que tienen entregas muy frecuentes de software pueden requerir una conciencia u orientación del tipo DevOps. La adopción de DevOps está siendo impulsada por factores tales como, se describe en [26] y se puede evidenciar en la Figura 5.

Figura 5. Modelo del enfoque DevOps. Fuente diseño propio de los autores del TT.

En 2012, Debois elaboró cuatro áreas clave para indicar los aspectos relevantes en DevOps. En la Figura 6 se muestra como la interacción entre desarrollo y operaciones es bidireccional y se incide en las tareas que potencian el intercambio de conocimiento y la evaluación entre equipos de trabajo [25].

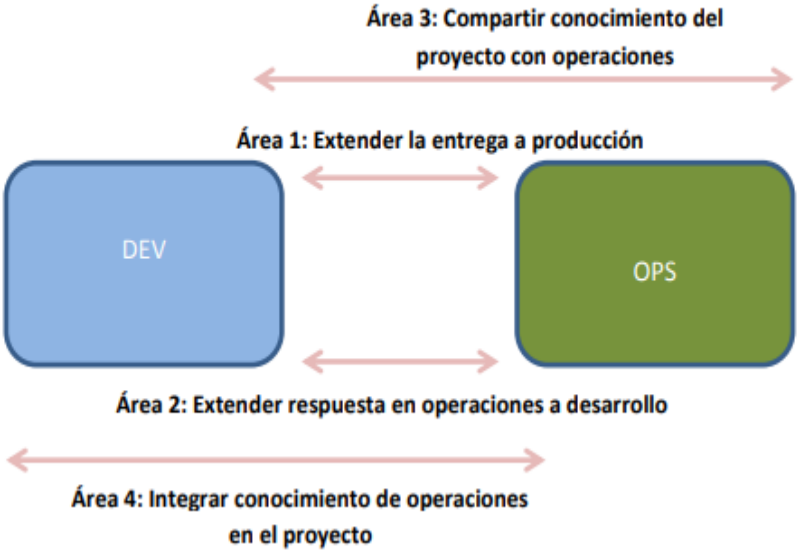


Figura 6. Interacción entre desarrollo y operaciones [17].

Área 1: Extender la entrega a producción: Los equipos de desarrollo y operaciones colaboran para mejorar el proceso de entrega de un proyecto al entorno de producción. Área 2: Extender la respuesta del sistema en operaciones al proyecto: Hacer extensible toda la información relevante en producción al equipo de desarrollo.

Área 3: Compartir todo el conocimiento del proyecto al equipo de operaciones: El equipo de desarrollo comparte la responsabilidad de todo lo que ocurre en el entorno de producción.

Área 4: Integrar el conocimiento de operaciones en el equipo de desarrollo: Operaciones debe involucrarse en el proyecto desde el inicio.

4.7.1. Herramientas principales DevOps

Dentro de las herramientas principales DevOps podemos destacar las siguientes:

- **Aprovisionamiento y configuración de ambientes**

  Cualquier administrador de sistemas ha vivido la frustrante experiencia de que una aplicación falle en producción, y al preguntarle al desarrollador al respecto

nos conteste “pues funciona en mi máquina” ... (sí, en efecto dan ganas de ahorcarlos). Las herramientas de aprovisionamiento y configuración ayudan a evitar estas situaciones, ya que nos permiten especificar a gran detalle los ambientes de ejecución, de tal manera que puedan ser replicados de forma automatizada y repetible por medio de scripts. A esto es a lo que se refiere el concepto de “administrar la infraestructura como código” [27]. Entre de las herramientas más conocidas para aprovisionamiento y configuración de sistemas están: Chef, Puppet, Ansible y Salt.

- **Integración continua**

  La Integración Continua es la forma en la que el equipo de desarrollo de software integra su trabajo parcial o total, en un determinado tiempo establecido por el equipo de trabajo [28]. Requiere de herramientas de automatización que son únicas para todo el equipo de desarrolladores. Estas herramientas ayudan a integrar en forma continua partes de código que son validados por pruebas automáticas, lo cual vuelve más eficiente el trabajo del equipo de desarrollo, ya que permite detectar fallos en etapas tempranas del ciclo de desarrollo. La Integración continua de cada uno de los integrantes del equipo va a favorecer en los tiempos y calidad del producto [27]. Su objetivo es que los problemas de integración surjan lo antes posible y se mejore el entendimiento de las dependencias entre el código que cada quien está trabajando. Para lograr esto requerimos usar un servidor de integración continua [28].

  Entre los servidores de integración continua más conocidos están: Jenkins. Atlassian Bamboo, Travis CI y JetBrains TeamCity.

- **Gestión de despliegue**

  El despliegue continuo es una práctica que permite llevar los resultados de un proceso de desarrollo a un entorno similar al de producción donde las pruebas funcionales puedan darse a escala completa. El objetivo es detectar problemas en producción lo más rápido posible. Es el momento temprano en que el usuario interactúa con la aplicación, revisa sus requerimientos y se puede volver atrás en el desarrollo. El Despliegue Continuo exige una configuración del ambiente de trabajo, que permita un funcionamiento efectivo de las versiones candidatas por parte de los usuarios. Se empieza con una pre-configuración durante todo el proceso de desarrollo y una configuración final

antes que se termine la versión candidata. La entrega continua requiere de una aprobación manual previo a su paso a producción; en el despliegue continuo el paso a producción se realiza de forma automática cuando se han satisfecho todos los criterios definidos. El Despliegue Continuo visualiza tempranamente los fallos que puedan existir en la aplicación. Realizando Despliegues Continuos, se logra entregar productos al usuario que pueden ser diarios o semanales, y así analizar fallos que van a ser detectados en forma temprana, debido al poco código que hay que revisar [27].

- **Soporte al flujo de trabajo.**

  Además de las herramientas en las categorías descritas anteriormente, existe un gran número de herramientas que facilitan la comunicación, colaboración y calidad a lo largo del ciclo de desarrollo de software y que por lo tanto también se pueden considerar como parte de la "caja de herramientas para DevOps". Ejemplos de esto son herramientas de control de versiones, pruebas (en sus distintas variantes), análisis de código, gestión de issues, e incluso comunicación entre miembros del equipo [28].

  Por lo tanto, podemos resumir que un DevOps es un profesional IT cuyo objetivo es ayudar al desarrollo y producción de productos creando un puente entre el departamento encargado del desarrollo y de las operaciones**.**

4.8. Metodologías de desarrollo ágil

4.8.1. Metodología de Desarrollo XP

XP es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico [29].

Razones principales para el uso de la metodología XP [29][30]:

- Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software.
- Diseño sencillo: Solo se efectúa el diseño necesario para cumplir con los requerimientos actuales, es decir, no se abordan requerimientos futuros.
- Entregas pequeñas: Se desarrolla primero la más mínima parte útil que le proporcione funcionalidad al sistema, y poco a poco se efectúan incrementos que añaden funcionalidad a la primera entrega, cada ciclo termina con una entrega del sistema.
- Promueve el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.
- Se basa en realimentación continua entre el cliente y el equipo de desarrollo.
- Proporciona una comunicación fluida entre todos los participantes.
- Ofrece simplicidad en las soluciones implementadas y coraje para enfrentar los cambios.

4.8.2. Fases de la metodología XP

La metodología XP consta de 4 fases las cuales son [31]:

4.8.2.1. Planeación

- La Metodología XP plantea la planificación como un diálogo continuo entre las partes involucradas en el proyecto, incluyendo al cliente, a los programadores y a los coordinadores. El proyecto comienza recopilando las historias de usuarios, las que constituyen a los tradicionales casos de uso. Una vez

obtenidas estas historias de usuarios, los programadores evalúan rápidamente el tiempo de desarrollo de cada una. Los Conceptos básicos de la planificación son:

- **Las Historias de Usuarios**, las cuales son descritas por el cliente, en su propio lenguaje, como descripciones cortas de lo que el sistema debe realizar.
- **Roles,** personal involucradro en el desarrollo del proyecto.
- **El Plan de Entregas (Release Plan)**, establece que las historias de usuarios serán agrupadas para conformar una entrega y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto.
- **Plan de Iteraciones (Iteration Plan)**, las historias de usuarios seleccionadas para cada entrega son desarrolladas y probadas en un ciclo de iteración, de acuerdo al orden preestablecido.
- **Reuniones Diarias de Seguimiento (Stand – Up Meeting)**, el objetivo es mantener la comunicación entre el equipo y compartir problemas y soluciones.
- Velocidad del proyecto, tiempo empleado en el desarrollo de cada interación.

4.8.2.2. Diseño

- La Metodología XP hace especial énfasis en los diseños simples y claros. Los conceptos más importantes de diseño en esta metodología son los siguientes: **Simplicidad**, Un diseño simple se implementa más rápidamente que uno complejo. Por ello XP propone implementar el diseño más simple posible que funcione.
- **Diagramas de secuencia,** diagrama usado para modelar la interacción entre objetos en un sistema.
- **Diagama de clases,** diagrama que permite describir la estructura de un sistema mostrando sus clases, atributos, operaciones y relaciones entre objetos.
- **Herramientas de desarrollo,** se describen las herramientas utilizadas para el desarrollo del sistema.
- **Soluciones "Spike"**, Cuando aparecen problemas técnicos, o cuando es difícil de estimar el tiempo para implementar una historia de usuario, pueden utilizarse pequeños programas de prueba (llamados "Spike"), para explorar diferentes soluciones.
- **Recodificación ("Refactoring")**, Consiste en escribir nuevamente parte del código de un programa, sin cambiar su funcionalidad, a los efectos de crearlo

más simple, conciso y entendible. Las metodologías de XP sugieren re codificar cada vez que sea necesario.

- **Metáforas**, XP sugiere utilizar este concepto como una manera sencilla de explicar el propósito del proyecto, así como guiar la estructura del mismo. Una buena metáfora debe ser fácil de comprender para el cliente y a su vez debe tener suficiente contenido como para que sirva de guía a la arquitectura del proyecto.

4.8.2.3. Codificación

- **Disponibilidad del Cliente**, Uno de los requerimientos de XP es tener al cliente disponible durante todo el proyecto. No solamente como apoyo a los desarrolladores, sino formando parte del grupo. El Involucramiento del cliente es fundamental para que pueda desarrollarse un proyecto con la metodología XP. Al comienzo del proyecto, el este debe proporcionar las historias de usuarios. Pero, dado que estas historias son expresamente cortas y de "alto nivel", no contienen los detalles necesarios para realizar el desarrollo del código. Estos detalles deben ser proporcionados por el cliente, y discutidos con los desarrolladores, durante la etapa de desarrollo.
- **Uso de Estándares**, XP promueve la programación basada en estándares, de manera que sea fácilmente entendible por todo el equipo, y que facilite la re codificación.
- **Programación Dirigida por las Pruebas ("Test-Driven Programming")**,En las metodologías tradicionales, la fase de pruebas, incluyendo la definición de los test, es usualmente realizada sobre el final del proyecto, o el final del desarrollo de cada módulo. La metodología XP propone un modelo inverso, primero se escribe los test que el sistema debe pasar. Luego, el desarrollo debe ser el mínimo necesario para pasar las pruebas previamente definidas. Las pruebas a los que se refiere esta práctica, son las pruebas unitarias, realizados por los desarrolladores. La definición de estos test al comienzo, condiciona o "dirige" el desarrollo.
- **Programación en Pares**, XP propone que se desarrolle en pares de programadores, ambos trabajando juntos en un mismo ordenador. Si bien parece que ésta práctica duplica el tiempo asignado al proyecto (y por ende, los costos en recursos humanos), al trabajar en pares se minimizan los errores y se logran mejores diseños, compensando la inversión en horas. El producto obtenido es por lo general de mejor calidad que cuando el desarrollo se realiza por programadores individuales.

- **Integraciones Permanentes**, Todos los desarrolladores necesitan trabajar siempre con la “última versión”. Realizar cambios o mejoras sobre versiones antiguas causan graves problemas, y retrasan al proyecto. Es por eso que XP promueve publicar lo antes posible las nuevas versiones, aunque no sean las últimas, siempre que estén libres de errores. Idealmente, todos los días deben existir nuevas versiones publicadas. Para evitar errores, solo una pareja de desarrolladores puede integrar su código a la vez.
- **Propiedad Colectiva del Código**, En un proyecto XP, todo el equipo puede contribuir con nuevas ideas que apliquen a cualquier parte del proyecto. Asimismo, una pareja de programadores puede cambiar el código que sea necesario para corregir problemas, agregar funciones o re codificar.
- **Pruebas Unitarias**, Todos los módulos deben de pasar las pruebas unitarias antes de ser liberados o publicados. Por otra parte, como se mencionó anteriormente, las pruebas deben ser definidas antes de realizar el código (“Test-Driven Programmming”). Que todo código liberado pase correctamente las pruebas unitarias, es lo que habilita que funcione la propiedad colectiva del código.

4.8.2.4. Pruebas

- **Detección y Corrección de Errores**, Cuando se encuentra un error (“Bug”), éste debe ser corregido inmediatamente, y se deben tener precauciones para que errores similares no vuelvan a ocurrir. Asimismo, se generan nuevas pruebas para verificar que el error haya sido resuelto.
- **Pruebas de Aceptación**, Son creadas en base a las historias de usuarios, en cada ciclo de la iteración del desarrollo. El Cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Asimismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta que pase correctamente todas las pruebas de aceptación.

4.9. Trabajos Relacionados

La TABLA V, nos presentan los mecanismos y herramientas a utilizar para el funcionamiento del presente TT, en base a un conjunto de artículos científicos.

TABLA V.
TRABAJOS RELACIONADOS PARA EL PRESENTE TT.

Nro	Título	Resultados
[32]	A1. The Research and Design of Unified Authentication System Based on CAS	• Jasig CAS • SSO • Protocolo LDAP • Https
[33]	A2. Towards Scalability for Federated Identity Systems for Cloud-Based Environments	• Jasig CAS • SSO • Protocolo LDAP • Https
[34]	A3. Implementasi Sistem Single Sign On / Single Sign Out Berbasis Central Authentication Service Protocol Pada Jaringan Lightweight Directory Access Protocol Universitas Diponegoro	• Jasig CAS • SSO • Protocolo LDAP • Https
[35]	A4. Analisis Teknologi Single Sign On (Sso) Dengan Penerapan Central Authentication Service (Cas) Pada Universitas Bina Darma	• Jasig CAS • SSO • Protocolo LDAP • Https
[36]	A5. Sistem Single Sign On Universitas Berbasis Cas-Ldap.	• Jasig CAS • SSO • Protocolo LDAP • Https
[37]	A6. Sistema Integrado de Autenticación para la Universidad Tecnológica de Bolívar- MiUTB.	• Jasig CAS • SSO • Protocolo LDAP • Https
[8]	A7. Central Authentication Service (CAS) SSO For EMC® Documentum® Rest Services.	• Jasig CAS • SSO • Protocolo LDAP • Https
[38]	A8. Linux PAM to LDAP Authentication Migration.	• Biblioteca adicional PHP para LDAP
[39]	A9. Intelligent agents applied to the management of ldap user profiles.	• Biblioteca adicional PHP para LDAP
[17]	A10. User Management with LDAP (Lightweight Directory Access Protocol) for access to technology and Information Services in Companies.	• Biblioteca adicional PHP para LDAP

5. Materiales y Métodos

Para llevar a cabo correctamente el desarrollo del TT, en la fase de análisis se utilizó el entorno de la Universidad Nacional de Loja - UNL, donde se realizó dos SLR, una de ellas derivando en un artículo científico que se dio a conocer en un entorno académico como lo es la Escuela Superior Politécnica de Chimborazo (ESPOCH), en la fase de diseño, prototipado y pruebas se utilizó un ambiente simulado local, en donde se implementó el prototipo propuesto para su correcta validación y funcionamiento, posteriormente se aplicó en el entorno de la Unidad de Telecomunicaciones e Información - UTI, los cuales son los encargados de la parte administrativa y técnica de la UNL, derivando en la implantación del prototipo propuesto en el TT. Como caso de estudio se realizó el experimento del TT, en toda la comunidad universitaria de la UNL.

Para el desarrollo del Servicio de Autenticación Único de Usuarios en Aplicaciones Web, se estableció una metodología de desarrollo adaptable a la naturaleza del presente TT, en la cual se incorporan varias áreas de estudio: sistemas de información, ingeniería de software, fundamentos informáticos, arquitectura de computadores, programación y análisis y diseño de sistemas. la misma que se define en 3 fases, correspondientes a cada uno de los objetivos:

5.1. Fase 1: Análisis

En esta fase, se realizó una SLR para el análisis del protocolo CAS para una autenticación única y centralizada y una SLR para analizar el protocolo LDAP como mecanismo centralizado, esta última derivando en la publicación de un artículo científico “Revisión Sistemática de Literatura sobre el protocolo LDAP como mecanismo centralizado para la autentificación de usuarios en múltiples sistemas” [40], en la plataforma Knowledge E, que se lo puede visualizar ingresando al siguiente link “https://knepublishing.com/”. Mediante las SLR identificamos y evaluamos todas las investigaciones disponibles acerca de las preguntas de investigación, que determinaron las características principales de cada uno de los protocolos a utilizar y las cuales se deben tener en cuenta para el desarrollo del TT, se las puede evidenciar completamente en la (Objetivo 1: Analizar la autenticación del protocolo CAS, como mecanismo centralizado).

5.1.1. Revisión Sistemática

Para fundamentar el desarrollo del presente TT se sigue el esquema propuesto de acuerdo al artículo de Bárbara kitchenham, que propone un método para la realización de revisiones sistemáticas en el contexto de la Ingeniería de Software [1]. A continuación, en la TABLA VI se puede observar las principales etapas y actividades utilizadas en las revisiones sistemáticas y su desarrollo completo se encuentran en el (Anexo 5. SLR del Protocolo CAS) y (Anexo 6. Artículo indexado en Knowledge E).

TABLA VI.
ESQUEMA UTILIZADO PARA LAS SLR.

Planificación de la Revisión Sistemática de Literatura	1. Objetivo de la Revisión Sistemática de Literatura
	2. Formulación de la pregunta de investigación
	3. Palabras Claves
	4. Selección de fuentes y estrategias de búsqueda
	5. Cadena de búsqueda
	6. Criterios de inclusión
	7. Criterios de exclusión
Ejecución de la Revisión Sistemática de literatura	1. Criterios de selección de estudios
	2. Extracción de la información
Análisis de resultados y hallazgos	1. Hallazgos

5.2. Fase 2: Diseño

En esta fase, se estableció la estructura jerárquica del directorio centralizado en la cual se definen las clases y atributos que va a tener cada grupo, subgrupo y usuarios, que nos sirvió de modelo para su implementación en el servidor OpenLDAP, así mismo se desarrolló un servicio Web con funciones administrativas y de autenticación, utilizando el lenguaje de programación PHP y el API que ofrece el protocolo CAS para aplicaciones o sistemas web que no tienen soporte directo con el protocolo antes mencionado. Garantizando el correcto funcionamiento del mismo, se desarrolló SAC, en el cual se utilizó la metodología XP, que promueve el trabajo en equipo, brindando un buen ambiente basado en la comunicación fluida entre el cliente y el equipo de desarrollo, donde se definen 4 subfases:

- **Planeación**

  En esta subfases, se definieron las historias de usuario que van a describir las características y funcionalidades del servicio de administración.
- **Diseño**

  En esta subfases, se estableció los diseños los cuales deben ser simples y sencillos para facilitar el desarrollo.
- **Codificación**

  En esta subfases, se diseñó y desarrollo las pruebas de unidad que ejecutará cada historia de usuario.
- **Pruebas**

  En esta subfases, se implementó un marco de trabajo que permite automatizar las pruebas de integración, validación diaria y aceptación final, esto proporcionará al equipo un indicador del progreso y revelaran a tiempo si existe alguna falla en el sistema.

Para la autenticación con el protocolo CAS, se implementó y configuro las funciones de autenticación única, utilizando el servidor OpenLDAP y el Sistema Jasig CAS personalizando su interfaz principal.

Finalizando la etapa de diseño, se integró el servicio Web, SAC y el Sistema Jasig CAS, permitiendo al usuario acceder a distintas aplicaciones Web, mediante una autenticación única, la cual se desarrolla en la (Objetivo 2: Diseñar y desarrollar un prototipo de servicio de autenticación central, para múltiples aplicaciones Web.).

5.3. Fase 3: Prototipado y Pruebas

En esta fase, se hizo uso del enfoque DevOps, la cual facilita la comunicación entre la parte de desarrollo y administración de operaciones en cada sistema, de igual manera se utilizó el ciclo DevOps, por su facilidad en el flujo de trabajo, este ciclo cuenta de 6 subfases [41]:

Requisitos

Se empleó un método para la recolección de la información, denominado “Revisión de Registros”, donde se establecieron las características principales que debe incorporar cada sistema y un método técnico para la selección de los sistemas, denominado “Evaluación del desempeño”.

Desarrollo

Se hizo uso de todos los sistemas, servidores y protocolos definidos en la fase anterior.

Construcción

Se definió la interfaz principal del Sistema de Gestión Único CAS (SiGUCAS) permitiendo la iteración individual y grupal de los sistemas, servidores y protocolos, para verificar la interoperabilidad entre las diferentes aplicaciones Web.

Pruebas

Se definió las características principales, el enfoque de pruebas; funcional y basado en escenarios, los criterios de aprobación, las necesidades ambientales, responsabilidades, resultados, tareas y aprobaciones.

Despliegue

Se desplegó el presente TT en la UTI de la UNL, realizando un control de versiones en cada despliegue realizado, dando como resultado la implantación del mismo.

Monitoreo

Se realizó un control de SiGUCAS y SAC para toda la comunidad universitaria de UNL, mediante el uso de correos electrónicos, redes sociales y demás medios digitales.

5.4. Recursos

5.4.1. Bibliográficos

- Revisión de Literatura: mediante un análisis bibliográfico, permitió obtener información relevante del problema y sustentar la parte teórica del TT, como se desarrolló en nuestra Revisión de Literatura.
- SLR: se utilizó el esquema de Bárbara kitchenham, que nos facilitó las características principales para la implementación del prototipo propuesto en el TT, la cual se la utilizó en la fase 1.
- Revisión de registros: nos dio lugar a la examinación y extracción de información de las aplicaciones Web, obteniendo las características principales, haciendo uso de la misma en la fase 3.

5.4.2. Técnicos

- Observación: mediante reuniones, se apreció de mejor manera los problemas que se encuentran y darnos una idea clara y concreta de los requerimientos del sistema, y así realizar una transferencia de conocimientos.

- Metodología XP: nos guio en el desarrollo del sistema de administración para el servidor OpenLDAP, la cual se realizó en la fase 2.
- Encuestas: se realizaron encuesta de tipo físicas; donde se usó preguntas de funcionalidad y aceptación para SiGUCAS y SAC, las cuales se realizaron en la fase 3.
- Evaluación del desempeño: Nos permitió la observación de las características principales, de un conjunto de aplicaciones Web y así hacer un juicio justo sobre su calidad para la correcta integración con el Protocolo CAS, la misma que se utilizó en la fase 3.
- Ciclo DevOps: Nos permitió la correcta selección de las aplicaciones Web, la definición del ambiente de pruebas y resultados mediante su ciclo de 6 subfases. Así mismo la administración de perfiles o roles de usuario por su filosofía de Desarrollo y Operaciones.
- Lucidchart: Herramienta de diagramación basada en la Web, nos permitió la creación de: diagramas de flujo, organigramas, esquemas de sitios Web, diseños Unified Modeling Language (UML - en español, Lenguaje Unificado de Modelado), prototipos de software y otros diagramas en tiempo real, la cual se utilizó en la fase 2.

5.5. Participantes

Se involucró al personal administrativo y técnico que laboran en la UTI de la UNL, en el periodo octubre – marzo del 2019, y al director del presente TT (Anexo 1. Personal involucrado) y (Anexo 2. Listado de participantes para las pruebas del presente TT.). La **TABLA VII**, describe el personal involucrado por cada uno de los departamentos.

TABLA VII.
PERSONAL INVOLUCRADO

Cargo	Cantidad	Rol
Director General de la UTI.	1	Administrador
Subdirector de Redes y Equipos Informáticos	1	Usuario Común
Subdirector de Desarrollo de Software	1	Usuario Común
Analistas de Sistemas Informáticos 2	3	Usuario Común
Director del TT.	1	Usuario Común
Supervisores del TT.	2	Desarrollador

La comunidad universitaria de la UNL, participó en los casos de estudio:

- Servicio de Administración Central (SAC)
- Sistema de Gestión Único CAS (SiGUCAS)

Para los cuales se realizó su implantación en la UNL (Anexo 3. Certificado de implantación del Sistema de Administración Central (SAC).) y (Anexo 4. Certificado de implantación del Sistema de Gestión Único CAS (SiGUCAS)).

6. Resultados

En este apartado se plasma el desarrollo de un prototipo de autenticación central para usuarios en aplicaciones Web, el mismo que se llevó a cabo de acuerdo a los objetivos planteados estableciendo una fase de desarrollo por cada uno de ellos.
En el objetivo 1, se realiza un análisis sobre el protocolo CAS y LDAP, mediante dos SLR, para determinar los requerimientos de implementación de cada uno de los protocolos antes mencionados.

En el objetivo 2, se realiza el diseño y desarrollo de un prototipo de servicio de autenticación, desarrollando una estructura jerárquica para el acceso a la información, la configuración de un servidor OpenLDAP y un servicio Web con funciones administrativas y de autenticación que van a permitir controlar la información de los usuarios, mediante la interfaz principal del Sistema Jasig CAS.
En el objetivo 3, se realiza un escenario de pruebas que permitan comprobar el funcionamiento del prototipo propuesto en el presente TT.

6.1. Objetivo 1: Analizar la autenticación del protocolo CAS, como mecanismo centralizado

6.1.1. Análisis del protocolo CAS, para una autenticación única y centralizada.

Para el análisis del protocolo CAS, se realizó una SLR, donde se analizó 10 artículos científicos para poder determinar la cadena de búsqueda y un análisis de criterios de exclusión e inclusión de diferentes estudios publicados en dos bases de datos científicas (IEEEXplorer y Scientific.net), un buscador académico (Google Scholar) y otros buscadores Web. Obteniendo un resultado de 16 artículos científicos con conclusiones relevantes, que se describen a continuación (Anexo 5. SLR del Protocolo CAS):

- El protocolo CAS, es una solución libre, robusta y probada para escenarios de autenticación centralizada, que suscribe varias aplicaciones compartiendo un formulario de autenticación común.
- Incorpora un mecanismo denominado SSO, que garantiza que un usuario ingrese sus credenciales (Usuario y contraseña), una sola vez.

- Utiliza HTTPS, método para garantizar una comunicación segura entre el navegador de un usuario y un servidor Web. A menudo se reconoce por una barra de direcciones verde o un candado en la ventana del navegador, que indica que la conexión es segura.
- Utiliza un mecanismo de Tickets basado en cookies para guardar las credenciales de un usuario que, a iniciado sesión con éxito, y que estará visible para las aplicaciones o sistemas a las que el usuario tiene acceso.
- Tiene un soporte de biblioteca cliente amplio para Java, PHP, Perl, Apache, uPortal, y otros.
- Se puede integrar con otros esquemas de autenticación como: LDAP, Kerberos SSO, SAML SSO, etc.
- Jasig CAS, es la interfaz principal para el acceso a múltiples sistemas mediante una autenticación única, siendo esta la conclusión más relevante y para lo cuál se hace uso para el presente TT.

6.1.2. Analizar el protocolo LDAP, como directorio centralizado.

Para el análisis del protocolo LDAP, se uso una SLR, donde se analizó 10 artículos científicos para poder determinar la cadena de búsqueda y un análisis de criterios de exclusión e inclusión de diferentes estudios publicados en cuatro bases de datos científicas (Scopus, IEEEXplorer, Scientific.net, DBLP), un buscador académico (Google Scholar) y dos revistas académicas (Revista Energía de la UNL, Revista científica de la UTB). Obteniendo un resultado de 14 artículos científicos con conclusiones relevantes, que se describen a continuación (Anexo 6. Artículo indexado en Knowledge E):

- Permite la creación y eliminación de grupos de usuarios.
- Almacena y gestiona datos de usuarios asignados como: identidades de usuario, contraseñas, datos personales, etc.
- Funcionalidad para restablecer contraseñas.
- Mejora la seguridad de la información, al proporcionar distintos algoritmos de cifrado.
- Multiplataforma (Puede ejecutarse en varios sistemas operativos).
- Permite agregar, modificar y borrar información junto con operaciones de búsqueda.
- Mejora la administración y almacenamiento de la información de usuarios, por su estructura jerárquica y su optimización de lectura rápida.

- Cuenta con una biblioteca adicional de PHP para la correcta conexión a LDAP, siendo esta la conclusión más sobresaliente por sus funciones de gestión para la administración de servidores OpenLDAP, la cual se hace uso para el presente TT.

El algoritmo de cifrado SHA, es utilizado para el cifrado de contraseñas en el servidor OpenLDAP [38] se usó el algoritmo antes mencionado para el desarrollo del TT, por la recomendación brindada en la Guía de Administración OpenLDAP [13] y un conjunto de reuniones o socializaciones con personal de la UTI, donde se desarrolló la implantación del TT (Anexo 23. Plan de Trabajo y Cambios para el Sistema SAC con personal de la UTI.).

6.2. Objetivo 2: Diseñar y desarrollar un prototipo de servicio de autenticación central, para múltiples aplicaciones Web.

6.2.1. Diseño de un árbol jerárquico en base a una estructura ordenada, para el acceso a múltiples aplicaciones Web.

Para el diseño de un árbol jerárquico que permita el acceso a múltiples sistemas Web, se lo realizó en base a una estructura ordenada; donde se creó 2 subgrupos principales, el primer grupo denominado **"personal"**, basado en la LOSEP, LOES y el Código de trabajo del Ecuador [42], [43], [44], en el cual ya se define los perfiles que manejará cada aplicación Web para una entidad de educación superior, siendo este donde se realizó la autenticación única y centralizada de los usuarios para múltiples aplicaciones Web, como podemos evidenciar en la Figura 7 y el segundo grupo denominado **"universidad_implementación"**, el cual se lo puede adaptar a la entidad de educación superior u otras organizaciones que deseen implementar el prototipo propuesto, indicado en la Figura 8.

Se estructura el servidor OpenLDAP basado en un diseño jerárquico y ordenado, el cual nos sirvió para una lectura de información más rápida y eficiente.

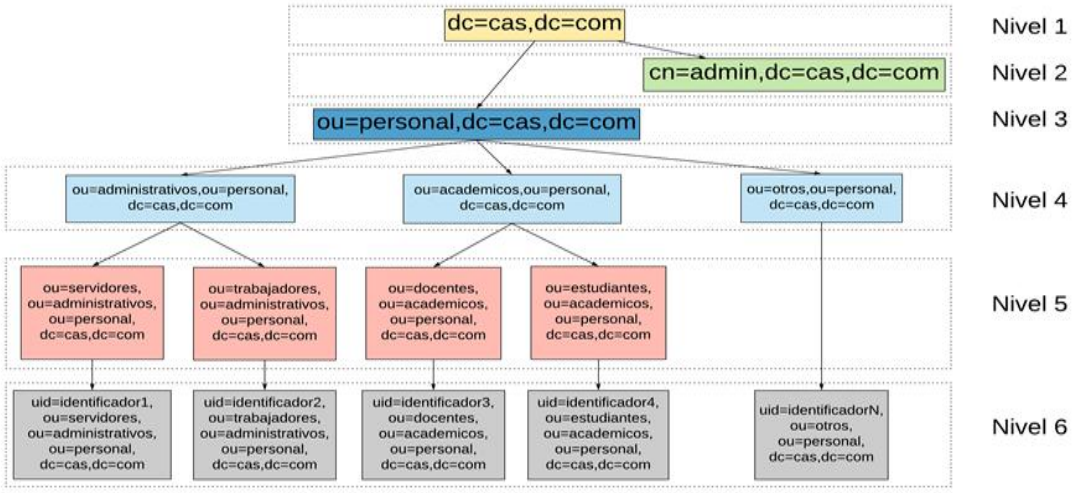


Figura 7. Diseño de la unidad organizacional "personal" y sus niveles.

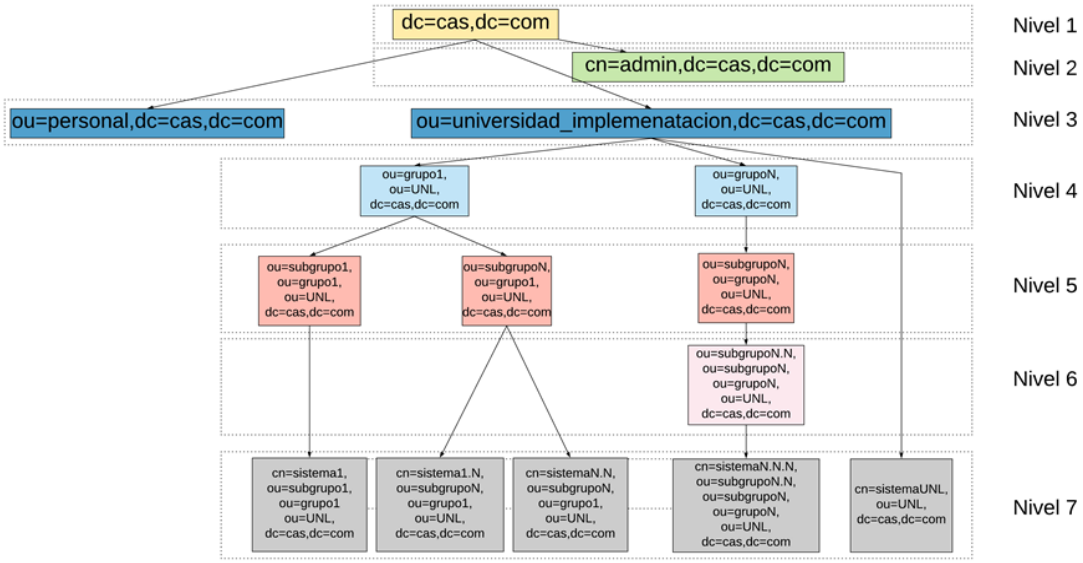


Figura 8. Diseño unidad organizacional "universidad_implementacion" y sus niveles.

La Figura 7, establece una estructura de 6 niveles:

- **Nivel 1:** dominio del servidor OpenLDAP
- **Nivel 2:** dominio del administrador del servidor OpenLDAP
- **Nivel 3:** unidades organizacionales principales (personal y universidad_implementación).
- **Nivel 4:** sub-unidades organizacionales de grupo personal (académicos, administrativos y otros).
- **Nivel 5:** sub-unidades organizacionales del nivel 4 (trabajadores, servidores, docentes y estudiantes).

- **Nivel 6:** información de los usuarios pertenecientes a las sub-unidades organizacionales (trabajadores, servidores, docentes, estudiantes y otros).
-

La Figura 8, establece una estructura de 7 niveles, en la cual los 3 primeros niveles son los mismos definidos anteriormente:

- **Nivel 4:** sub-unidades organizacionales de universidad_implementación que la entidad desee implementar.
- **Nivel 5:** sub-unidades organizacionales del nivel 4 de universidad_implementación que la entidad desee implementar.
- **Nivel 6:** sub-unidades organizacionales del nivel 5 de universidad_implementación que la entidad desee implementar.
- **Nivel 7:** usuarios vinculados a una de las sub-unidades organizacionales del nivel 5 o nivel 6 de universidad_implementación que la entidad desee implementar.

En estos niveles se definen las clases, atributos y campos obligatorios que deben incorporarse en cada uno de ellos, para el correcto funcionamiento del servidor OpenLDAP (Anexo 8. Niveles del árbol Jerárquico para el presente TT) y que se encuentra implementada en la UNL (Anexo 9. Certificado de implantación del diseño jerárquico del servidor OpenLDAP para la autenticación de diferentes sistemas de la UNL).

6.2.2. Configuración del servidor OpenLDAP

Para la configuración e instalación del servidor OpenLDAP, es necesarios tener instalado un sistema operativo basado en GNU/LINUX también conocido como Linux, en el presente TT se utilizó el Sistema Operativo Ubuntu 18.04 LTS y la instalación del servidor antes mencionado se la puede evidenciar en el (Anexo 10. Configuración del servidor OpenLDAP).

En el servidor implementado en el punto anterior, se creó la estructura propuesta en la Figura 7 y Figura 8, donde se usó el navegador multiplataforma JXplorer, cuya instalación se encuentra en el (Anexo 11. Instalación JXplorer) y que se puede evidenciar en la Figura 9.

Figura 9. Estructura jerárquica en el servidor OpenLDAP.

En la Figura 9 se enmarca confidencial debido a que dicha estructura fue implantada en la UNL y la información no puede ser divulgada.

6.2.3. Desarrollo de un Web service, con métodos administrativos y de autenticación para el servidor CAS.

6.2.3.1. Desarrollo del Web service y el sistema de administración SAC

Para el desarrollo del Web service se utilizó el protocolo de intercambio para servicios Web basado en XML SOAP, el lenguaje de descripción de servicios Web WSDL para definir los mensajes creados y para su implementación se hace uso de la librería NuSOAP la cual nos permitió crear el servicio Web en PHP, tomando en cuenta todos los hallazgos encontrados en el punto (Analizar el protocolo LDAP, como directorio centralizado.), la implementación del servicio Web con sus funciones administrativas y de autenticación para el servidor OpenLDAP se encuentra en el (Anexo 12. Desarrollo de un Web Service con métodos administrativos y de autenticación).

El desarrollo del servicio Web se lo realizó utilizando SOAP que es un formato de mensaje XML utilizado en interacciones de servicios Web. Los mensajes SOAP se enviarán sobre HTTP que se describe mediante la definición WSDL, en la Figura 10 se puede observar el funcionamiento general del servicio Web desarrollado para el presente TT y las principales funciones de autenticación y administración para el servidor OpenLDAP.

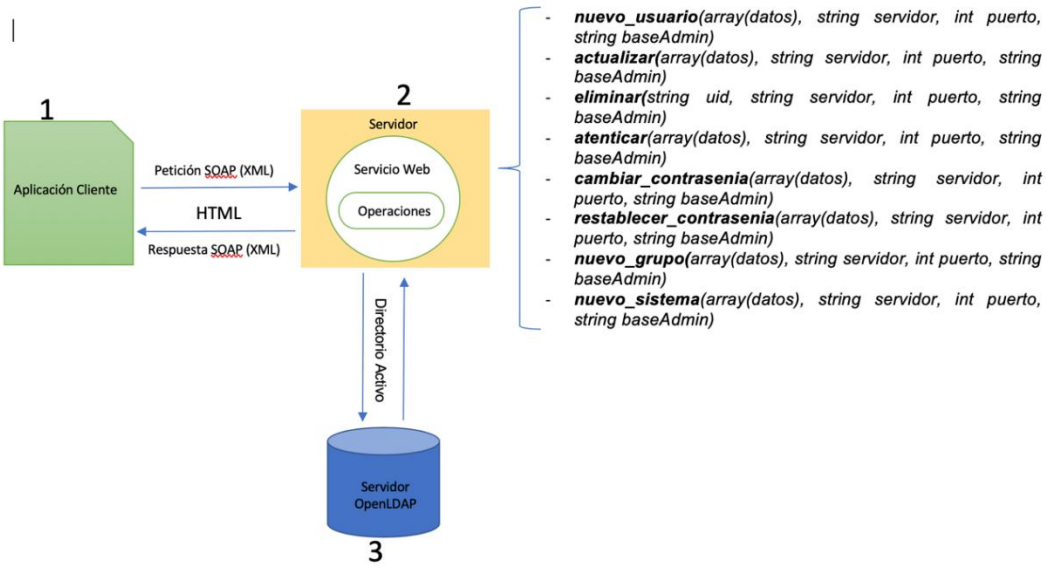


Figura 10. Esquema de Funcionamiento del Servico Web con el Protocolo SOAP.

Para comprobar el correcto funcionamiento e integración del servicio Web se desarrolló un sistema Web denominado SAC, haciendo uso de la metodología XP, la cual está definida en el (Anexo 13. Metodología XP para el desarrollo del sistema de administración SAC.) y define 4 subfases que se deben abordar y se describen a continuación:

- **Planeación**

  En este punto se definieron un total de 23 historias de usuario, la cuales se las dividió en 4 iteraciones cada una de ellas con un producto funcional obteniendo un total de 76 tareas a desarrollarse, también se definieron los roles de los participantes que van a intervenir en el desarrollo de SAC, la velocidad con la que se va a ejecutar cada iteración el plan y el cronograma de entregas.

- **Diseño**

  En este punto se definió un diagrama simple que permitió orientar al cliente como va a quedar la vista general de SAC, se diseñaron 26 diagramas de secuencia uno por cada historia de usuario para indicar las interacciones entre los objetos y la línea de vida de cada uno de ellos, también se definió el diagrama de clases que se usó como base para la creación de los módulos de SAC y el diagrama de procesos para especificar el orden de los componentes y las dependencias entre ellos, para finalizar se definieron las herramientas y tecnologías con las cuales se desarrolló SAC.

- **Desarrollo**

  En este punto primero se definieron las reglas de programación a seguir para la correcta codificación de SAC, el desarrollo se lo hizo en parejas por parte de los autores del TT para obtener un código de mejor calidad y más organizado, siempre teniendo una buena comunicación con el cliente. Se llevaron a cabo las pruebas unitarias para comprobar la correcta ejecución de cada historia de usuario y poder integrarlas en el menor tiempo posible y en los límites de tiempo establecidos.

- **Pruebas**

  Para comprobar el correcto funcionamiento de SAC, se lo desplegó en un servidor local y se procedió a realizar las pruebas formales de aceptación que determinaron que el cliente estaba conforme con la aplicación desarrollada.

La Figura 11, nos indica la interfaz de acceso para el administrador del SAC la cual es la encargada de gestionar la información almacenada de los usuarios en el servidor OpenLDAP, si el acceso se lo hace mediante el perfil de administrador y si el acceso es mediante el perfil de usuario común le permitirá gestionar su información personal, como es el cambio y recuperación de contraseña mediante la interfaz de acceso del Sistema Jasig CAS, que se muestra en la Figura 12, esto debido a la integración de la API del protocolo CAS con el lenguaje de desarrollo PHP, para el acceso único al igual que las aplicaciones Web integradas con la Autenticación única.

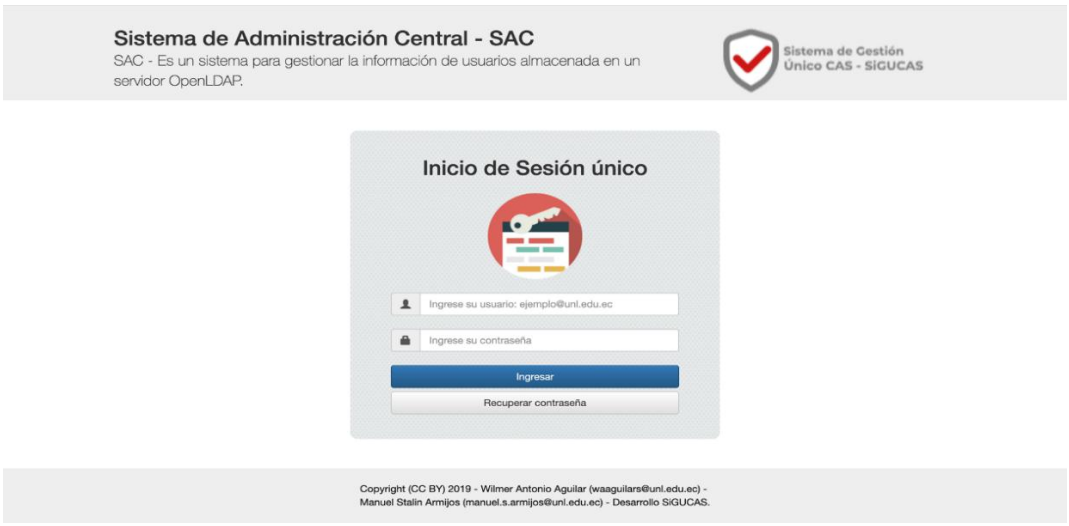


Figura 11. Interfaz de acceso del SAC

6.2.3.2. Implementación de Jasig CAS

Se realizó la configuración de los métodos de autenticación única, utilizando el sistema Jasig CAS, tomando en cuenta todos los hallazgos encontrados en el punto (Análisis del protocolo CAS, para una autenticación única y centralizada.). Jasig CAS, ofrece una interfaz de acceso, la cual se la puede modificar respetando sus políticas, por esto se realizó la personalización de la interfaz de acceso enfocada a nuestro TT. Las propiedades y archivos de configuración para el correcto funcionamiento del inicio de sesión único, se define en el (Anexo 14. Configuración y personalización del Sistema Jasig CAS.).

El Sistema Jasig CAS, nos permite hacer uso de su interfaz para el ingreso de credenciales, el cual en caso de éxito generará un ticket seguro que será compartido para todas las aplicaciones o sistemas Web que utilicen el protocolo CAS, de la misma manera se personalizo su interfaz principal en base a las necesidades de imagen del TT, respetando las políticas establecidas en su licencia, como podemos observar en la Figura 12.

Figura 12. Interfaz principal y personalizada del Sistema Jasig CAS

6.3. Objetivo 3: Evaluar el Servicio de Autenticación Central desarrollado a través de los DevOps.

6.3.1. Seleccionar las herramientas DevOps, para la integración con el Servicio de Autenticación Central.

El desarrollo de esta tarea, se la realizó en la etapa de requisitos del ciclo DevOps, la cual nos sirvió para obtener las aplicaciones Web finales que se integraron con el protocolo de autenticación CAS, tomando un total de 13 aplicaciones, de los cuales se seleccionaron 5 (Moodle, GitLab, WordPress, Drupal y Jenkins), que cumplen con las características principales, establecidas en el (Anexo 16. Ciclo DevOps).
Las aplicaciones Web que se seleccionaron, utilizan la filosofía DevOps, la cual nos permite manejar los perfiles de usuarios en cada aplicativo, evitando así conflictos de acceso a los diferentes roles que se establece en los usuarios.

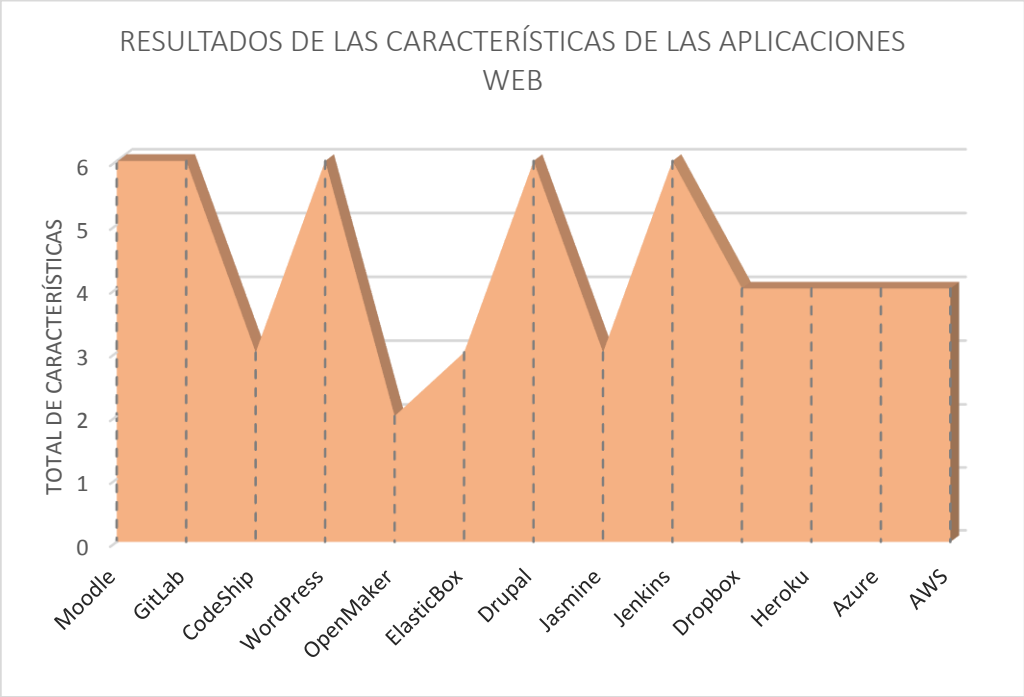


Figura 13. Resultados selección de las aplicaciones Web

La Figura 13, muestra el resultado de las aplicaciones Web que se integrarán con el Protocolo CAS, por ser las que cumplen con todas las características establecidas en esta etapa, cada una de las aplicaciones seleccionados maneja una arquitectura distinta, es por ellos que para la configuración e instalación de los sistemas con el protocolo CAS se trabajó de la siguiente manera:

- Configuración e implementación de Moodle (Anexo 17. Configuración de la Aplicación Web Moodle).
- Configuración e implementación de GitLab (Anexo 18. Configuración de la Aplicación Web GitLab).
- Configuración e implementación de WordPress (Anexo 19. Configuración de la Aplicación Web WordPress).
- Configuración e implementación de Drupal (Anexo 20. Configuración de la Aplicación Web Drupal).
- Configuración e implementación de Jenkins (Anexo 21. Configuración de la Aplicación Web Jenkins).

6.3.2. Determinar un prototipo para el ambiente de pruebas con la integración de los DevOps a un Servicio de Autenticación Central.

Para definir esta tarea, se usó la etapa de desarrollo del ciclo DevOps, la cual nos sirvió para establecer: el servidor OpenLDAP, SAC y el Sistema Jasig CAS, mismos que se utilizaron en la siguiente etapa de construcción, donde se implementó el sistema final SiGUCAS como se puede ver en la Figura 14, que integra los sistemas seleccionados en la etapa de requisitos, y los sistemas definidos en la etapa de desarrollo, permitiendo comprobar la interoperabilidad de los mismos, mediante la autenticación única y centralizada.

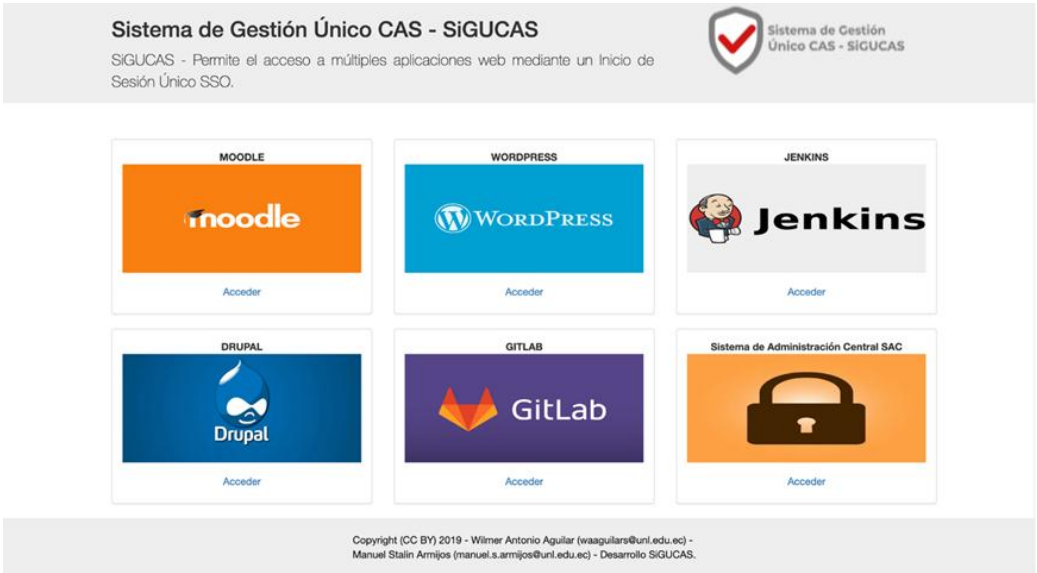


Figura 14. Interfaz principal de SiGUCAS

El prototipo propuesto para el ambiente de pruebas, nos permite tener una perspectiva clara, de la forma en que funcionará el sistema desarrollado. Incorporando el trabajo

conjunto de: SiGUCAS el cual contiene todos los sistemas que interactuaran entre sí, el sistema Jasig CAS que nos permite ingresar las credenciales de acceso mediante su interfaz principal previamente personalizada respetando sus políticas de uso, el controlador con sus métodos de autenticación, cambio y recuperación de contraseñas,el servidor OpenLDAP donde se almacena toda la información de los usuarios y el sistema que se procedió a seleccionar para su ingreso, como podemos observar en la Figura 15.

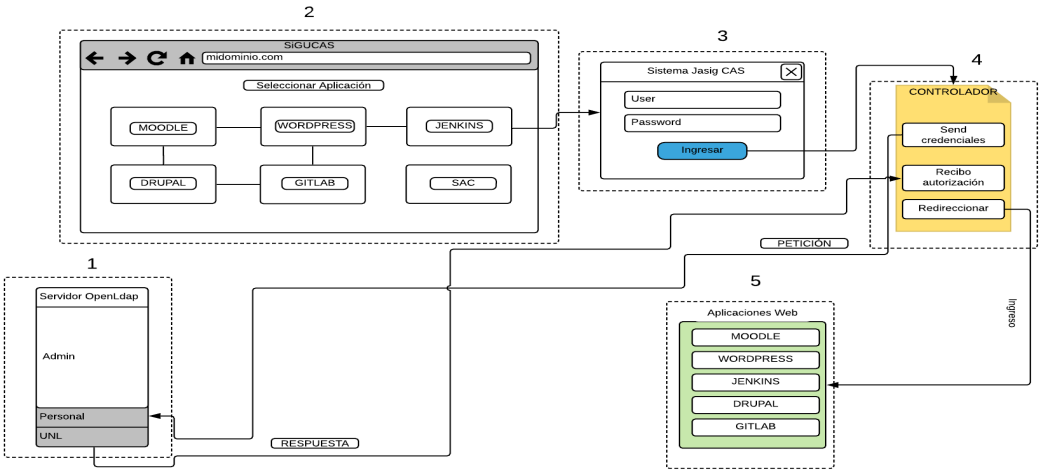


Figura 15. Prototipo propuesto para el ambiente de pruebas del TT.

6.3.3. Realizar pruebas del ambiente antes seleccionado.

Para la etapa de pruebas del ciclo DevOps se definieron una serie de puntos que se consideran importantes para comprobar las pruebas tanto funcionales como de aceptación, con la aplicación de una secuencia de 25 tareas a un conjunto de 7 participantes de la UTI de la UNL.

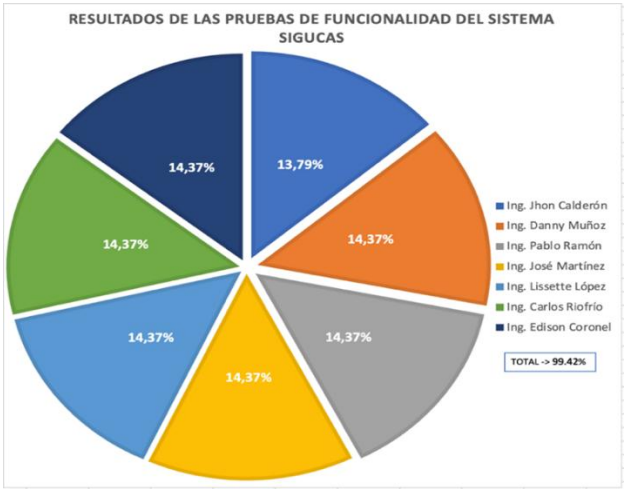


Figura 16. Resultados funcionales del sistema SiGUCAS

La Figura 16, indica los resultados obtenidos en las encuestas aplicadas a los 7 participantes, la misma muestra que el sistema SiGUCAS tiene una funcionalidad del 99,42%, lo cual quiere decir; que todos los participantes establecieron que SiGUCAS cumple las 25 tareas establecidas; donde la mayoría de ellas obtuvo una calificación máxima de 5 puntos y en una de ellas la calificación de 4 puntos por parte de uno de los usuarios.

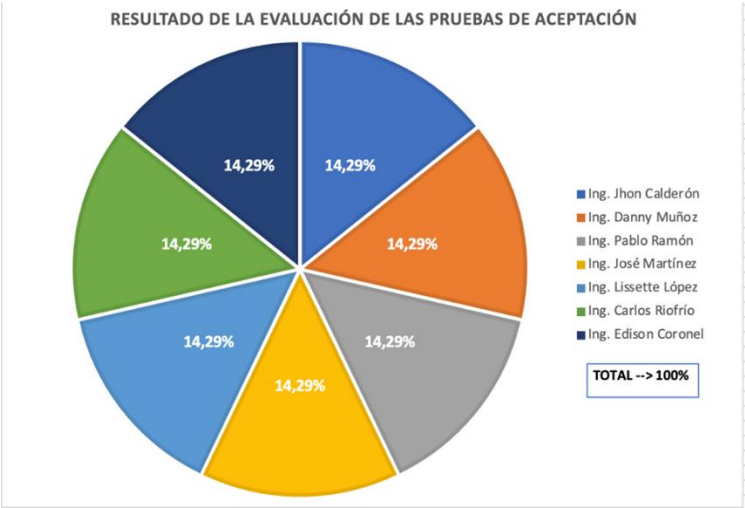


Figura 17. Resultados de la aceptación del sistema SiGUCAS.

La Figura 17, indica que de un total del 100% de participantes que equivale a 7, muestran que el sistema SiGUCAS tiene una aceptación absoluta, esto quiere decir; que todos los participantes aceptaron las tareas establecidas para el desarrollo del sistema SiGUCAS, concluyendo junto a las pruebas de funcionalidad que el sistema es apto para su implantación en cualquier entidad.

Se utilizó la etapa de despliegue del ciclo DevOps, en la cual se utilizó la UTI de la UNL para la correcta implementación del prototipo propuesto en el TT, con las condiciones técnicas y de seguridad solicitadas por esta entidad.

La Figura 18, muestra la validez, operatividad de las actividades y cambios que se desarrollaron para definir el entorno final de la implantación de SiGUCAS y SAC en la UNL.

Figura 18. Despliegue de SiGUCAS y SAC en la UNL

La Figura 19, muestra la validez, operatividad de las actividades y cambios que se desarrollaron para definir el entorno final de la implantación del Sistema Jasig CAS en la UNL.

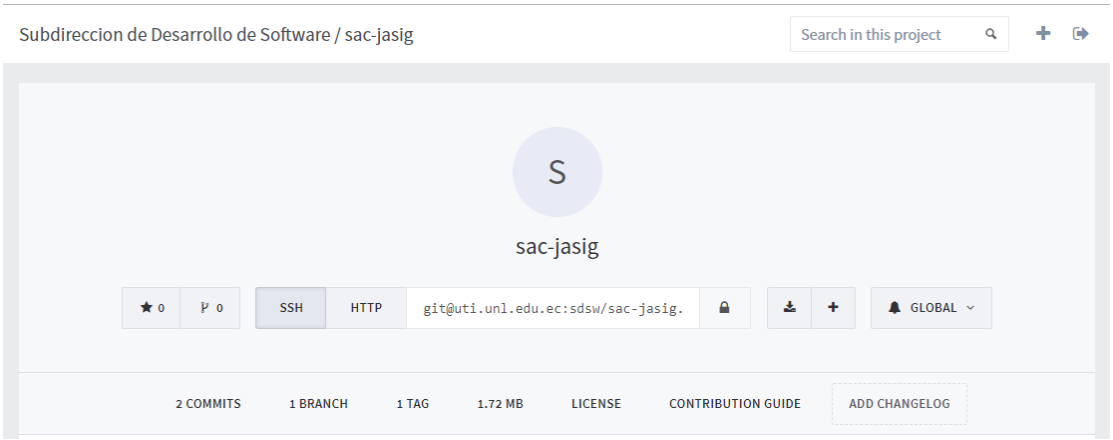


Figura 19. Despliegue del Sistema Jasig CAS en la UNL

Con la etapa de monitoreo del ciclo DevOps, nos permitió la corrección de errores presentadas por la comunidad universitaria, a través de correos electrónicos, visitas técnicas y medios de comunicación digitales, como podemos observar en la Figura 20, la cual nos indica las necesidades que se tomo en cuenta para la correcta implantación del presente TT en la UNL.

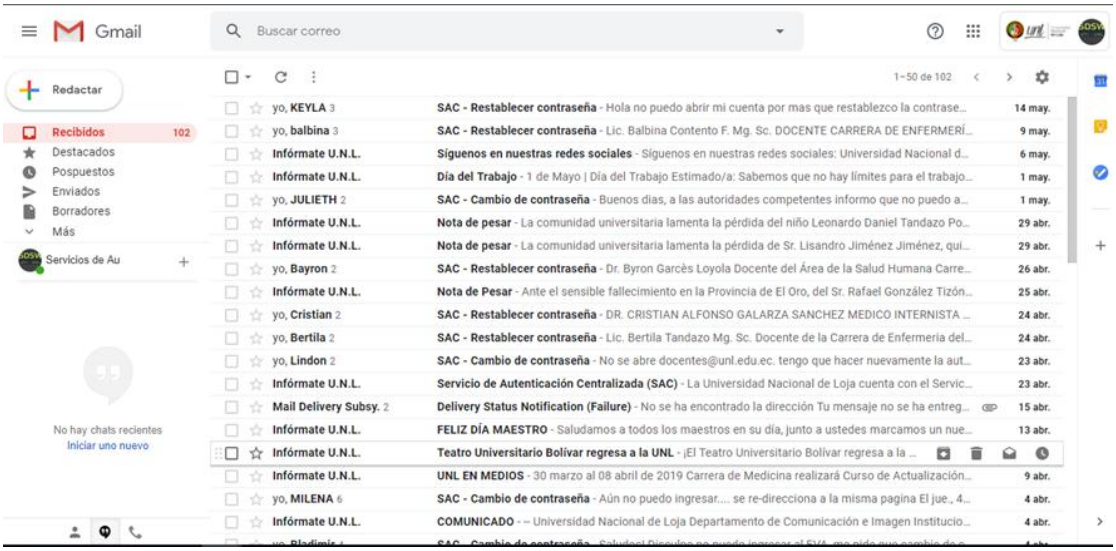


Figura 20. Monitoreo de la implantación del presente TT en la UNL.

Sugerencias encontradas

Luego de haber analizado los resultados de la encuesta realizada al personal de la UTI de la UNL, que se encuentra en el (Anexo 2. Listado de participantes para las pruebas del presente TT.), las cuales ayudaron durante las pruebas realizadas y las observaciones que hicieron cada uno de ellos a SiGUCAS, se identificaron las siguientes recomendaciones:

- Aplicar el cierre de sesión único en cada uno de los sistemas integrados con la Autenticación única mediante SiGUCAS.

Mejoras realizadas

Luego de haber identificado los problemas o necesidades de los usuarios al interactuar con el sistema, se los resolvieron de la siguiente forma:

- Se cambió el cierre de sesión por cada aplicación o sistema Web a un cierre de sesión único para todas las aplicaciones que se seleccionaron, tomando en cuenta la arquitectura de cada una de ellas y que permiten utilizar esta opción, como se puede evidenciar en el (Anexo 22. Configuración del Cierre de Sesión Único).

El protocolo CAS, envía un aviso indicando que el ticket generado al iniciar sesión ya no es válido y en la mayoría de los casos depende de la aplicación o sistema web para tomar esta información y anular su inicio de sesión en cada uno que lo este utilizando.

7. Discusión

El presente TT, tiene como propósito general promover la autenticación única y centralizada, mediante la interoperabilidad de aplicaciones Web; es decir, permitir manejar un solo identificador de usuario y contraseña para el acceso a diferentes aplicaciones. Dando como resultado final, la construcción del sistema SiGUCAS, el cual integra distintas aplicaciones Web para la autenticación única mediante un directorio activo, como es el Servidor OpenLDAP y un sistema de administración SAC, para el servidor antes mencionado; es así que, mediante la construcción de este prototipo, se obtuvo las habilidades adecuadas para su implementación en la UNL.

7.1. Desarrollo de la propuesta alternativa

La propuesta alternativa se desarrolló en base al cumplimiento de cada uno de los objetivos específicos que fueron abarcados en su totalidad, y como se describen a continuación:

Objetivo 1: Analizar la autenticación del protocolo CAS, como mecanismo centralizado.

El cumplimiento de este objetivo se realizó mediante dos SLR, considerando el esquema propuesto por Kitchenham[1], que es una de las cuales orienta a la ingeniería, contando con un protocolo definido claramente y estandarizado para garantizar la claridad y transparencia en el transcurso de la revisión:

La primera sobre casos de estudio relativos al protocolo CAS, dando como resultado los siguientes mecanismos que se utilizaron: Método de comunicación segura HTTPS para garantizar la transferencia de datos, mediante la generación de un certificado de seguridad autofirmado y el Sistema Jasig CAS para el ingreso de credenciales del usuario mediante su interfaz principal, personalizado de acuerdo la estreuctura del presente TT y respectando las políticas establecidas en su licencia, así también por su método de autenticación única SSO para la validación del inicio de sesión una sola vez, cuyo desarrollo completo se encuentra en el (Análisis del protocolo CAS, para una autenticación única y centralizada.).

La segunda sobre casos de estudio relativos al protocolo LDAP, dando como resultado los siguientes mecanismos que se utilizaron: Servidor OpenLDAP para el almacenamiento jerárquico y centralizado de la información, mediante el uso de todas

sus clases y atributos, el algoritmo de cifrado SHA para garantizar la seguridad, confidencialidad e integridad de la información, establecido por defecto en el Servidor OpenLDAP y el lenguaje de desarrollo PHP mediante su biblioteca adicional con LDAP, para el uso de sus funciones administrativas y de autenticación para la gestión del servidor OpenLDAP, cuyo desarrollo completo se encuentra en (Analizar el protocolo LDAP, como directorio centralizado.).

Se generó un artículo científico que fue presentado en el III SIMPOSIO IBEROAMERICANO EN PROGRAMACIÓN INFORMÁTICA (SIIPRING'2018), en la ciudad de Riobamba el día 30 de noviembre del año 2018 (Anexo 7. Certificado de Participación en el SIIPRIN'2018) y publicado en la plataforma Knowledge E, que se lo puede visualizar ingresando al siguiente link "https://knepublishing.com/index.php/KnE-Engineering/article/view/3652/7626".

Objetivo 2: Diseñar y desarrollar un prototipo de servicio de autenticación central, para múltiples aplicaciones Web.

El diseño del árbol jerárquico y estructural nos permitió organizar y almacenar la información de los usuarios de una manera ordenada y más eficaz en la consulta de información, debido a que cada registro tiene una ruta específica. Es por ello que se definieron varios niveles en donde se establece las clases y atributos que se necesitan, tomando como base la LOSEP, LOES y Código de Trabajo [42], [43], [44] para la creación de las unidades organizacionales, las cuales permiten hacer uso de esta estructura jerárquica en las diferentes entidades de Educación Superior; con el fin de realizar la autenticación centralizada al servidor OpenLDAP, cuyo desarrollo completo se encuentra en (Diseño de un árbol jerárquico en base a una estructura ordenada, para el acceso a múltiples aplicaciones Web.).

La configuración del servidor OpenLDAP, se la hizo en el Sistema Operativo Ubuntu 18.04 LTS, al terminar con su configuración se realizó la integración del schema eduPerson, para hacer uso de la estandarización de sus atributos de personas y organización utilizados en la educación superior, cuyo desarrollo completo se encuentra en (Configuración del servidor OpenLDAP).

El diseño del árbol jerárquico, propuesto en el punto anterior se lo implementó en el servidor OpenLDAP, utilizando el navegador multiplataforma JXplorer, por su facilidad para la gestión del servidor antes mencionado, en consecuencia, se obtuvo las habilidades adecuadas para realizar la propuesta del diseño del árbol jerárquico en la

UTI de la UNL, siendo aceptado y posteriormente se realizó su implantación en dicha entidad.

En el desarrollo del servicio Web, se utilizó el protocolo SOAP por su comunicación entre dos objetos de diferentes procesos sin importar el lenguaje y plataforma, dando como resultado una correcta comunicación con el servidor OpenLDAP, mediante el desarrollo de sus funciones administrativas y de autenticación; desarrollados en el lenguaje de programación PHP utilizando la librería NuSOAP, misma que se encuentra en una fase estable de desarrollo, así mismo se integró en el Servicio de Administración Central (desarrollado en lenguaje PHP) el API del protocolo CAS para el acceso único de los usuarios comunes, su desarrollo completo se encuentra en (Desarrollo de un Web service, con métodos administrativos y de autenticación para el servidor CAS.).

Conjuntamente se desarrolló un sistema de administración Web denominado SAC, mismo que nos permitió validar el servicio Web descrito anteriormente, siguiendo las etapas descritas en la metodología XP, obteniendo como resultado un sistema Web parametrizado que funciona en cualquier dispositivo de escritorio, portátiles y móviles, siendo suficiente la configuración inicial al momento de su implementación para ser desplegado en cualquier organización o entidad, su desarrollo completo se encuentra en el (Desarrollo del Web service y el sistema de administración SAC).

Terminando con el objetivo propuesto, se usó el Sistema Jasig CAS para el ingreso de credenciales de los usuarios, la cual se personalizó respetando las políticas establecidas en su licencia y que será el encargado de validar la información ingresada en el servidor OpenLDAP, mediante la configuración de sus atributos y métodos de autenticación, su desarrollo completo se encuentra en la (Implementación de Jasig CAS).

Se realizó la propuesta de la estructura antes mencionada, en la UTI de la UNL, la cual fue aceptada por dicha entidad y se procedió a su implantación en el servidor (ldap.unl.edu.ec), misma que está siendo utilizada por la comunidad universitaria en los siguientes sistemas (Anexo 9. Certificado de implantación del diseño jerárquico del servidor OpenLDAP para la autenticación de diferentes sistemas de la UNL).

De igual manera se realizó la implementación de los sistemas:

- Servicio de Administración Central (SAC), el cual es un sistema de gestión del servidor OpenLDAP.

- Sistema Jasig CAS, el cual es un sistema para el ingreso de credenciales y generación de tickets seguros.

En la UTI de la UNL, los cuales fueron aceptados por dicha entidad y se procedió a su implementación (Anexo 3. Certificado de implantación del Sistema de Administración Central (SAC).) y (Anexo 4. Certificado de implantación del Sistema de Gestión Único CAS (SiGUCAS)).

Objetivo Específico 3: Evaluar el Servicio de Autenticación Central desarrollado a través de los DevOps.

En la selección de herramientas DevOps, tomamos una muestra por conveniencia de 13 sistemas, y un método de Revisión de Registros, para la recolección de información de cada uno, luego verificamos que cada aplicación seleccionada cumple con las características establecidas para la integración con el protocolo CAS aplicando un método de Evaluación de Desempeño; donde obtuvimos un total de 5 sistemas (Moodle, GitLab, WordPress, Drupal y Jenkins), que nos permitirán verificar la interoperabilidad y acceso único, cuyo desarrollo completo se encuentra en (Seleccionar las herramientas DevOps, para la integración con el Servicio de Autenticación Central.).

Se desarrolló e implementó en un servidor local una interfaz contenedora de las aplicaciones Web y de selección única denominada SiGUCAS, que contiene las 5 aplicaciones antes seleccionados para su respectiva integración, configurados previamente para así hacer uso de la autenticación única mediante la interfaz personalizada del Sistema Jasig CAS, lo que nos dio como resultado la interoperabilidad entre los sistemas, compartiendo las credenciales de acceso mediante la generación de Tickets seguros, cuyo desarrollo completo se encuentra en (Determinar un prototipo para el ambiente de pruebas con la integración de los DevOps a un Servicio de Autenticación Central.).

Se realizó las pruebas de funcionalidad y aceptación del Prototipo propuesto en la UTI de la UNL, mediante una encuesta que contiene 25 tareas, aplicada a 7 profesionales que laboran en la entidad antes mencionada, dando como resultado un 99,42% y un 100% respectivamente, cuyo desarrollo completo se encuentra en (Realizar pruebas del ambiente antes seleccionado.), lo que generó un interés por el TT y se procedió a su despliegue en la entidad antes mencionada, donde actualmente se encuentra funcionando para toda la comunidad universitaria.

Se realizó la implantación del Sistema de Gestión único CAS (SiGUCAS), en la UTI de la UNL (Anexo 4. Certificado de implantación del Sistema de Gestión Único CAS (SiGUCAS)).

La Figura 21 describe el total de usuarios que se encuentran registrados en el servidor "ldap.unl.edu.ec" hasta la fecha de corte del 17 de mayo del 2019 y que pueden hacer uso de los sistemas informáticos con el uso de una credencial de acceso única.

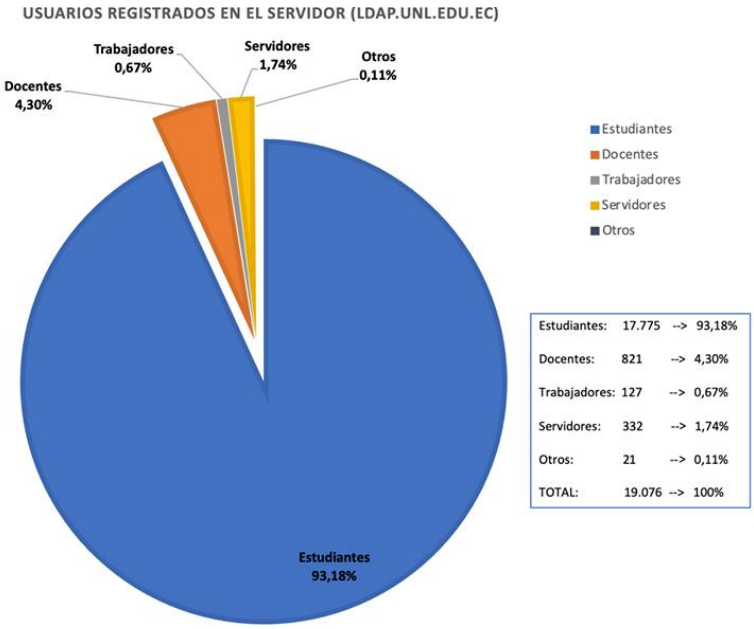


Figura 21. Usuarios de la UNL que utilizan el presente TT.

7.2. Valoración Social, Técnica, Económica y Científica.

Se expresa la valoración del TT describiendo los beneficios presentados en 4 aspectos:

Valoración Social

- Conocer el procedimiento adecuado para la realización de las SLR.
- Comprender los tipos de mecanismos existentes para el inicio de sesión único y centralizado.
- Investigar sobre las metodologías ágiles para el desarrollo de un proyecto de software.
- Entender el enfoque DevOps para que los sistemas manejen la parte de desarrollo y operaciones.

Valoración Técnica

- A través del gestor bibliográfico Mendeley se ahorró tiempo ya que este permite organizar las referencias de manera sencilla desde las fuentes y de distintos modos.
- Con la utilización de los servicios de Google como: Drive, Docs y Classroom se facilitó la revisión del presente TT, puesto que están diseñados para permitir la fácil y rápida colaboración de varios usuarios a la vez en un mismo proyecto.
- El uso del correo electrónico y redes sociales permitió la constate comunicación entre los investigadores y el director del presente TT.

Valoración Económica

- Uno de los principales beneficios es el aporte de la UNL con el control y seguimiento del presente TT, ya que cubre los gastos del Tutor o Director de Tesis.
- El uso de herramientas tecnológicas colaboro al ahorro de tiempo y dinero pues se evitó realizar impresiones innecesarias, así como asistencias personales a la UNL.

Valoración Científica

- El beneficio en el aspecto científico radica en el aporte que presta a la realización de trabajos futuros ya que el presente TT contiene una variedad de literatura que es relevante en este tema y una variada bibliografía lo que permitirá agilizar la búsqueda de documentos que aportan conocimientos sobre el mismo.
- El manuscrito publicado en el SIIPRIN 2018 que está disponible en la plataforma digital de Knowledge E de manera gratuita aporta nuevos conocimientos a la comunidad científica.

Para la elaboración del presupuesto se ha tomado en cuenta: talento humano, recursos técnicos y tecnológicos, recursos materiales y servicios para llevar a cabo los objetivos que demanda el presente TT.

En las siguientes (TABLA VIII. – TABLA XII.) se detalla cada aspecto y costos que involucra el desarrollo del presente TT.

Talento Humano

Para el desarrollo del TT se necesitó los recursos que se describen en la TABLA VIII.

TABLA VIII.
TALENTO HUMANO PARA EL DESARROLLO DEL PRESENTE TT.

Rol	Tiempo (Prop)	Tiempo (Real)	Precio/ Hora $	Valor Total	Valor Real
Responsable 1	400	600	8,00	3200,00	4800,00
Responsable 2	400	600	8,00	3200,00	4800,00
Director de Proyecto de Titulación	50	70	0,00	0,00	0,00
Profesionales	14	20	0,00	0,00	0,00
Total				**6400,00**	**9600,00**

Recursos Técnicos y Tecnológicos

Los siguientes recursos técnicos y tecnológicos requeridos se detallan en la TABLA IX.

TABLA IX.
RECURSOS TÉCNICOS PARA EL DESARROLLO DEL PRESENTE TT.

Descripción	Cantidad (Prop)	Cantidad Real	Valor Unitario $	Valor Total $	Valor Real $
HARDWARE					
Laptop	2	3	1200,00	2400,00	3600,00
Impresora	1	1	250,00	250,00	250,00
Flash	2	2	20,00	20,00	20,00
SOFTWARE					
Protocolo CAS	1	1	0	0	0
Servidor OpenLdap	1	1	0	0	0
OS Linux	1	1	0	0	0
Google Drive	1	2	0	0	0
Office 2016	1	2	32,00	32,00	64,00
Lenguaje PHP	1	1	0,00	0,00	0,00
Total				**2702,00**	**3934,00**

Materiales y Servicios

Para el desarrollo del TT se necesitaron los siguientes materiales y servicios detallados en la TABLA X. y la

TABLA XI.

TABLA X.
RECURSOS MATERIALES PARA EL DESARROLLO DEL PRESENTE TT.

Descripción	Cantidad	Valor Unitario $	Valor Total $
Resma de Papel Hojas A4	2	5,00	10,00
Cartuchos	4	20,00	80,00
Fotocopias	300	0,02	6,00
Anillados	6	1,00	6,00
Total			**102,00**

TABLA XI.
SERVICIOS PARA EL DESARROLLO DEL PRESENTE TT.

Descripción	Cantidad	Valor Unitario $	Valor Total $
Telefonía Celular	5 (horas)	10,00	50,00
Internet	450 (horas)	0,50	225,00
Transporte	100 c/u	0,30	60,00
Total			**335,00**

Presupuesto Total

Para los imprevistos fue necesario tomar el 10 % del valor total del presupuesto, agregado al valor total del TT, como se muestra en la TABLA XII.

TABLA XII.
PRESUPUESTO GENERAL PARA EL DESARROLLO DEL PRESENTE TT.

RECURSOS	SUBTOTAL	SUBTOTAL REAL
Recursos Humanos	6400,00	9600,00
Recursos Técnicos y Tecnológicos	2702,00	3934,00
Recursos Materiales	102,00	102,00

Servicios	335,00	335,00
Subtotal	9437,00	13971,00
Imprevistos 10%	943,70	1397,1
Total	**10.380,70**	**15.368,1**

Al ser un TT, los gastos que se presenten serán asumidos en su totalidad por los responsables del mismo. Cabe recalcar que el costo que demanda el Director de Tesis y los profesionales que participaron en las pruebas del presente TT, serán cubiertas por la Universidad Nacional de Loja.

8. Conclusiones

En esta sección se expone el punto de vista del investigador sobre los sucesos más relevantes presentes en el TT, entre ellos tenemos:

- El desarrollo del prototipo propuesto en el presente TT, nos permitió llevar a cabo la creación del Sistema de Gestión Único CAS (SiGUCAS), pues con esto, como equipo de trabajo se permitió gestionar y comprobar la interoperabilidad de las diferentes aplicaciones Web mediante un inicio de sesión único.

- Las Revisiones Sistemáticas de Literatura, nos permitieron establecer las características principales (SSO, Protocolo CAS, Servidor OpenLDAP y OpenSource) que deben cumplir las Aplicaciones Web para su correcta integración con una autenticación única.

- El uso del prototipo de autenticación central fortalece el proceso de acceso único para los usuarios mediante la interfaz de inicio de sesión del Sistema Jasig CAS, reduciendo el uso de múltiples credenciales de acceso a un único identificador de usuario y contraseña asociada, así mismo la optimización en la gestión de información en un único directorio centralizado.

- El uso de Aplicaciones Web con el enfoque DevOps y soporte para el protocolo CAS permite la administración y configuraciones necesarias para el uso de la autenticación única, mediante su interfaz amigable de conexión y segura.

- Con la implantación del Sistema de Administración Central (SAC), se logró automatizar el proceso que efectuaban de forma manual el personal de la UTI de la UNL, durante el proceso de cambio y restablecimiento de contraseñas, permitiendo reducir el tiempo que se empleaba en estás actividades.

- La implantación del presente TT en las diferentes aplicaciones y sistemas Web que maneja la comunidad universitaria de la UNL, permitió que 19,076 usuarios hasta la fecha de corte del 17 de mayo del 2019, utilizan el prototipo propuesto.

- El presente TT, empezó como un prototipo técnico - académico, generó un gran impacto en la UTI por lo cual se procedió a su implementación en dicha entidad para beneficio de toda la comunidad universitaria de la UNL.

9. Recomendaciones

En esta sección se expone recomendaciones, que los investigadores evidenciaron durante el desarrollo del TT para mejorar y optimizar el mismo, entre ellos tenemos:

- Utilizar versiones estables y actuales de las aplicaciones Web para que no existan problemas de implementación e incompatibilidad durante la conexión con el protocolo CAS.

- Hacer uso de Aplicaciones Web que utilicen el enfoque DevOps, las cuales nos permitirán la optimización en la gestión de usuarios por su filosofía de desarrollo y administración de recursos mediante los perfiles de usuario.

- La adaptación de una infraestructura necesaria para la implementación de un mecanismo centralizado que permitan el respaldo, almacenamiento y gestión de la información en los directorios centralizados.

- Las aplicaciones o sistemas Web que no cuentan con soporte directo para el protocolo CAS o el uso de sus métodos de autenticación mediante el API del Sistema Jasig CAS, pueden conectarse directamente al directorio centralizado, para hacer uso de las mismas credenciales de acceso.

- Tomar en cuenta la adaptación de certificados de seguridad mediante una Autoridad de Certificación (CA) de jerárquia superior que garantice la confidencialidad, integridad y autenticación segura de los usuarios.

9.1. Trabajos Futuros

- Hacer uso del API que el protocolo CAS ofrece para la autenticación única mediante el Sistema Jasig CAS, para las aplicaciones o sistemas Web que no cuentan con soporte directo con el protocolo antes mencionado.

- Las aplicaciones o sistemas Web, que utilicen el Sistema Jasig Cas para el ingreso de credenciales deben incorporar la multi-identificación para garantizar el acceso a los recursos de las distintas aplicaciones web en caso de existir inconvenientes en la interfaz principal de acceso.

- Para manejar la seguridad en el inicio de sesión único, mediante el uso del protocolo CAS en diferentes aplicaciones o sistemas web, tomar en cuenta la verificación mediante hardware, verificación en dos pasos, mensaje de alerta al iniciar sesión o gestión de notificaciones de acceso.

10. Bibliografía

[1] B. Kitchenham, “Procedures for performing systematic reviews,” *Keele, UK, Keele Univ.*, 2004.

[2] J. María and G. Torres, “Implantación de un SSO,” 2018.

[3] F. M. Vargas, “Sistema Integrado de Autenticación para la Universidad Tecnológica de Bolívar- MiUTB,” 2016.

[4] J. María and G. Torres, “Implantación de un SSO,” 2018.

[5] V. Radha and D. H. Reddy, “A Survey on Single Sign-On Techniques,” *Procedia Technol.*, vol. 4, no. May, pp. 134–139, 2012.

[6] “CAS | Apereo.” [Online]. Available: https://www.apereo.org/projects/cas. [Accessed: 27-Mar-2019].

[7] P. Aubry, V. Mathieu, and J. Marchal, “ESUP-Portail : open source Single Sign-On with CAS (Central Authentication Service).”

[8] W. Paper, “CENTRAL AUTHENTICATION SERVICE (CAS) SSO FOR EMC ® DOCUMENTUM ® REST SERVICES.”

[9] “CAS - Home.” [Online]. Available: https://apereo.github.io/cas/4.2.x/index.html. [Accessed: 24-May-2019].

[10] A. Aguilar Santos, J. C. Pérez Pérez, and L. E. Cornejo Tello, “Implementación de un Sistema de Autenticación usando LDAP para control de Acceso a una RED”,” Sep. 2014.

[11] M. A. V. P. Jairo Alberto Cifuentes Fuentes, “Ldap_Seguridad,” 2013.

[12] D. Aníbal and O. Bravo, “‘Análisis de las Implementaciones del Protocolo LDAP. Caso práctico: Implantación de un Sistema de Autenticación Aplicado a los Aaboratorios de la EIS,’” 2013.

[13] “OpenLDAP Software 2.4 Administrator’s Guide.” [Online]. Available: https://www.openldap.org/doc/. [Accessed: 18-Mar-2019].

[14] G. Wang, “Unified Identity Authentication between Heterogeneous Systems Based on LDAP and RBAC,” vol. 9, no. 10, pp. 2858–2865, 2014.

[15] R. A. A. Martínez and A. A. G. Hernández, “Marco de trabajo para el desarrollo de herramientas orientadas a la gestión e integración de servicios telemáticos de infraestructura en GNU,” *Rev. Cuba. Ciencias Informáticas*, vol. 7, no. 2, pp. 157–168, 2013.

[16] I. J. Lázaro Ramos Alfonso and M. Martín Mesa, “Implementación de un Servidor Samba con autenticación LDAP como alternativa Libre a los Servidores de Dominio Windows.”

[17] M. Jose, M. Gonzáles, and Á. Epaña, “User Management with LDAP

(Lightweight Directory Access Protocol) for access to technology and Information Services in Companies ´," *J. Sci. Res. Rev. Cienc. E Investig. ´ ON, E-ISSN 2528-8083, VOL. 1, CITT, PP. 10-15*, vol. 1, pp. 10–15, 2016.

[18] "eduPerson & eduOrg Documentation | Internet2." [Online]. Available: https://www.internet2.edu/products-services/trust-identity/eduperson-eduorg/eduperson-eduorg-documentation/. [Accessed: 09-May-2019].

[19] R. P. Hielscher and V. Delgado, "Introducción a la Criptografía: tipos de algoritmos," *An. Mecánica Y Electr.*, vol. 83, no. 1, pp. 42–46, 2006.

[20] J. D. Pino, "Organización del Computador II," Buenos Aires, 2016.

[21] J. Fermín and D. Amado, "Distribución de un Entorno de Modelado Utilizando Servicios Web," Cartagena, 2014.

[22] C. Gil, "Los WEB SERVICES y CARACTERÍSTICAS DE CALIDAD," 2009.

[23] C. A. Morales, *Estado del Arte: Servicios Web*. 2010, p. 4.

[24] A. Los and S. Aransay, "URevisión de los Servicios Web SOAP/REST: Características y Rendimiento," 2009.

[25] Guillermo Jiménez Marco, "DevOps, la nueva tendencia en el desarrollo de sistemas TI, un caso práctico en el análisis de incidencias de software," p. 102, 2016.

[26] "Qué es Devops ?," 2012.

[27] "Autor : Alberto Belalcázar Director : Lcdo . Javier Díaz," pp. 1–200, 2017.

[28] M. Hüttermann, "DevOps for Developers."

[29] J. H. Canós, P. Letelier, and M. C. Penadés, "Métodologías Ágiles en el Desarrollo de Software."

[30] O. A. P. A, "Cuatro enfoques metodológicos para el desarrollo de Software RUP – MSF – XP - SCRUM," no. 10, pp. 64–78, 2011.

[31] N. P. Sintya Meléndez, Maria Gaitan, "METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION EXTREMA.," 2016.

[32] Y. Wang, J. Tian, C. Yang, and Y. Zhu, "The Research and Design of Unified Authentication System Based on CAS," 2017.

[33] A. Pereira, J. Sobral, and C. Westphall, "Towards scalability for federated identity systems for cloud-Based environments," *2014 6th Int. Conf. New Technol. Mobil. Secur. - Proc. NTMS 2014 Conf. Work.*, 2014.

[34] M. Y. A. Saputro, K. I. Satoto, and A. F. Rochim, "Implementasi Sistem Single Sign On / Single Sign Out Berbasis Central Authentication Service Protocol Pada Jarigan Lightweight Directory Access Protocol Universitas Diponegoro," vol. 1, no. 3, p. 36, 2012.

[35] G. Ramadhan, "Analisis Teknologi Single Sign On (SSO) dengan Penerapan

Central Authentication Service (CAS) Pada Universitas Bina Darma," *Uma ética para quantos?*, vol. XXXIII, no. 2, pp. 81–87, 2012.

[36] Y. N. Kunang and I. Z. Yadi, "Sistem Single Sign on Universitas Berbasis Cas-Ldap," no. 12, pp. 1–7, 2014.

[37] F. Mendoza Vargas, "Sistema Integrado de Autenticación para la Universidad Tecnológica de Bolívar- MiUTB," 2016.

[38] J. Andjarwirawan, H. N. Palit, and J. C. Salim, "Linux PAM to LDAP Authentication Migration," *2017 Int. Conf. Soft Comput. Intell. Syst. Inf. Technol.*, pp. 155–159, 2017.

[39] G. Espinoza, P. Ortega, C. Palacios, and S. Junior, "Intelligent agents applied to the management of ldap user profiles," *Revista Energía*, Loja, p. 104, 2014.

[40] M. Cueva-Hurtado, R. Figueroa-Diaz, W. Aguilar-Soto, and M. Armijos-Ordoñez, "Systematic Literature Review on the LDAP Protocol As a Centralized Mechanism for the Authentication of Users in Multiple Systems," *KnE Eng.*, vol. 3, no. 9, p. 144, 2018.

[41] L. Z. Len Bass, Ingo Weber, *DevOps: A Software Architect's Perspective - Len Bass, Ingo Weber, Liming Zhu - Google Libros*, Pearson Education. United States of America, 2015.

[42] R. Oficial Suplemento, "LEY ORGANICA DE SERVICIO PUBLICO, LOSEP Estado: Vigente PRESIDENCIA DE LA REPUBLICA."

[43] P. D. E. L. A. Republica, H. Enrique, and D. Pozo, "Ley Organica De Educacion Superior (Loes)," *Regist. Of. Supl. 298 del 12-oct-2010*, pp. 1–63, 2010.

[44] M. P. Maya, "CODIGO DEL TRABAJO."

[45] R. Marrero, "Sistema centralizado de gestión de usuarios para Innova7.," 2014.

[46] M. L. Ramos and O. F. Díaz, "Authentication Component for Dynamic Report Generator." Santo Domingo, 2015.

[47] S. Y. Guo and X. X. Zhang, "Research on the Integration of Mail System and Digital Campus Platform," *Adv. Mater. Res.*, vol. 998–999, pp. 1709–1712, 2014.

[48] H. Hu and Z. Guo, "The application of cross-domain single sign-on in municipal portal," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, 2013.

[49] W. Li, Y. B. Zhu, and M. Zou, "Applied research of single sign on technology in cloud services," *Appl. Mech. Mater.*, vol. 602–605, pp. 3552–3555, 2014.

[50] F. Huang, C. Wang, and J. Long, "Design and Implementation of Single Sign On System with Cluster CAS for Public Service Platform of Science and

Technology Evaluation," 2011.

[51] X. Chen, "Research on User Identity Authentication Technology for Virtual Laboratory System," *Proc. - 2015 6th Int. Conf. Intell. Syst. Des. Eng. Appl. ISDEA 2015*, pp. 688–691, 2016.

[52] H. Arslan and H. D. Karkı, "Examining of Single Sign on Protocols and A Model of Business Application."

[53] Z. Wu, W. Huang, and L. Yu, "Design and Implementation of unified Identity Authentication System Based on LDAP in Digital Campus," pp. 1213–1217, 2014.

[54] F. Yinglan, J. Hao, and H. Bing, "Single Sign-On Research and Expansion Based On CAS," *Open Cybern. Syst. J.*, vol. 8, pp. 200–207, 2014.

[55] A. Amarudin, "Implementation of CAS Server as Authentication Protocol on Single Sign-On (SSO) Network With PHP Programming," no. December 2014, 2017.

[56] E. Bahit, "Programador PHP," 2010.

[57] A. Marzal Varó, I. Gracia Luengo, and P. García Sevilla, *Introducción a la programación con Python 3*. 2014.

[58] T. Groussard, *JAVA 7 : los fundamentos del lenguaje Java*. Ediciones ENI, 2012.

11. Anexos

Anexo 1. Personal involucrado en el presente TT.

Figura A1. 1. Introducción del presente TT

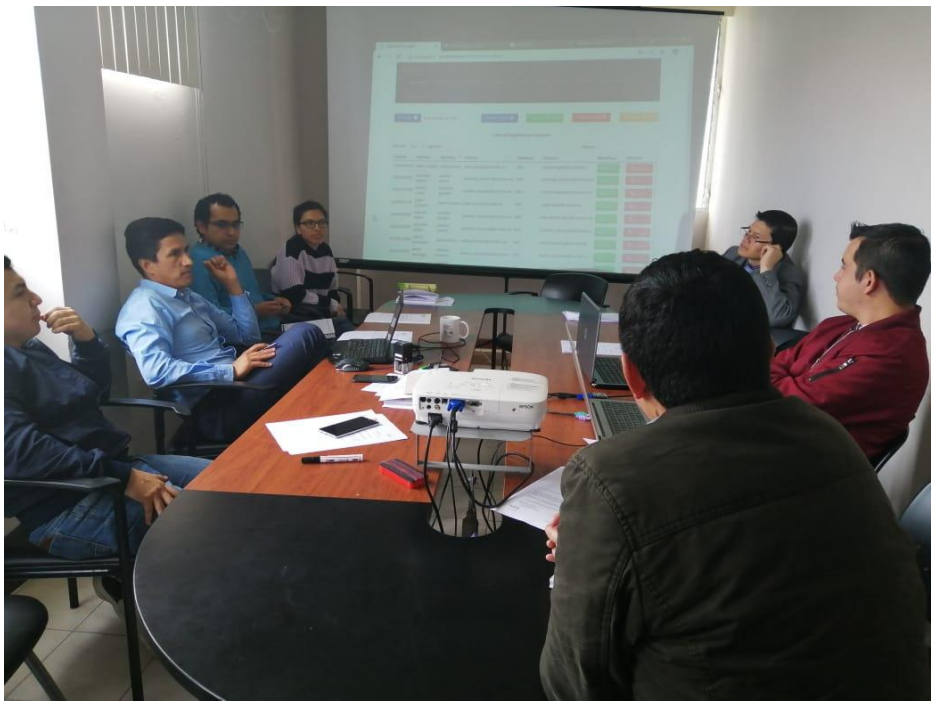


Figura A1. 2. Presentación final del presente TT

Anexo 2. Listado de participantes para las pruebas del presente TT.

UNIVERSIDAD NACIONAL DE LOJA

CONV-CIS-UNL

Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja

Ciudad. –

Convocatoria:

Se realiza la cordial invitación, al personal administrativo de la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja, a las pruebas de aceptabilidad y funcionalidad del tema de titulación ***"Desarrollo de un Prototipo para el Servicio de Autenticación Central de Usuario en Aplicaciones Web"***, desarrollado por los estudiantes: **Antonio Aguilar y Manuel Armijos**. La cual se llevará a cabo el día 12 de marzo del 2019.

Adjunto a la presente, la lista de la población participante.

Por la favorable atención a la presente, le anticipo mis más sinceros agradecimientos.

Atentamente.

Wilmer Antonio Aguilar Soto
1900481878

Manuel Stalin Armijos Ordóñez
1105593238

UNIVERSIDAD NACIONAL DE LOJA

INF-CIS-UNL

Listado de la población presente en las pruebas del Trabajo de Titulación (TT).

Población:	Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja			
Nombre:	Antonio Aguilar, Manuel Armijos	Rol:	Supervisores del TT.	FIRMA
Nombre:	Ing. Jhon Calderón	Rol:	Administrador del SiGUCAS	
Nombre:	Ing. Danny Muñoz	Rol:	Usuario Común	
Nombre:	Ing. Juan Pablo Ramón	Rol:	Usuario Común	
Nombre:	Ing. José Martínez	Rol:	Usuario Común	
Nombre:	Ing. Lissette López	Rol:	Usuario Común	
Nombre:	Ing. Juan Carlos Riofrio	Rol:	Usuario Común	
Nombre:	Ing. Edison Coronel	Rol:	Usuario Común	

Anexo 3. Certificado de implantación del Sistema de Administración Central (SAC).

Universidad Nacional de Loja

Unidad de Telecomunicaciones e Información

DIRECTOR DE TELECOMUNICACIONES E INFORMACIÓN

CERTIFICA:

Que el señor **Wilmer Antonio Aguilar Soto** con cédula de ciudadanía número **1900481878** y el señor **Manuel Stalin Armijos Ordóñez** con cédula de ciudadanía número **1105593238**; egresados de la Carrera de Ingeniería en Sistemas, han finalizado lo referente a la implantación del ***Servicio de Administración Central – SAC***, de su proyecto de titulación denominado ***"Desarrollo de un Prototipo para el Servicio de Autenticación Central de Usuarios en Aplicaciones Web"*** en la Unidad de Telecomunicaciones e Información, bajo los lineamientos y requerimientos establecidos por esta unidad administrativa de la Universidad Nacional de Loja, el cual en la actualidad se encuentran bajo el nombre ***Servicio de Autenticación Centralizada SAC.***

Es cuanto puedo indicar en honor a la verdad, facultando al interesado, hacer uso del presente documento.

Loja, 21 de marzo del 2019

Ing. Jhon Alexander Calderón Sanmartín
DIRECTOR DE TELECOMUNICACIONES E INFORMACIÓN
UNL

072 -54 7252 Ext. 125
Ciudad Universitaria "Guillermo Falconí Espinosa",
Casilla letra "S", Sector La Argelia - Loja - Ecuador

Anexo 4. Certificado de implantación del Sistema de Gestión Único CAS (SiGUCAS) y el Sistema Jasig CAS.

Universidad Nacional de Loja

Unidad de Telecomunicaciones e Información

DIRECTOR DE TELECOMUNICACIONES E INFORMACIÓN

CERTIFICA:

Que el señor **Wilmer Antonio Aguilar Soto** con cédula de ciudadanía número **1900481878** y el señor **Manuel Stalin Armijos Ordóñez** con cédula de ciudadanía número **1105593238**; egresados de la Carrera de Ingeniería en Sistemas, han finalizado lo referente a la implantación de los siguientes sistemas:

- Sistema de Gestión Único CAS (SiGUCAS)
- Sistema Jasig CAS

De su proyecto de titulación denominado ***"Desarrollo de un Prototipo para el Servicio de Autenticación Central de Usuarios en Aplicaciones Web"*** en la Unidad de Telecomunicaciones e Información, bajo los lineamientos y requerimientos establecidos por esta unidad administrativa de la Universidad Nacional de Loja, los cuales en la actualidad se encuentran bajo el nombre ***Servicio de Autenticación Centralizada SAC.***

Es cuanto puedo indicar en honor a la verdad, facultando al interesado, hacer uso del presente documento.

Loja, 15 de abril del 2019

Ing. Jhon Alexander Calderón Sanmartín
DIRECTOR DE TELECOMUNICACIONES E INFORMACIÓN

072 -54 7252 Ext. 125
Ciudad Universitaria "Guillermo Falconí Espinosa",
Casilla letra "S", Sector La Argelia - Loja - Ecuador

Anexo 5. SLR del Protocolo CAS

A. Planificación de la Revisión Sistemática de Literatura

En esta etapa identificamos cuáles son las necesidades de la revisión para lo cual es necesario plantear y responder algunos puntos claves descritos a continuación.

1. Objetivo de la Revisión Sistemática de Literatura

El objetivo es poder identificar y seleccionar los artículos científicos que tengan relevancia sobre el protocolo CAS y la autenticación única, que otorguen información fundamental al tema de titulación.

2. Formulación de la pregunta de investigación

A partir de tema de titulación “Desarrollo de un Prototipo Para el Servicio de Autenticación Central de Usuarios en Aplicaciones Web”, se plantea las siguientes preguntas de investigación referente al tema antes mencionado, descritas en la TABLA A5. I.

TABLA A5. I.
PREGUNTAS DE INVESTIGACIÓN

P1. ¿Por qué se utiliza el protocolo CAS como mecanismo centralizado para la autenticación de usuarios en múltiples sistemas?
P2. ¿La autenticación única con el protocolo CAS, mejoro la interoperabilidad de los múltiples sistemas?

3. Palabras claves

Se identifican las palabras claves de los artículos científicos propuestos por el grupo de investigadores, las cuales nos servirán para poder plantear la cadena de búsqueda, se encuentran en la TABLA A5. II.

TABLA A5. II.
ARTÍCULOS PRELIMINARES Y PALABRAS CLAVES

#	Artículo	Palabras Claves
A1	Central Authentication Service (Cas) Sso For Emec [8].	Single Sign-on (SSO), Central Authentication Service (CAS), Jasig, Ehcache.
A2	Analisis Teknologi Single Sign On (Sso) Dengan Penerapan Central Authentication Service (Cas) Pada Universitas Bina Darma [35].	Single Sign On (SSO), Lightweight Directory Acces Protocol (LDAP), Central Authentication Service (CAS).
A3	Implementasi Sistem Single Sign On / Single Sign Out Berbasis Central Authentication Service Protocol Pada Jaringan Lightweight Directory Access Protocol Universitas Diponegoro [34].	CAS, Single Sign On, Single Sign Out, Otentikasi, LDAP.

A4	Sistema centralizado de gestión de usuarios para Innova7 [45].	Sistema centralizado, usuarios, estructura, organización, innova7, aplicación, Web, sso, autenticación, autorización.
A5	Authentication Component for Dynamic Report Generator [46].	GDR, authentication, SSO, CAS.
A6	Research on the Integration of Mail System and Digital Campus Platform [47].	Mail system; digital campus platform; integration.
A7	The Application of Cross-domain Single Sign-on in Municipal Portal [48].	Single Sign-on, Cross-domain, Portal.
A8	Applied Research of Single Sign On Technology in Cloud Services [49].	Cloud Service; Identity Authentication; CAS protocol; Single sign-on.
A9	ESUP-Portail: open source Single Sign-On with CAS (Central Authentication Service) [7].	Single Sign-On, open-source, authentication.
A10	Design and Implementation of Single Sign On System with Cluster CAS for Public Service Platform of Science and Technology Evaluation [50].	public service platform of science and technology evaluation, cluster CAS, Single Sign On.

4. Selección de fuentes y estrategias de búsqueda

Se seleccionó un conjunto de bases de datos científicas, revistas académicas, tesis y páginas oficiales en donde se procedió a buscar la información más sobresaliente, como se indica en la TABLA A5. III.

TABLA A5. III.
FUENTES DE MOTORES DE BÚSQUEDA

Fuentes	URL
IEEEXPlorer	http://ieeexplore.ieee.org/
Scientific.net	https://www.scientific.net
Scholar Google	http://scholar.google.es/
Navegadores Web	http://www.google.com

5. Cadena de Búsqueda

La TABLA A5. IV. muestra la manera de generar la cadena de búsqueda donde se utilizó los conectores lógicos “AND” y “OR”, con el fin de obtener la cadena resultante.

TABLA A5. IV.
CADENA DE BÚSQUEDA

((“CAS”) OR (“Central Authentication Service”) AND (“SSO OR Single Sign On”) OR (“Authentication OR protocol)).

6. Criterios de inclusión

Se realizó un estudio de los artículos más relevantes, excluyendo a los demás tomando en cuenta los siguientes criterios, descritos en la TABLA A5. V.

TABLA A5. V.
CRITERIOS DE INCLUSIÓN

Idioma	• Ingles • Español • Indonesio
Motores de búsqueda	• IEEEXplorer • Scientific.net • Google Scholar • Navegadores Web o Conferencias Académicas
Fecha de publicación	2013 - 2018
Tipo de producciones	Artículos científicos, tesis y conferencias científicas.

7. Criterios de exclusión

Los estudios que no han sido relevantes se los descarta tomando en consideración el siguiente criterio:

- Título.
- Abstract (Resumen).
- Texto completo del documento.
- Palabras claves.
- Resultados.
- Conclusiones

B. Ejecución de la Revisión Sistemática de literatura

A continuación, especificamos los criterios de selección de estudios más importantes, extracción de la información y las cadenas de búsqueda avanzada aplicadas en las fuentes científicas.

Ejecución de la selección de fuentes en IeeeXplorer

((CAS) AND (AUTHENTICATION))

La ejecución de la búsqueda en IEEEXplorer nos arrojó 130 resultados. Tras aplicar el criterio de inclusión nos quedamos con 41 documentos relevantes, de los cuales, aplicando el criterio de exclusión, se consideran los descritos en la TABLA A5. VI. como estudios primarios.

TABLA A5. VI.
ESTUDIOS PRIMARIOS DE IEEEXPPLORER

#	art. cit.	Título y Publicación
1	[32]	**The research and design of unified authentication system based on CAS.** 2017 2nd IEEE International Conference on Computational Intelligence and Applications.
2	[48]	**The application of cross-domain single sign-on in municipal portal.** IEEE Region 10 Annual International Conference, Proceedings/TENCON.
3	[51]	**Research on User Identity Authentication Technology for Virtual Laboratory System.** Proceedings - 2015 6th International Conference on Intelligent Systems Design and Engineering Applications, ISDEA 2015.
4	[33]	**Towards Scalability for Federated Identity Systems for Cloud-Based Environments.** 2014 6th International Conference on New Technologies, Mobility and Security - Proceedings of NTMS 2014 Conference and Workshops.
5	[52]	**Examining of single sign on protocols and a model of business application.**

Ejecución de la selección de fuentes en Scientific.net

((CAS) AND (AUTHENTICATION))

La ejecución de la búsqueda en Scientific.net nos arrojó 27 resultados. Tras aplicar el criterio de inclusión nos quedamos con 16 documentos relevantes, de los cuales, aplicando el criterio de exclusión, se consideran los descritos en la TABLA A5. VII. como estudios primarios.

TABLA A5. VII.
ESTUDIOS PRIMARIOS DE SCIENTIFIC.NET

#	art. cit.	Título y Publicación
6	[49]	**Applied Research of Single Sign On Technology in Cloud Services.** Applied Mechanics and Materials Vols. 602-605 (2014) pp 3552-3555.
7	[47]	**Research on the Integration of Mail System and Digital Campus Platform** Advanced Materials Research Vols. 998-999 (2014) pp 1709-1712.
8	[53]	**Design and Implementation of unified Identity Authentication System Based on LDAP in Digital Campus** Advanced Materials Research Vols. 912-914 (2014) pp 1213-1217.

Ejecución en el motor de búsqueda científico Google Scholar

((CAS AND SSOAND AUTHENTICATION))

La ejecución de la búsqueda en Google Scholar nos arrojó 4.620 resultados. Tras aplicar el criterio de inclusión nos quedamos con 1.450 documentos relevantes, de los cuales, aplicando el criterio de exclusión, se consideran los descritos en la TABLA A5. VIII. como estudios primarios.

TABLA A5. VIII.
ESTUDIOS PRIMARIOS DE GOOGLE SCHOLAR

#	art. cit.	Título Y Publicación
9	[34]	**Implementasi Sistem Single Sign On / Single Sign Out Berbasis Central Authentication Service Protocol Pada Jaringan Lightweight Directory Access Protocol Universitas Diponegoro.** TRANSIENT, VOL. 1, NO. 3, SEPTEMBER 2012, ISSN: 2302-9927, 36.
10	[35]	**Analisis Teknologi Single Sign On (Sso) Dengan Penerapan Central Authentication Service (Cas) Pada Universitas Bina Darma.** Jurnal Ilmiah Teknik Informatika Ilmu Komputer Vol. xx No.x Oktober 2013: 1-13.
11	[36]	**Sistem Single Sign On Universitas Berbasis Cas-Ldap.** Seminar Nasional Inovasi dan Tren (SNIT) 2014.
12	[54]	**Single Sign-On Research And Expansion Based On CAS.** The Open Cybernetics & Systemics Journal.
13	[37]	**Sistema Integrado De Autenticación Para La Universidad Tecnológica De Bolívar- Miutb.** Trabajo de Grado.

Ejecución de la selección de fuentes en Navegadores Web o Conferencias Académicas

Para realizar la consulta en esta fuente, seleccionamos la búsqueda por título introduciendo palaras claves como (Autenticación única con CAS o Manual CAS), nos dio como resultado 2, aplicando el criterio de inclusión nos quedamos con dos y aplicando el criterio de exclusión y página de referencia, se consideran los descritos en la TABLA A5. IX. como estudios primarios.

TABLA A5. IX.
ESTUDIOS PRIMARIOS DE NAVEGADORES WEB Y CONFERENCIAS ACADÉMICAS

#	art. cit.	Título y Publicación
14	[46]	**Authentication Component for Dynamic Report Generator.**

		13th LACCEI Annual International Conference: "Engineering Education Facing the Grand Challenges, What Are We Doing?".
15	[8]	**Central Authentication Service (Cas) Sso For Emc® Documentum® Rest Services.** Manual CAS.
16	[55]	**Implementation of CAS Server as Authentication Protocol on Single Sign-On (SSO) Network With PHP Programming.** ICETIA 2014 ISSN 2407-4330.

1. Criterios de selección de estudios

Para el cumplimiento del objetivo principal de resultados de la búsqueda deben cumplir el siguiente criterio de selección: Los artículos deben destacar la importancia y beneficios del uso del protocolo CAS para la autenticación de usuarios.

2. Extracción de la información

Los criterios dados de inclusión, exclusión y de selección, permitieron identificar los diferentes artículos, patentes, libros, páginas oficiales y revistas digitales con el fin de cumplir el objetivo planteado en esta investigación. Para la extracción importante de cada estudio se utilizó los siguientes elementos:

- Información relevante del Protocolo CAS
- Características claves de la autentificación del Protocolo CAS
- Autenticación única e interoperabilidad

C. Análisis de resultados y hallazgos

Se realizó un análisis previo donde se evalúa cada estudio, discriminando artículos que tienen criterios comunes e información no muy trascendental, estos fueron descartados quedándose con los artículos más relevantes.

Se enlistan 16 estudios con las etiquetas A1 a la A16, que son los estudios seleccionados de acuerdo a los criterios indicados, como se pueden observar en la TABLA A5. X.

TABLA A5. X.
RESULTADOS DE LA SLR

A1. The Research and Design of Unified Authentication System Based on CAS	
Resumen	CAS es un SSO de Open Source, razonable y ajustable al principio y realización básica de inicio de sesión único, debido a su eficiencia y sencillez de aplicar en base de datos centralizada y directorios activos. Se utilizan para gestionar centralmente identidades con el fin de lograr la autenticación unificada y la gestión unificada. La autenticación de identidad unificada, tales como el uso de

	control de acceso centralizado, proporciona a los usuarios un servicio personalizado, brindando permisos de acceso para diferentes sistemas de aplicación Web.

Características claves del uso del protocolo CAS.	- Jasig CAS
	  ▪ CAS CLIENT
	  ▪ CAS SERVER
	- Servicio personalizado
	- Comprobación de credenciales
	- Redireccionamiento
	- Protocolo LDAP
	- Protocolo Secure Sockets Layer (SSL)
	- SSO

Mecanismos de Autenticación única e interoperabilidad.	- Basado en cookies. - Las aplicaciones deben hacerse bajo el protocolo HTTPS para asegurar la información en la transmisión de datos. - URL con cadena del ticket de servicio. - TGT Ticket de acceso a través de un login exitoso. - ST Ticket de servicio, se utiliza sólo una vez para acceder a un servicio específico. - TGC Ticket que se almacena en la cookie CASTGC, representa una sesión de SSO para un usuario.
A2. The Application of Cross-domain Single Sign-on in Municipal Portal.	
Resumen	CAS permite al usuario conectarse una vez para acceder a otras aplicaciones, a fin de mejorar la facilidad de uso, seguridad y estabilidad de la información del sistema. CAS no solo puede proporcionar la infraestructura de SSO en varias aplicaciones Web, sino que también proporcionan la función SSO para aplicaciones no Web con la función de servicio funcional cliente Web. CAS puede centralizar la autenticación de usuario para la aplicación Web única y simplificar la gestión de contraseñas, por lo tanto, mejorar la seguridad.
Características claves del uso del protocolo CAS.	- CAS CLIENT - CAS SERVER - SSO - Protocolo LDAP
Mecanismos de Autenticación única e interoperabilidad.	- Basado en cookies, los cuales permiten a los servidores Web, comprobar si los usuarios ya se han autenticado. - ST Ticket de servicio. - TGC Ticket que se almacena en la cookie CASTGC, representa una sesión de SSO para un usuario.
A3. Research on User Identity Authentication Technology for Virtual Laboratory System.	

Resumen	CAS es una autenticación de identidad uniforme y utiliza el modelo mejorado de autenticación, con el fin de garantizar la transmisión de información de los usuarios, mediante una autenticación única y centralizada, basada en el protocolo de seguridad de los datos.
Características claves del uso del protocolo CAS.	- YALE CAS 3.3.3 (Jasig CAS) ▪ CAS CLIENT ▪ CAS SERVER - Re direccionamiento a través de Https - MS SQL SERVER 2000 - Comprobación de credenciales.
Mecanismos de Autenticación única e interoperabilidad.	- Basado en cookies. - TGT Ticket de acceso a través de un login exitoso. - ST Ticket de servicio, se utiliza sólo una vez para acceder a un servicio específico. - TGC Ticket que se almacena en la cookie CASTGC, representa una sesión de SSO para un usuario. - URL con cadena del ticket de servicio. - JSSE Java Secure Socket Extension, para el procesamiento seguro de datos. - JDBC Java Database Connectivity, permite ejecutar operaciones sobre la base de datos. CAS lo toma como un controlador adicional para la autenticación.
A4. Towards Scalability for Federated Identity Systems for Cloud-Based Environments	
Resumen	CAS proporciona componentes para actuar como un servicio de autenticación, conocido como Servidor CAS. La solución implementa una conexión HTTPSprotocolo basado para SSO, y proporciona herramientas de cliente, en varias plataformas, por lo que los sistemas pueden participar en la infraestructura de autenticación.
Características claves del uso del protocolo CAS.	- Jasig CAS - SSO (Inicio de sesión único) - SSO (Cierre de sesión único) - SAML (Lenguaje de Marcado para Confirmaciones de Seguridad). Redirecionamiento a los diferentes servicios.
Autenticación única e interoperabilidad.	- HTTPSprotocolo basado para SSO. - Utiliza Cookies para guardar información. Genera un ticket para verificar si la autenticación fue exitosa.
A5. Examining of Single Sign on Protocols and A Model of Business Application.	
Resumen	CAS proporciona la accesibilidad a distintas aplicaciones con un único inicio de sesión SSO. Elimina la necesidad de múltiples credenciales para diferentes aplicaciones,

	utilizando el mecanismo de autenticación única, donde el usuario tiene acceso a otros recursos, mediante un redireccionamiento de sus credenciales, de una manera segura.
Características claves del uso del protocolo CAS.	- CAS Clients - CAS Server - Comprobación de credenciales - Variedad de entornos. - Soporte para protocolos como SAML, AOuth y OpenID tanto en la flexibilidad como en la infraestructura de la aplicación.
Mecanismos de Autenticación única e interoperabilidad.	- Basado en cookies. - TGT Ticket de acceso a través de un login exitoso. - ST Ticket de servicio, se utiliza sólo una vez para acceder a un servicio específico.
A6. Applied Research of Single Sign On Technology in Cloud Services.	
Resumen	CAS permite a un usuario acceder a múltiples aplicaciones al tiempo que proporciona sus credenciales (como ID y contraseña) sólo una vez. La autenticación única SSO utilizando el protocolo CAS, esta diseñado para reducir al mínimo los tiempos que cada usuario utiliza en las diversar aplicaciones Web. Reduciendo costos de asistencia técnica y mejorando la seguridad.
Características claves del uso del protocolo CAS.	- Mecanimos de Autenticación soportados: LDAP, Active Directory, Kerberosy RDBMS. - Variedad de entornos como: Java, .Net, PHP, Perl, uPortal, etc. - Soporte y preferible conexión TCP/IP - Redireccionamiento - Gestión de credenciales.
Mecanismos de Autenticación única e interoperabilidad.	- Basado en cookies, los cuales permiten a los servidores Web, comprobar si los usuarios ya se han autenticado. - Utiliza peticiones de autenticación y autorización. - ST Ticket de servicio, se utiliza sólo una vez para acceder a un servicio específico.
A7. Research on the Integration of Mail System and Digital Campus Platform	
Resumen	El propósito de CAS es crear un sistema de autenticación unificada para gestionar y verificar la identidad del usuario para lograrlo utiliza el modelo de confianza SSO (inicio de sesión único).

Características claves del uso del protocolo CAS.	- Servidor CAS. - SSO.
Autenticación única e interoperabilidad.	- Utiliza un Servicio Web para interoperar con otros componentes compatibles, toma principalmente la ventaja de HTTP y el protocolo SOAP para realizar la transmisión de datos en la Web.

A8. Design and Implementation of unified Identity Authentication System Based on LDAP in Digital Campus.	
Resumen	CAS es un sistema de autenticación de identidad independiente, seguro, eficiente y confiable. Basado en una autenticación única SSO o sistema de autenticación de identidad unificada. Con este sistema, los usuarios acceden a los recursos de todos los sistemas de aplicación Web con los permisos adecuados simplemente entrando en una sola vez.
Características claves del uso del protocolo CAS.	- CAS Clients - CAS Server - Protocolo LDAP - Variedad de entornos - Navegador Web del cliente - Redireccionamiento
Mecanismos de Autenticación única e interoperabilidad.	- Basado en cookies. - TGT Ticket de acceso a través de un login exitoso. - ST Ticket de servicio, se utiliza sólo una vez para acceder a un servicio específico. - TGC Ticket que se almacena en la cookie CASTGC, representa una sesión de SSO para un usuario.

A9. Implementasi Sistem Single Sign On / Single Sign Out Berbasis Central Authentication Service Protocol Pada Jaringan Lightweight Directory Access Protocol Universitas Diponegoro	
Resumen	CAS se utiliza como una página de inicio de sesión centralizada para servicios Web basados en CMS, multiblogging, Webcloud y Webmail. Mientras que la cuenta para iniciar sesión proviene de LDAP. El éxito se determina mediante el proceso de inicio de sesión y cierre de sesión en una aplicación. Si una aplicación tiene éxito en iniciar sesión / cerrar sesión, otras aplicaciones iniciarán sesión / cerrarán sesión automáticamente.
Características claves del uso del protocolo CAS.	- Jasig CAS - SSO (Inicio de sesión único) - SSO (Cierre de sesión único) - Protocolo LDAP - Servidor OpenLDAP - SSL

Autenticación única e interoperabilidad.	- El sistema de inicio de sesión en el cliente CAS utiliza un mecanismo de verificación de validez de ticket para obtener datos de nombre de usuario a través del protocolo CAS, que se usa para iniciar sesión en las aplicaciones y crear sesiones de aplicaciones locales. - CAS Servidor de autenticación utilizando tickets mediante el almacenamiento de cookies de concesión de vales (TGC) en la cookie de autenticación como prueba en el navegador y envía el vale de servicio (ST) para la aplicación Web cliente CAS como la autenticación de la prueba y la sustitución de contraseñas. - Aplicaciones Web basadas en PHP.
A10. Analisis Teknologi Single Sign On (Sso) Dengan Penerapan Central Authentication Service (Cas) Pada Universitas Bina Darma	
Resumen	El protocolo CAS SSO está destinado a otorgar al usuario permiso para acceder a múltiples aplicaciones, al tiempo que proporciona credenciales de usuario (como identificación de usuario y contraseña) una sola vez y permite que las aplicaciones Web autentiquen usuarios sin hacer clic. Esto puede facilitar al usuario el uso de las aplicaciones existentes y también para facilitar la organización de los datos del usuario, de modo que la seguridad de los datos del usuario sea más segura, ya que utiliza un almacenamiento centralizado de los datos del usuario.
Características claves del uso del protocolo CAS.	- Jasig CAS ▪ CAS SERVER - SSO - Protocolo LDAP - Servidor OpenLDAP - Requiere HTTPS
Autenticación única e interoperabilidad.	- Utiliza Cookies - TGC Ticket que se almacena en la cookie CASTGC, representa una sesión de SSO para un usuario. - Base de datos de las aplicaciones administrada por PhpMyAdmin. - Biblioteca PhpCAS que se utiliza como una biblioteca de cliente.
A11. Sistem Single Sign On Universitas Berbasis Cas-Ldap.	
Resumen	CAS es un sistema de autenticación que está creado originalmente por la Universidad de Yale para proporcionar un camino seguro para que una aplicación pueda volver a autenticar a un usuario automaticamente. En la fase de fabricación, CAS tiene algunas características básicas del servicio, incluyendo:

	SSO para facilitar el inicio de sesión único para diversas aplicaciones Web.
Características claves del uso del protocolo CAS.	- Jasig CAS - SSO (Inicio de sesión único). - SSO (Cierre de sesión único). - Protocolo LDAP - Servidor OpenLDAP
Autenticación única e interoperabilidad.	- Uso de cookies de otorgamiento de tickets, que identificará al usuario después que realice un inicio de sesión. - Utiliza HTTPS - Biblioteca PhpCAS que se utiliza como una biblioteca de cliente.
A12. Single Sign-On Research And Expansion Based On CAS.	
Resumen	SSO se ha convertido en una solución importante para la integración empresarial de negocios populares en la actualidad. CAS tiene una arquitectura razonable y un soporte para la interfaz rica, también se admite el uso compartido de cookies de varios dominios. Este sistema permite una expansión del protocolo de autenticación CAS para resolver las dificultades de integración de negocio complejas para el logro de un solo sistema de inicio de sesión SSO.
Características claves del uso del protocolo CAS.	- YaleCAS (Jasig CAS) - SSO - Protocolo LDAP
Autenticación única e interoperabilidad.	- Uso de Cookies de otorgamiento de tickets - Protocolo HTTPS - Java Servlet - Entradas incluyendo TGC, ST
A13. Sistema Integrado de Autenticación para la Universidad Tecnológica de Bolívar- MiUTB.	
Resumen	CAS utiliza un inicio de sesión único SSO, permite a un usuario acceder a múltiples servicios, o sistemas de aplicación después de ser autenticado solo una vez. Integrando múltiples servidores de base de datos o directorios activos.

Características claves del uso del protocolo CAS.	- Jasig CAS
	  ▪ CAS Clients
	  ▪ CAS Server
	- Integra múltiples servidores: LDAP y Base de datos (Mysql, Postgresql), X.509, SPNEGO.
	- Soporte de clientes multiplataforma (Moodle, Wordpress, ASP, Java, .Net, PHP, Perl, Apache, etc).

	- Integrar multiples aplicaciones en distintas tecnologías o ambientes de desarrollo.
Mecanismos de Autenticación única e interoperabilidad.	- TGT Ticket de acceso a través de un login exitoso. - ST Ticket de servicio emitido por petición del usuario mediante el navegador. - Las aplicaciones deben hacerse bajo el protocolo HTTPS.
A14. Authentication Component for Dynamic Report Generator.	
Resumen	CAS es una solución libre, robusta y probada para escenarios de autenticación centralizada, que suscribe varias aplicaciones compartiendo un formulario de autenticación común. La autenticación centralizada permite a los usuarios acceder a varias aplicaciones introduciendo sus credenciales una sola vez por medio de un formulario de autenticación único.
Características claves del uso del protocolo CAS.	- Multiprotocolo - Librerías implementadas en varios lenguajes de programación. - Servidor LDAP - Servidor CAS
Mecanismos de Autenticación única e interoperabilidad.	Como política para asegurar el SSO el formulario de inicio de sesión, debe estar publicado bajo el protocolo HTTPS.
A15. Central Authentication Service (CAS) SSO For EMC® Documentum® Rest Services.	
Resumen	CAS es un protocolo de inicio de sesión único para la Web. Su propósito es permitir a un usuario acceder a múltiples aplicaciones al tiempo que proporciona sus credenciales (tales como identificación de usuario y contraseña) sólo una vez.
Características claves del uso del protocolo CAS.	- Jasig CAS - Servidor REST - Servidor LDAP - Navegador Web del cliente - Protocolo Secure Sockets Layer (SSL) - Se puede integrar con otros esquemas de autenticación como: LDAP, Kerberos SSO, SAML SSO y así sucesivamente.
Mecanismos de Autenticación única e interoperabilidad.	- TGT Un ticket producido por CAS y retenido por el usuario como una identidad autenticada. - ST Cada ST se utiliza sólo una vez para acceder a un servicio específico. - TGC Ticket que se almacena en la cookie CASTGC, representa una sesión de SSO para un usuario. - PGT Proxy de concesión.

	- PGTIOU Un ticket enviado por CAS solo en una respuesta de validación del servicio, y con un PGT a la URL de devolución de llamada. - PT Utilizable por un proxy para acceder a un sistema haciéndose pasar por un único usuario. - CT Token recibido por el servidor para no pasar por el CAS SSO de nuevo. - LT Contraseña temporal para un usuario autenticado.
A16. Implementation of CAS Server as Authentication Protocol on Single Sign-On (SSO) Network With PHP Programming.	
Resumen	El servicio de autenticación central (CAS) es un sistema de autenticación creado originalmente por la Universidad de Yale para proporcionar la autenticación única a un usuario. El CAS, implementado como un componente de servidor Java de código abierto y con un soporte de biblioteca cliente para Java, PHP, Perl, Apache, uPortal, y otros. - CAS es un diseño de código abierto de sesión único que describe sobre el uso del cifrado de protocolo https para cifrar la comunicación como medida de seguridad, el ahorro de la identidad de las facturas en el centro de autentificación, sin el navegador para guardar las cookies.
Características claves del uso del protocolo CAS.	- SSO (Inicio de sesión único). - SSO (Cierre de sesión único). - Servidor CAS desarrollado en el lenguaje de programación PHP - Base de datos MySQL
Autenticación única e interoperabilidad.	- Utiliza Cookies para guardar información de un usuario - Utiliza HTTPS - Apache Banch para comprobar el rendimiento de servidor CAS

Anexo 6. Artículo indexado en Knowledge E

SLR DEL Protocolo LDAP

KnE Engineering

SIIPRIN-CITEGC
Ibero-American Symposium on Computer Programming jointly held with
the International Congress on Technology Education and Knowledge Management
Volume 2018

Knowledge E
Engaging minds

Conference Paper

Systematic Literature Review on the LDAP Protocol As a Centralized Mechanism for the Authentication of Users in Multiple Systems

Revisión Sistemática de Literatura sobre el protocolo LDAP como mecanismo centralizado para la autentificación de usuarios en múltiples sistemas

Mario Cueva-Hurtado, Roberth Figueroa-Diaz, Wilmer Aguilar-Soto, and Manuel Armijos-Ordoñez

Universidad Nacional de Loja, Carrera de Ingeniería en Sistemas, Av. Pío Jaramillo Alvarado, La Argelia, Loja, Ecuador

Corresponding Author:
Mario Cueva-Hurtado
mecueva@unl.edu.ec

Received: 4 December 2018
Accepted: 5 December 2018
Published: 27 December 2018

Publishing services provided by Knowledge E

Selection and Peer-review under the responsibility of the SIIPRIN-CITEGC Conference Committee.

Abstract

The protocol LDAP (Lightweight Directory Access Protocol) allows centralized identity authentication, where the information of the directory is faster and easier to read. This article carries out a systematic literature review (SLR) according to what is proposed in the article by Bárbara Kitchenham [1], aimed to identify different methods for users' authentication in multiple systems using LDAP protocol, an analysis of criteria is carried out about different studies published in five digital libraries (Scopus, IEEEXplorer, Scientific.net, Google Scholar, DBLP), and two academic magazines (Revista Energía of UNL, Revista Científica of UTB), making relevant conclusions of the use of four mechanisms for the authentication of users of multiple systems such as: Languaje PHP, SSO (Single sign-on), IAM (Identity and Access Management), and T-RBAC (Access control based on roles and tasks), predominantly the use of the PHP language for its administrative tools for managing LDAP servers.

Resumen

El protocolo LDAP (Protocolo Ligero de Acceso a Directorios), permite la autentificación de identidad centralizada, donde la información del directorio es más rápida y fácil de leer. El presente artículo realiza una revisión sistemática de literatura (SLR) de acuerdo a lo propuesto en el artículo de Bárbara kitchenham [1], con el fin de identificar los diferentes métodos para la autentificación de usuarios en múltiples sistemas usando el protocolo LDAP, se realiza un análisis de criterios de diferentes estudios publicados en 5 bibliotecas digitales (Scopus, IEEEXplorer, Scientific.net, Google Scholar, DBLP), y 2 revistas académicas (Revista Energía de la UNL, Revista científica de la UTB), realizando conclusiones relevantes del uso de cuatro mecanismos para la autentificación de usuarios de múltiples sistemas como son: lenguaje PHP, SSO (Sistema de autentificación única), IAM (Sistema de gestión de identidades) y T-RBAC

OPEN ACCESS

(Control de acceso basado en tareas y roles); predominando el uso del lenguaje PHP por sus herramientas administrativas para la administración de servidores LDAP.

Keywords: LDAP, authentication, user management, systematic literature review, security

Palabras clave: LDAP, autentificación, gestión de usuarios, revisión sistemática de literatura, seguridad

1. Introduction

En los últimos años el uso y servicios de las tecnologías de la información y comunicación (TIC), han evolucionada constantemente, dichos servicios tienen su impacto directamente en la seguridad de la información, acceso a los recursos y datos personales.

En la actualidad cuando un usuario desea ingresar a una aplicación, sistema o información relevante, este debe autentificarse para poder verificar su identidad, generalmente ingresando un ID de usuario y una contraseña, mediante el cual se puede autorizar o denegar el permiso, pero el problema principal radica en que si un usuario utiliza o maneja varios sistemas o aplicaciones este deberá utilizar o manejar muchos ID de usuario y contraseñas, lo cual es bastante incómodo para los mismos.

LDAP (Protocolo Ligero de Acceso a Directorios), permite el acceso a un servicio de directorio ordenado y distribuido especialmente basado en el estándar X.500 para compartir directorios [2], LDAP se ejecuta sobre TCP/IP u otros servicios de transferencia orientados a conexión, utiliza un modelo de cliente - servidor. LDAP maneja su estructura de forma jerárquica, la forma de representar su directorio se llama DIT (Árbol de Información del Directorio) como lo indica [2] y se amplia en [3].

En la actualidad se usa LDAPv3 que se desarrolló a finales de la década de 1990 para reemplazar a LDAPv2. LDAPv3 agrega las siguientes características como se indica en [2]:

- Fuertes servicios de autenticación y seguridad de datos a través de SASL (Capa de Seguridad y Autenticación Simple)
- Certificación de autenticación y servicios de seguridad de datos a través de TLS (Seguridad de la Capa de Transporte)

- Internacionalización mediante el uso de Unicode
- Referencias y continuaciones
- Descubrimiento de esquema
- Extensibilidad (controles, operaciones extendidas entre otros).

En este sentido en [2, 3] se amplia el protocolo LDAP como mecanismo centralizado, permitiendo la integración de diferentes sistemas con una autentificación única y brindando la proteccion a los datos confidenciales mediante protocolos de comunicación segura como: SASL (Capa de autenticación y seguridad), TLS (Seguridad de la capa de transporte) y SSL (Capa de conexión segura).

El propósito de este artículo muestra el resultado de una revisión sistemática de literatura, donde enfatiza principalmente la forma en que LDAP se integra a múltiples sistemas o aplicaciones, sin dejar de lado la seguridad que conlleva el proceso. Las siguientes secciones están organizadas de la siguiente manera: En la sección 2 se describe la metodología a seguir, en la sección 3 se presentan los resultados de la revisión sistemática de literatura, en la sección 4 se presenta la discusión e interpretación de los resultados y en la sección 5 se presentan las conclusiones que responden a nuestras preguntas de investigación.

2. Metodología

En el presente artículo se aplica la metodología propuesta por Bárbara Kitchenham [1], la cual parte del estudio de las pruebas disponibles sobre una determinada intervención, con el objeto de responder a cuestiones concretas, siguiendo una metodología explícita y rigurosa, descrita en la Tabla 1, donde se puede visualizar el esquema a seguir.

3. Resultados

3.1. Planificación de la Revisión Sistemática de Literatura

En esta etapa se identifica cuáles son las necesidades de la revisión, para lo cual es necesario plantear y responder algunos puntos claves descritos a continuación.

1. Objetivo de la Revisión Sistemática de Literatura

Tabla 1: Proceso de Revisión Sistemática de Literatura.

Fase	Paso
A. Planificación de la Revisión Sistemática de Literatura	1. Objetivo de la Revisión Sistemática de Literatura
	2. Formulación de la pregunta de investigación
	3. Palabras Claves
	4. Selección de fuentes y estrategias de búsqueda
	5. Cadena de búsqueda
	6. Criterios de inclusión
	7. Criterios de exclusion
B. Ejecución de la Revisión Sistemática de literatura	1.Criterios de selección de estudios
	2. Extracción de la información
C. Análisis de resultados y hallazgos	1. Hallazgos

El objetivo es poder identificar y seleccionar los artículos científicos que tengan relevancia sobre el protocolo LDAP y la autentificación de usuarios, que otorguen información fundamental al artículo de revisión.

2. Formulación de la pregunta de investigación

A partir de tema de investigación "Protocolo LDAP como mecanismo centralizado para la autentificación de usuarios en múltiples sistemas", se plantea las siguientes preguntas de investigación referente al tema antes mencionado, descrita en la Tabla 2, donde P1 es la primera pregunta y P2 la segunda pregunta.

Tabla 2: Preguntas de investigación.

Preguntas
P1. ¿Por qué se utiliza el protocolo LDAP como mecanismo centralizado para la autentificación de usuarios en múltiples sistemas?
P2. ¿La autentificación con el protocolo LDAP, mejoró la seguridad y administración de usuarios?

3. Palabras claves

Se identifican las palabras claves de los artículos científicos propuestos por el grupo de investigadores, las cuales servirán para poder plantear la cadena de búsqueda, descritos en la Tabla 3, que enumera los 10 artículos seleccionados.

4. Selección de fuentes y estrategias de búsqueda

Se seleccionó un conjunto de bases de datos científicas y revistas académicas, en donde se procedió a buscar los artículos científicos descritos en la Tabla 4, visualizando el nombre y URL de cada una.

Tabla 3: Artículos Preliminares y Palabras Claves.

No.	Artículo	Palabras claves
A1	Profile Management and Authentication using LDAP. [4]	LDAP, client, server, SLADP, schemas, database, ACLs, authentication, LDIF, Web Server, VPN, RAS, SendMail.
A2	Inventions on LDAP-A study based on US Patents. [5]	LDAP, Directory Protocol, Inventions, Software Inventions, LDAP inventions, Software Patents.
A3	User Management with LDAP (Lightweight Directory Access Protocol) for access to technology and Information Services in Companies. [6]	LDAP, user management, integration, synchronization services.
A4	User identity & lifecycle management using LDAP directory server on distributed network. [7]	LDAP, authentication, security, policies, servers, internet, identity management systems, user access
A5	Intelligent agents applied to the management of LDAP user profiles. [8]	LDAP, interface, intelligent agents, Smart.
A6	Vulnerabilities of LDAP as an Authentication Service. [9]	LDAP, Authentication service, Denial.of-service, SYN Flooding.
A7	External authentication approach for virtual private network using LDAP. [10]	Authentication, LDAP, servers, protocolos, access protocol, external authentication, internet protocol security.
A8	Improve data security in cloud environment by using LDAP and two way encryption algorithm. [11]	Cloud computing, Encryption, Servers, Authentication, IP networks, Algorithm desing and analysis, LDAD, authorisation, security.
A9	Authentication using LDAP in Wireless Body Area Network. [12]	Authentication, security, Wireless Body Area Network, Attack.
A10	OCL Fault Injection-Based Detection of LDAP Query Injection Vulnerabilities.[13]	LDAP, Servers, authentication, detection, privilege, security.

Tabla 4: Fuentes de motores de búsqueda científica y revistas académicas.

Fuentes	URL
Scopus	https://www.scopus.com/
IEEEXPlore	http://ieeexplore.ieee.org/
Scientific.net	https://www.scientific.net
Scholar Google	http://scholar.google.es/
DBLP	http://dblp.uni-trier.de/
Revista Energía de la UNL	http://revistas.unl.edu.ec
Revista Científica de la UTB	http://revistas.utb.edu.ec

5. Cadena de Búsqueda

Para generar la cadena de búsqueda se utilizó los conectores lógicos "AND" y "OR", descrita en la Tabla 5, donde se puede visualizar la cadena resultante.

Tabla 5: Cadena de búsqueda.

Cadena
(("LDAP") AND ("Authentication") AND ("User management OR User") AND ("Security OR internet protocol security")).

6. Criterios de Inclusión

Se realizó un estudio de los artículos más relevantes, excluyendo a los demás, se tomó en cuenta los descritos en la Tabla 6, donde se puede visualizar los criterios de inclusión.

Tabla 6: Criterios de Inclusión.

Criterio	Valor
Idioma	Inglés, español.
Motores de búsqueda	Scopus, IEEEXplorer, Scientific.net, Google Scholar, DBLP, Revistas académicas.
Fecha de publicación	2013 – 2018.
Tipo de producciones	Artículos científicos.

7. Criterios de Exclusión

Los estudios que no han sido relevantes se los descarta tomando en consideración el siguiente criterio: Título, abstract (resumen), texto completo del documento, palabras claves, resultados y conclusiones.

3.2. Ejecución de la Revisión Sistemática de literatura

A continuación, se describe los criterios de selección de los estudios más importantes, extracción de la información y las cadenas de búsqueda aplicadas en las fuentes científicas.

3.2.1. Ejecución en la base de datos Scopus

(("LDAP") AND ("Authentication") AND ("Security")).

La ejecución de la cadena de búsqueda en Scopus arrojó 5459 resultados. Luego al aplicar el criterio de inclusión se obtuvo 18 documentos relevantes, de los cuales, aplicando el criterio de exclusión se consideran los descritos en la Tabla 7, que muestra los estudios primarios y sus referencias bibliográficas.

Tabla 7: Estudios primarios de Scopus.

#	art. cit.	Título y Publicación
1	[7]	User identity & lifecycle management using LDAP directory server on distributed network. 2015 International Conference on Pervasive Computing: Advance Communication Technology and Application for Society, ICPC 2015.
2	[10]	External Authentication Approach for Virtual Private Network using LDAP. 2014 First International Conference on Networks & Soft Computing (ICNSC2014).

3.2.2. Ejecución en la base de datos IeeeXplorer

(("LDAP") AND ("Authentication") AND ("Security")).

La ejecución de la cadena de búsqueda en IEEEXplorer arrojó 36 resultados. Luego al aplicar el criterio de inclusión se obtuvo 14 documentos relevantes, de los cuales, aplicando el criterio de exclusión se consideran los descritos en la Tabla 8, que muestra los estudios primarios y sus referencias bibliograficas.

Tabla 8: Estudios primarios de la IEEEXplorer.

#	art. cit.	Título y Publicación
3	[11]	Improve data security in cloud environment by using LDAP and two way encryption algorithm. 2016 Symposium on Colossal Data Analysis and Networking (CDAN).
4	[14]	Research on data and workflow security of electronic military systems. Proceedings of the 2013 International Conference on Intelligent Control and Information Processing, ICICIP 2013.
5	[15]	Study and design of enterprise public security platform based on PKI. Proceedings - 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science, DCABES 2014.
6	[16]	Classification of Lightweight Directory Access Protocol Query Injection Attacks and Mitigation Techniques. 978-1-4673-7648-8/15/$31.00 ©2015 IEEE
7	[17]	Linux PAM to LDAP Authentication Migration. 2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT).

3.2.3. Ejecución de la selección de fuentes en Scientific.net

((LDAP) AND (Authentication)).

La ejecución de la cadena de búsqueda en Scientific.net arrojó 44 resultados. Luego al aplicar el criterio de inclusión se obtubo 25 documentos relevantes, de los cuales, aplicando el criterio de exclusión se consideran los descritos en la Tabla 9, que muestra los estudios primarios y sus referencias bibliograficas.

Tabla 9: Estudios primarios de Scientific.net.

#	art. cit.	Título y Publicación
8	[18]	Research of Unified Authentication System Based on LDAP. Published by Atlantis Press, Paris, France.
9	[19]	Design and Implementation of unified Identity Authentication System Based on LDAP in Digital Campus. Advanced Materials Research

3.2.4. Ejecución en el motor de búsqueda científico Google Scholar

Para realizar la consulta en esta fuente, seleccionamos la búsqueda por título introduciendo palaras claves como (Autentificación con LDAP), el cuál dio como resultado 2.800 documentos, aplicando el criterio de inclusión se obtubo 1.290 documentos relevantes y aplicando el criterio de exclusión y página de referencia se considera el descrito en la Tabla 10, que muestra el estudio primario y su referencia bibliografica.

Tabla 10: Estudios primarios de Google Scholar.

#	art. cit.	Título y Publicación
10	[20]	Sistem Autentikasi Hotspot Menggunakan LDAP dan Radius pada Jaringan Internet Wireless Prodi Teknik Sistem Komputer. Jurnal Teknologi dan Sistem Komputer

3.2.5. Ejecución en la base de datos DBLP

(LDAP).

La ejecución de la cadena de búsqueda en DPLB arrojó 163 resultados. Luego al aplicar el criterio de inclusión se obtuvo 8 documentos relevantes, de los cuales, aplicando el criterio de exclusión se considera el descrito en la Tabla 11, que muestra el estudio primario y su referencia bibliografica.

Tabla 11: Estudios primarios de DBLP.

#	art. cit.	Título y Publicación
11	[21]	Selective LDAP Multi-Master Replication. Proceedings Open Identity Summit 2013. Open Identity Summit (OID-2013), September 9-11, Kloster Banz, Germany.

3.2.6. Ejecución en las revistas académicas de la UNL y UTB.

La ejecución de la cadena de búsqueda en las revistas académicas, aplicando el criterio de inclusión y exclusión arrojó 3 resultados relevantes descritos en la Tabla 12, que muestra los estudios primarios y sus referencias bibliograficas.

TABLA 12: Estudios primarios de Revista Energía de la UNL y Revista Científica de la UTB.

#	art. cit.	Título y Publicación
12	[8]	Intelligent agents applied to the management of ldap user profiles. Revista Energía ISSN: 1390-9037.
13	[22]	Implemetation of Eduroam as Wireless Infraestructure on the Campus of National University of Loja. Revista Energía ISSN: 1390-9037.
14	[6]	User Management with LDAP (Lightweight Directory Access Protocol) for access to technology and Information Services in Companies. Journal Of Science And Research: Revista Ciencia E Investigación, E-Issn: 2528-8083, Vol. 1, Citt, Pp. 10-15.

1. Criterios de selección de estudios

   Para el cumplimiento del objetivo principal, los resultados de búsqueda deben cumplir el siguiente criterio de selección: Los artículos deben destacar la importancia y beneficios del uso del protocolo LDAP para la autenticación de usuarios.

2. Extracción de la información

   Los criterios dados de inclusión, exclusión y de selección, permitieron identificar los diferentes artículos con el fin de cumplir el objetivo planteado en esta investigación. Para la extracción importante de cada estudio se utilizó los siguientes elementos:

   - Características claves de la autentificación de LDAP
   - Mecanismos de interrelación con diferentes sistemas.

3.3. Análisis de resultados y hallazgos

Se realizó un análisis previo donde se evalúa cada estudio, discriminando artículos que tienen criterios comunes e información no trascendental, estos fueron descartados quedándose con los artículos más relevantes.

Se enlistan 14 artículos con las etiquetas A1 a A14, que son los estudios seleccionados de acuerdo a los criterios indicados, presentados en la Tabla 13.

Tabla 13: Estudios seleccionados de la Revisión Sistemática de Literatura.

A1. User identity & lifecycle management using LDAP directory server on distributed network. [7]	
Características claves del uso del protocolo LDAP	Almacena y gestiona datos de roles asignados como: identidades de usuario, contraseñas y políticas.
Mecanismos de interrelación con otros sistemas.	Utiliza un sistema de gestión de identidades (IAM), el cual determina quién puede ingresar a sus sistemas protegidos, a qué nivel puede acceder el usuario y también asegurará que los usuarios accedan sólo a lo que necesitan para sus tareas comerciales mediante un protocolo DAML (Lenguaje de marcado de acceso al directorio), para transferir datos entre el servidor de Identity Manager y los servicios de LDAP.
A2. External Authentication Approach for Virtual Private Network using LDAP. [10]	
Características claves del uso del protocolo LDAP	-Inicio de sesión con credenciales de administrador. -Almacena y gestiona datos de roles asignados como identidades de usuario, contraseñas y políticas. -Para la autenticación utiliza los protocolos: SSL (Capa de sockets seguros) y TLS (Seguridad de la capa de transporte). -Autenticación externa mediante VPN.
Mecanismos de interrelación con otros sistemas.	Se implementa una Red privada virtual (VPN), utilizando el protocolo de tunelización punto a punto (PPTP) para mejorar la seguridad de las credenciales del administrador, enviando una solicitud al servidor LDAP para una correcta conexión a los sistemas.
A3. Improve data security in cloud environment by using LDAP and two way encryption algorithm. [11]	
Características claves del uso del protocolo LDAP	-Almacena y gestiona datos de roles asignados como: identidades de usuario, contraseñas y políticas. -Protocolos de Autenticación. -Multiplataforma (puede ejecutarse en varios sistemas operativos).
Mecanismos de interrelación con otros sistemas.	Se implementan 2 métodos descritos a continuación: El primer método lo realiza mediante un enlace simple y TLS (Seguridad de la capa de transporte), para evitar la divulgación de contraseñas en la red. El segundo método lo realiza mediante la autenticación simple y capa de seguridad SASL (Capa de seguridad y autenticación simple), que junto con los certificados del lado del cliente y TLS (Seguridad de la capa de transporte) proporcionan la protección más completa.
A4. Research on data and workflow security of electronic military systems. [14]	
Características claves del uso del protocolo LDAP	-Protocolos de Autenticación -Almacena y gestiona datos de roles asignados como: identidades de usuario, contraseñas y políticas.
Mecanismos de interrelación con otros sistemas.	Para conectarse al servidor LDAP se implementa un mecanismo de control de acceso T-RBAC (control de acceso basado en tareas y roles), y un componente de autenticación independiente basado en PKI (Infraestructura de clave pública), obteniendo así una conexión segura. T-RBAC es el mecanismo encargado de las funciones de envío, edición y composición de roles y tareas, garantizando un acceso correcto en la conexión con el LDAP y la seguridad del mismo.
A5. Study and design of enterprise public security platform based on PKI. [15]	

Características claves del uso del protocolo LDAP	-Almacena y gestiona datos de roles asignados como: identidades de usuario, contraseñas y políticas. -Eficiencia de consulta -Marco de despliegue distribuido -Control de acceso flexible y preciso.
Mecanismos de interrelación con otros sistemas.	Para la conexión de las aplicaciones móviles con el servidor LDAP, utiliza KMC SERVER Y CA SERVER que permiten la autenticación segura de las apps, las cuales son compatibles con el certificado digital X.509 de tecnología PKI (Infraestructura de Clave Pública) y el certificado digital X.509 de WPKI (Infraestructura de clave pública inalámbrica).
A6. Classification of Lightweight Directory Access Protocol Query Injection Attacks and Mitigation Techniques. [16]	
Características claves del uso del protocolo LDAP	-LDAP se puede usar para agregar, modificar y borrar información junto con operaciones de búsqueda. -Almacena y gestiona datos de roles asignados como: identidades de usuario, contraseñas y políticas. -Restringir el acceso mediante el uso de IP
A7. Linux PAM to LDAP Authentication Migration. [17]	
Características claves del uso del protocolo LDAP	-Almacena y gestiona datos de roles asignados como: identidades de usuario, contraseñas y políticas. -Para la encriptación de mensajes se utiliza los protocolos: SSL (Capa de sockets seguros) y TLS (Seguridad de la capa de transporte). -Multiplataforma (Puede ejecutarse en varios sistemas operativos). -LDIF (Formato de intercambio de datos en LDAPv3). -LDAP acepta métodos de almacenamiento de contraseñas y transformación de claves con valores hash MD5 que son: SMD5, - Crypt, SHA y SSHA este es el más seguro. -Sistema de base de datos backend -OID Identificador único de objeto para asignar los atributos. -LDAP utiliza AC que es la configuración de permiso para cuenta existente. -Apache Bench herramienta para determinar el tiempo de respuesta del servidor LDAP.
Mecanismos de interrelación con otros sistemas.	-LDAP cuenta con una biblioteca adicional para PHP el cual tiene un módulo de conexión a las aplicaciones creadas en PHP estás pueden acceder a la información de los usuarios que se encuentran en el servidor LDAP, para las nuevas contraseñas utiliza una función de PHP que incorpora un generador SHA1 (Algoritmo de Hash Seguro 1). -Para la migración de contraseñas de un servidor a otro se utiliza la herramienta Migrationtools que permite trabajar con LDAP.
A8. Research of Unified Authentication System Based on LDAP. [23]	
Características claves del uso del protocolo LDAP	-Almacena y gestiona datos de roles asignados como: identidades de usuario, contraseñas y políticas. -Directorio centralizado -Protocolos de autenticación. -Directorio Base de la Información (DIB) ofrece acceso a directorios estándar para todo tipo de aplicaciones basada en LDAP. -LDIF (Formato de intercambio de datos)

Mecanismos de interrelación con otros sistemas.	-Se puede acceder a las diferentes aplicaciones con SSO (Sistema de autenticación única). -Para importar y exportar la información de los usuarios, utiliza ICE Novel que es un kit de herramientas para la importación y exportación de datos de código abierto (como el Navegador JXplore de interfaz gráfica), que se incorporan con LDAP. -Utiliza un IDM (Novell Identity Management) que es una guía de identidad sincronizada con la base de datos, directorio y aplicación estándar.
A9. Design and Implementation of unified Identity Authentication System Based on LDAP in Digital Campus. [24]	
Características claves del uso del protocolo LDAP	-Estándar de información unificada -Plataforma de aplicaciones unificada -LDAP trabaja con 4 modelos básicos: modelo de información (representa la información), modelo de organización de datos, modelo de acceso-operación de datos y modelo de seguridad. -Bloquear y desbloquear cuentas de usuario. -Bloqueo y desbloqueo de tiempo. -Restablecer contraseñas. -Creación y eliminación de grupos de usuarios. -Estándar x.500 -LDAPv3
Mecanismos de interrelación con otros sistemas.	-Se utiliza ApacheDS (Servidor de Directorios) para conectarse con el servidor LDAP y se puede acceder a las diferentes aplicaciones con SSO (Sistema de autenticación única). -LDAP simplifica operaciones utilizando el patrón Spring's JdbcTemplat para la gestión de operaciones y base de datos al conectar los sistemas con LDAP, este patrón utiliza el Interprete de Ordenes Seguro SSH2 (Struts2 + Spring + Hibernate) que es una estructura de 3 niveles y sirve para una conexión segura.
A10. Sistem Autentikasi Hotspot Menggunakan LDAP dan Radius pada Jaringan Internet Wireless Prodi Teknik Sistem Komputer. [20]	
Características claves del uso del protocolo LDAP	-Almacena y gestiona datos de roles asignados como: identidades de usuario, contraseñas y políticas. - Interfaz phpLdapadmin. - Protocolo de autenticación. - Directorio centralizado
Mecanismos de interrelación con otros sistemas.	Para la conexión de diversas computadoras mediante la red inalámbrica Wireless se utiliza FreeRADIUS este incluye un servidor RADIUS, cuenta con un archivo de configuración llamado users donde se agregan los usuarios para acceso a la red wireless con el servidor LDAP.
A11. Selective LDAP Multi-Master Replication. [21]	
Características claves del uso del protocolo LDAP	-Coherencia de datos duplicados -Almacena y gestiona datos de roles asignados como: identidades de usuario, contraseñas y políticas.
A12. Intelligent agents applied to the management of ldap user profiles. [8]	
Características claves del uso del protocolo LDAP	-LDAPv3 -Multiplataforma (Puede ejecutarse en varios sistemas operativos). -Almacena y gestiona datos de roles asignados como: identidades de usuario, contraseñas y políticas. -Protocolos de autenticación.
Mecanismos de interrelación con otros sistemas.	LDAP cuenta con una biblioteca adicional para PHP el cual tiene un módulo de conexión a las aplicaciones creadas en PHP estás pueden acceder a la información de los usuarios que se encuentran en el servidor LDAP.

A13. Implemetation of Eduroam as Wireless Infraestructure on the Campus of National University of Loja. [22]	
Características claves del uso del protocolo LDAP	-Para la autenticación se utiliza los protocolos: SSL (Capa de sockets seguros) y TLS (Seguridad de la capa de transporte). -Almacena y gestiona datos de roles asignados como: identidades de usuario, contraseñas y políticas. -Multiplataforma (Puede ejecutarse en varios sistemas operativos).
Mecanismos de interrelación con otros sistemas.	Para la conexión de diversas computadoras mediante la red inalámbrica Wireless se utiliza FreeRADIUS este incluye un servidor RADIUS, cuenta con un archivo de configuración llamado users donde se agregan los usuarios para acceso a la red wireless con el servidor LDAP.
A14. User Management with LDAP (Lightweight Directory Access Protocol) for access to technology and Information Services in Companies. [6]	
Características claves del uso del protocolo LDAP	-Almacena y gestiona datos de roles asignados como: identidades de usuario, contraseñas y políticas. -Proporciona seguridad que impide el acceso no autorizado mediante protocolos de comunicación segura como: SSL (Capa de sockets seguros) y TLS (Seguridad de la capa de transporte). -Permite realizar replicas permitiendo tener varios servidores LDAP que almacenan contenido del mismo directorio. -Multiplataforma (Puede ejecutarse en varios sistemas operativos).
Mecanismos de interrelación con otros sistemas.	-LDAP cuenta con una biblioteca adicional para PHP, el cual tiene un módulo de conexión para los CRM, ERP y aplicaciones. Estás pueden acceder a la información de los usuarios que se encuentran en el servidor LDAP. El módulo desarrollado en PHP requiere lo siguiente para la conexión: *La URL del servidor LDAP *Definir el puerto que va a usar para la comunicación (Puerto 389 por defecto). -Para la integración de correo electrónico y LDAP se utiliza un servidor de correos ZIMBRA y se lo configura de la siguiente manera para poder establecer la conexión con el servidor LDAP: *Se define la URL del servidor LDAP y el puerto que va a utilizar para la comunicación (Por defecto 389 por defecto). *Se debe marcar en usar SSL (Capa de sockets seguros). -Para la conexión de diversas computadoras mediante la red inalámbrica Wireless se utiliza FreeRADIUS este incluye un servidor RADIUS, cuenta con un archivo de configuración llamado *users* donde se agregan los usuarios para acceso a la red wireless con el servidor LDAP.

4. Discusión

A continuación, se presentan los principales hallazgos que se encontraron al realizar la Revisión Sistemática:

Los artículos A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13 y A14 que se detallan en la Tabla 13, presentan las características claves de la autentificación con el protocolo LDAP, las cuales son: Almacenamiento y gestión de perfiles de usuario asignados, multiplataforma (puede ejecutarse en varios sistemas operativos); encriptación de

mensajes en la autentificación mediante los protocolos SSL (Capa de sockets seguros) y TLS (Seguridad de la capa de transporte). Las características claves de LDAP en los artículos revisados para mecanismos de interrelación con diferentes sistemas son: Biblioteca adicional de PHP incorporada en LDAP para una correcta conexión; FreeRADIUS que incluye un servidor RADIUS para la conexión inalámbrica con el servidor LDAP. Estos artículos especifican a LDAP como un mecanismo centralizado para compartir directorios y acceso a un servicio distribuido.

Los artículos A8 y A9 recomienda la utilización de un SSO (Sistema de Autenticación única) para acceder a las diferentes aplicaciones conectadas al servidor LDAP.

El artículo A7, menciona además que para mejorar la seguridad hace uso del algoritmo SHA-1(actualmente ya es obsoleto) para generar contraseñas mediante una función de PHP. A partir del año 2017 las comunicaciones usan el algoritmo SHA-2.

El artículo A6 indica la importancia de usar ICE Novel que es un kit de herramientas para la importación y exportación de datos que se incorporan con LDIF (Formato de intercambio de datos), el cual contiene los registros que se envían a un servidor LDAP.

El artículo A9 sugiere un aspecto alternativo para la simplificación de operaciones utilizando el patrón Spring's JdbcTemplat para la gestión de operaciones y base de datos al conectar los sistemas con LDAP.

Los artículos A7, A12 y A13 puntualizan la importancia que tiene LDAP al contar con una biblioteca adicional para PHP el cual tiene un módulo de conexión a las aplicaciones, estás pueden acceder a la información de los usuarios que se encuentran en el servidor LDAP.

Los artículos A2, A3, A7, A13 y A14 precisan el mérito de LDAP al trabajar con protocolos de seguridad en la autentificación y encriptación de mensajes como son: SSL (Capa de sockets seguros) y TLS (Seguridad de la capa de transporte).

El artículo A2 indica que el uso del protocolo LDAP mejoró la seguridad mediante una red virtual privada VPN, junto al protocolo de tunelización punto a punto (PPTP), así mismo el artículo A4 puntualiza la seguridad al conectarse al servidor LDAP, referente a un mecanismo de control de acceso T-RBACK y un componente de autenticación independiente basado en PKI.

5. Conclusiones

La utilización del protocolo LDAP, permite la integración de múltiples sistemas o aplicaciones desarrollados en diferentes lenguajes de programación, mediante librerías y módulos que cuentan con métodos para su integración con el servidor OpenLDAP.

El protocolo LDAP, garantiza la seguridad de los datos mediante la encriptación de mensajes asegurando la confidencialidad, integridad y autentificación en la comunicación usando protocolos SSL (Capa de sockets seguros) y TLS (Seguridad de la capa de transporte).

El uso del protocolo LDAP, como mecanismo centralizado para la integración de distintos sistemas o aplicaciones, ofrece varios mecanismos de conexión como son: lenguaje PHP, SSO (Sistema de autentificación única), IAM (Sistema de gestión de identidades) y T-RBAC (Control de acceso basado en tareas y roles); siendo más relevante el uso del lenguaje PHP por sus herramientas de gestión para la administración de servidores OpenLDAP.

El protocolo LDAP mejoró la administración y almacenamiento de la información de usuarios, debido a su mecanismo centralizado y jerárquico, el cual optimiza las operaciones de lectura rápida y de gran volumen; respecto a una base de datos relacional que se encuentra optimizada para manejo de transacciones.

Referencias

[1] Kitchenham, B.: Procedures for performing systematic reviews. Keele, UK, Keele Univ. 33, 28 (2004).

[2] OpenLDAP Software 2.4 Administrator's Guide, http://www.openldap.org/doc/admin24/guide.html#What is a directory service.

[3] Butcher, M.: Mastering OpenLDAP. (2007).

[4] Qadeer, M.A., Salim, M., Sana Akhtar, M.: Profile management and authentication using LDAP. Proc. - 2009 Int. Conf. Comput. Eng. Technol. ICCET 2009. 2, 247–251 (2009).

[5] Mishra, U.: Inventions on LDAP-A study based on US Patents. 1–15 (2014).

[6] Jose, M., Gonzáles, M., Epaña, Á.: User Management with LDAP (Lightweight Directory Access Protocol) for access to technology and Information Services in Companies ´. J. Sci. Res. Rev. Cienc. E Investig. ´ ON, E-ISSN 2528-8083, VOL. 1, CITT, PP. 10-15. 1, 10–15 (2016).

[7] Thakur, M.A., Gaikwad, R.: User identity & lifecycle management using LDAP directory server on distributed network. 2015 Int. Conf. Pervasive Comput. Adv. Commun. Technol. Appl. Soc. ICPC 2015. 00, 1–3 (2015).

[8] Espinoza, G., Ortega, P., Palacios, C., Junior, S.: Intelligent agents applied to the management of ldap user profiles, https://issuu.com/universidadnacionaldeloja/docs/revista_energ__a/91, (2014).

[9] Obimbo, C.: Vulnerabilities of LDAP As An Authentication Service. J. Inf. Secur. 02, 151–157 (2011).

[10] Shrivastava, A., Rizvi, M.A.: External authentication approach for virtual private network using LDAP. In: 2014 First International Conference on Networks & Soft Computing (ICNSC2014). pp. 50–54. IEEE (2014).

[11] Raipurkar, K. V., Deorankar, A. V.: Improve data security in cloud environment by using LDAP and two way encryption algorithm. In: 2016 Symposium on Colossal Data Analysis and Networking (CDAN). pp. 1–4. IEEE (2016).

[12] Dharme, W.S.: Authentication using LDAP in Wireless Body Area Network. 4, 235–239 (2017).

[13] Shahriar, H., Haddad, H.M., Bulusu, P.: OCL Fault Injection-Based Detection of LDAP Query Injection Vulnerabilities. Proc. - Int. Comput. Softw. Appl. Conf. 2, 455–460 (2016).

[14] Wang, W., Luo, H., Deng, H.: Research on data and workflow security of electronic military systems. Proc. 2013 Int. Conf. Intell. Control Inf. Process. ICICIP 2013. 705–709 (2013).

[15] Xiao, Y., Zhao, Y.: Study and design of enterprise public security platform based on PKI. Proc. - 13th Int. Symp. Distrib. Comput. Appl. to Business, Eng. Sci. DCABES 2014. 258–262 (2014).

[16] Bulusu, P., Shahriar, H., Haddad, H.M.: Classification of Lightweight Directory Access Protocol Query Injection Attacks and Mitigation Techniques. 337–344 (2015).

[17] Andjarwirawan, J., Palit, H.N., Salim, J.C.: Linux PAM to LDAP Authentication Migration. 2017 Int. Conf. Soft Comput. Intell. Syst. Inf. Technol. 155–159 (2017).

[18] Ming, J.: Research of Unified Authentication System Based on LDAP. 1044–1047 (2012).

[19] Zhiyuan Wu1, Z. edu. c., Weiping Huang1, H. edu. c., Lei Yu1, Y. edu. c.: Design and Implementation of unified Identity Authentication System Based on LDAP in Digital Campus. Adv. Mater. Res. 1213–1217 (2014).

[20] Muttaqin, A.H., Rochim, A.F., Widianto, E.D.: Sistem Autentikasi Hotspot Menggunakan LDAP dan Radius pada Jaringan Internet Wireless Prodi Teknik Sistem Komputer. J. Teknol. dan Sist. Komput. 4, 282–288 (2016).

[21] Bauereiß, T., Gohmann, S., Hutter, D., Kläser, A.: Selective LDAP Multi-Master Replication. Proc. Open Identity Summit 2013. Open Identity Summit (OID-2013), Sept. 9-11, Kloster Banz, Ger. 94–105 (2013).

[22] Loayza J, J., Castillo, J, F., Chamba, L, A.: Implemetation of Eduroam as Wirelless Infraestructure on the Campus of National University of Loja., https://issuu.com/universidadnacionaldeloja/docs/revista_energ__a/91, (2014).

[23] Ming, J.: Research of Unified Authentication System Based on LDAP. 1044–1047 (2012).

[24] Wu, Z., Huang, W., Yu, L.: Design and Implementation of unified Identity Authentication System Based on LDAP in Digital Campus. 1213–1217 (2014).

Anexo 7. Certificado de Participación en el SIIPRIN'2018

Otorgan el presente

CERTIFICADO

A: *Mario Cueva Hurtado, Roberth Figueroa Díaz, Wilmer Aguilar Soto y Manuel Armijos Ordóñez*

Como: **AUTORES DEL ARTICULO** titulado *"Revisión Sistemática de Literatura sobre el Protocolo LDAP como Mecanismo Centralizado para la Autentificación de Usuarios en Múltiples Sistemas"*, expuesto en el **III SIMPOSIO IBEROAMERICANO EN PROGRAMACIÓN INFORMÁTICA 2018 – CONGRESO INTERNACIONAL DE TECNOLOGÍA EDUCATIVA Y GESTIÓN DEL CONOCIMIENTO 2018**, evento realizado el 29 y 30 de noviembre del 2018.

Riobamba, 30 de noviembre de 2018

Ing. Washington Luna, Ph.D.
Decano FIE

Ing. Omar Gómez, Ph.D.
Coordinador de SIIPRIN' 18

Ing. Cosme Ortega, Ph.D.
Coordinador CITEGC'18

Ing. Luis Flores, Ph.D.
Director de Publicaciones

Ing. Tony Coronel Vásquez
Presidente del CIEECH

Evento aprobado según Resolución 430.CP.2018 / CÓDIGO: SI-PI-FIE-ESPOCH-2018-0018DP-015

JORNADAS ACADÉMICAS 2018

Anexo 8. Niveles del árbol Jerárquico para el presente TT.

Niveles de la Unidad Organizacional “personal”.

1. Nivel 1

En este nivel se define el dominio de nuestro servidor OpenLdap, y se establecen los siguientes ObjectClass y atributos descritos a continuación en la TABLA A8. I.

TABLA A8. I.
DESCRIPCIÓN DEL NIVEL 1 DEL ÁRBOL JERÁRQUICO

ObjectClass	Descripción	
top	Raíz de jerarquía. Heredan de esta clase	
organization	Se utiliza esta clase para poder agregar atributos y crear una entrada	
dcObject	Es una clase que empaqueta el atributo dc, para poder tomar el DN. Se la utiliza con la clase organization.	
Atributos	**Descripción**	**Campo**
dc	Componente del dominio, es un valor único. En nuestra propuesta es dc=cas,dc=com.	Obligatorio
o	Para organizaciones asociadas al dominio.	Obligatorio

2. Nivel 2

En este nivel se define el dominio de nuestro administrador para el servidor OpenLdap, y se establecen los siguientes ObjectClass y atributos, descritos a continuación en la TABLA A8. II.

TABLA A8. II.
DESCRIPCIÓN DEL NIVEL 2 DEL ÁRBOL JERÁRQUICO

ObjectClass	Descripción	
simpleSecurityObject	Es una clase para seguridad, se usa para permitir que una entrada tenga un atributo userPassword.	
organizationalRole	Define los procesos de trabajo con miembros de su organización y sus roles organizativos. En nuestra propuesta se lo utiliza para los roles del administrador.	
Atributos	**Descripción**	**Campo**
cn	Nombre común, que se definirá para el administrador.	Obligatorio
userPassword	Contraseña del administrador.	Obligatorio
description	Contiene la descripción del administrador.	Opcional

3. Nivel 3

En este nivel se establecen 2 unidades organizacionales: la primera denominada **personal** y la segunda denominado **universidad_implementacion**, el cual se lo puede adaptar al Orgánico Estructural de la UNL u otra organización que implemente el prototipo propuesto, como se describe en la TABLA A8. III.

TABLA A8. III.
DESCRIPCIÓN DEL NIVEL 3 DEL ÁRBOL JERÁRQUICO

ObjectClass	Descripción	
top	Raíz de jerarquía. Heredan de esta clase	
organizationalUnit	Clase para crear una Unidad Organizativa dentro de nuestro directorio.	
Atributos	**Descripción**	**Campo**
ou	Nombre con el cual se va a definir nuestra unidad organizativa dentro del directorio.	Obligatorio
description	Contiene la descripción de la unidad organizativa	Opcional

4. Nivel 4

En este nivel se establecen 3 unidades organizacionales: la primera denominada administrativos, la segunda denominado académicos, y la tercera denominada otros, como se describe en la TABLA A8. IV.

TABLA A8. IV.
DESCRIPCIÓN DEL NIVEL 4 DEL ÁRBOL JERÁRQUICO

ObjectClass	Descripción	
top	Raíz de jerarquía. Heredan de esta clase	
organizationalUnit	Clase para crear una Unidad Organizativa dentro de nuestro directorio.	
Atributos	**Descripción**	**Campo**
ou	Nombre con el cual se va a definir nuestra unidad organizativa dentro del directorio.	Obligatorio
description	Contiene la descripción de la unidad organizativa	Opcional

5. Nivel 5

En este nivel se establecen 4 unidades organizacionales: la primera denominada servidores, la segunda denominado trabajadores, la tercera denominada docentes, y la cuarta denominada estudiantes, como se describe en la TABLA A8. V.

TABLA A8. V.
DESCRIPCIÓN DEL NIVEL 5 DEL ÁRBOL JERÁRQUICO

ObjectClass	Descripción	
top	Raíz de jerarquía. Heredan de esta clase	
organizationalUnit	Clase para crear una Unidad Organizativa dentro de nuestro directorio.	
Atributos	**Descripción**	**Campo**
ou	Nombre con el cual se va a definir nuestra unidad organizativa dentro del directorio.	Obligatorio
description	Contiene la descripción de la unidad organizativa	Opcional

6. Nivel 6

En este nivel se establece el identificador único para cada usuario, y poder usarlo para su inicio de sesión a los distintos sistemas, como se describe en la TABLA A8. VI.

TABLA A8. VI.
DESCRIPCIÓN DEL NIVEL 6 DEL ÁRBOL JERÁRQUICO

ObjectClass	Descripción	
person	Hereda de la clase top.	
inetOrgPerson	Es una clase de propósito general los atributos que posee fueron elegidos para acomodar los requisitos de información que se encuentran en Internet típico e implementaciones del servicio de directorio de Intranet. Hereda de la clase oganizationalPerson.	
organizationalPerson	Hereda de la clase person. Aquí termina la jerarquía de estos objectClass	
top	Raíz de jerarquía. Heredan de esta clase	
eduPerson	Diseñados para incluir atributos de persona y organización ampliamente utilizados en la educación superior.	
shadowAccount	Se lo utiliza para que la cuenta del usuario se integre con la configuración de su password.	
Atributos	**Descripción**	**Campo**
uid	Se establece este atributo para definir el identificador único de cada usuario, en nuestra propuesta el identificador será creado automáticamente por el sistema.	Obligatorio
cn	Nombre común, en nuestra propuesta se ingresa los 2 nombres del usuario.	Obligatorio
sn	Apellidos del usuario	Obligatorio
eduPersonPrincipalName	Identificador de red que el usuario va a utilizar para la autenticación inter-institucional. Debería ser de la forma usuario@organización.com. En nuestra la propuesta será el uid.	Opcional

eduPersonTargetedId	Especifica un identificador único para una entrada al directorio. En nuestra propuesta será el DNI o cédula.	Opcional
givenName	Almacena el nombre de la persona, sin sus apellidos	Opcional
mail	Correo, en nuestra propuesta se debe ingresar un correo valido.	Opcional
ou	Nombre de la unida organizativa a la que pertenece el usuario.	Opcional
telephoneNumber	Número de teléfono del usuario	Opcional
userPassword	Contraseña del usuario. En nuestra propuesta este campo se lo considera Obligatorio. Además, utiliza el algoritmo de encriptación SSHA.	Obligatorio

Niveles de la Unidad Organizacional "universidad_implementación".

Los 3 primeros niveles, ya se encuentran definidos en el paso anterior, debido a que es la misma estructura. Los niveles 4 y 5 manejan la misma descripción de sus campos y los cuales se los define a continuación en el nivel 6:

1. Nivel 6

En este nivel se establecen unidades organizacionales, en base a los sub-grupos que pertenecen al nivel 5, se los definirá dependiendo de la organización, que desee implementar el prototipo propuesto, como se describe en la TABLA A8. VII.

TABLA A8. VII.

DESCRIPCIÓN DEL NIVEL 7 DEL ÁRBOL JERÁRQUICO

ObjectClass	Descripción	
top	Raíz de jerarquía. Heredan de esta clase	
organizationalUnit	Clase para crear una Unidad Organizativa dentro de nuestro directorio.	
Atributos	**Descripción**	**Campo**
ou	Nombre con el cual se va a definir nuestra unidad organizativa dentro del directorio.	Obligatorio
description	Contiene la descripción de la unidad organizativa	Opcional

2. Nivel 7

En este nivel se establece el nombre común para cada sistema, como se describe en la TABLA A8. VIII.

TABLA A8. VIII.

DESCRIPCIÓN DEL NIVEL 8 DEL ÁRBOL JERÁRQUICO

ObjectClass	Descripción	
posixGroup	Se utiliza esta clase para poner generar una lista de usuarios que tienen acceso a ese sistema agregando el atributo memberUid, el cual será el mismo que el identificador único y así no repetir la información de los usuarios en cada sistema.	
top	Raíz de jerarquía. Heredan de esta clase	
Atributos	**Descripción**	**Campo**
cn	En nuestra propuesta es el nombre común del sistema	Obligatorio
gidNumber	Es un identificador secundario, para identificar los grupos.	Obligatorio
memberUid	En el prototipo propuesto, corresponde con el uid de los usuarios a los que queremos que pertenezcan a este grupo, de esta forma, los usuarios formarán parte de este grupo como un grupo secundario.	Obligatorio
description	Contiene la descripción del cn o sistema.	Opcional

Anexo 9. Certificado de implantación del diseño jerárquico del servidor OpenLDAP para la autenticación de diferentes sistemas de la UNL.

Universidad Nacional de Loja

Unidad de Telecomunicaciones e Información

DIRECTOR DE TELECOMUNICACIONES E INFORMACIÓN

CERTIFICA:

Que el señor **Wilmer Antonio Aguilar Soto** con cédula de ciudadanía número **1900481878** y el señor **Manuel Stalin Armijos Ordóñez** con cédula de ciudadanía número **1105593238**; egresados de la Carrera de Ingeniería en Sistemas, han finalizado lo referente a la implantación del ***Diseño de una Estructura Jerárquica en el servidor (ldap.unl.edu.ec)*** y que actualmente se encuentran haciendo uso de la información almacenada los siguientes sistemas:

- Sistema de Información Académico Administrativo Financiero SIAAF
- Sistema de Gestión Académico - Estudiantes
- Sistema de Gestión Académico - Docentes
- Página web de la UNL
- EDUROAM
- OpenVPN
- Servicios de Cedia
- (FileSender, ZOOM)
- Sistema Jasig CAS

De su proyecto de titulación denominado ***"Desarrollo de un Prototipo para el Servicio de Autenticación Central de Usuarios en Aplicaciones Web"*** en la Unidad de Telecomunicaciones e Información, bajo los lineamientos y requerimientos establecidos por esta unidad administrativa de la Universidad Nacional de Loja.

Es cuanto puedo indicar en honor a la verdad, facultando al interesado, hacer uso del presente documento.

Loja, 15 de abril del 2019

Ing. Jhon Alexander Calderón Sanmartín
DIRECTOR DE TELECOMUNICACIONES E INFORMACIÓN UNL

072 -54 7252 Ext. 125
Ciudad Universitaria "Guillermo Falconí Espinosa",
Casilla letra "S", Sector La Argelia · Loja - Ecuador

Anexo 10. Configuración del servidor OpenLDAP

A. Instalación del servidor OpenLdap

A continuación, se realizan los siguientes pasos para la configuración del servidor OpenLdap:

PASO 1

Actualizar la lista de paquetes disponibles y sus versiones, ingresando el siguiente comando en la terminal:

sudo apt-get update

PASO 2

Instalamos el demonio slapd del servidor OpenLdap, también instalaremos el paquete que contiene las utilidades de administración de LDAP, como es ldap-utils. Ambos paquetes se encuentran en los repositorios oficiales de Ubuntu, ingresar el siguiente comando en la terminal:

sudo apt-get install slapd ldap-utils

Durante la instalación, nos aparece un mensaje que nos solicita la contraseña de administración para LDAP. Esta deberá ser una contraseña segura.

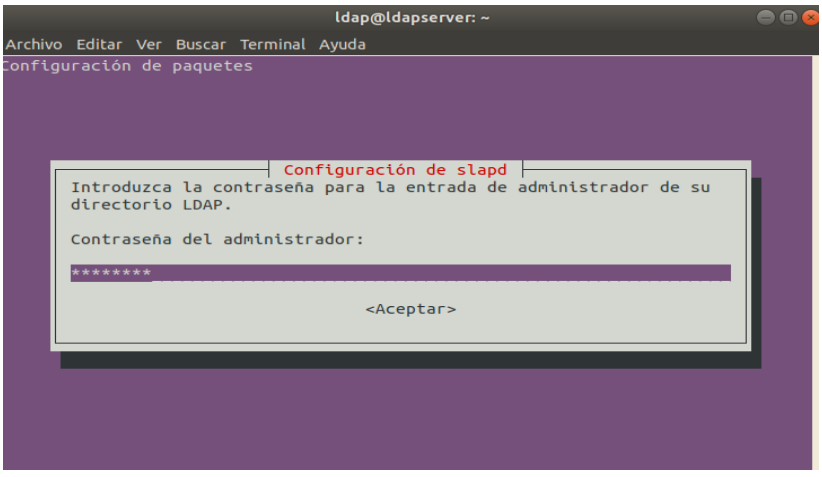


Figura A10. 1. Contraseña del administrador

Para evitar algún error tipográfico el cual luego nos impida entrar, el sistema nos solicita que volvamos a escribir la contraseña para su verificación.

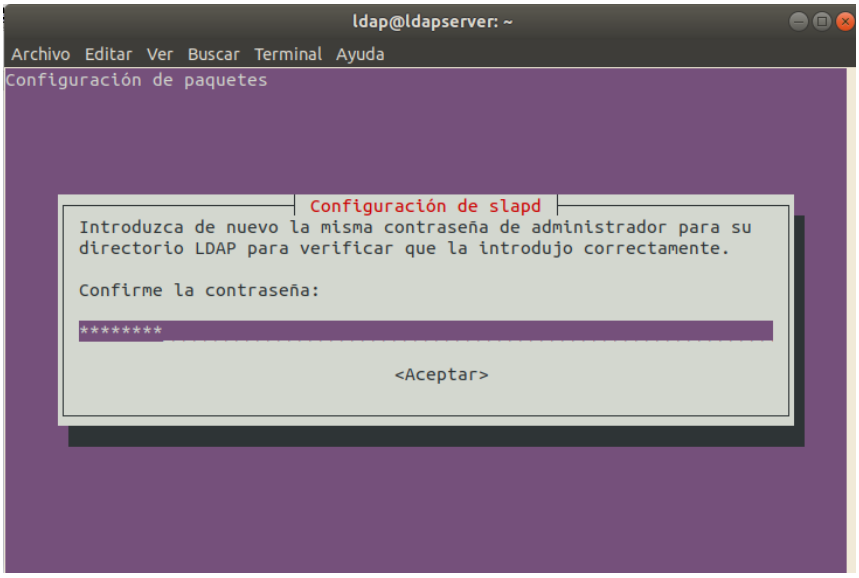


Figura A10. 2. Confirmación contraseña del administrador

Luego de confirmar la contraseña del administrador LDAP, volvemos al aspecto normal de la terminal, comprobando que la instalación sigue su curso correctamente, iniciando el servidor OpenLdap.

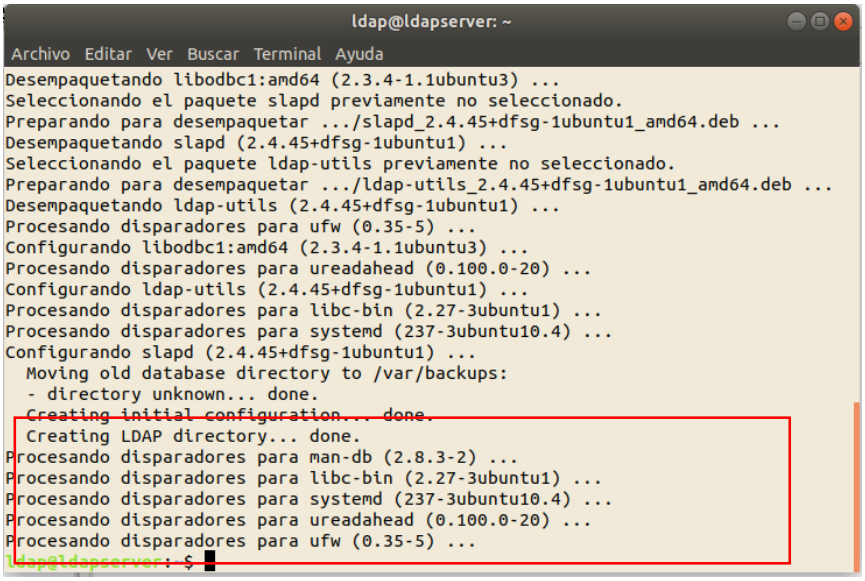


Figura A10. 3. Instalación correcta del servidor

PASO 3

Verificamos que la instalación e iniciación del servidor es correcta y el puerto del servicio se encuentra abierto, ingresando el siguiente comando en la terminal:

sudo netstat –natup | grep slapd

PASO 4

La instalación de **slapd** por defecto crea una plantilla de configuración para poder trabajar. El sufijo (o base DN) de esta plantilla se determina a partir del nombre de dominio del host local. Por lo cual procedemos a modificar nuestro dominio local, en base a nuestro servidor LDAP, con el objetivo, de cuando hagamos referencia al nombre de dominio establecido, nuestro sistema entienda que nos estamos refiriendo a nuestro servidor LDAP.

sudo nano /etc/hosts

Puede ser el comando anterior o el que está a continuación:

sudo gedit /etc/hosts

Dentro del archivo, agregamos una nueva línea la cual relacione la dirección IP estática del servidor con los nombres lógicos que tenemos previsto utilizar.

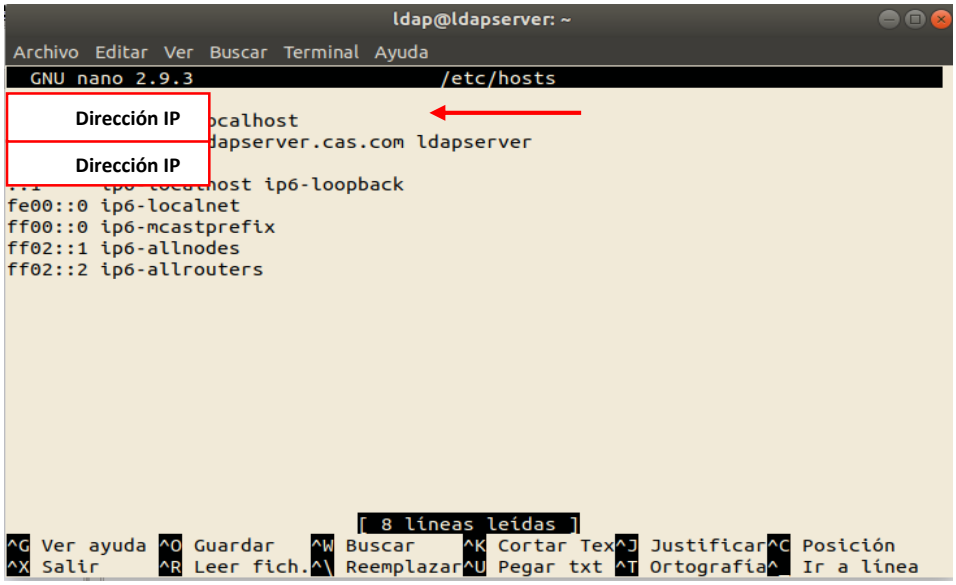


Figura A10. 4. Configuración de nuestro host o dominio

Cuando terminemos, pulsamos **Ctrl + x** para salir y nos aseguramos de guardar los cambios en el archivo.

PASO 5

Se configurar el demonio **SLAPD** (Standalone LDAP Daemon), el cual es un programa multiplataforma, que se ejecuta en segundo plano, atendiendo las solicitudes de autenticación LDAP que se reciban en el servidor. El último paso en la

configuración del servidor LDAP será establecer algunos parámetros en la configuración de este demonio. Para conseguirlo, ingresamos el siguiente comando en la terminal:

sudo dpkg-reconfigure slapd

Se nos presentará un asistente cuyo objetivo es evitar que tengamos que cambiar a mano el archivo **slapd.conf.** El primer mensaje que nos indica, actúa como medida de seguridad, para asegurarse de que no hacemos cambios por error. Hay que tener cuidado porque la pregunta se hace al revés, es decir, nos pregunta si queremos omitir la configuración del servidor. En este caso, lógicamente, deberemos elegir la opción **No.** Ya que precisamente lo que queremos es configurar el servidor LDAP.

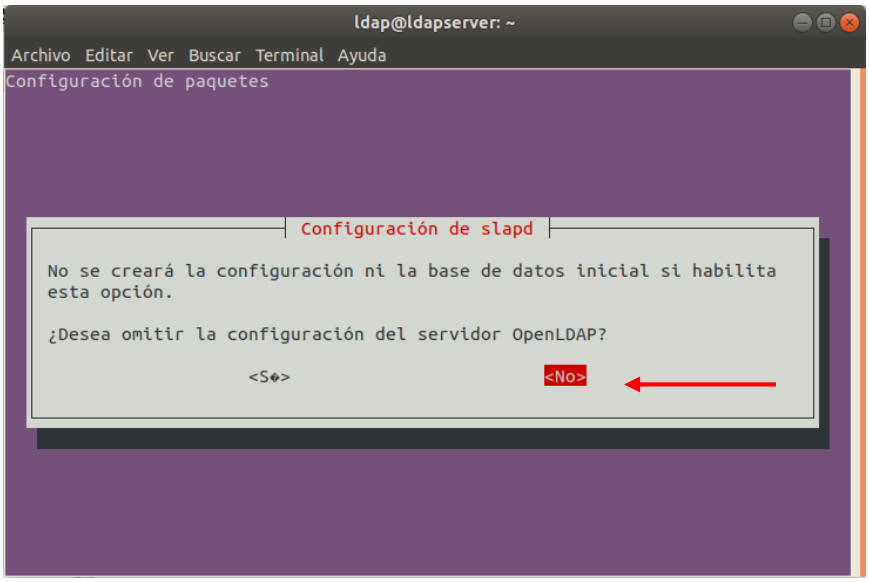


Figura A10. 5. Configuración de Slapd

A continuación, deberemos escribir el nombre DNS que utilizamos para crear el DN base (**Distinguished Name**), esta opción determinará la estructura base de la ruta del directorio LDAP (**dc=mi,dc=dominio**).

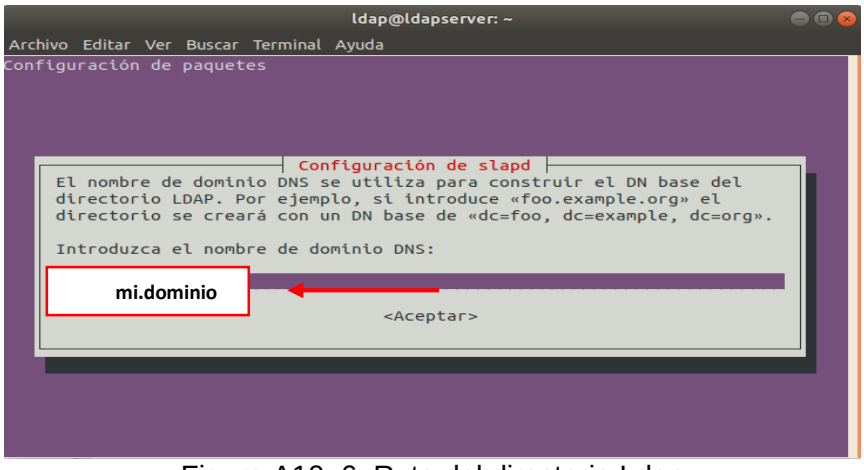


Figura A10. 6. Ruta del directorio Ldap

Posterior, escribiremos el nombre de la entidad u organización en la que estamos instalando el directorio LDAP (**ldapserver**).

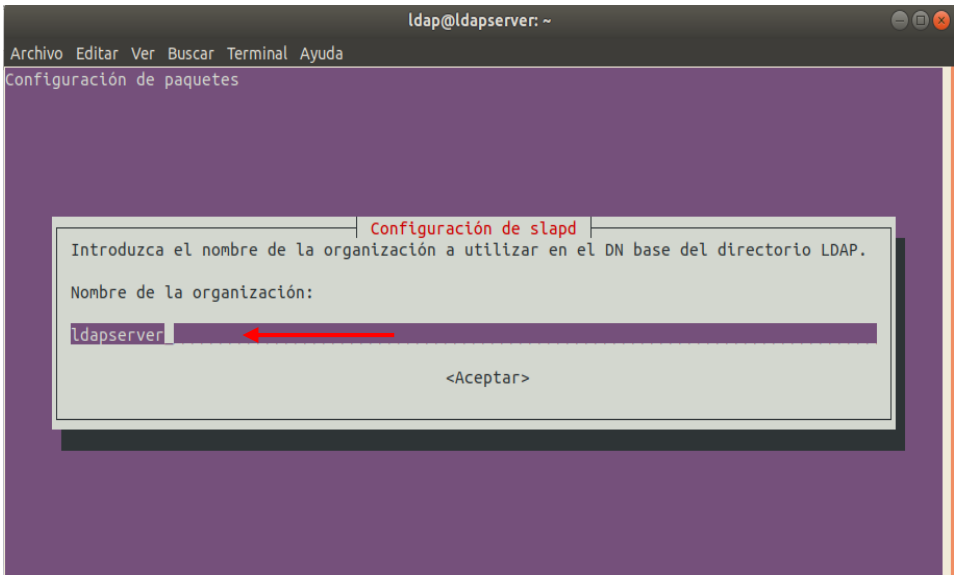


Figura A10. 7. Nombre entidad del directorio Ldap

En el siguiente paso, debemos escribir la contraseña de administración del directorio. La contraseña puede ser la misma que escribimos en el aparto de instalación del servidor OpenLdap o una contraseña nueva, la cual se sobrescribirá sobre la anterior que se utilizó.

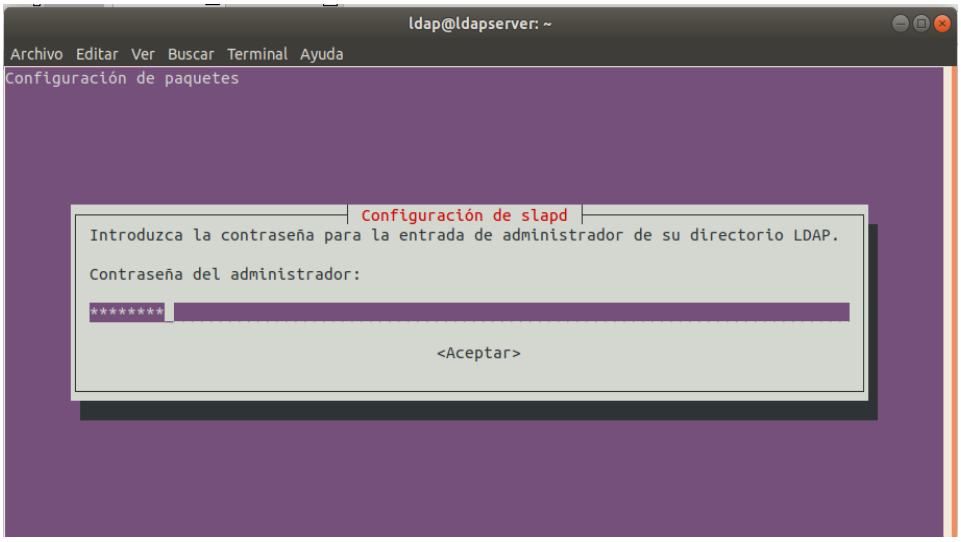


Figura A10. 8. Contraseña administrador del directorio

Como es habitual, para evitar algún error tipográfico el cual luego nos impida entrar, el sistema nos solicita que volvamos a escribir la contraseña para su verificación.

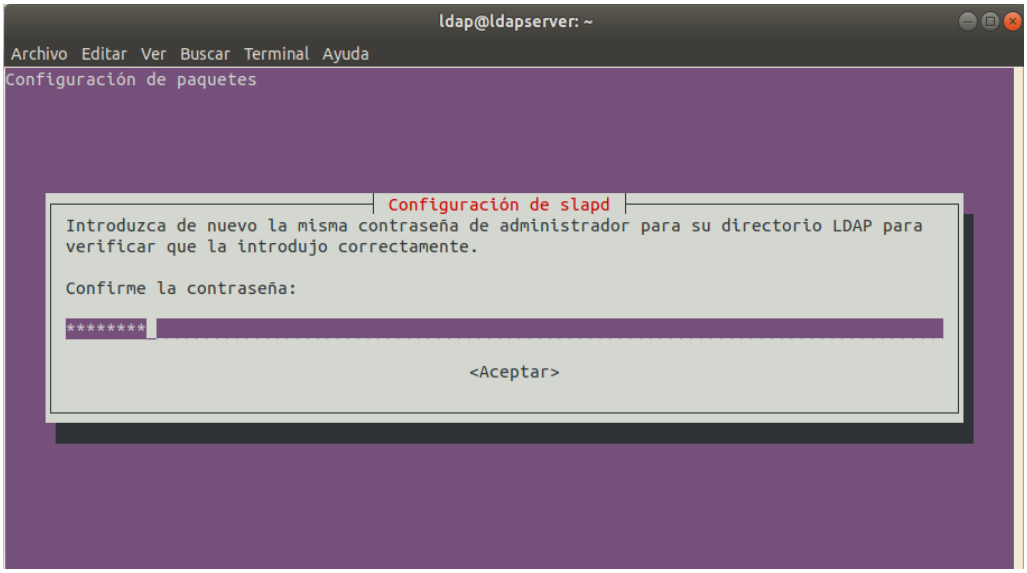


Figura A10. 9. Confirmar contraseña administrador del directorio

A continuación, nos indica un cuadro de información acerca de los motores de base de datos, y así poder seleccionar el adecuado en el siguiente paso.

En este paso, elegiremos el motor de la base de datos que usaremos para el directorio. Se recomienda HDB, porque es jerárquica y soporta el renombrado de sub-árboles si fuese necesario.

HDB es una variante del backend BDB (Berkeley Database).

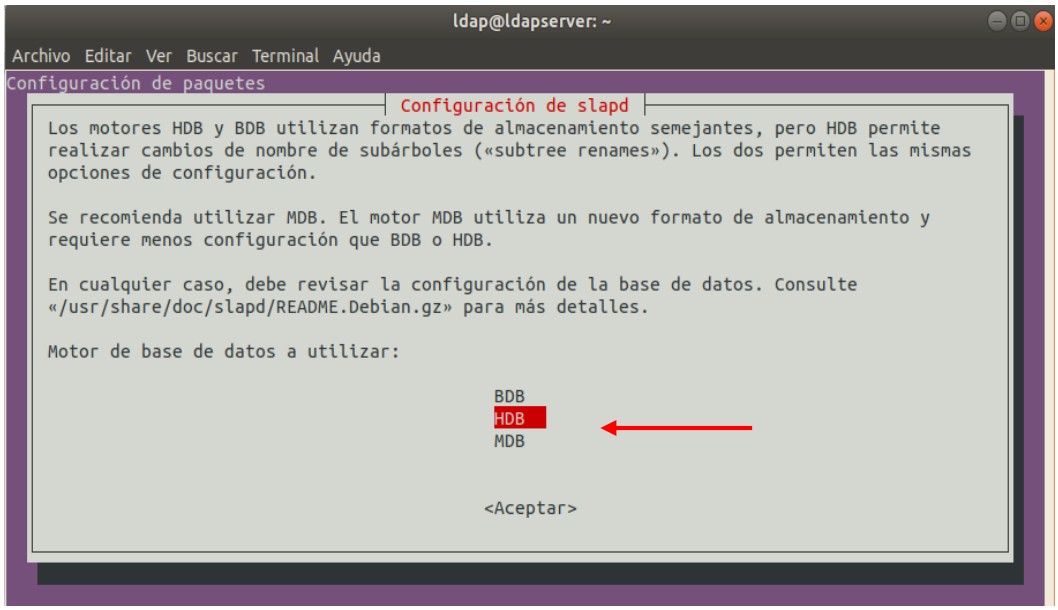


Figura A10. 10. Seleccionar motor de la base de datos

La siguiente pregunta del asistente es, si queremos que se borre la base de datos anterior del directorio cuando terminemos la configuración de slapd.

Se nos presentará una nueva ventana. En este paso, seleccionamos la opción **NO**.

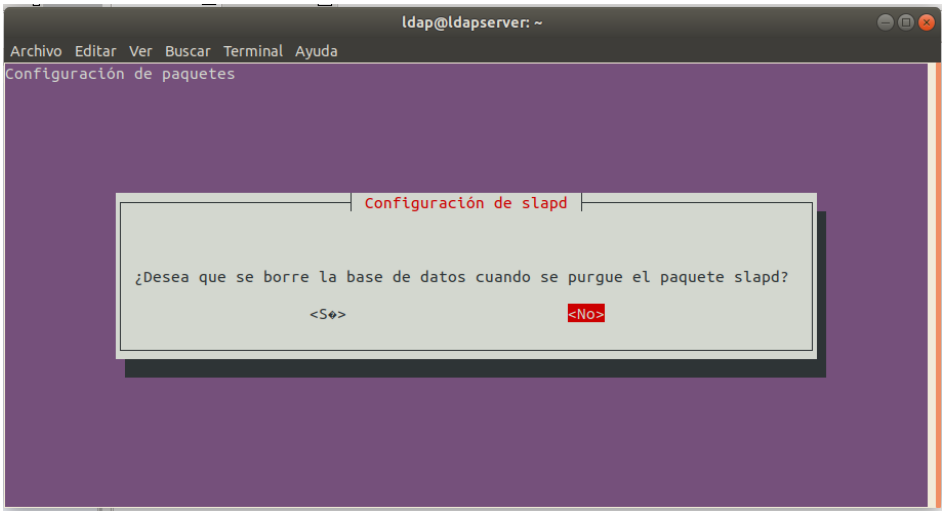


Figura A10. 11. Base de datos antigua

A continuación, como hemos decidido no borrar la base de datos antigua, el asistente nos pregunta si queremos cambiarla de sitio. Para evitar confusiones entre las dos bases de datos (nueva y antigua), elegiremos la opción **SI**.

Figura A10. 12. Mover base de datos antigua

Como paso final, el asistente suele indicar si queremos permitir el protocolo LDAPv2, esto debido a que, en algunas redes u entidades, con clientes muy antiguos, puede ser necesario mantener la versión 2 del protocolo LDAP. En la mayoría de los casos, la respuesta será **NO**.

Luego del último paso, se cierra el asistente de LDAP y volvemos a la consola. Donde podemos observar que la configuración se realizó con éxito. Ahora LDAP está listo para crear los directorios y autenticar los usuarios.

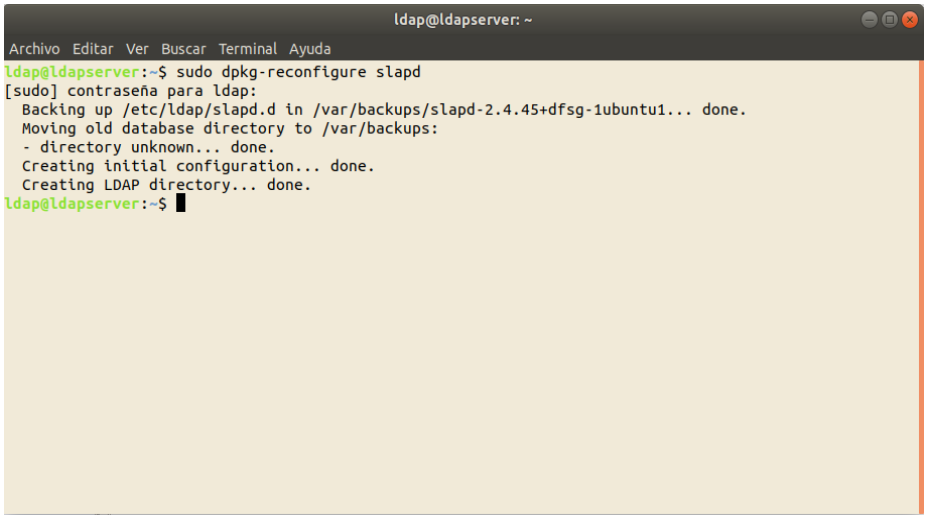


Figura A10. 13. Creación del directorio Ldap

Con los pasos que se mencionó anteriormente, se realizó con éxito la configuración del servidor OpenLdap, en el sistema operativo Ubuntu 18.04LTS.

El dominio de nuestro servidor LDAP en base a la configuración anterior será el nombre de la organización seguido de su dominio (**ldapserver.mi.dominio**) y el dominio de su administrador será en formato LDAP **(cn=admin,dc=mi,dc=dominio).**

B. Configuración del esquema eduPerson.

PASO 1

Crear y editar un archivo con el nombre **eduperson.ldif** la extensión del archivo se la denomina LDIF, debido a que son archivos estándar con formato de intercambio de datos sencillo que representan contenido de directorios LDAP.

El archivo se lo debe agregar en la ruta de esquemas para el directorio LDAP, ingresamos el siguiente comando en la terminal:

sudo nano /etc/ldap/schema/eduperson.ldif

Puede ser el comando anterior o el que está a continuación:

sudo gedit /etc/ldap/schema/eduperson.ldif

Se crea el archivo, en el cual procedemos a trabajar

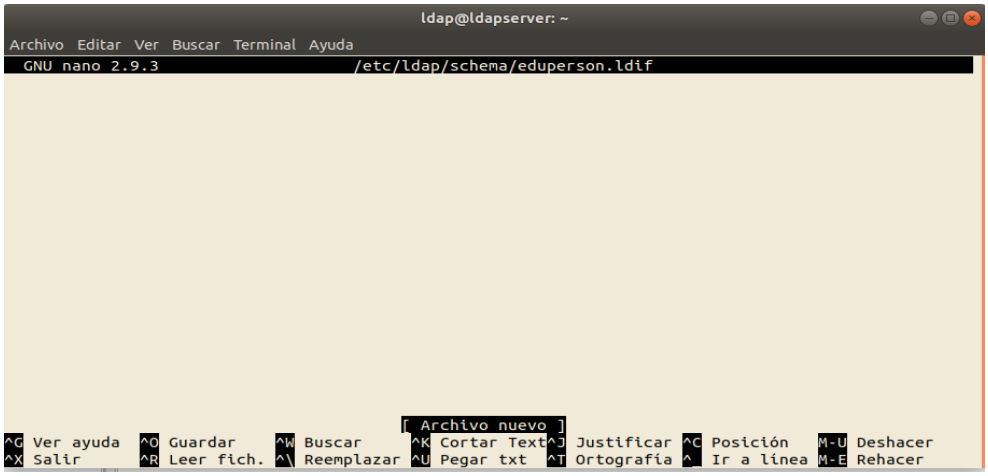


Figura A10. 14. Nuevo archivo eduPerson

PASO 2

A continuación, se dirige a la siguiente dirección https://github.com/Antonio-Cis/Sistema-de-Gestion-Unico-CAS--SiGUCAS.git donde se descarga el archivo y se copia la información del mismo, para proceder a pegarla en el archivo antes creado **eduperson.ldif**.

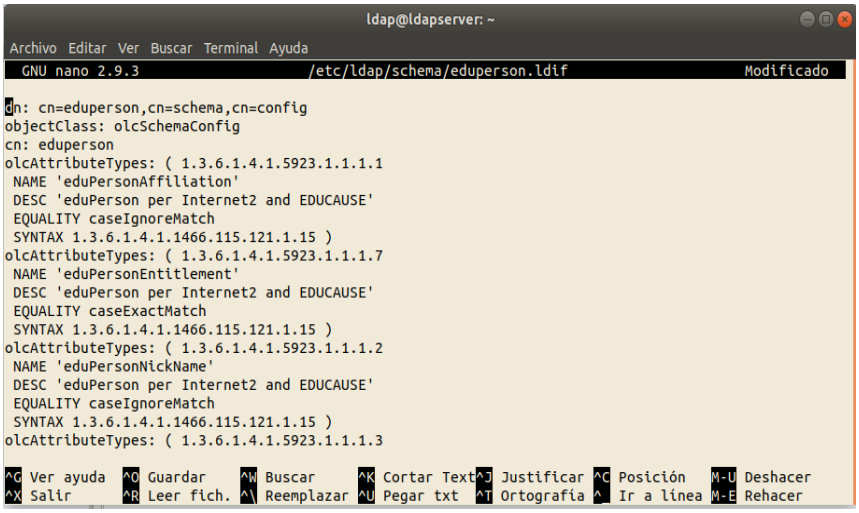


Figura A10. 15. Información del esquema eduPerson

Cuando terminemos, pulsamos **Ctrl + x** para salir y nos aseguramos de guardar los cambios en el archivo.

PASO 3

En el siguiente paso debemos agregar el esquema, que hemos creado anteriormente, haciendo uso de **ldapadd** que es una herramienta del paquete **openldap-clients** para agregar entradas en un directorio LDAP.

sudo ldapadd -Q -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/eduperson.ldif

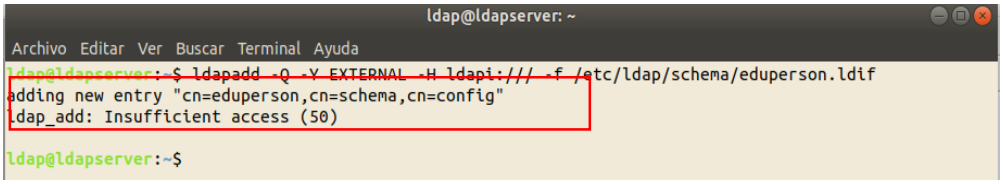


Figura A10. 16. Agregar esquema eduPerson

Luego del último paso, nos presentará un mensaje dando a conocer que el esquema se agregó correctamente. Ahora podemos hacer uso del schema eduPerson y todos sus atributos en el servidor OpenLDAP.

Anexo 11. Instalación JXplorer

Se utilizó el navegador multiplataforma JXplorer, siendo una herramienta de administración para el servidor LDAP, que brinda estándares que se puede usar para: buscar, leer y editar cualquier directorio LDAP con una excelente interfaz gráfica.

A continuación, se detallan los pasos para la correcta instalación, para lo cual recomendamos contar con Java en nuestro equipo:

a) Paso 1

Actualizar la lista de paquetes disponibles y sus versiones, ingresando el siguiente comando en la terminal:

sudo apt-get update

b) Paso 2

Procedemos a instalar JXplorer e inicializarlo con el siguiente comando:

sudo apt-get install jxplorer

Se nos presentará un mensaje donde debemos confirmar que continúe con la instalación.

Hemos terminado con la instalación de JXplorer, podemos hacer uso para administrar nuestro servidor OpenLDAP.

Figura A11. 1. Vista principal de JXplorer

Anexo 12. Desarrollo de un Web Service con métodos administrativos y de autenticación

El desarrollo del servicio Web se lo realizó utilizando SOAP que es un formato de mensaje XML utilizado en interacciones de servicios Web. Los mensajes SOAP se enviarán sobre HTTP que se describe mediante la definición WSDL, en la Figura A12. 1, se puede observar el funcionamiento general del servicio Web desarrollado para el presente TT y las principales funciones de autenticación y administración para el servidor OpenLDAP.

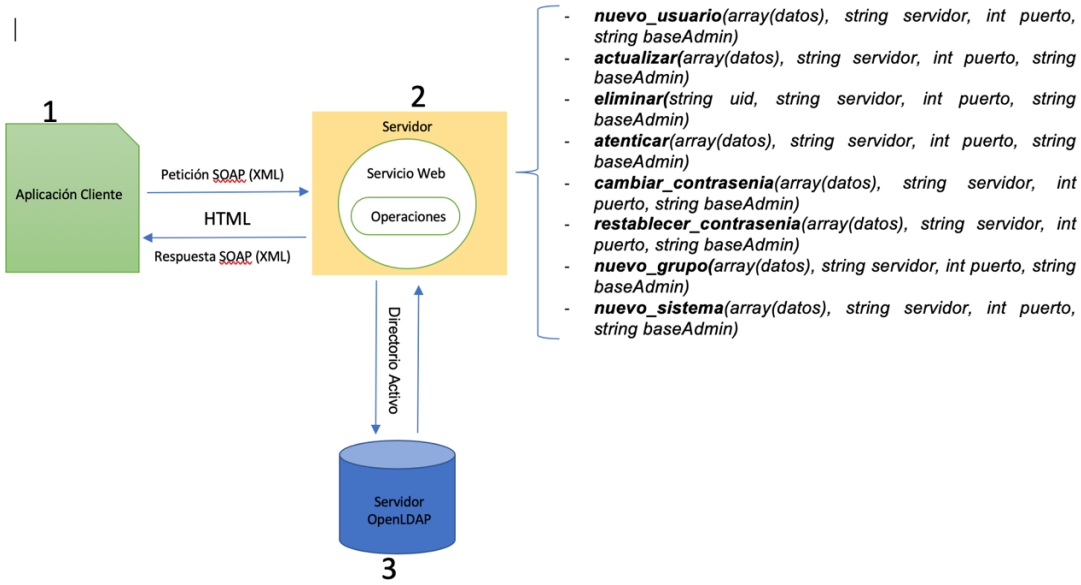


Figura A12. 1. Funcionamiento general del servicio Web propuesto para el presente TT.

1) El cliente o usuario de una aplicación Web solicita una petición utilizando el protocolo SOAP y compartiendo los datos XML sobre HTTP, como se puede observar el la Figura A12. 2. se define el cliente del servicio Web que necesita el WSDL para identificar la ruta del servicio Web, y se llama al método cuyo valor de salida se guarda en una variable para verificar el proceso que se llevará a cabo dependiendo de la respuesta obtenida.

```
$cliente = new nusoap_client("http://localhost:8888/sacfinals/servicio.php",false);
$respuesta = $cliente->call("nuevo_usuario",array('datos' => $cadena, 'servidor' =>
$host,'puerto' => $port,'baseAdmin'=>$baseAdmin));
```

Figura A12. 2. Petición de un cliente para acceder a un método de un servicio Web SOAP.

2) El servicio Web recibe la petición hecha por el cliente y busca la operación específica solicitada, una vez identificada ejecuta el método y devuelve una respuesta nuevamente al cliente mediante HTTP, como se puede observar en la Figura A12. 3. se encuentran todos los métodos que el servicio Web tendrá incorporados y a los que el cliente puede acceder.

```
require_once("lib/nusoap.php");
$servicio = new soap_server();
$ns = "urn:miserviciowsdl";
$servicio->configureWSDL("miprimerservicio",$ns);
$servicio->schemaTargetNamespace = $ns;

$servicio->register("nuevo_usuario", array('datos' => 'xsd:string','servidor'=> 'xsd:string',
'puerto' => 'xsd:int','baseAdmin'=> 'xsd:string'),
array('return' => 'xsd:string'), $ns);

$servicio->register("actualizar", array('datos' => 'xsd:string','servidor'=> 'xsd:string',
'puerto' => 'xsd:int','baseAdmin'=> 'xsd:string'),
array('return' => 'xsd:json'), $ns);

$servicio->register("eliminar", array('datos' => 'xsd:string','servidor'=> 'xsd:string',
'puerto' => 'xsd:int','baseAdmin'=> 'xsd:string'),
array('return' => 'xsd:string'), $ns);

$servicio->register("autenticar", array('datos' => 'xsd:string','servidor'=> 'xsd:string',
'puerto' => 'xsd:int','baseAdmin'=> 'xsd:string'),
array('return' => 'xsd:string'), $ns);

$servicio->register("actualizar_contrasenia", array('datos' => 'xsd:string','servidor'=> 'xsd:string',
'puerto' => 'xsd:int','baseAdmin'=> 'xsd:string'),
array('return' => 'xsd:string'), $ns);
```

Figura A12. 3. Servicio Web con los métodos establecidos para ser accedidos por los clientes.

3) El servidor OpenLDAP contendrá toda la información de los usuarios que pertenecen a una entidad u organización, el servicio Web está conectado directamente al directorio el mismo que recibirá una petición por parte del servicio Web y devolverá una respuesta.

1. Descripción del método nuevo_usuario

El método nuevo_usuario será utilizado para ingresar un nuevo registro en el servidor OpenLDAP, cuya estructura se puede observar en la Figura A12. 4. y se define en la TABLA A12. I.

```
$servicio->register("nuevo_usuario", array('datos' => 'xsd:string','servidor'=> 'xsd:string',
'puerto' => 'xsd:int','baseAdmin'=> 'xsd:string'),
array('return' => 'xsd:string'), $ns);
```

Figura A12. 4. Método nuevo_usuario para el ingreso de un nuevo registro en el servidor OpenLDAP.

TABLA A12. I.
DESCRIPCIÓN DEL MÉTODO NUEVO_USUARIO

Nombre:	**nuevo_usuario**	
Entrada:	array	string usuario_sesion, string password_sesion, string nombre1, string nombre2, string apellido1, string apellido2, string cédula, string correo, string teléfono, string tipoU.
	string	servidor (ruta del servidor OpenLDAP).
	int	puerto (puerto del servidor OpenLDAP).
	string	baseAdmin (base del administrador del servidor OpenLDAP).
Salida:	JSON	int token (Bandera de éxito o error), string status (nombre del usuario creado).

2. Descripción del método actualizar

El método actualizar será utilizado para modificar la información de un registro en el servidor OpenLDAP, cuya estructura se puede observar en la Figura A12. 5. y se define en la TABLA A12. II.

```
$servicio->register("actualizar", array('datos' => 'xsd:string','servidor'=> 'xsd:string',
'puerto' => 'xsd:int','baseAdmin'=> 'xsd:string'),
array('return' => 'xsd:string'), $ns);
```

Figura A12. 5. Método actualizar para modificar la información de un registro en el servidor OpenLDAP.

TABLA A12. II.
DESCRIPCIÓN DEL MÉTODO ACTUALIZAR

Nombre:	**actualizar**	
Entrada:	array	string usuario_sesion, string password_sesion, string nombre1, string nombre2, string apellido1, string apellido2, string cédula, string correo, string teléfono, string tipoU, string contrasenia.
	string	servidor (ruta del servidor OpenLDAP).
	int	puerto (puerto del servidor OpenLDAP).
	string	baseAdmin (base del administrador del servidor OpenLDAP).
Salida:	JSON	int token (Bandera de éxito o error).

3. Descripción del método eliminar

El método eliminar será utilizado para eliminar la información de un registro en el servidor OpenLDAP, cuya estructura se puede observar en la Figura A12. 6. y se define en la TABLA A12. III.

```
$servicio->register("eliminar", array('datos' => 'xsd:string','servidor'=> 'xsd:string',
'puerto' => 'xsd:int','baseAdmin'=> 'xsd:string'),
array('return' => 'xsd:string'), $ns);
```

Figura A12. 6. Método eliminar para borrar la información de un registro en el servidor OpenLDAP.

TABLA A12. III.
DESCRIPCIÓN DEL MÉTODO ELIMINAR

Nombre:	**actualizar**	
Entrada:	string	string uid_usuario.
	string	servidor (ruta del servidor OpenLDAP).
	int	puerto (puerto del servidor OpenLDAP).
	string	baseAdmin (base del administrador del servidor OpenLDAP).
Salida:	JSON	int token (Bandera de éxito o error).

4. Descripción del método autenticar

El método autenticar será utilizado para verificar las credenciales de un usuario en el servidor OpenLDAP, cuya estructura se puede observar en la Figura A12. 7. y se define en la TABLA A12. IV.

```
$servicio->register("autenticar", array('datos' => 'xsd:string','servidor'=> 'xsd:string',
'puerto' => 'xsd:int','baseAdmin'=> 'xsd:string'),
array('return' => 'xsd:string'), $ns);
```

Figura A12. 7. Método de autenticar para verificar las credenciales de un usuario en el servidor OpenLDAP.

TABLA A12. IV.
DESCRIPCIÓN DEL MÉTODO AUTENTICAR

Nombre:	**actualizar**	
Entrada:	array	string usuario, string password.
	string	servidor (ruta del servidor OpenLDAP).

	int	puerto (puerto del servidor OpenLDAP).
	string	baseAdmin (base del administrador del servidor OpenLDAP).
Salida:	JSON	int token (Bandera de éxito o error), string usuario, string password, string nombres_usuario, string apellidos_usuario, string correo_usuario.

5. Descripción del método actualizar_contrasenia

El método actualizar_contrasenia será utilizado para cambiar la contraseña de un usuario en el servidor OpenLDAP, cuya estructura se puede observar en la Figura A12. 8. y se define en la TABLA A12. V.

```
$servicio->register("actualizar_contrasenia", array('datos' => 'xsd:string','servidor'=> 'xsd:string',
'puerto' => 'xsd:int','baseAdmin'=> 'xsd:string'),
array('return' => 'xsd:string'), $ns);
```

Figura A12. 8. Método actualizar_contrasenia para cambiar la contraseña de un usuario en el servidor OpenLDAP.

TABLA A12. V.
DESCRIPCIÓN DEL MÉTODO ACTUALIZAR_CONTRASENIA

Nombre:	**actualizar**	
Entrada:	array	string usuario_sesion, string password_sesion, string password_actual, string password_nueva, string password_nueva_confirmar.
	string	servidor (ruta del servidor OpenLDAP).
	int	puerto (puerto del servidor OpenLDAP).
	string	baseAdmin (base del administrador del servidor OpenLDAP).
Salida:	JSON	int token (Bandera de éxito o error).

Anexo 13. Metodología XP para el desarrollo del sistema de administración SAC.

1. FASE 1: Planeación

Es la Fase inicial de la metodología XP, se establece una comunicación continua entre el equipo de desarrollo y el cliente, para obtener principalmente los requisitos del sistema. Además, nos permitió establecer el alcance del proyecto y fechas de entrega del sistema, tomando en cuenta la prioridad y tiempo estimado para el desarrollo de cada historia de usuario.

1.1. Historias de usuarios.

Las Historias de Usuarios del Sistema SAC son las siguientes:

- Administrador, Usuario: Selección del sistema SAC.
- Administrador: Acceso al módulo de administración.
- Usuario: Acceso al módulo de usuario.
- Administrador: Visualizar información general de cada usuario.
- Usuario: Cambio de contraseña.
- Administrador, Usuario: Cerrar sesión.
- Administrador: Añadir nuevo usuario.
- Administrador: Actualizar información de usuarios.
- Administrador: Eliminar usuario.
- Administrador: Buscar usuario.
- Administrador: Cargar usuarios (Archivo csv).
- Administrador: Vincular usuario.
- Administrador: Vincular varios usuarios.
- Administrador: Visualizar información de grupos y sistemas.
- Administrador: Crear nuevos grupos.
- Administrador: Eliminar grupos.
- Usuario: Restablecer contraseña.
- Administrador: Crear nuevos sistemas.
- Administrador: Eliminar sistemas.
- Administrador: Generar reportes.
- Administrador: Eliminar todos los registros.
- Administrador: Crear nuevos administradores de lectura.

- Administrador: Visualizar usuarios vinculados.

La Plantilla a utilizarse para la elaboración de las historias de usuario se muestra en la TABLA A13. I. y cada uno de sus componentes se explica a continuación:

TABLA A13. I.
PLANTILLA PARA LA ELABORACIÓN DE LAS HISTORIAS DE USUARIO.

Historia de Usuario	
Número: Permite identificar a una historia de usuario.	**Usuario:** Persona que utilizará la funcionalidad del sistema descrita en la historia de usuario.
Nombre de Historia: Describe de manera general a una historia de Usuario.	
Prioridad en negocio: Grado de importancia que el cliente asigna a una historia de usuario.	**Riesgo en desarrollo**: Valor de complejidad que una historia de usuario representa al equipo de desarrollo.
Puntos estimados: Número de días o semanas que se necesitara para el desarrollo de una historia de usuario.	**Iteración asignada**: Número de iteración, en que el cliente desea que se implemente una historia de usuario.
Programador responsable: Persona encargada de programar cada historia.	
Descripción: Información detallada de una historia de Usuario.	
Observaciones: Campo opcional utilizado para aclarar, si es necesario, el requerimiento descrito de una historia de usuario.	

A continuación, en las (TABLAS A13. II – A13. XXIV) se muestran las historias de usuario, las cuales fueron utilizadas para llevar a cabo el desarrollo del sistema.

TABLA A13. II.
HISTORIA DE USUARIO PARA LA SELECCIÓN DEL SISTEMA SAC.

Historia de Usuario	
Número: 1	**Usuario:** Administrador, Usuario
Nombre de Historia: Selección del Sistema SAC	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados:2 días	**Iteración asignada:** 1
Programador responsable: Antonio Aguilar – Manuel Armijos	
Descripción: El Administrador, Usuario seleccionará el sistema, haciendo clic en el enlace SAC, que redirecciona a la página de acceso (login) del sistema antes mencionado.	

Observaciones: El Administrador/Usuario debe seleccionar el logotipo del sistema para poder redireccionar correctamente.

TABLA A13. III.
HISTORIA DE USUARIO PARA EL ACCESO AL MÓDULO DE ADMINISTRACIÓN.

Historia de Usuario	
Número: 2	**Usuario:** Administrador
Nombre de Historia: Acceso al módulo de Administración	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados: 2 días	**Iteración asignada:** 1
Programador responsable: Antonio Aguilar - Manuel Armijos	
Descripción: El administrador deberá iniciar sesión ingresando sus credenciales que son: usuario y contraseña. Luego presionará aceptar, para que el sistema verifique si las credenciales son correctas y procedes a dar acceso a la interfaz de administración del servidor.	
Observaciones: El administrador debe ingresar correctamente sus credenciales, caso contrario se le mostrará un mensaje de error, indicando que los datos son incorrectos. La interfaz de acceso correcto presenta una vista general de los tipos de usuarios que existen en el servidor, separados en sus respectivos grupos.	

TABLA A13. IV.
HISTORIA DE USUARIO PARA EL ACCESO AL MÓDULO DEL USUARIO.

Historia de Usuario	
Número: 3	**Usuario:** Usuario
Nombre de Historia: Acceso al módulo del Usuario	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados: 2 días	**Iteración asignada: 1**
Programador responsable: Antonio Aguilar - Manuel Armijos	
Descripción: El Usuario deberá ingresar sus credenciales que son: usuario y contraseña. Luego presionará aceptar, para que el sistema verifique si las credenciales son correctas y procedes a dar acceso al módulo de información del usuario identificado.	
Observaciones: El usuario debe ingresar correctamente sus credenciales, caso contrario se le mostrará un mensaje de error, indicando que los datos son incorrectos. La interfaz de acceso correcto presenta la información relevante del usuario identificado y un campo para la opción de cambio de contraseña.	

TABLA A13. V.
HISTORIA DE USUARIO PARA VISUALIZAR LA INFORMACIÓN DEL USUARIO.

Historia de Usuario	
Número: 4	**Usuario:** Administrador
Nombre de Historia: Visualizar la información del usuario	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados: 2 días	**Iteración asignada:** 2
Programador responsable: Antonio Aguilar – Manuel Armijos	
Descripción: El administrador podrá visualizar la información de todos los usuarios registrados en el servidor OpenLDAP. El sistema le mostrará la siguiente información: (Número de cédula o DNI, Nombres, Apellidos, Correo electrónico, Número de teléfono, Nombre de usuario y Grupo de Autenticación), en la que se encuentra almacena en el servidor y es correspondiente al usuario antes seleccionado.	
Observaciones: El administrador no podrá visualizar las contraseñas de los usuarios por motivos de seguridad.	

TABLA A13. VI.
HISTORIA DE USUARIO PARA EL CAMBIO DE CONTRASEÑA.

Historia de Usuario	
Número: 5	**Usuario:** Usuario
Nombre de Historia: Cambio de contraseña	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados:3 día	**Iteración asignada:** 2
Programador responsable: Antonio Aguilar - Manuel Armijos	
Descripción: El usuario deberá ingresar en el primer campo su contraseña actual, luego ingresará la nueva contraseña que desea asignar y por último volver a ingresar su nueva contraseña, posterior deberá dar clic en cambiar contraseña. El sistema le indicará si el proceso se realizó con éxito o si se encontró un problema. Si el cambio se realizó con éxito el sistema lo redirige al módulo de acceso al sistema para que ingrese sus nuevas credenciales.	
Observaciones: El usuario debe ingresar correctamente su contraseña actual o el método de cambio de contraseña le indicará un mensaje de error, posterior la contraseña actual que desea ingresar debe coincidir las dos veces que se la ingresa o de la misma manera el método le indicará un mensaje de error en el proceso que desea cumplir.	

TABLA A13. VII.
HISTORIA DE USUARIO PARA EL CIERRE DE SESIÓN DEL SISTEMA.

Historia de Usuario	
Número: 6	**Usuario**: Administrador, Usuario
Nombre de Historia: Cierre de sesión del sistema	
Prioridad en negocio: ALTA	**Riesgo en desarrollo:** Media
Puntos estimados: 1 día	**Iteración asignada:** 2
Programador responsable: Antonio Aguilar - Manuel Armijos	
Descripción: El Administrador, Usuario deberá dar clic en cerrar sesión, para que el sistema destruya la sesión creada por el usuario logueado y lo redireccione a la página de acceso al sistema nuevamente.	
Observaciones: Si el usuario no destruye cesión corre con el riesgo de que terceras personas puedan acceder al sistema y robar información.	

TABLA A13. VIII.
HISTORIA DE USUARIO PARA AÑADIR UN NUEVO USUARIO.

Historia de Usuario	
Número: 7	**Usuario:** Administrador
Nombre de Historia: Añadir nuevo Usuario	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados: 3 días	**Iteración asignada:** 3
Programador responsable: Antonio Aguilar – Manuel Armijos	
Descripción: El administrador presionará el botón añadir nuevo e ingresa los datos personales (Número de Cédula, Nombres, Apellidos, Correo Electrónico, Número de Teléfono y Grupo), y presionará guardar. El sistema confirmará el registro del nuevo usuario y le envía un correo electrónico indicándole que su cuenta se ha creado con éxito e proporcionando sus credenciales de acceso (Usuario, Contraseña).	
Observaciones: El Administrador debe proporcionar una dirección de correo electrónico válida debido a que el sistema la usará para notificarle al usuario que a sido registrado en el directorio centralizado.	

TABLA A13. IX.
HISTORIA DE USUARIO PARA ACTUALIZAR LA INFORMACIÓN DE UN USUARIO.

Historia de Usuario	
Número: 8	**Usuario:** Administrador
Nombre de Historia: Actualizar información de un Usuario	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados: 2 días	**Iteración asignada:** 3
Programador responsable: Antonio Aguilar – Manuel Armijos	
Descripción: El administrador presionará el botón actualizar referente a un usuario seleccionado, el sistema le presentará los datos personales (Número de Cédula, Nombres, Apellidos, Correo Electrónico, Número de Teléfono, Contraseña y Grupos), para que modifique los campos que crea conveniente. El sistema confirmará la actualización de la información.	
Observaciones: El sistema presentará el número de cédula como dato bloqueado, únicamente como referencia al usuario seleccionado, debido a que este campo es un identificador único y no se puede modificar.	

TABLA A13. X.
HISTORIA DE USUARIO PARA ELIMINAR UN USUARIO.

Historia de Usuario	
Número: 9	**Usuario:** Administrador
Nombre de Historia: Eliminar Usuario	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados: 1 días	**Iteración asignada:** 3
Programador responsable: Antonio Aguilar – Manuel Armijos	
Descripción: El administrador presionará el botón eliminar referente a un usuario, el sistema le presentará una notificación para ratificar la eliminación del usuario.	
Observaciones: La eliminación de un usuario también lo eliminará de los grupos a los que se encuentre vinculado.	

TABLA A13. XI. 7
HISTORIA DE USUARIO PARA BUSCAR USUARIOS.

Historia de Usuario	
Número: 10	**Usuario:** Administrador

Nombre de Historia: Buscar Usuario	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados: 1 día	**Iteración asignada:** 3
Programador responsable: Antonio Aguilar – Manuel Armijos	
Descripción: El administrador ingresará un criterio de búsqueda (Número de Cédula, Nombres o Usuario), para filtrar el resultado de los Usuarios registrados. El sistema recuperará todas las coincidencias referentes al criterio de búsqueda del administrador y presenta toda la información detallada en una tabla.	
Observaciones: Si la información ingresada no coincide con ningún registro en el directorio centralizado el sistema mostrará una tabla vacía.	

TABLA A13. XII.
HISTORIA DE USUARIO PARA CARGAR VARIOS USUARIOS.

Historia de Usuario	
Número: 11	**Usuario:** Administrador
Nombre de Historia: Cargar Usuarios	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados: 2 días	**Iteración asignada:** 3
Programador responsable: Antonio Aguilar – Manuel Armijos	
Descripción: El administrador presionará el botón cargar usuarios, el sistema le presentará una ventana donde debe seleccionar el archivo con la extensión .csv, y presionará cargar. El sistema verificará si el archivo es correcto y confirmará la carga de los usuarios.	
Observaciones: Si existen errores en la estructura del archivo seleccionado no se cargará ningún usuario hasta que se corrija el error. Si existen usuarios que ya están ingresados se los descarta y únicamente se cargará a los que no se en encuentren en el directorio centralizado.	

TABLA A13. XIII.
HISTORIA DE USUARIO PARA VINCULAR USUARIOS.

Historia de Usuario	
Número: 12	**Usuario:** Administrador
Nombre de Historia: Vincular Usuario	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados: 2 días	**Iteración asignada:** 3

Programador responsable: Antonio Aguilar – Manuel Armijos
Descripción: El administrador presionará el botón vincular referente a un usuario seleccionado, el sistema le presentará una ventana con los grupos que se encuentren registrados en el directorio centralizado a las que el usuario aún no ha sido vinculado, seleccionará uno o más grupos y presiona vincular. El sistema confirma la vinculación del Usuario.
Observaciones: El sistema mostrará únicamente los grupos a los que el usuario aún no pertenece, por lo que no sería un error no poder visualizar todos los grupos registrados en el directorio centralizado.

TABLA A13. XIV.
HISTORIA DE USUARIO PARA VINCULAR VARIOS USUARIOS.

Historia de Usuario	
Número: 13	**Usuario:** Administrador
Nombre de Historia: Vincular varios Usuarios	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados: 1 días	**Iteración asignada:** 3
Programador responsable: Antonio Aguilar – Manuel Armijos	
Descripción: El administrador presionará el botón vincular usuarios, el sistema le presentará una tabla con todos los usuarios registrados por cada grupo asignado referente a la LOES, el administrador seleccionará varios usuarios y presiona vincular, el sistema le presentará una ventana con los grupos que se encuentren registrados en el directorio centralizado, selecciona uno o más grupos y presionará nuevamente vincular. El sistema confirmará la vinculación de los Usuarios.	
Observaciones: El administrador podrá seleccionar uno o varios grupos para vincular a los usuarios.	

TABLA A13. XV.
HISTORIA DE USUARIO PARA VISUALIZAR INFORMACIÓN DE GRUPOS Y SISTEMAS.

Historia de Usuario	
Número: 14	**Usuario:** Administrador
Nombre de Historia: Visualizar información de grupos y sistemas	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados: 1 día	**Iteración asignada:** 4
Programador responsable: Antonio Aguilar – Manuel Armijos	

Descripción: El administrador seleccionará el ítem grupos, el sistema recuperará la información de todos los grupos y sistemas registrados en el directorio centralizado, el administrador podrá visualizar la información de cada grupo y cada sistema.
Observaciones: El administrador podrá navegar dentro de cada grupo para visualizar sus subgrupos o sus subsistemas, dependiendo de la estructura definida de cada uno.

TABLA A13. XVI.
HISTORIA DE USUARIO PARA CREACIÓN NUEVOS GRUPOS.

Historia de Usuario	
Número: 15	**Usuario:** Administrador
Nombre de Historia: Crear nuevos grupos	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados: 3 días	**Iteración asignada:** 4
Programador responsable: Antonio Aguilar – Manuel Armijos	
Descripción: El administrador presionará el botón crear nuevo grupo, el sistema le presentará una ventana para que ingrese la información del mismo (Nombre, Subgrupo(s), sistema(s), descripción), y presionará guardar. El sistema confirmará la creación de un nuevo grupo.	
Observaciones: Cuando se crea un nuevo grupo será necesario la creación de un sistema y opcionalmente la creación de uno o varios subgrupos. La creación del sistema se lo hará en el grupo principal o en el último subgrupo que se crea.	

TABLA A13. XVII.
HISTORIA DE USUARIO PARA ELIMINAR GRUPOS.

Historia de Usuario	
Número: 16	**Usuario:** Administrador
Nombre de Historia: Eliminar grupo	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Baja
Puntos estimados: 2 días	**Iteración asignada:** 4
Programador responsable: Antonio Aguilar - Manuel Armijos	

Descripción: El administrador deberá dar clic en el ítem grupos y dará clic en el botón eliminar grupo y se le presentará una modal con todos los grupos registrados en el directorio centralizado, el administrador seleccionará el grupo y presiona nuevamente el botón eliminar. El sistema confirmará la eliminación del grupo.
Observaciones: El administrador podrá eliminar uno o varios grupos a la vez.

TABLA A13. XVIII.
HISTORIA DE USUARIO PARA RESTABLECER LA CONTRASEÑA.

Historia de Usuario	
Número: 17	**Usuario:** Administrador, Usuario
Nombre de Historia: Restablecer Contraseña	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados: 3 días	**Iteración asignada:** 4
Programador responsable: Antonio Aguilar - Manuel Armijos	
Descripción: El administrador, usuario deberá dar clic en la opción de restablecer contraseña en el login, el sistema le presentará una ventana para la verificación de credenciales (Cédula, usuario), si el sistema encuentra un registro que coincida con las credenciales ingresadas le mostrará al usuario un mensaje indicándole que se le a enviado a su correo un link para que pueda ingresar una nueva contraseña de acceso.	
Observaciones: El administrador, usuario podrán tener acceso al link enviado al correo únicamente por un corto periodo de tiempo por seguridad.	

TABLA A13. XIX.
HISTORIA DE USUARIO PARA CREAR NUEVOS SISTEMAS.

Historia de Usuario	
Número: 18	**Usuario:** Administrador
Nombre de Historia: Crear nuevos sistemas	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados: 1 día	**Iteración asignada:** 4
Programador responsable: Antonio Aguilar - Manuel Armijos	
Descripción: El administrador presionará el botón crear nuevo sistema, el sistema le presentará una ventana para que ingrese la información del mismo (Nombre, subsistemas(s), descripción), y presionará guardar. El sistema confirmará la creación de un nuevo sistema.	

Observaciones: Cuando se crea un nuevo subsistema se lo hará en el grupo principal en el que se encuentre actualmente.

TABLA A13. XX.
HISTORIA DE USUARIO PARA ELIMINAR SISTEMAS.

Historia de Usuario	
Número: 19	**Usuario:** Administrador
Nombre de Historia: Eliminar sistema	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados: 1 día	**Iteración asignada:** 4
Programador responsable: Antonio Aguilar - Manuel Armijos	
Descripción: El administrador deberá dar clic en el ítem grupos y dará clic en el botón eliminar sistema y se le presentará una modal con todos los sistemas registrados en el directorio centralizado, el administrador seleccionará el sistema y presiona nuevamente el botón eliminar. El sistema confirmará la eliminación del sistema.	
Observaciones: El administrador podrá eliminar uno o varios sistemas a la vez.	

TABLA A13. XXI.
HISTORIA DE USUARIO PARA GENERAR REPORTES.

Historia de Usuario	
Número: 20	**Usuario:** Administrador
Nombre de Historia: Generar reportes	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados: 1 día	**Iteración asignada:** 4
Programador responsable: Antonio Aguilar - Manuel Armijos	
Descripción: El administrador deberá dar clic en el botón generar reporte, el sistema le presentará una ventana con los grupos de usuarios registrados en el directorio centralizado, el administrador deberá seleccionar un grupo y automáticamente el sistema le descarga un archivo con la extensión .csv con los usuarios del grupo seleccionado.	
Observaciones: Una vez descargado el archivo .csv el sistema le mostrará una modal indicándole que su reporte está listo.	

TABLA A13. XXII.
HISTORIA DE USUARIO PARA ELIMINAR TODOS LOS REGISTROS.

Historia de Usuario	
Número: 21	**Usuario:** Administrador
Nombre de Historia: Eliminar todos los registros	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Baja
Puntos estimados: 1 día	**Iteración asignada:** 4
Programador responsable: Antonio Aguilar - Manuel Armijos	
Descripción: El Administrador deberá dar clic en el botón eliminar todo, el sistema le mostrará una ventana con los grupos de usuarios registrados en el directorio centralizado, el administrador deberá elegir un grupo y el sistema le mostrará una ventana de confirmación con las opciones de aceptar o cancelar, una vez finalizado el proceso el sistema la notificará de la eliminación de todos los registros.	
Observaciones: Dependiendo del número de usuarios registrados dependerá el tiempo de eliminación.	

TABLA A13. XXIII.
HISTORIA DE USUARIO PARA CREAR NUEVOS ADMINISTRADORES DE LECTURA.

Historia de Usuario	
Número: 22	**Usuario:** Administrador
Nombre de Historia: Crear nuevos administradores de lectura.	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Alta
Puntos estimados: 2 días	**Iteración asignada:** 4
Programador responsable: Antonio Aguilar - Manuel Armijos	
Descripción: El Administrador deberá dar clic en el ítem Administrador, el sistema le mostrará una ventana para que ingrese la información requerida (Usuario, contraseña y descripción), el sistema le mostrará una notificación indicando la creación de un nuevo administrador solo de lectura.	
Observaciones: Para la consulta de información sobre los administrador de lectura así como su eliminación solo se lo podrá hacer con otras herramientas por cuestiones de seguridad.	

TABLA A13. XXIV.
HISTORIA DE USUARIO PARA VISUALIZAR USUARIOS VINCULADOS.

Historia de Usuario	
Número: 23	**Usuario:** Administrador
Nombre de Historia: Visualizar usuarios vinculados	
Prioridad en negocio: Alta	**Riesgo en desarrollo:** Media
Puntos estimados: 2 días	**Iteración asignada:** 4
Programador responsable: Antonio Aguilar - Manuel Armijos	
Descripción: El Administrador deberá dar clic en el ítem grupos, y dar clic en el nombre de cualquier sistema para que pueda visualizar a los usuarios vinculados en el mismo, el sistema le presentará una nueva página con la descripción del sistema y una tabla con todos los usuarios.	
Observaciones: El sistema le va a permitir imprimir, generar un reporte o cancelar la visualización de los usuarios.	

1.2. ROLES

En la TABLA A13. XXV, se muestra la asignación de los roles para el presente proyecto.

TABLA A13. XXV.
ROLES PARA EL DESARROLLO DEL SISTEMA SAC.

Roles:	Asignado A:
Programadores:	- Wilmer Antonio Aguilar Soto - Manuel Stalin Armijos Ordóñez
Cliente:	- Director del proyecto de titulación - Personal de la UTI
Encargado de pruebas (Tester):	- Director del proyecto de titulación - Personal de la UTI
Encargado de seguimiento:	- Director del proyecto de titulación - Personal de la UTI

1.3. PLAN DE ITERACIONES

El proyecto fue dividido en 4 iteraciones, por consiguiente, se obtuvo un total de cuatro entregas para las cuales se desarrollaron partes de la aplicación completamente funcionales**.**

1.3.1. Primera Iteración

En la TABLA A13. XXVI, se puede observar la primera iteración que consta de tres historias de usuario.

TABLA A13. XXVI.
PRIMERA ITERACIÓN PARA EL DESARROLLO DE SAC.

Iteración 01		
# Historia de Usuario	**Nombre de Historia de Usuario**	**Tarea**
1	Selección del Sistema SAC	Desarrollo de la interfaz gráfica de selección.
		Diseño de la modal de autenticación
2	Acceso al módulo de Administración	Diseño de la interfaz de Administración.
		Desarrollo del método para verificar las credenciales del administrador.
3	Acceso al módulo del Usuario	Diseño de la modal de usuarios.
		Desarrollo del método para verificar las credenciales del usuario.

1.3.2. Segunda Iteración

En la TABLA A13. XXVII, se puede observar la segunda iteración que consta de tres historias de usuario.

TABLA A13. XXVII.
SEGUNDA ITERACIÓN PARA EL DESARROLLO DE SAC.

Iteración 02		
# Historia de Usuario	**Nombre de Historia de Usuario**	**Tarea**
4	Visualizar información del usuario	Desarrollo del método para consultar y devolver la información de los usuarios del directorio centralizado.
		Diseño de la vista para presentar los datos personales del usuario
5	Cambio de contraseña	Verificación de la contraseña actual y nueva contraseña.

		Desarrollo del método de cambio de contraseña
		Diseño de la notificación de confirmación
		Diseño de la notificación de error
6	Cierre de sesión del sistema	Desarrollo del método para el cierre de sesión

1.3.3. Tercera Iteración

En la TABLA A13. XXVIII, se puede observar la tercera iteración que consta de siete historias de usuario.

TABLA A13. XXVIII.
TERCERA ITERACIÓN PARA EL DESARROLLO DE SAC.

Iteración 03		
# Historia de Usuario	**Nombre de Historia de Usuario**	**Tarea**
7	Añadir nuevo usuario	Diseño de la modal para ingresar los datos de usuario.
		Desarrollo del método para validar los datos.
		Desarrollo del método para añadir los datos.
		Desarrollo de la notificación de confirmación.
		Desarrollo de la notificación de error.
8	Actualizar información de un usuario	Diseño de la modal para presentar los datos de usuario.
		Desarrollo del método para validar los datos.
		Desarrollo del método para actualizar los datos.
		Desarrollo de la notificación de confirmación.
		Desarrollo de la notificación de error.

9	Eliminar Usuario	Desarrollo del método para eliminar a un usuario.
		Desarrollo de la notificación de confirmación.
		Desarrollo de la notificación de error.
10	Buscar Usuario	Desarrollo de la entrada de datos para buscar.
		Desarrollo del método para buscar a un usuario.
11	Cargar usuarios	Desarrollo de la modal para seleccionar el archivo de usuarios.
		Desarrollo del método para cargar a los usuarios.
		Desarrollo de la notificación de confirmación.
		Desarrollo de la notificación de error.
12	Vincular usuario	Desarrollo de la modal para vincular a un usuario.
		Desarrollo de un método para consultar los sistemas.
		Desarrollo de un método para vincular un usuario.
		Desarrollo de la notificación de confirmación.
		Desarrollo de la notificación de error.
13	Vincular varios usuarios	Desarrollo de la interfaz para vincular Usuarios.
		Desarrollo de la notificación de confirmación.
		Desarrollo de la notificación de error.

1.3.4. Cuarta Iteración

En la TABLA A13. XXIX, se puede observar la cuarta iteración que consta de diez historias de usuario.

TABLA A13. XXIX.
CUARTA ITERACIÓN PARA EL DESARROLLO DE SAC.

Iteración 04		
# Historia de Usuario	**Nombre de Historia de Usuario**	**Tarea**
14	Visualizar información de grupos y sistemas	Desarrollo de la interfaz para presentar los datos de cada grupo, sistema, subgrupos y subsistemas.
		Desarrollo del método para consultar los grupos, sistemas, subgrupos y subsistemas.
15	Crear nuevos grupos	Diseño de la modal para ingresar la información del grupo, subgrupo, sistema y descripción.
		Desarrollo del método para validar los datos.
		Desarrollo del método para añadir los datos.
		Desarrollo de la notificación de confirmación.
		Desarrollo de la notificación de error.
16	Eliminar grupo	Desarrollo de la modal para elegir un grupo.
		Desarrollo del método para eliminar un grupo.
		Desarrollo de la notificación de confirmación.
		Desarrollo de la notificación de error.
17	Restablecer contraseña	Implementación de Tokens para la generación de los links de acceso
		Desarrollo del método para reemplazar una contraseña
		Desarrollo de una notificación de confirmación.
		Desarrollo de una notificación de error.
18	Crear nuevos sistemas	Diseño de la modal para ingresar la información del sistema, subsistema(s) y descripción.

		Desarrollo del método para validar los datos.
		Desarrollo del método para añadir los datos.
		Desarrollo de la notificación de confirmación.
		Desarrollo de la notificación de error.
19	Eliminar sistemas	Desarrollo de la modal para elegir un sistema.
		Desarrollo del método para eliminar un sistema.
		Desarrollo de la notificación de confirmación.
		Desarrollo de la notificación de error.
20	Generar Reportes	Desarrollo de la modal para seleccionar un grupo.
		Desarrollo del método para descargar el archivo en formato .csv
		Desarrollo de una notificación de confirmación de la generación de un reporte.
21	Eliminar todos los registros.	Desarrollo de una modal para seleccionar un grupo.
		Desarrollo de un modal para verificar la eliminación de los registros.
		Desarrollo de una notificación de confirmación.
		Desarrollo de la notificación de error.
22	Crear nuevos administradores de lectura.	Desarrollo de la modal para ingresar los datos del nuevo administrador.
		Desarrollo de la modal para verificar los datos ingresados.
		Desarrollo de una notificación de confirmación.
		Desarrollo de la notificación de error.

23	Visualizar usuarios vinculados.	Desarrollo de una interfaz para presentar la lista de los usuarios vinculados.
		Desarrollo del método para consultar los usuarios vinculados.
		Desarrollo de una notificación de confirmación.
		Desarrollo de una notificación de error.

1.4. TAREAS DE INGENIERÍA

A continuación, en las (TABLAS A13. XXX – A13. CV), se muestran las tareas de ingeniería las cuales fueron utilizadas para llevar a cabo el desarrollo del sistema.

TABLA A13. XXX.
TAREA DE INGENIERÍA 1 - HISTORIA DE USUARIO 1.

TAREA	
Número Tarea: 1	**Número de Historia:** 1
Nombre de la Tarea: Desarrollo de la interfaz gráfica de selección.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 15/10/2018	**Fecha fin:** 15/10/2018
Descripción: Se realizará la interfaz principal, para que el usuario y el administrador puedan seleccionar al sistema SAC.	

TABLA A13. XXXI.
TAREA DE INGENIERÍA 2 - HISTORIA DE USUARIO 1.

TAREA	
Número Tarea: 2	**Número de Historia:** 1
Nombre de la Tarea: Diseño de la modal de autenticación.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 15/10/2018	**Fecha fin:** 15/10/2018
Descripción: Se realizará la interfaz para el inicio de sesión, para que el usuario y el administrador puedan ingresar sus credenciales (Usuario y Contraseña).	

TABLA A13. XXXII.
TAREA DE INGENIERÍA 3 - HISTORIA DE USUARIO 2.

TAREA	
Número Tarea: 3	**Número de Historia:** 2
Nombre de la Tarea: Diseño de la interfaz de Administración.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 16/10/2018	**Fecha fin:** 16/10/2018
Descripción: Se realizará la interfaz para que el administrador pueda manipular las funcionalidades del sistema SAC.	

TABLA A13. XXXIII.
TAREA DE INGENIERÍA 4 - HISTORIA DE USUARIO 2.

TAREA	
Número Tarea: 4	**Número de Historia:** 2
Nombre de la Tarea: Desarrollo del método para verificar las credenciales del administrador.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 17/10/2018	**Fecha fin:** 17/10/2018
Descripción: Se realizará un método que permita recoger las credenciales ingresadas por el administrador y verificar si coinciden con las registradas en el directorio centralizado.	

TABLA A13. XXXIV.
TAREA DE INGENIERÍA 5 - HISTORIA DE USUARIO 3.

TAREA	
Número Tarea: 5	**Número de Historia:** 3
Nombre de la Tarea: Diseño de la modal de Usuarios.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 18/10/2018	**Fecha fin:** 18/10/2018
Descripción: Se realizará la interfaz para que el usuario pueda ver su información personal registrada en el directorio centralizado.	

TABLA A13. XXXV.
TAREA DE INGENIERÍA 6 - HISTORIA DE USUARIO 3.

TAREA	
Número Tarea: 6	**Número de Historia:** 3
Nombre de la Tarea: Desarrollo del método para verificar las credenciales del usuario.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 19/10/2018	**Fecha fin:** 19/10/2018
Descripción: Se realizará un método que permita recoger las credenciales ingresadas por el usuario y verificar si coinciden con las registradas en el directorio centralizado.	

TABLA A13. XXXVI.
TAREA DE INGENIERÍA 7 - HISTORIA DE USUARIO 4.

TAREA	
Número Tarea: 7	**Número de Historia:** 4
Nombre de la Tarea: Desarrollo del método para consultar y devolver la información de los usuarios del directorio centralizado.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 20/10/2018	**Fecha fin:** 21/10/2018
Descripción: Se realizará un método que permita consultar a todos los usuarios registrados en cada grupo dentro del directorio centralizado.	

TABLA A13. XXXVII.
TAREA DE INGENIERÍA 8 - HISTORIA DE USUARIO 4.

TAREA	
Número Tarea: 8	**Número de Historia:** 4
Nombre de la Tarea: Diseño de la vista para presentar los datos personales del usuario.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 21/10/2018	**Fecha fin:** 21/10/2018

Descripción: Se realizará la vista para presentarle al administrador la información de los usuarios (Cédula, Nombres, Apellidos, Correo, Teléfono, Usurario y opciones generales).

TABLA A13. XXXVIII.
TAREA DE INGENIERÍA 9 - HISTORIA DE USUARIO 5.

TAREA	
Número Tarea: 9	**Número de Historia:** 5
Nombre de la Tarea: Verificación de la contraseña actual y nueva contraseña.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 21/10/2018	**Fecha fin:** 22/10/2018
Descripción: Se realizará un método que permita verificar si la contraseña actual ingresada por el usuario coincide con la que tiene registrada en el directorio centralizado y posteriormente verificar si las nuevas contraseñas ingresadas coinciden entre sí.	

TABLA A13. XXXIX.
TAREA DE INGENIERÍA 10 - HISTORIA DE USUARIO 5.

TAREA	
Número Tarea: 10	**Número de Historia:** 5
Nombre de la Tarea: Desarrollo del método de cambio de contraseña	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 22/10/2018	**Fecha fin:** 23/10/2018
Descripción: Se realizará un método que permita recoger las contraseñas ingresadas por el usuario y cambiarlas en el directorio centralizado.	

TABLA A13. XL.
TAREA DE INGENIERÍA 11 - HISTORIA DE USUARIO 5.

TAREA	
Número Tarea: 11	**Número de Historia:** 5
Nombre de la Tarea: Diseño de la notificación de confirmación.	

Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 23/10/2018	**Fecha fin:** 23/10/2018
Descripción: Se realizará una notificación que permita informar al usuario que el cambio de contraseña se realizó con éxito.	

TABLA A13. XLI.
TAREA DE INGENIERÍA 12 - HISTORIA DE USUARIO 5.

TAREA	
Número Tarea: 12	**Número de Historia:** 5
Nombre de la Tarea: Diseño de la notificación de error.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 23/10/2018	**Fecha fin:** 23/10/2018
Descripción: Se realizará una notificación que permita informar al usuario que el cambio de contraseña no se pudo realizar con éxito.	

TABLA A13. XLII.
TAREA DE INGENIERÍA 13 - HISTORIA DE USUARIO 6.

TAREA	
Número Tarea: 13	**Número de Historia:** 6
Nombre de la Tarea: Desarrollo del método para el cierre de sesión.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 24/10/2018	**Fecha fin:** 24/10/2018
Descripción: Se realizará un método que le permita al administrador o al usuario destruir la sesión y re direccionar a la interfaz de autenticación.	

TABLA A13. XLIII.
TAREA DE INGENIERÍA 14 - HISTORIA DE USUARIO 7.

TAREA	
Número Tarea: 14	**Número de Historia:** 7
Nombre de la Tarea: Diseño de la modal para ingresar los datos de usuario.	

Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 25/10/2018	**Fecha fin:** 25/10/2018
Descripción: Se realizará una modal que le permita al administrador ingresar los datos de un nuevo usuario (Cédula, Nombres, Apellidos, Correo, Teléfono, Grupo).	

TABLA A13. XLIV.
TAREA DE INGENIERÍA 15 - HISTORIA DE USUARIO 7.

TAREA	
Número Tarea: 15	**Número de Historia:** 7
Nombre de la Tarea: Desarrollo del método para validar los datos.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 25/10/2018	**Fecha fin:** 26/10/2018
Descripción: Se realizará un método que recogerá los datos ingresados del nuevo usuario (Cédula, Nombres, Apellidos, Correo, Teléfono, Grupo), y valide si están correctos.	

TABLA A13. XLV.
TAREA DE INGENIERÍA 16 - HISTORIA DE USUARIO 7.

TAREA	
Número Tarea: 16	**Número de Historia:** 7
Nombre de la Tarea: Desarrollo del método para añadir los datos.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 26/10/2018	**Fecha fin:** 27/10/2018
Descripción: Se realizará un método que recogerá los datos ingresados del nuevo usuario (Cédula, Nombres, Apellidos, Correo, Teléfono, Grupo) y lo registre en el directorio centralizado.	

TABLA A13. XLVI.
TAREA DE INGENIERÍA 17 - HISTORIA DE USUARIO 7.

TAREA	
Número Tarea: 17	**Número de Historia:** 7

Nombre de la Tarea: Desarrollo de la notificación de confirmación.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 27/10/2018	**Fecha fin:** 27/10/2018
Descripción: Se realizará una notificación que permita informar al administrador que se ha registrado al nuevo usuario con éxito.	

TABLA A13. XLVII.
TAREA DE INGENIERÍA 18 - HISTORIA DE USUARIO 7.

TAREA	
Número Tarea: 18	**Número de Historia:** 7
Nombre de la Tarea: Desarrollo de la notificación de error.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 27/10/2018	**Fecha fin:** 27/10/2018
Descripción: Se realizará una notificación que permita informar al administrador que no se ha registrado al nuevo usuario con éxito.	

TABLA A13. XLVIII.
TAREA DE INGENIERÍA 19 - HISTORIA DE USUARIO 8.

TAREA	
Número Tarea: 19	**Número de Historia:** 8
Nombre de la Tarea: Diseño de la modal para presentar los datos de usuario.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 27/10/2018	**Fecha fin:** 28/10/2018
Descripción: Se realizará una modal que le permita visualizar al administrador los datos de un usuario (Cédula, Nombres, Apellidos, Correo, Teléfono, Contraseña, Grupo).	

TABLA A13. XLIX.
TAREA DE INGENIERÍA 20 - HISTORIA DE USUARIO 8.

TAREA	
Número Tarea: 20	**Número de Historia:** 8

Nombre de la Tarea: Desarrollo del método para validar los datos.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 27/10/2018	**Fecha fin:** 28/10/2018
Descripción: Se realizará un método que recogerá los datos actualizados de un usuario (Nombres, Apellidos, Correo, Teléfono, Contraseña, Grupo), y valide si están correctos.	

TABLA A13. L.
TAREA DE INGENIERÍA 21 - HISTORIA DE USUARIO 8.

TAREA	
Número Tarea: 21	**Número de Historia:** 8
Nombre de la Tarea: Desarrollo del método para actualizar los datos.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 27/10/2018	**Fecha fin:** 28/10/2018
Descripción: Se realizará un método que recogerá los datos actualizados de un usuario (Nombres, Apellidos, Correo, Teléfono, Contraseña, Grupo), y los actualice en el directorio centralizado.	

TABLA A13. LI.
TAREA DE INGENIERÍA 22 - HISTORIA DE USUARIO 8.

TAREA	
Número Tarea: 22	**Número de Historia:** 8
Nombre de la Tarea: Desarrollo de la notificación de confirmación.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 28/10/2018	**Fecha fin:** 28/10/2018
Descripción: Se realizará una notificación que permita informar al administrador que se a actualizado los datos de un usuario con éxito.	

TABLA A13. LII.
TAREA DE INGENIERÍA 23 - HISTORIA DE USUARIO 8.

TAREA	
Número Tarea: 23	**Número de Historia:** 8

Nombre de la Tarea: Desarrollo de la notificación de error.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 28/10/2018	**Fecha fin:** 28/10/2018
Descripción: Se realizará una notificación que permita informar al administrador que no se a actualizado los datos de un usuario con éxito.	

TABLA A13. LIII.
TAREA DE INGENIERÍA 24 - HISTORIA DE USUARIO 9.

TAREA	
Número Tarea: 24	**Número de Historia:** 9
Nombre de la Tarea: Desarrollo del método para eliminar a un usuario.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 29/10/2018	**Fecha fin:** 29/10/2018
Descripción: Se realizará un método que recogerá los datos de un usuario (Cédula y Usuario), y lo elimine del directorio centralizado.	

TABLA A13. LIV.
TAREA DE INGENIERÍA 25 - HISTORIA DE USUARIO 9.

TAREA	
Número Tarea: 25	**Número de Historia:** 9
Nombre de la Tarea: Desarrollo de la notificación de confirmación.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 29/10/2018	**Fecha fin:** 29/10/2018
Descripción: Se realizará una notificación que permita informar al administrador que se a eliminado a un usuario con éxito.	

TABLA A13. LV.
TAREA DE INGENIERÍA 26 - HISTORIA DE USUARIO 9.

TAREA	
Número Tarea: 26	**Número de Historia:** 9
Nombre de la Tarea: Desarrollo de la notificación de error.	

Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 29/10/2018	**Fecha fin:** 29/10/2018
Descripción: Se realizará una notificación que permita informar al administrador que no se a eliminado a un usuario con éxito.	

TABLA A13. LVI.
TAREA DE INGENIERÍA 27 - HISTORIA DE USUARIO 10.

TAREA	
Número Tarea: 27	**Número de Historia:** 10
Nombre de la Tarea: Desarrollo de la entrada de datos para buscar.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 30/10/2018	**Fecha fin:** 30/10/2018
Descripción: Se realizará una entrada de texto que le permita al administrador ingresar el criterio de búsqueda de un usuario.	

TABLA A13. LVII.
TAREA DE INGENIERÍA 28 - HISTORIA DE USUARIO 10.

TAREA	
Número Tarea: 28	**Número de Historia:** 10
Nombre de la Tarea: Desarrollo del método para buscar a un usuario.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 30/10/2018	**Fecha fin:** 30/10/2018
Descripción: Se realizará un método que permita recoger el criterio de búsqueda ingresado por el administrador y buscarlo en el directorio centralizado.	

TABLA A13. LVIII.
TAREA DE INGENIERÍA 29 - HISTORIA DE USUARIO 11.

TAREA	
Número Tarea: 29	**Número de Historia:** 11
Nombre de la Tarea: Desarrollo de la modal para seleccionar el archivo de usuarios.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1

Fecha de inicio: 31/10/2018	**Fecha fin:** 31/10/2018
Descripción: Se realizará una modal que le permita al administrador seleccionar un archivo con la extensión .csv desde su ordenador.	

TABLA A13. LIX.
TAREA DE INGENIERÍA 30 - HISTORIA DE USUARIO 11.

TAREA	
Número Tarea: 30	**Número de Historia:** 11
Nombre de la Tarea: Desarrollo del método para cargar a los usuarios.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 31/10/2018	**Fecha fin:** 01/10/2018
Descripción: Se realizará un método que lea a los usuarios del archivo .csv y los registre en el directorio centralizado.	

TABLA A13. LX.
TAREA DE INGENIERÍA 31 - HISTORIA DE USUARIO 11.

TAREA	
Número Tarea: 31	**Número de Historia:** 11
Nombre de la Tarea: Desarrollo de la notificación de confirmación.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 01/10/2018	**Fecha fin:** 01/11/2018
Descripción: Se realizará una notificación que permita informar al administrador que se han registrado a los usuarios del archivo .csv con éxito.	

TABLA A13. LXI.
TAREA DE INGENIERÍA 32 - HISTORIA DE USUARIO 11.

TAREA	
Número Tarea: 32	**Número de Historia:** 11
Nombre de la Tarea: Desarrollo de la notificación de error.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 01/11/2018	**Fecha fin:** 01/11/2018

Descripción: Se realizará una notificación que permita informar al administrador que no se han registrado a los usuarios del archivo .csv con éxito.

TABLA A13. LXII.
TAREA DE INGENIERÍA 33 - HISTORIA DE USUARIO 12.

TAREA	
Número Tarea: 33	**Número de Historia:** 12
Nombre de la Tarea: Desarrollo de un método para consultar los sistemas.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 02/11/2018	**Fecha fin:** 02/11/2018
Descripción: Se realizará un método que permita recoger los datos de un usuario (Cédula, Usuario), y permita buscar los sistemas referentes a ese usuario en el directorio centralizado.	

TABLA A13. LXIII.
TAREA DE INGENIERÍA 34 - HISTORIA DE USUARIO 12.

TAREA	
Número Tarea: 34	**Número de Historia:** 12
Nombre de la Tarea: Desarrollo de un método para vincular un usuario.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 02/11/2018	**Fecha fin:** 03/11/2018
Descripción: Se realizará un método que permita recoger los datos de un usuario (Usuario), y permita vincularlo a los sistemas que el administrador ha seleccionado.	

TABLA A13. LXIV.
TAREA DE INGENIERÍA 35 - HISTORIA DE USUARIO 12.

TAREA	
Número Tarea: 35	**Número de Historia:** 12
Nombre de la Tarea: Desarrollo de la notificación de confirmación.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 03/11/2018	**Fecha fin:** 03/11/2018

Descripción: Se realizará una notificación que permita informar al administrador que un usuario fue vinculado con éxito.

TABLA A13. LXV.
TAREA DE INGENIERÍA 36 - HISTORIA DE USUARIO 12.

TAREA	
Número Tarea: 36	**Número de Historia:** 12
Nombre de la Tarea: Desarrollo de la notificación de error.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 03/11/2018	**Fecha fin:** 03/11/2018
Descripción: Se realizará una notificación que permita informar al administrador que un usuario no fue vinculado con éxito.	

TABLA A13. LXVI.
TAREA DE INGENIERÍA 37 - HISTORIA DE USUARIO 13.

TAREA	
Número Tarea: 37	**Número de Historia:** 13
Nombre de la Tarea: Desarrollo de la interfaz para vincular Usuarios.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 04/11/2018	**Fecha fin:** 04/11/2018
Descripción: Se realizará una interfaz que le permita visualizar al administrador la información de todos los usuarios registrados en el directorio centralizado.	

TABLA A13. LXVII.
TAREA DE INGENIERÍA 38 - HISTORIA DE USUARIO 13.

TAREA	
Número Tarea: 38	**Número de Historia:** 13
Nombre de la Tarea: Desarrollo de la notificación de confirmación.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 04/11/2018	**Fecha fin:** 04/11/2018
Descripción: Se realizará una notificación que permita informar al administrador que los usuarios fueron vinculados con éxito.	

TABLA A13. LXVIII.
TAREA DE INGENIERÍA 39 - HISTORIA DE USUARIO 13.

TAREA	
Número Tarea: 39	**Número de Historia:** 13
Nombre de la Tarea: Desarrollo de la notificación de error.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 04/11/2018	**Fecha fin:** 04/11/2018
Descripción: Se realizará una notificación que permita informar al administrador que los usuarios no fueron vinculados con éxito.	

TABLA A13. LXIX.
TAREA DE INGENIERÍA 40 - HISTORIA DE USUARIO 14.

TAREA	
Número Tarea: 40	**Número de Historia:** 14
Nombre de la Tarea: Desarrollo de la interfaz para presentar los datos de cada grupo, sistema, subgrupos y subsistemas.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 05/11/2018	**Fecha fin:** 05/11/2018
Descripción: Se realizará una interfaz que le permita al administrador navegar por los diferentes grupos, sistemas, subgrupos y subsistemas que se encuentren registrados en el directorio centralizado.	

TABLA A13. LXX.
TAREA DE INGENIERÍA 41 - HISTORIA DE USUARIO 14.

TAREA	
Número Tarea: 41	**Número de Historia:** 14
Nombre de la Tarea: Desarrollo del método para consultar los grupos, sistemas, subgrupos y subsistemas.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 05/11/2018	**Fecha fin:** 05/11/2018
Descripción: Se realizará un método que permita consultar todos los grupos, sistemas, subgrupos y subsistemas del directorio centralizado.	

TABLA A13. LXXI.
TAREA DE INGENIERÍA 42 - HISTORIA DE USUARIO 15.

TAREA	
Número Tarea: 42	**Número de Historia:** 15
Nombre de la Tarea: Diseño de la modal para ingresar la información del grupo, subgrupo, sistema y descripción.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 06/11/2018	**Fecha fin:** 07/11/2018
Descripción: Se realizará una modal para que el administrador pueda ingresar la información del grupo, sistema, subgrupo y subsistema.	

TABLA A13. LXXII.
TAREA DE INGENIERÍA 43 - HISTORIA DE USUARIO 15.

TAREA	
Número Tarea: 43	**Número de Historia:** 15
Nombre de la Tarea: Desarrollo del método para validar los datos.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 07/11/2018	**Fecha fin:** 07/11/2018
Descripción: Se realizará un método que recoja la información ingresada por el administrador y valide si son correctos.	

TABLA A13. LXXIII.
TAREA DE INGENIERÍA 44 - HISTORIA DE USUARIO 15.

TAREA	
Número Tarea: 44	**Número de Historia:** 15
Nombre de la Tarea: Desarrollo del método para añadir los datos.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 07/11/2018	**Fecha fin:** 08/11/2018
Descripción: Se realizará un método que recoja la información ingresada por el administrador y los registre en el directorio centralizado.	

TABLA A13. LXXIV.
TAREA DE INGENIERÍA 45 - HISTORIA DE USUARIO 15.

TAREA	
Número Tarea: 45	**Número de Historia:** 15
Nombre de la Tarea: Desarrollo de la notificación de confirmación.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 08/11/2018	**Fecha fin:** 08/11/2018
Descripción: Se realizará una notificación para indicarle al administrador que el registro del grupo se realizó con éxito.	

TABLA A13. LXXV.
TAREA DE INGENIERÍA 46 - HISTORIA DE USUARIO 15.

TAREA	
Número Tarea: 46	**Número de Historia:** 15
Nombre de la Tarea: Desarrollo de la notificación de error.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 08/11/2018	**Fecha fin:** 08/11/2018
Descripción: Se realizará una notificación para indicarle al administrador que no se registró al grupo con éxito.	

TABLA A13. LXXVI.
TAREA DE INGENIERÍA 47 - HISTORIA DE USUARIO 16.

TAREA	
Número Tarea: 47	**Número de Historia:** 16
Nombre de la Tarea: Desarrollo de la modal para elegir un grupo.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 09/11/2018	**Fecha fin:** 09/11/2018
Descripción: Se realizará una modal que le permita al administrador seleccionar uno o más grupos para su eliminación.	

TABLA A13. LXXVII.
TAREA DE INGENIERÍA 48 - HISTORIA DE USUARIO 16.

TAREA	
Número Tarea: 48	**Número de Historia:** 16
Nombre de la Tarea: Desarrollo del método para eliminar un grupo.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 09/11/2018	**Fecha fin:** 10/11/2018
Descripción: Se realizará un método que recoja la información de uno o varios grupos y lo elimine del directorio centralizado.	

TABLA A13. LXXVIII.
TAREA DE INGENIERÍA 49 - HISTORIA DE USUARIO 16.

TAREA	
Número Tarea: 49	**Número de Historia:** 16
Nombre de la Tarea: Desarrollo de la notificación de confirmación.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 10/11/2018	**Fecha fin:** 10/11/2018
Descripción: Se realizará una notificación para indicarle al administrador que el o los grupos se eliminaron con éxito.	

TABLA A13. LXXIX.
TAREA DE INGENIERÍA 50 - HISTORIA DE USUARIO 16.

TAREA	
Número Tarea: 50	**Número de Historia:** 16
Nombre de la Tarea: Desarrollo de la notificación de error.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 10/11/2018	**Fecha fin:** 10/11/2018
Descripción: Se realizará una notificación para indicarle al administrador que el o los grupos no se eliminaron con éxito.	

TABLA A13. LXXX.
TAREA DE INGENIERÍA 51 - HISTORIA DE USUARIO 17.

TAREA	
Número Tarea: 51	**Número de Historia:** 17
Nombre de la Tarea: Implementación de Tokens para la generación de los links de acceso.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 3
Fecha de inicio: 11/11/2018	**Fecha fin:** 11/11/2018
Descripción: Se implementará una librería JWT que permita la generación de Tokens encriptados para que el usuario pueda restablecer su contraseña.	

TABLA A13. LXXXI.
TAREA DE INGENIERÍA 52 - HISTORIA DE USUARIO 17.

TAREA	
Número Tarea: 52	**Número de Historia:** 17
Nombre de la Tarea: Desarrollo del método para reemplazar una contraseña.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 11/11/2018	**Fecha fin:** 12/11/2018
Descripción: Se desarrollará un método que permita verificar las contraseñas ingresadas y reemplazarla en el directorio centralizado.	

TABLA A13. LXXXII.
TAREA DE INGENIERÍA 53 - HISTORIA DE USUARIO 17.

TAREA	
Número Tarea: 53	**Número de Historia:** 17
Nombre de la Tarea: Desarrollo de una notificación de confirmación.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 12/11/2018	**Fecha fin:** 13/11/2018
Descripción: Se realizará una notificación para indicarle al usuario que su contraseña fue restablecida con éxito.	

TABLA A13. LXXXIII.
TAREA DE INGENIERÍA 54 - HISTORIA DE USUARIO 17.

TAREA	
Número Tarea: 54	**Número de Historia:** 17
Nombre de la Tarea: Desarrollo de una notificación de error.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 12/11/2018	**Fecha fin:** 13/11/2018
Descripción: Se realizará una notificación para indicarle al usuario que su contraseña no fue restablecida con éxito.	

TABLA A13. LXXXIV.
TAREA DE INGENIERÍA 55 - HISTORIA DE USUARIO 18.

TAREA	
Número Tarea: 55	**Número de Historia:** 18
Nombre de la Tarea: Diseño de la modal para ingresar la información del sistema, subsistema(s) y descripción.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio:	**Fecha fin:**
Descripción: Se realizará una modal que le permita al administrador ingresar la información del sistema.	

TABLA A13. LXXXV.
TAREA DE INGENIERÍA 56 - HISTORIA DE USUARIO 18.

TAREA	
Número Tarea: 56	**Número de Historia:** 18
Nombre de la Tarea: Desarrollo del método para validar los datos.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 14/11/2018	**Fecha fin:** 14/11/2018
Descripción: Se realizará un método que permita recoger la información ingresada por el administrador y validar si es correcta.	

TABLA A13. LXXXVI.
TAREA DE INGENIERÍA 57 - HISTORIA DE USUARIO 18.

TAREA	
Número Tarea: 57	**Número de Historia:** 18
Nombre de la Tarea: Desarrollo del método para añadir los datos.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 14/11/2018	**Fecha fin:** 14/11/2018
Descripción: Se realizará un método que permita recoger la información ingresada por el administrador e ingresarla en el directorio centralizado.	

TABLA A13. LXXXVII.
TAREA DE INGENIERÍA 58 - HISTORIA DE USUARIO 18.

TAREA	
Número Tarea: 58	**Número de Historia:** 18
Nombre de la Tarea: Desarrollo de la notificación de confirmación.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 14/11/2018	**Fecha fin:** 14/11/2018
Descripción: Se realizará una notificación para indicarle al administrador que un sistema fue creado con éxito.	

TABLA A13. LXXXVIII.
TAREA DE INGENIERÍA 59 - HISTORIA DE USUARIO 18.

TAREA	
Número Tarea: 59	**Número de Historia:** 18
Nombre de la Tarea: Desarrollo de la notificación de error.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 14/11/2018	**Fecha fin:** 14/11/2018
Descripción: Se realizará una notificación para indicarle al administrador que un sistema no fue creado con éxito.	

TABLA A13. LXXXIX.
TAREA DE INGENIERÍA 60 - HISTORIA DE USUARIO 19.

TAREA	
Número Tarea: 60	**Número de Historia:** 19
Nombre de la Tarea: Desarrollo de la modal para elegir un sistema.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 15/11/2018	**Fecha fin:** 15/11/2018
Descripción: Se realizará una modal para que el administrador pueda seleccionar uno o más sistemas a ser eliminados.	

TABLA A13. XC.
TAREA DE INGENIERÍA 61 - HISTORIA DE USUARIO 19.

TAREA	
Número Tarea: 61	**Número de Historia:** 19
Nombre de la Tarea: Desarrollo del método para eliminar un sistema.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 15/11/2018	**Fecha fin:** 15/11/2018
Descripción: Se realizará un método que permite recoger la selección de los sistemas para eliminarlos en el directorio centralizado.	

TABLA A13. XCI.
TAREA DE INGENIERÍA 62 - HISTORIA DE USUARIO 19.

TAREA	
Número Tarea: 62	**Número de Historia:** 19
Nombre de la Tarea: Desarrollo de la modal de confirmación.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 15/11/2018	**Fecha fin:** 15/11/2018
Descripción: Se realizará una modal para indicarle al administrador que el sistema fue eliminado con éxito.	

TABLA A13. XCII.
TAREA DE INGENIERÍA 63 - HISTORIA DE USUARIO 19.

TAREA	
Número Tarea: 63	**Número de Historia:** 19
Nombre de la Tarea: Desarrollo de la notificación de error.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 15/11/2018	**Fecha fin:** 15/11/2018
Descripción: Se realizará una modal para indicarle al administrador que el sistema no fue eliminado con éxito.	

TABLA A13. XCIII.
TAREA DE INGENIERÍA 64 - HISTORIA DE USUARIO 20.

TAREA	
Número Tarea: 64	**Número de Historia:** 20
Nombre de la Tarea: Desarrollo del método para descargar el archivo en formato csv.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 16/11/2018	**Fecha fin:** 17/11/2018
Descripción: Se realizará un método para que permita descargar un archivo .cv con los usuarios de un grupo registrados en el directorio centralizado.	

TABLA A13. XCIV.
TAREA DE INGENIERÍA 65 - HISTORIA DE USUARIO 20.

TAREA	
Número Tarea: 65	**Número de Historia:** 20
Nombre de la Tarea: Desarrollo de una notificación de confirmación de la generación de un reporte.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 17/11/2018	**Fecha fin:** 17/11/2018
Descripción: Se realizará una notificación que le permita indicarle al administrador que el reporte está generado.	

TABLA A13. XCV.
TAREA DE INGENIERÍA 66 - HISTORIA DE USUARIO 21.

TAREA	
Número Tarea: 66	**Número de Historia:** 21
Nombre de la Tarea: Desarrollo de una modal para seleccionar un grupo.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 18/11/2018	**Fecha fin:** 18/11/2018
Descripción: Se realizará una modal para que el administrador seleccione un grupo.	

TABLA A13. XCVI.
TAREA DE INGENIERÍA 67 - HISTORIA DE USUARIO 21.

TAREA	
Número Tarea: 67	**Número de Historia:** 21
Nombre de la Tarea: Desarrollo de una modal para verificar la eliminación de los usuarios.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 18/11/2018	**Fecha fin:** 18/11/2018
Descripción: Se realizará una modal para que el administrador pueda saber cuántos usuarios a eliminado.	

TABLA A13. XCVII.
TAREA DE INGENIERÍA 68 - HISTORIA DE USUARIO 21.

TAREA	
Número Tarea: 68	**Número de Historia:** 21
Nombre de la Tarea: Desarrollo de una modal de confirmación.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 18/11/2018	**Fecha fin:** 18/11/2018
Descripción: Se realizará una modal para indicarle al administrador que confirmé la eliminación de todos los usuarios.	

TABLA A13. XCVIII.
TAREA DE INGENIERÍA 69 - HISTORIA DE USUARIO 21.

TAREA	
Número Tarea: 69	**Número de Historia:** 21
Nombre de la Tarea: Desarrollo de la notificación de confirmación.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 18/11/2018	**Fecha fin:** 18/11/2018
Descripción: Se realizará una notificación para indicarle al administrador que los usuarios fueron eliminados con éxito.	

TABLA A13. XCIX.
TAREA DE INGENIERÍA 70 - HISTORIA DE USUARIO 21.

TAREA	
Número Tarea: 70	**Número de Historia:** 21
Nombre de la Tarea: Desarrollo de la notificación de error.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 18/11/2018	**Fecha fin:** 18/11/2018
Descripción: Se realizará una notificación para indicarle al administrador que los usuarios no fueron eliminados con éxito.	

TABLA A13. C.
TAREA DE INGENIERÍA 71 - HISTORIA DE USUARIO 22.

TAREA	
Número Tarea: 71	**Número de Historia:** 22
Nombre de la Tarea: Desarrollo de la modal para ingresar los datos del nuevo administrador.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 19/11/2018	**Fecha fin:** 19/11/2018
Descripción: Se realizará una modal que le permita al administrador ingresar la información de un nuevo administrador de lectura (Usuario, Contraseña, Descripción).	

TABLA A13. CI.
TAREA DE INGENIERÍA 72 - HISTORIA DE USUARIO 22.

TAREA	
Número Tarea: 72	**Número de Historia:** 22
Nombre de la Tarea: Desarrollo del método para verificar los datos ingresados.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 19/11/2018	**Fecha fin:** 20/11/2018
Descripción: Se realizará un método para recoger los datos ingresados por al administrador y verificar si se encuentran correctos.	

TABLA A13. CII.
TAREA DE INGENIERÍA 73 - HISTORIA DE USUARIO 22.

TAREA	
Número Tarea: 73	**Número de Historia:** 22
Nombre de la Tarea: Desarrollo de una notificación de confirmación.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 20/11/2018	**Fecha fin:** 20/11/2018
Descripción: Se realizará una notificación para indicarle al administrador que se creó con éxito en nuevo registro.	

TABLA A13. CIII.
TAREA DE INGENIERÍA 74 - HISTORIA DE USUARIO 22.

TAREA	
Número Tarea: 74	**Número de Historia:** 22
Nombre de la Tarea: Desarrollo de una notificación de error.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 20/11/2018	**Fecha fin:** 20/11/2018
Descripción: Se realizará una notificación para indicarle al administrador que no se creó con éxito en nuevo registro.	

TABLA A13. CIV.
TAREA DE INGENIERÍA 75 - HISTORIA DE USUARIO 23.

TAREA	
Número Tarea: 75	**Número de Historia:** 23
Nombre de la Tarea: Desarrollo de una interfaz para presentar la lista de usuarios vinculados.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 1
Fecha de inicio: 21/11/2018	**Fecha fin:** 22/11/2018
Descripción: Se realizará una interfaz que permite presentarle al administrador la información general de un sistema (Nombre), y de los usuarios que se encuentran vinculados en el mismo (Usuario, Nombre del sistema, Ruta del sistema).	

TABLA A13. CV.
TAREA DE INGENIERÍA 76 - HISTORIA DE USUARIO 23.

TAREA	
Número Tarea: 76	**Número de Historia:** 23
Nombre de la Tarea: Desarrollo del método para consultar los usuarios vinculados.	
Tipo de Tarea: Desarrollo	**Puntos estimados:** 2
Fecha de inicio: 22/11/2018	**Fecha fin:** 22/11/2018
Descripción: Se realizará un método que permita recoger el nombre de un sistema y consultar a todos los usuarios que se encuentran vinculados en el mismo.	

1.5. VELOCIDAD DEL SISTEMA

La TABLA A13. CVI, describe el tiempo empleado en el desarrollo de cada iteración en horas y días con un tiempo total de 156 horas.

TABLA A13. CVI.
TIEMPO DE DESARROLLO DE CADA ITERACIÓN.

Velocidad del sistema					
	Iteración 1	Iteración 2	Iteración 3	Iteración 4	
Horas	4	4	4	4	

Días	5	5	11	18	**Total horas**
# Historias de Usuario	3	3	7	10	
Horas totales por iteración.	20	20	44	72	156

1.6. PLAN DE ENTREGAS

Partiendo de las historias de usuarios, se realizó una planificación de 4 iteraciones con una duración como se describe en la TABLA A13. CVI. fue al término de este plazo que se realizaron entregas, las cuales siempre fueron funcionales, lo que quiere decir que al momento de la entrega estaban en condiciones de ser puestas en funcionamiento para hacer uso mediante el cliente. De esta manera se generó un éxito en el desarrollo del proyecto ya que mantenía el interés del cliente en continuarlo debido a que estaba viendo resultados en el corto plazo.

Programa de entregas y reuniones

1. Se realizaron 4 reuniones iniciales.
2. Las historias de usuario se las realizó en conjunto por el grupo de trabajo, incluyendo al director y personal de la UTI que tomaron el cargo de clientes, lo cual no generó problemas en las entregas.
3. La clasificación de historias de usuarios, no se las realizó por su grado de importancia. Se tomó en cuenta desarrollar los módulos de cada iteración en un orden secuencial, para ir verificando las actividades del sistema.
4. Para aproximar el tiempo que tardaría cada iteración, se optó por separar cada iteración por 1 semana, tomando en cuenta todos los 7 días de la misma, en donde se trabajan 4 horas sin interrupciones y así cumplir con el cronograma propuesto en el anteproyecto del tema de titulación. Ver TABLA A15. CVII.
5. Se realizaron 4 entregas, es decir 1 entrega por cada iteración.

TABLA A13. CVII.
PROGRAMA DE ENTREGAS DE LAS HISTORIAS DE USUARIO.

# Historia	# Iteración	Prioridad	Fecha de inicio	Fecha final
1	1	Alta	15/10/2018	16/10/2018
2	1	Alta	17/10/2018	18/10/2018

3	1	Alta	18/10/2018	19/10/2018
4	2	Alta	20/10/2018	21/10/2018
5	2	Alta	21/10/2018	23/10/2018
6	2	Alta	24/10/2018	24/10/2018
7	3	Alta	25/10/2018	27/10/2018
8	3	Alta	27/10/2018	28/10/2018
9	3	Alta	29/10/2018	29/10/2018
10	3	Alta	30/10/2018	30/10/2018
11	3	Alta	31/10/2018	01/10/2018
12	3	Alta	02/11/2018	03/11/2018
13	3	Alta	04/11/2018	04/11/2018
14	4	Alta	05/11/2018	05/11/2018
15	4	Alta	06/11/2018	08/11/2018
16	4	Alta	09/11/2018	10/11/2018
17	4	Alta	11/11/2018	13/11/2018
18	4	Alta	14/11/2018	14/11/2018
19	4	Alta	15/11/2018	15/11/2018
20	4	Alta	16/11/2018	17/11/2018
21	4	Alta	18/11/2018	18/11/2018
22	4	Alta	19/11/2018	20/11/2018
23	4	Alta	21/11/2018	22/11/2018

1.7. CRONOGRAMA DE ENTREGAS

La TABLA A13. CVIII, describe la fecha en la que se entregó el desarrollo de cada iteración definida en las fases anteriores.

TABLA A13. CVIII.
CRONOGRAMA DE ENTREGAS.

Iteración	Fecha de Entrega
1	19/10/2018

2	24/10/2018
3	05/11/2018
4	22/11/2018

2. FASE 2: Diseño

2.1. DIAGRAMA SIMPLE.

La metodología XP sugiere que hay que conseguir diseños simples y sencillos. Por lo que se estableció un diseño fácilmente entendible e implementable que costará menos tiempo y esfuerzo desarrollar, como se puede observar el la Figura A13. 1., la interfaz principal de administración una vez desarrollada.

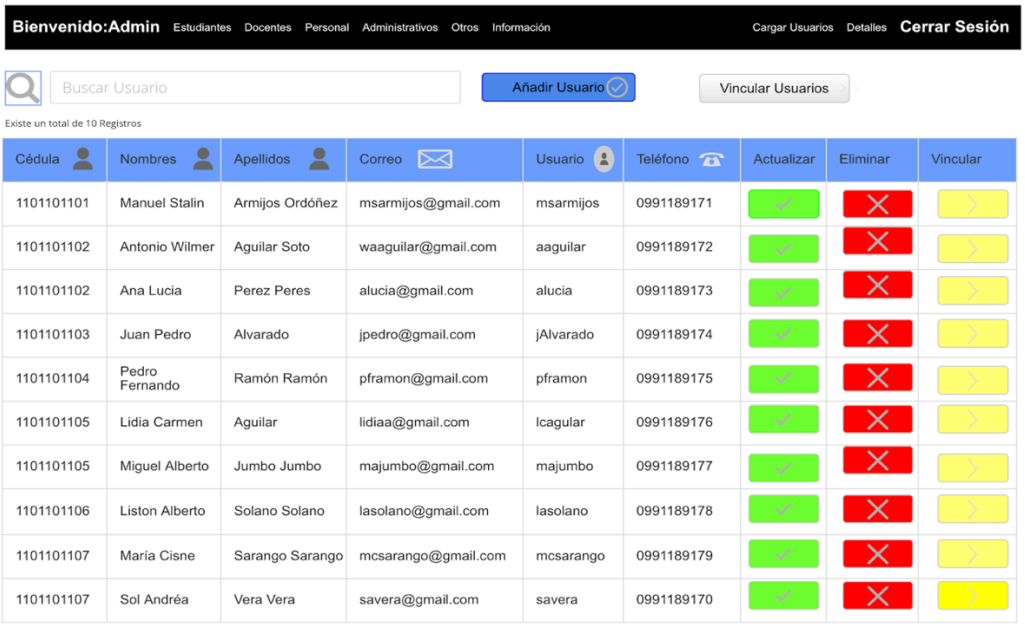


Figura A13. 1. Diagrama simple de la página principal del administrador de SAC.

Diagrama de secuencia.

En las (Figuras A13. 2 – A13. 18), se establecen los diagramas de secuencia que permitieron definir las interacciones de un conjunto de objetos de SAC a través del tiempo, en el cual se indican los módulos o clases que formaran parte del sistema y las llamadas que se hacen cada uno de ellos para realizar una tarea determinada.

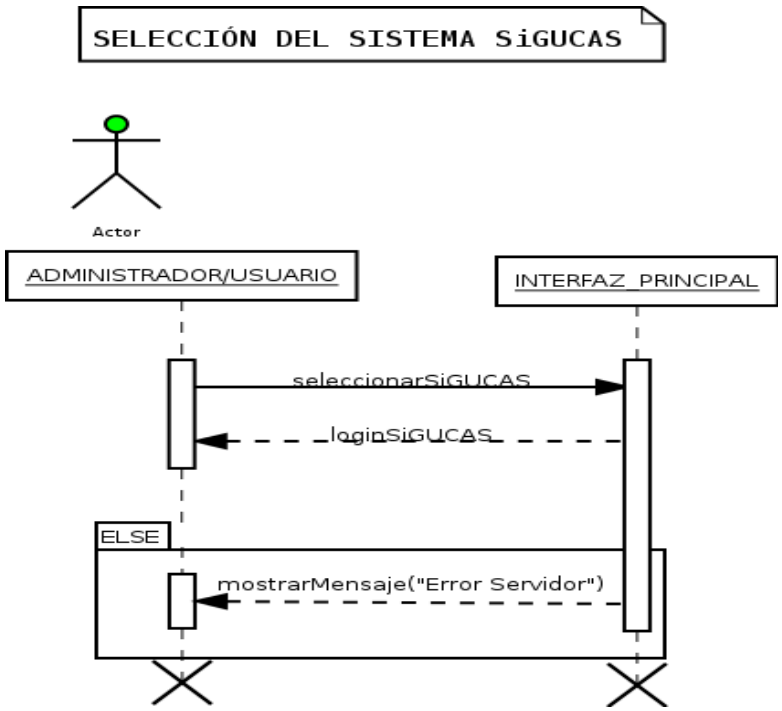


Figura A13. 2. Diagrama de secuencia de la selección del sistema SAC.

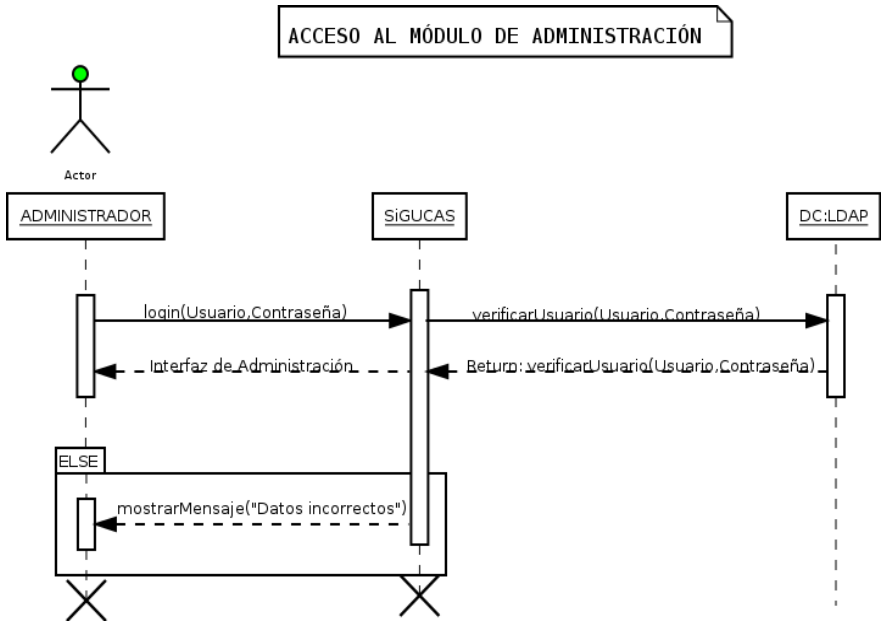


Figura A13. 3. Diagrama de secuencia sobre el acceso al módulo de administración.

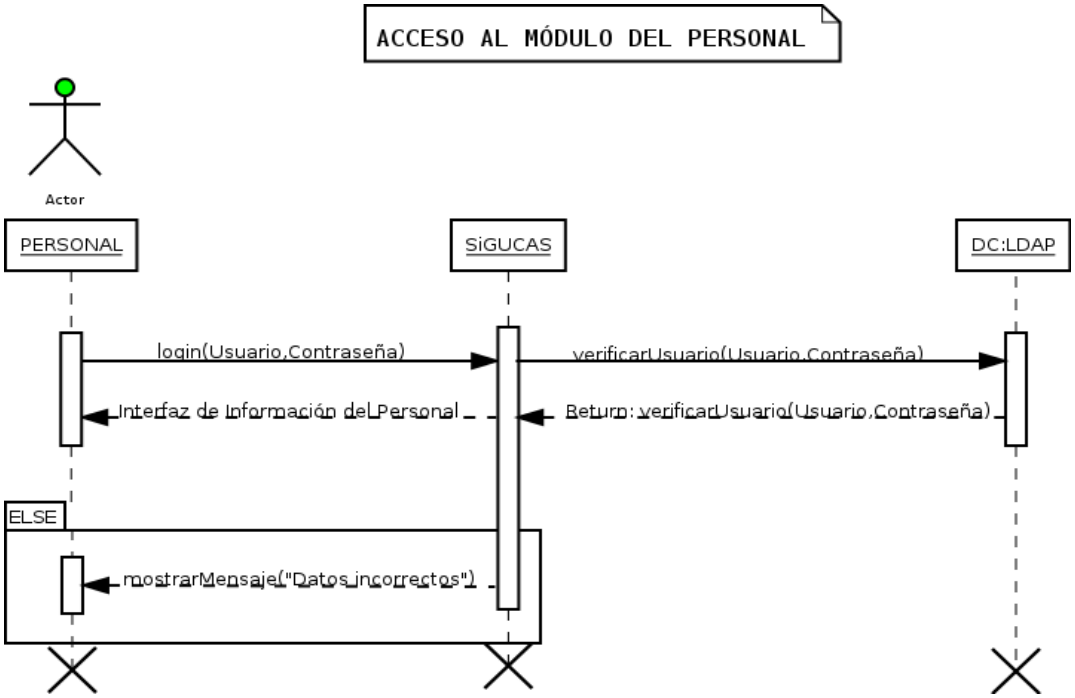


Figura A13. 4. Diagrama de secuencia sobre el acceso al módulo del personal.

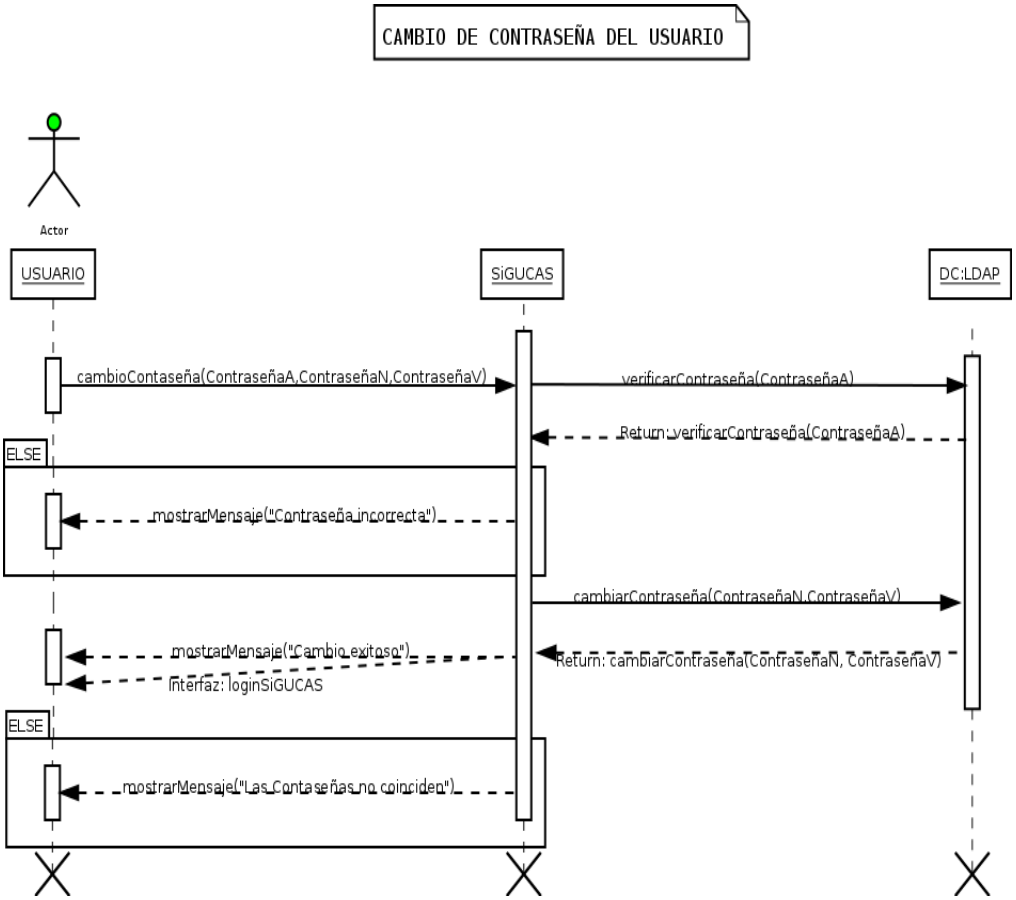


Figura A13. 5. Diagrama de secuencia para el cambio de contraseño de los usuarios.

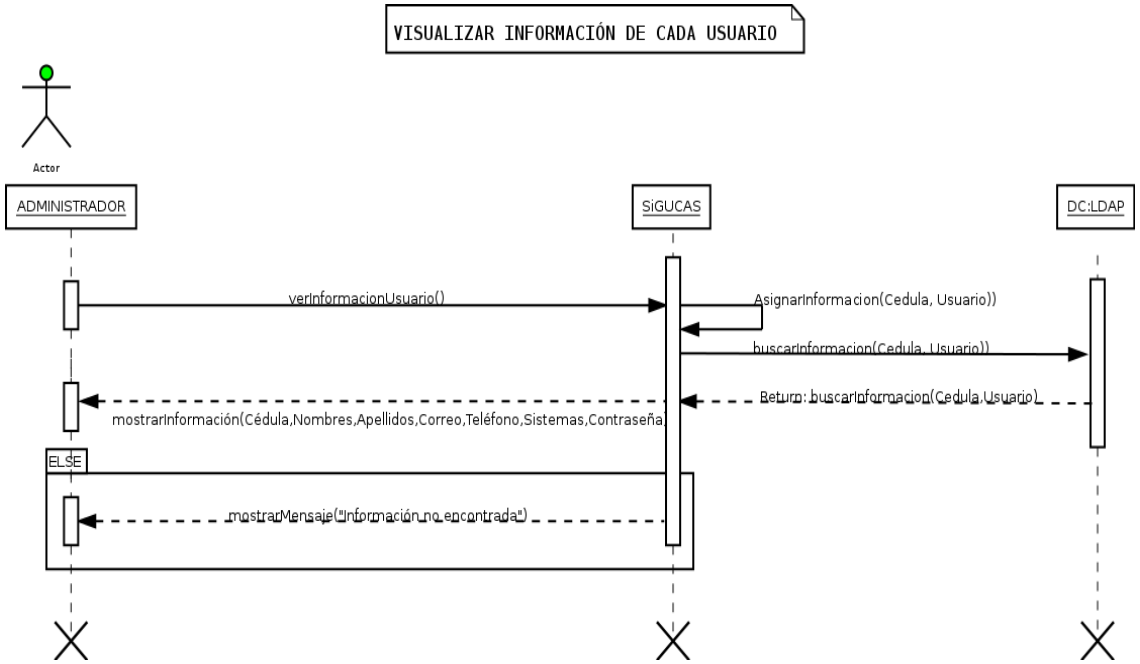


Figura A13. 6. Diagrama de secuencia para visualizar la información de cada usuario.

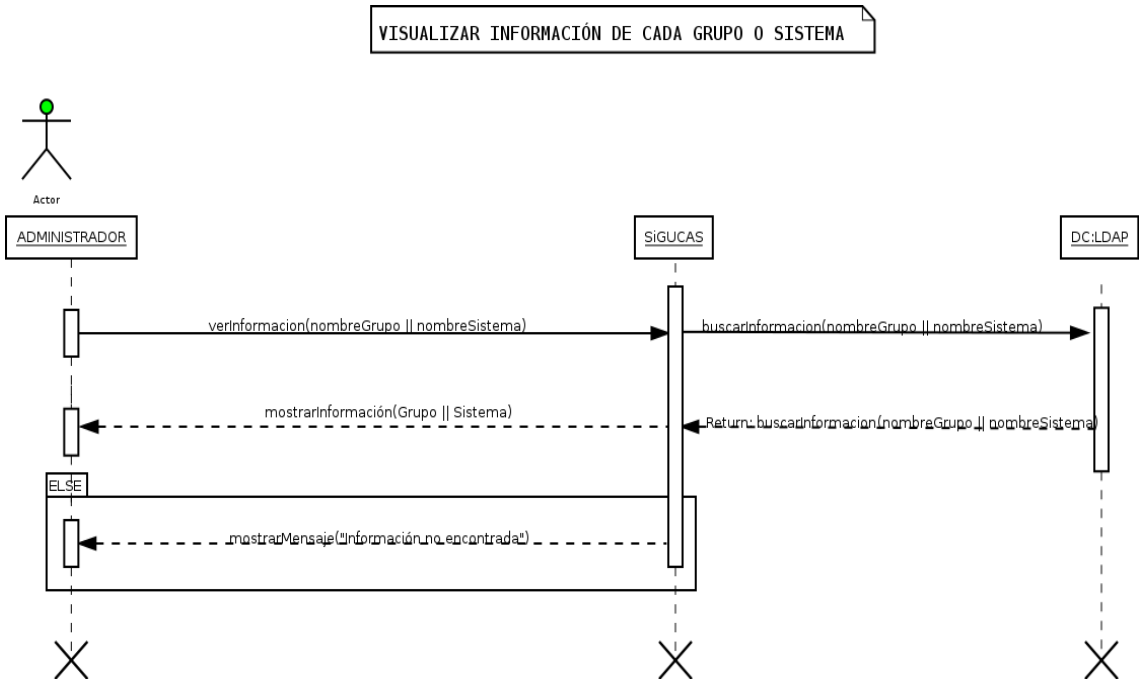


Figura A13. 7. Diagrama de secuencia para visualizar la información de cada grupo o sistema.

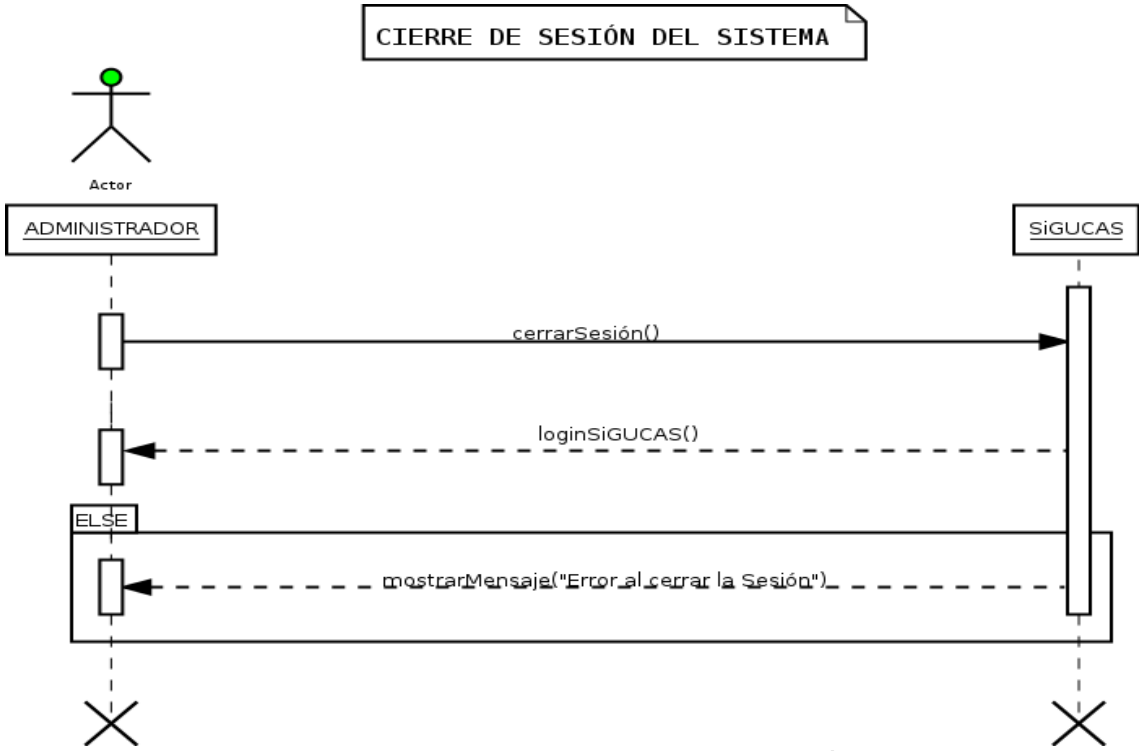


Figura A13. 8. Diagrama de secuencia para el cierre de sesión del sistema SAC.

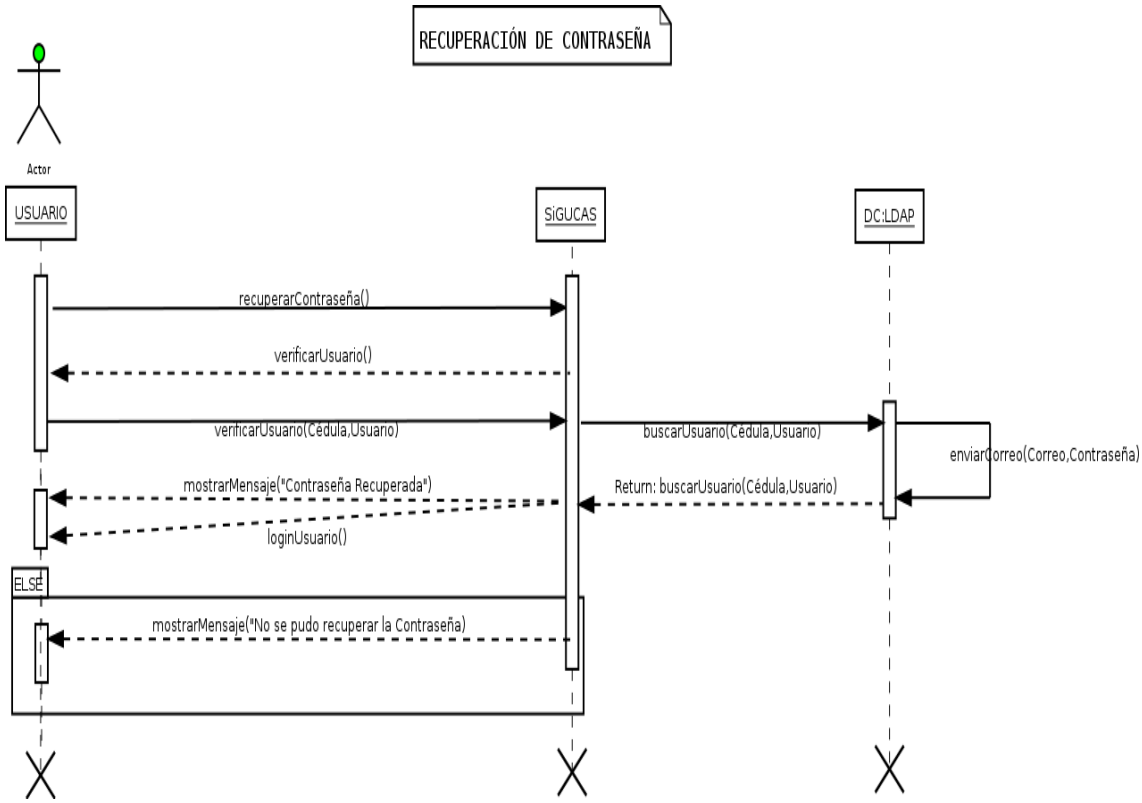


Figura A13. 9. Diagrama de secuencia para la recuperación de contraseña.

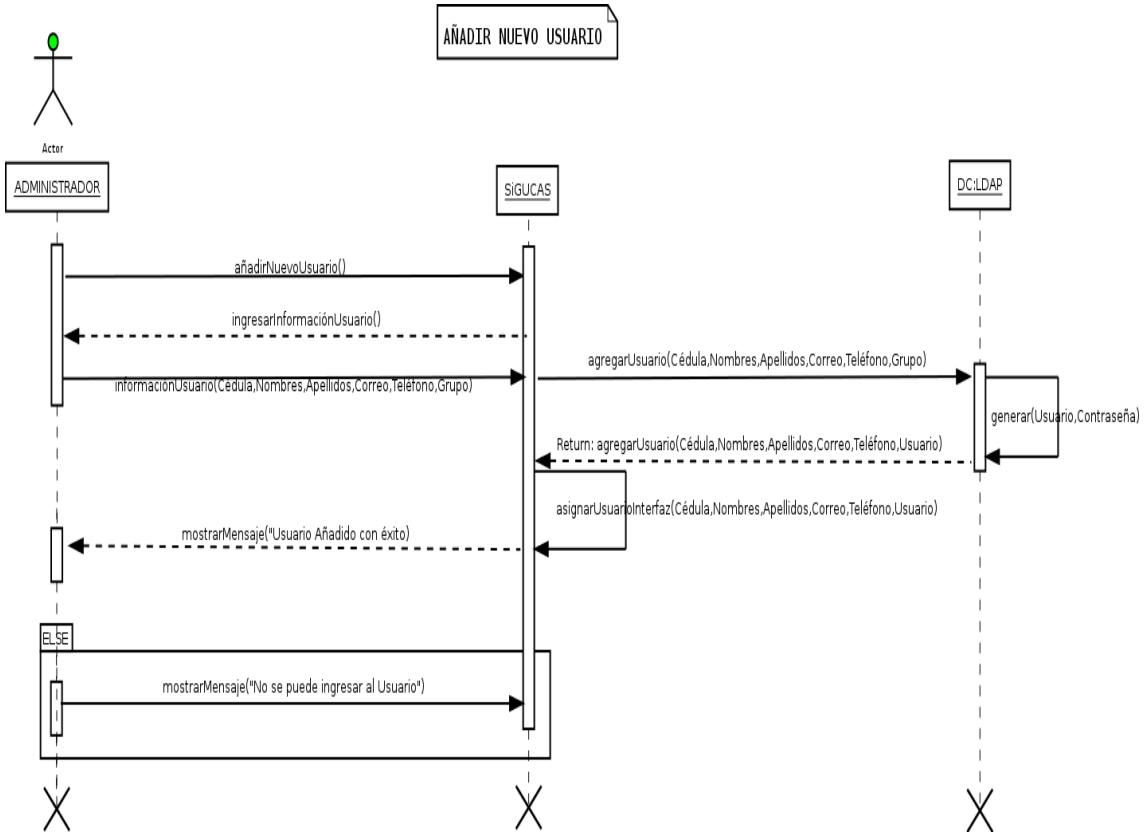


Figura A13. 10. Diagrama de secuencia para añadir un nuevo usuario.

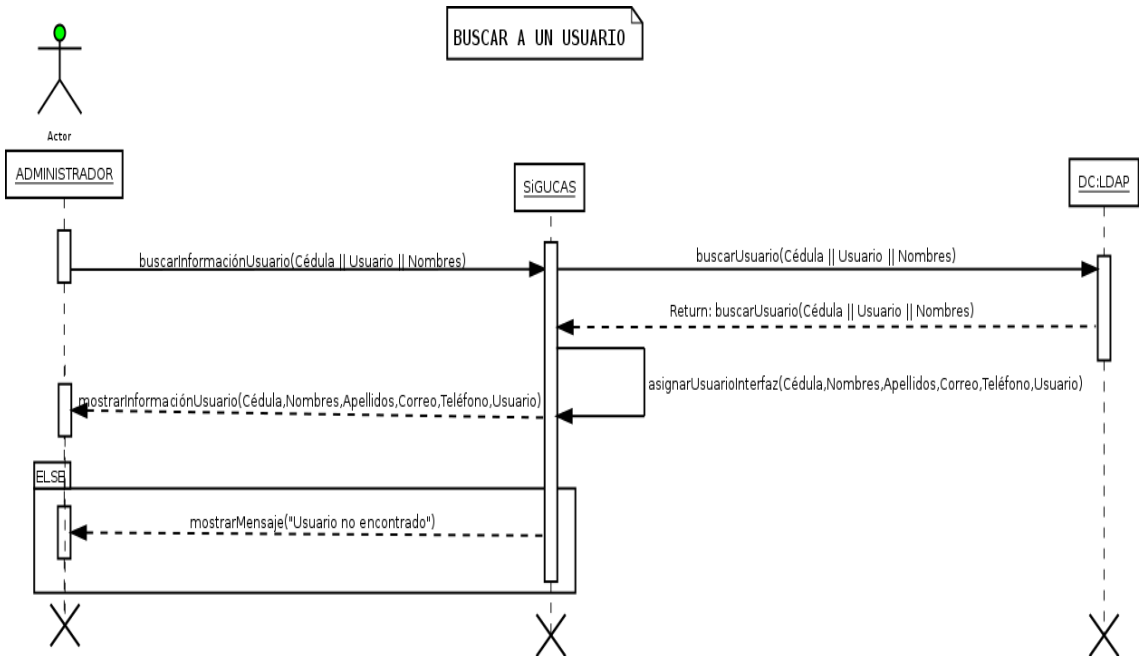


Figura A13. 11. Diagrama de secuencia para buscar a un usuario.

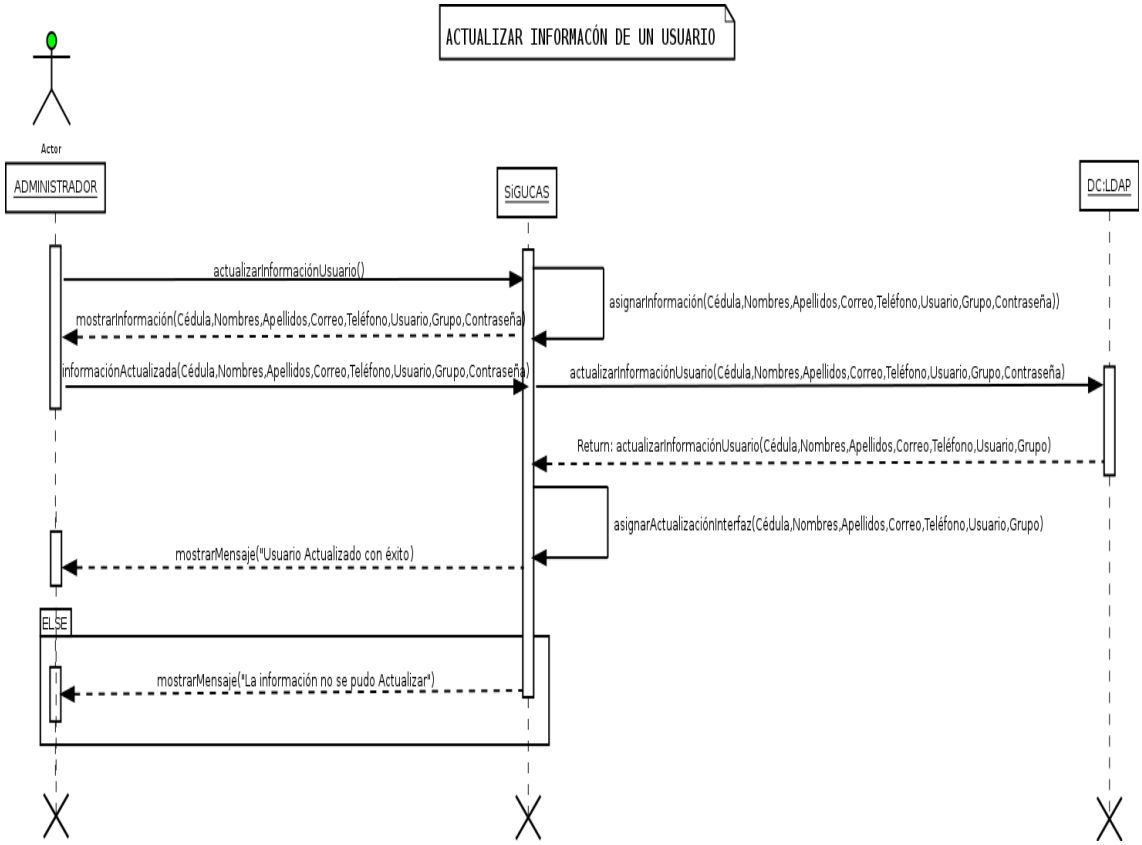


Figura A13. 12. Diagrama de secuencia para actualizar la información de un usuario.

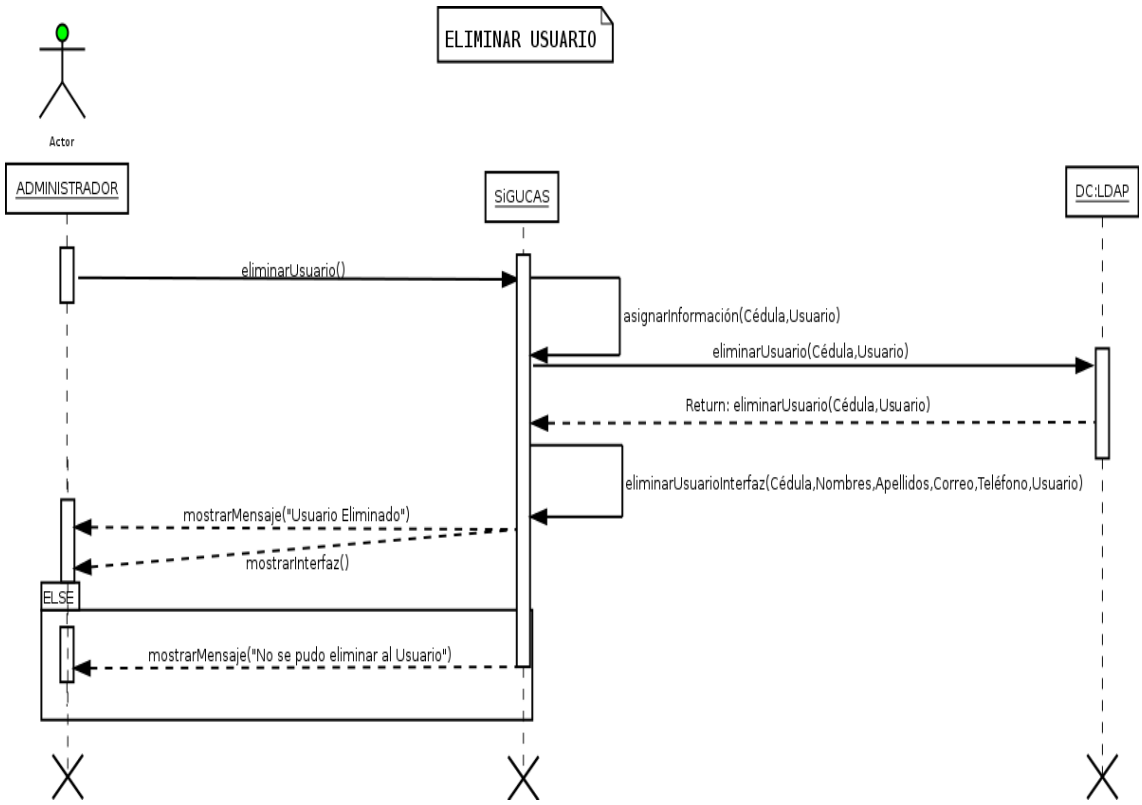


Figura A13. 13. Diagrama de secuencia para eliminar a un usuario.

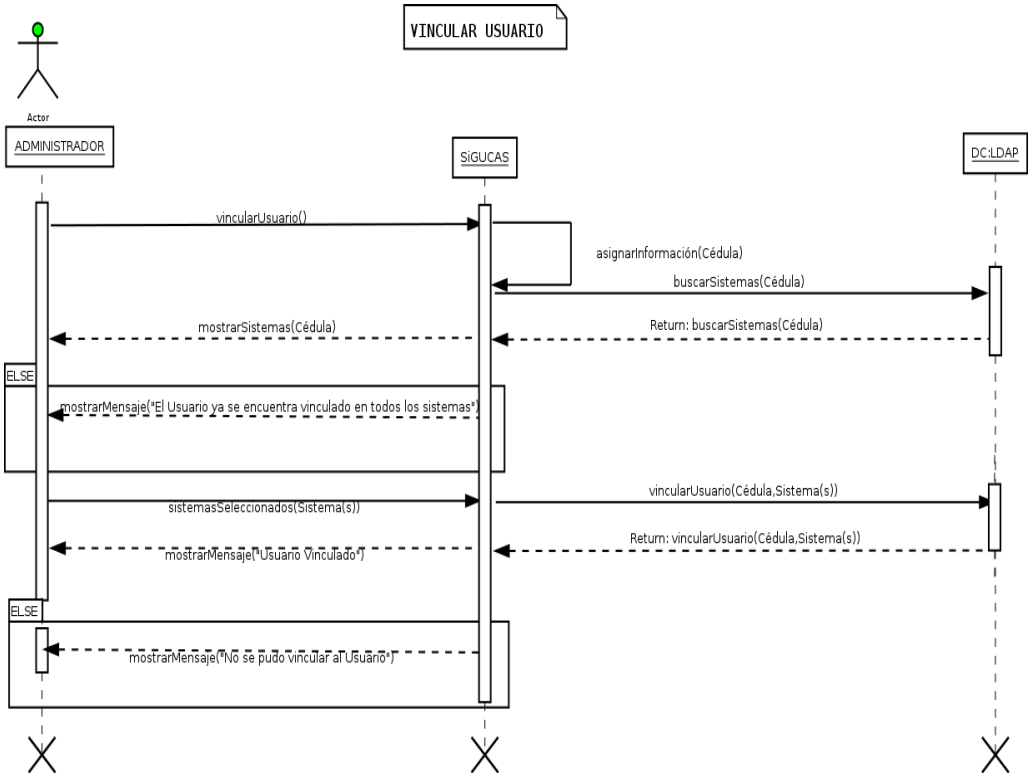


Figura A13. 14. Diagrama de secuencia para vincular a un usuario.

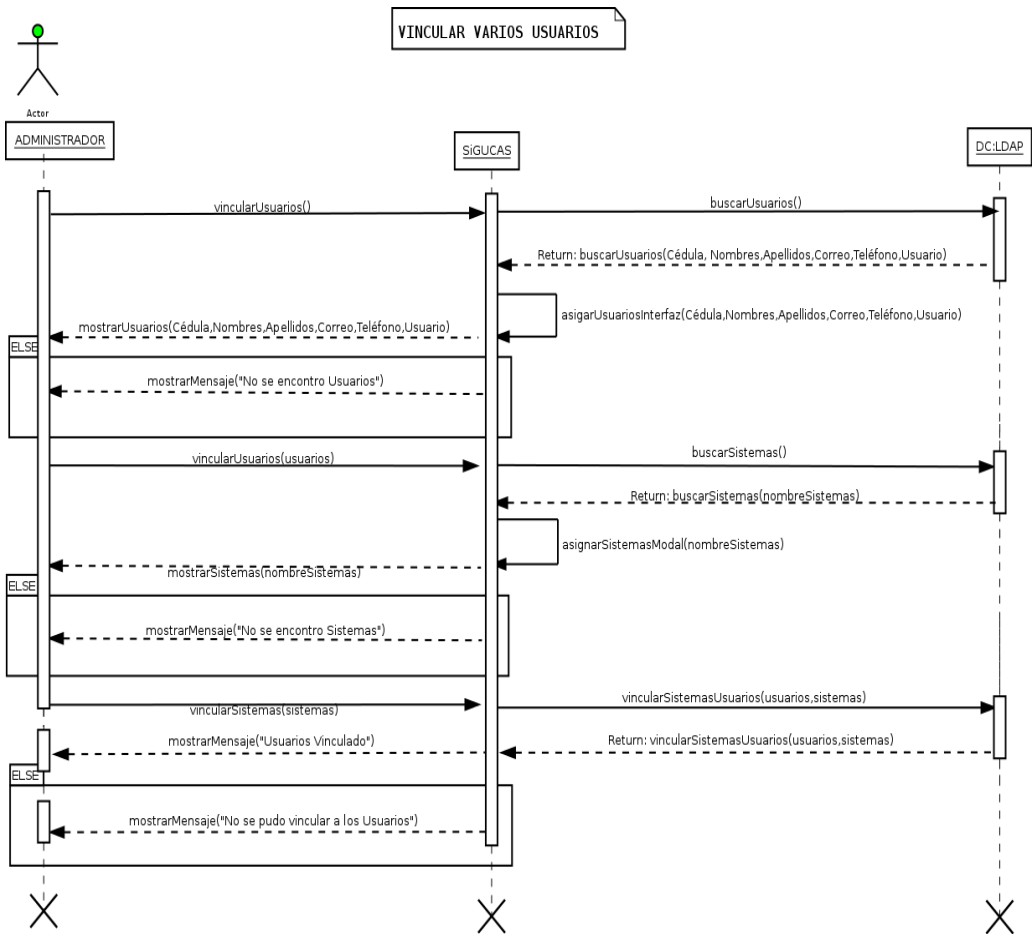


Figura A15. 15. Diagrama de secuencia para vincular varios usuarios.

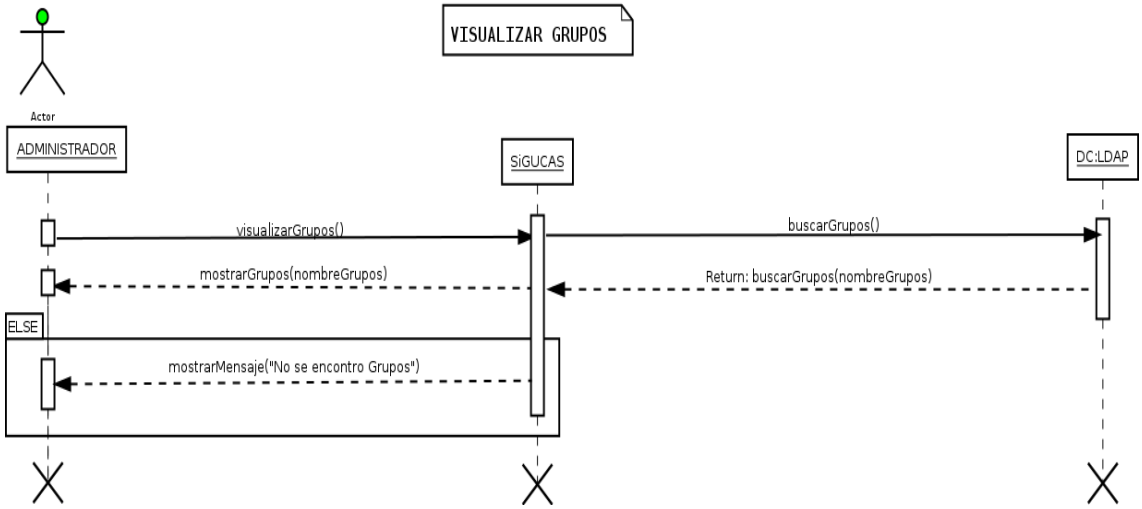


Figura A13. 16. Diagrama de secuencia para visualizar los grupos.

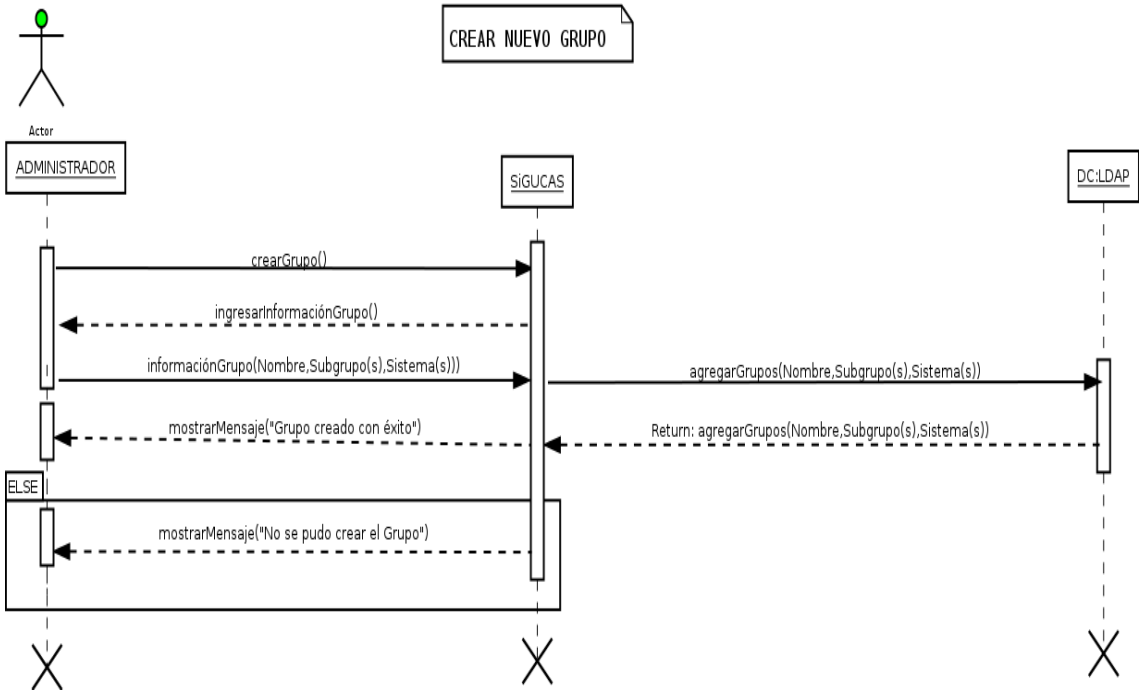


Figura A13. 17. Diagrama de secuencia para crear nuevos grupos.

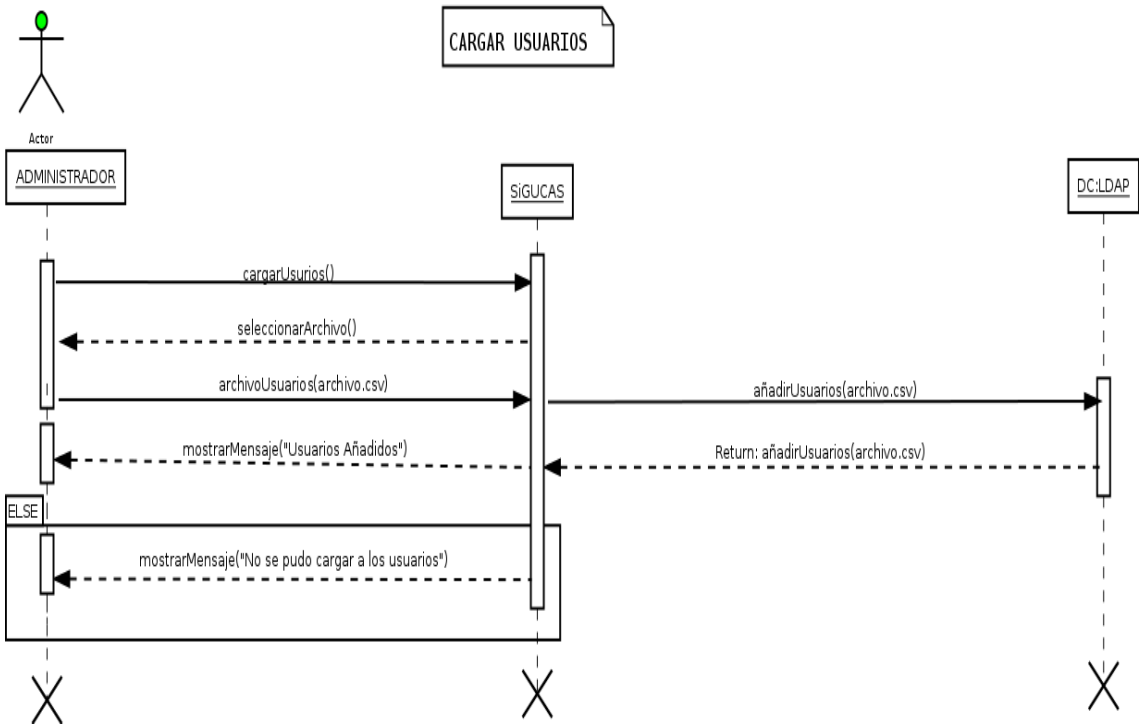


Figura A13. 18. Diagrama de secuencia para cargar un conjunto de usuarios.

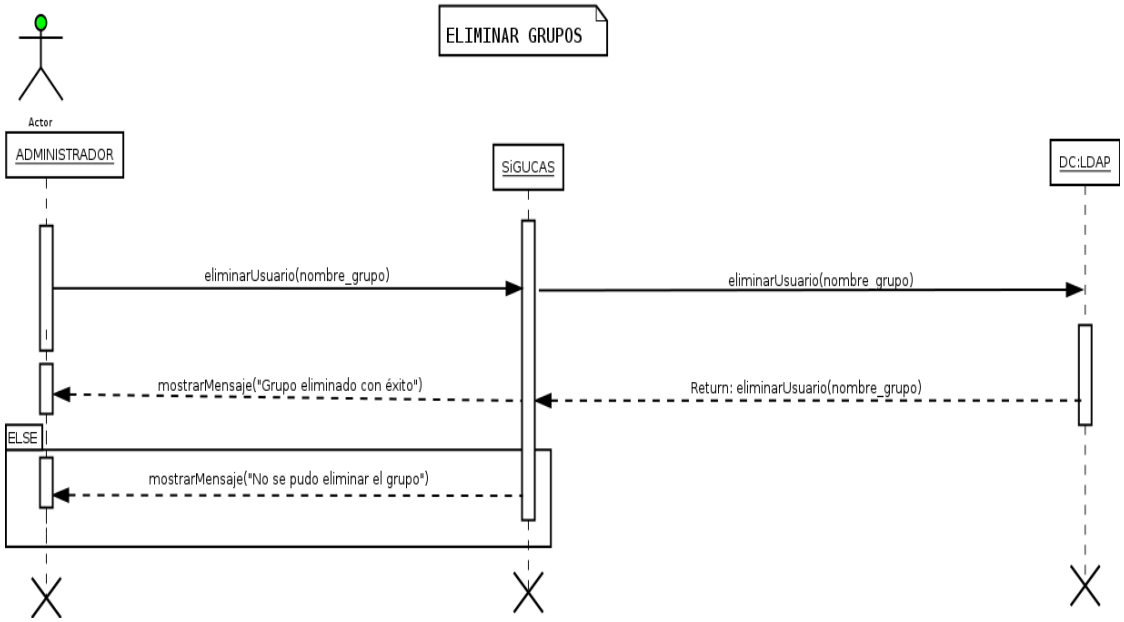


Figura A13. 19. Diagrama de secuencia para eliminar grupos.

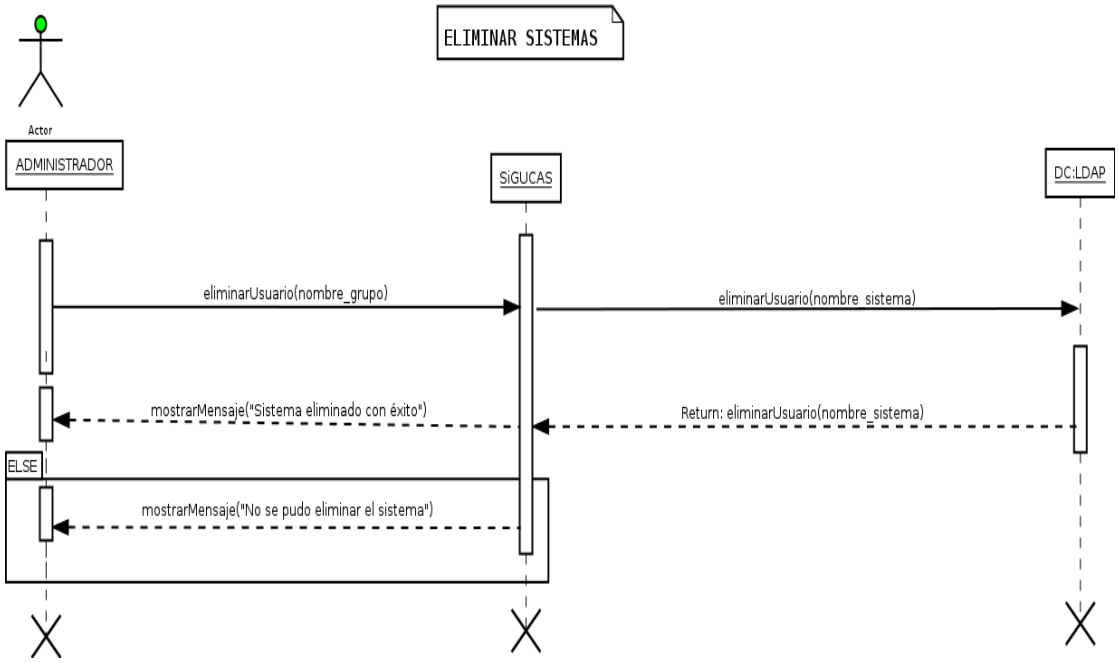


Figura A13. 20. Diagrama de secuencia para eliminar sistemas.

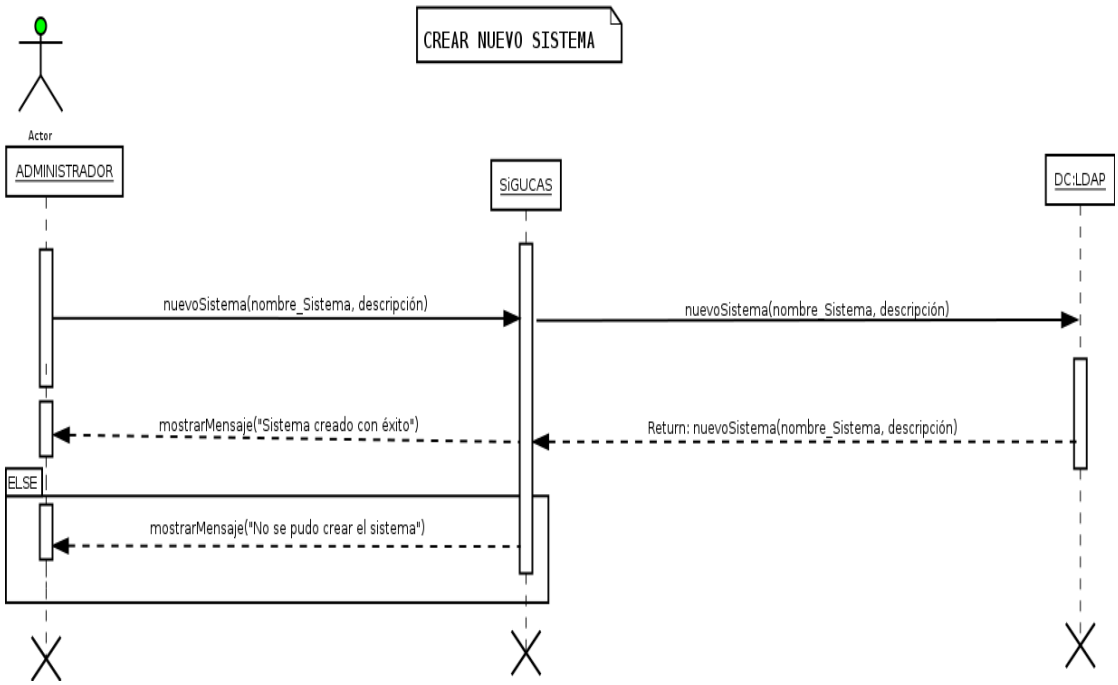


Figura A13. 21. Diagrama de secuencia para la crear nuevos sistemas.

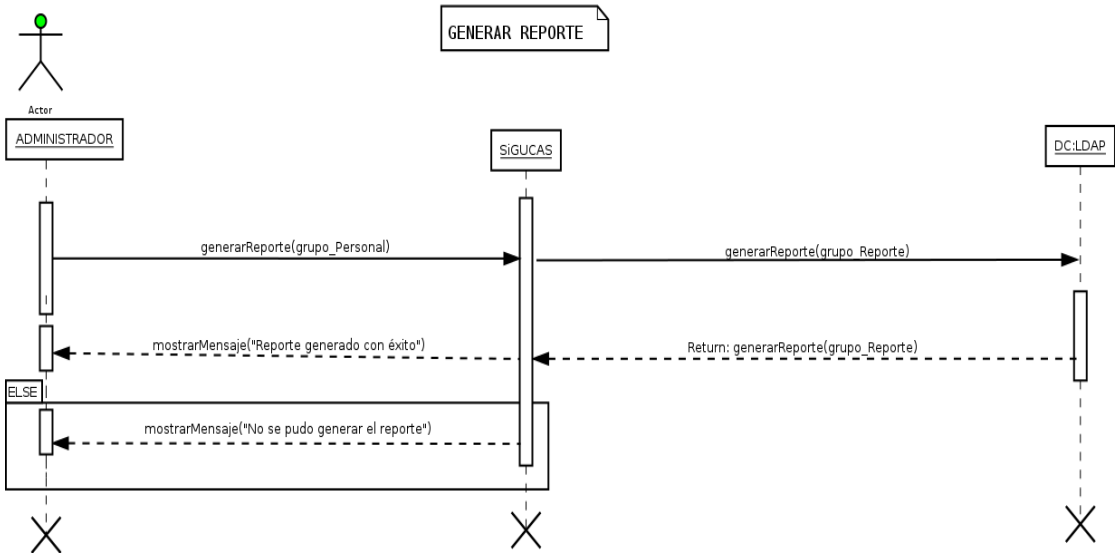


Figura A13. 22. Diagrama de secuencia para generar reportes

2.2. Diagrama de Clases

En la (Figura A15. 19), se muestra el diagrama de clases, el cual es una recopilación de información relativa en cuanto a los procesos que realiza el sistema SAC para administrar el servidor OpenLdap.

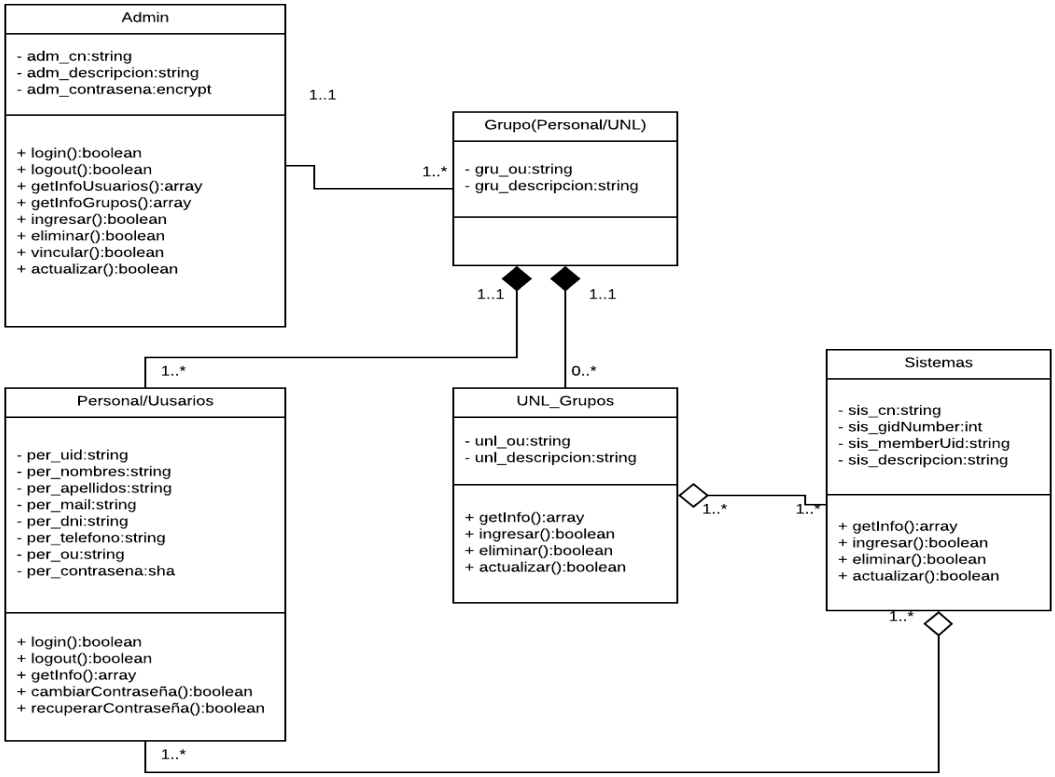


Figura A13. 23. Diagrama de clases del sistema SAC.

2.3. Diagrama interpretativo.

La Figura A13. 20, ilustra las relaciones entre los principales componentes de sistema de administración:

- Interfaz principal
- Login centralizado
- Controlador
- Web service
- Y servidor openLDAP.

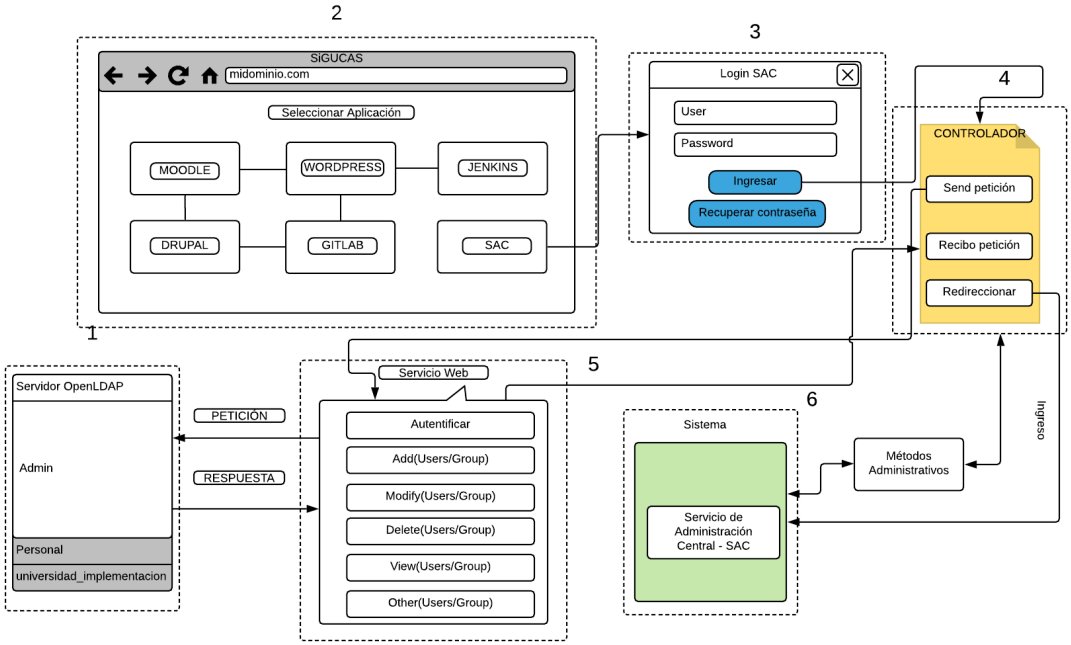


Figura A13. 24. Diagrama de componentes del sistema SAC.

2.4. DEFINICIÓN DE HERRAMIENTAS Y TECNOLOGÍAS

2.4.1. Herramientas de Desarrollo

Para el desarrollo del sistema SAC con métodos administrativos y de autenticación para CAS, se utilizó herramientas de software libre, programación orientada a objetos (POO), que permite la reutilización de código y el esquema modelo – vista – controlador (MVC), la cual es ideal para la programación Web y deja fácilmente cambiar cada parte sin afectar unas a otras.

Arquitectura MVC

El Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Modelo

Es la representación de la información que maneja el sistema (procesa y recibe datos del servidor) con los cuales el sistema realiza sus procesos o funcionalidad. Para el desarrollo de los métodos de conexión, administración y autenticación con el servidor OpenLdap se utilizó el siguiente lenguaje de programación:

- **PHP**, porque es un lenguaje interpretado del lado del servidor, de código libre (open source) adecuado para el desarrollo Web y genera código HTML dinámico. Se caracteriza por su velocidad, estabilidad, seguridad y simplicidad. Además, cuenta con herramientas de gestión para la administración de servidores OpenLDAP, previamente analizado en la RSL del Protocolo LDAP como mecanismo centralizado.
- **JSON Web Token**, estándar de la industria RFC 7519 para representar reclamos de forma segura entre dos partes.

Vista

Es la responsable de representar la interfaz gráfica, que se compone de la información que se envía al cliente, mediante los mecanismos que están interactuando. Esta puede contener expresiones PHP, pero se recomienda que estas expresiones no afecten los datos del modelo. Aquí se desarrolló la parte visual para el usuario, haciendo uso de las siguientes herramientas:

- **HTML**, ya que es un lenguaje de marcado el cual se escribe en forma de etiquetas para definir la estructura de una página Web, mediante un marcado de presentación que se encarga de señalar como se verá el texto más allá de su función. Se puede usar en conjunto con diversos lenguajes de programación para la crear páginas Web dinámicas, en nuestro caso con el lenguaje de programación PHP.
- **Framework BootStrap**, debido a que facilita la maquetación de sitios Web y adapta la interfaz del sitio Web al tamaño del dispositivo en que se visualice. Permite la integración de JavaScript, CSS y Html para obtener elementos Web

muy útiles, y la integración con JQuery para ofrecer resultados dinámicos, simplificando el trabajo y sin sacrificar resultados.

- **Alertify**, es un script escrito con Jquery, el cual nos permite utilizar los siguientes elementos Javascript personalizados: alert(), confirm() y prompt(). Además, también nos permite utilizar sus notificaciones, las cuales son muy agradables y sencillas de utilizar y modificar.

Controlador

Es la lógica del programa, la que actúa como intermediario entre el modelo y la vista, gestionando la información entre ellos y adaptando los datos a la necesidad de cada uno. Aquí se desarrolló el proceso para obtener información del modelo y presentarlo en la vista, haciendo uso de las siguientes herramientas:

- **JavaScript**, nos proporciona una mejora en la gestión de la interfaz y nos brinda dinamismo a las páginas Web, quiere decir que trabaja del lado del cliente por lo que sus efectos son muy rápidos y dinámicos. Incluye la librería JQuery que es una de las más usadas.
- **JQuery**, porque permite simplificar la manera de interactuar con HTML, a través de una librería de JavaScript, para trabajar con sitios Web dinámicos, validación de campos y funcionalidades más complejas, haciéndolo de una manera sencilla. Debido a que son un conjunto de utilidades ya programadas y probadas.
- **DataTable**, Es una herramienta altamente flexible, construida sobre los cimientos de la mejora progresiva, que agrega características avanzadas a cualquier tabla HTML.

Tecnologías

Las tecnologías que se emplearon en el desarrollo del sistema de administración de describen en la TABLA A13. CIX.

TABLA A13. CIX.
TECNOLOGÍAS UTILIZADAS EN EL DESARROLLO DEL SISTEMA SAC.

Nombre	Descripción	Tipo
SublimeText	Editor de texto y editor de código fuente.	Editor de texto

Ubuntu 18.04 LTS	Sistema operativo de código abierto GNU/LINUX.	Sistema Operativo
Servidor Web Apache	Servidor Web gratuito y de código abierto	Servidor Web
OpenLdap	Implementación libre y de código abierto del protocolo LDAP.	Directorio Activo
Mozila Firefox	Navegador Web libre y de código abierto.	Navegador Web
Chrome Google	Navegador Web de software privativo desarrollado por google.	Navegador Web
Safari	Navegador Web de software privativo desarrollado Apple.	Navegador Web

3. FASE 3: Codificación

3.1. REGLAS DE PROGRAMACIÓN PARA SAC

Para el presente sistema, se definieron unas reglas de programación basadas en la experiencia y conocimiento adquirido, las cuales permiten indicar la forma de escribir el código fuente del sistema, para que los demás pueden interpretar el código con facilidad y coherencia. Generando una mejor integración de los módulos del sistema y una correcta sincronización para la programación en parejas.

3.1.1. Nombre de las variables

- Los nombres que se usen deben ser significativos
- Los nombres deben empezar con minúscula, excepto la primera letra de la segunda palabra
- Se puede definir números al final de cada palabra cuando estas sean similares, la asignación será en orden ascendente, por ejemplo: $resultado; $resultado1; $resultado2.

3.1.2. Asignaciones

Debe existir un espacio entre las variables y los operadores, por ejemplo:

- $contador = 5;
- $nuevo = $contador;

3.1.3. Operadores

- “.” no llevan espacio entre los miembros de la izquierda y la derecha, por ejemplo:
  echo $prueba.$prueba1;
  $valor = $uno.“”.$dos;
- Se recomienda no exceder con el uso de la concatenación.
- “.=” necesita un espacio entre los miembros de la izquierda y la derecha ejemplo:
  $arreglo.= 'Array';

3.1.4. Declaraciones

- if, else if, while, for, do while: puede o no presentar un espacio entre la palabra clave if y el paréntesis
  if (<condición>)
  if(<condición>)
  while (<condición>)
  while(<condición>)
- Todas las condiciones se deben encontrar correctamente tabuladas.

3.1.5. Nombres de métodos o funciónes

Los nombres del método o función, deben ser explícitos, por lo tanto nombres de función o métodos como: “ar()” o “b()” están completamente prohibidos.

3.1.6. Devolución de valores

Las declaraciones de devolución no necesitan paréntesis, excepto cuando se trata de una expresión compuesta.

- return $respuesta;
- return ($uno + $dos);
- return false;
- return (ar() - b());

3.1.7. Cliente siempre presente

El rol de cliente lo asume el director del presente TT y el personal de la UTI, por tal motivo se llevó una comunicación constante, en la cual los desarrolladores debían de acudir a la UNL, para mantener una comunicación directa con el cliente, también se utilizó otros medios de comunicación como: redes sociales, correos electrónicos y llamadas telefónicas, para poder solucionar cualquier duda que se requiera en el proceso de implementación. De esta forma se logra una buena comunicación con el cliente.

3.1.8. Programación en parejas

La metodología XP, recalca que toda la producción de código debe ser hecha en parejas, donde se tiene un diseño de mejor calidad y un código más organizado y se soluciona los problemas más fácilmente. En el desarrollo de este proyecto se contó con un ambiente adecuado para cumplir con el objetivo de programación en parejas, así mismo fue de gran importancia la colaboración y compañerismo para realizar los distintos módulos y poder culminar el sistema propuesto.

En XP se tiene como salvedad para trabajar en parejas que uno o ambos de los programadores sean expertos en el uso de las herramientas para el desarrollo del sistema y también fue muy conveniente la estandarización del código desde el inicio. En el caso de ambos desarrolladores, tenían conocimientos elevados sobre el uso de las herramientas que se emplearon, por lo cual el nivel de autonomía fue superior y beneficioso para el cumplimiento del sistema.

3.1.9. Pruebas Unitarias

Las (TABLAS A13. CX – A15. CXXXII), muestran las pruebas unitarias de cada historia de usuario definidas anteriormente.

TABLA A13. CX.
PRUEBA UNITARIA DE LA SELECCIÓN DEL SISTEMA SAC.

Número de caso de prueba unitaria: 01
Caso de prueba: Selección del sistema SAC
Objetivo de la prueba: Seleccionar correctamente el sistema SAC para verificar que el usuario se redirigió al login del sistema.
Condiciones: - Tener levantado el servidor local de aplicaciones.

Datos de entrada: Evento de clic del mouse.
Salida esperada: Interfaz de login del sistema SAC.
Salida obtenida: Interfaz de login del sistema SAC.
Evaluación: Prueba Unitaria Exitosa.

TABLA A13. CXI.
PRUEBA UNITARIA SOBRE EL ACCESO AL MÓDULO DE ADMINISTRACIÓN.

Número de caso de prueba unitaria: 02
Caso de prueba: Acceso al módulo de Administración
Objetivo de la prueba: Verificar que el administrador pueda acceder al panel de administración del sistema SAC.
Condiciones: - Tener levantado el servidor OpenLDAP. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLdap, como administrador.
Datos de entrada: Usuario y contraseña.
Salida esperada: Interfaz gráfica del administrador.
Salida obtenida: Interfaz gráfica del administrador.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXII.
PRUEBA UNITARIA PARA EL ACCESO AL MÓDULO DEL USUARIO.

Número de caso de prueba unitaria: 03
Caso de prueba: Acceso al módulo del Usuario
Objetivo de la prueba: Verificar que el usuario acceda al panel de visualización de su información en el sistema SAC.
Condiciones: - Tener levantado el servidor OpenLDAP. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLdap, como usuario.
Datos de entrada: Usuario y contraseña.
Salida esperada: Interfaz gráfica del usuario.
Salida obtenida: Interfaz gráfica del usuario.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXIII.
PRUEBA UNITARIA PARA LA VISUALIZACIÓN DE INFORMACIÓN DE CADA USUARIO.

Número de caso de prueba unitaria: 04
Caso de prueba: Visualizar información de cada usuario
Objetivo de la prueba: Verificar que administrador pueda visualizar la información de los usuarios registrados en el directorio centralizado.
Condiciones: - Tener levantado el servidor OpenLdap. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLdap, como administrador.
Datos de entrada: Usuario y contraseña.
Salida esperada: Presentar la información personal del usuario como: Nombres, Apellidos, Teléfono y Correo.
Salida obtenida: Información relevante del usuario
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXIV.
PRUEBA UNITARIA PARA EL CAMBIO DE CONTRASEÑA.

Número de caso de prueba unitaria: 05
Caso de prueba: Cambio de contraseña.
Objetivo de la prueba: Realizar el cambio de contraseña de un usuario.
Condiciones: - Tener levantado el servidor OpenLdap. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLdap, como usuario.
Datos de entrada: Contraseña actual, Contraseñas nuevas.
Salida esperada: Notificación del cambio contraseña.
Salida obtenida: Notificación del cambio contraseña.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXV.
PRUEBA UNITARIA PARA EL CIERRE DE SESIÓN DEL SISTEMA.

Número de caso de prueba unitaria: 06
Caso de prueba: Cierre de sesión del sistema

Objetivo de la prueba: Verificar el cierre de sesión del sistema SAC por parte del administrador.
Condiciones: - Tener levantado el servidor OpenLdap. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLdap, como administrador o Usuario.
Datos de entrada: Evento de clic del mouse.
Salida esperada: Interfaz de login del sistema SAC.
Salida obtenida: Interfaz de login del sistema SAC.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXVI.
PRUEBA UNITARIA PARA AÑADIR UN NUEVO USUARIO.

Número de caso de prueba unitaria: 07
Caso de prueba: Añadir nuevo usuario.
Objetivo de la prueba: Verificar el correcto registro de un nuevo usuario en el servidor OpenLdap.
Condiciones: - Tener levantado el servidor OpenLDAP. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLDAP, como administrador.
Datos de entrada: Cédula, Nombres, Apellidos, Correo, Teléfono, Grupo.
Salida esperada: Notificación del registro de un nuevo usuario.
Salida obtenida: Notificación del registro de un nuevo usuario.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXVII.
PRUEBA UNITARIA PARA ACTUALIZAR LA INFORMACIÓN DE UN USUARIO.

Número de caso de prueba unitaria: 08
Caso de prueba: Actualizar información de un usuario.
Objetivo de la prueba: Verificar el correcto cambio de información de un usuario en el servidor OpenLDAP.
Condiciones: - Tener levantado el servidor OpenLDAP. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLdap, como administrador.

Datos de entrada: Cédula || Nombres || Apellidos || Correo || Teléfono || Grupo || Contraseña.
Salida esperada: Notificación de la actualización de la información.
Salida obtenida: Notificación de la actualización de la información.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXVIII.
PRUEBA UNITARIA PARA ELIMINAR A UN USUARIO.

Número de caso de prueba unitaria: 09
Caso de prueba: Eliminar usuario.
Objetivo de la prueba: Verificar la correcta eliminación de un usuario en el servidor OpenLdap.
Condiciones: - Tener levantado el servidor OpenLdap. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLdap, como administrador.
Datos de entrada: Nombre de Usuario (dni).
Salida esperada: Notificación de la eliminación del usuario.
Salida obtenida: Notificación de la eliminación del usuario.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXIX.
PRUEBA UNITARIA PARA LA BÚSQUEDA DE USUARIOS.

Número de caso de prueba unitaria: 10
Caso de prueba: Buscar usuario.
Objetivo de la prueba: Verificar la búsqueda de usuarios registrados en el servidor OpenLDAP..
Condiciones: - Tener levantado el servidor OpenLDAP. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLDAP, como administrador.
Datos de entrada: Nombre de Usuario.
Salida esperada: Tabla de usuarios que coincidan con el campo de búsqueda.
Salida obtenida: Tabla de usuarios que coincidan con el campo de búsqueda.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXX.
PRUEBA UNITARIA PARA LA CARGA DE UN CONJUNTO DE USUARIOS.

Número de caso de prueba unitaria: 11
Caso de prueba: Cargar Usuarios.
Objetivo de la prueba: Verificar el correcto registro de un conjunto de usuarios contenidos en archivo externo con la extensión .csv
Condiciones: - Tener un archivo .csv con un conjunto de usuarios. - Tener levantado el servidor OpenLdap. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLdap, como administrador.
Datos de entrada: Archivo .csv
Salida esperada: Notificación del total de usuarios registrados.
Salida obtenida: Notificación del total de usuarios registrados.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXXI.
PRUEBA UNITARIA PARA VINCULAR A UN USUARIO.

Número de caso de prueba unitaria: 12
Caso de prueba: Vincular usuario.
Objetivo de la prueba: Verificar la vinculación de un usuario a uno o varios sistemas registrados en el servidor OpenLDAP.
Condiciones: - Tener levantado el servidor OpenLDAP. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLDAP, como administrador. - Tener como mínimo un sistema registrado en el servidor OpenLDAP.
Datos de entrada: Nombre de Usuario, Nombre de los Sistema(s).
Salida esperada: Notificación de la vinculación de un usuario.
Salida obtenida: Notificación de la vinculación de un usuario.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXXII.
PRUEBA UNITARIA PARA VINCULAR VARIOS USUARIOS.

Número de caso de prueba unitaria: 13
Caso de prueba: Vincular varios usuarios.

Objetivo de la prueba: Verificar la vinculación de varios usuarios a uno o varios sistemas registrados en el servidor OpenLDAP.
Condiciones: - Tener levantado el servidor OpenLDAP. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLDAP, como administrador. - Tener registrado como mínimo un sistema en el servidor OpenLdap.
Datos de entrada: Nombre de los usuarios, nombre de los Sistema(s).
Salida esperada: Notificación de la vinculación de varios usuarios.
Salida obtenida: Notificación de la vinculación de varios usuarios.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXXIII.
PRUEBA UNITARIA PARA VISUALIZAR LA INFORMACIÓN DE GRUPOS Y SISTEMAS.

Número de caso de prueba unitaria: 14
Caso de prueba: Visualizar información de grupos y sistemas.
Objetivo de la prueba: Verificar la correcta visualización de todos los grupos y sistemas registrados en el servidor OpenLDAP.
Condiciones: - Tener registrado como mínimo un grupo en el servidor OpenLDAP. - Tener levantado el servidor OpenLDAP. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLDAP, como administrador.
Datos de entrada: Evento de clic del mouse.
Salida esperada: Lista con los grupos y sistemas del servidor OpenLDAP.
Salida obtenida: Lista con los grupos y sistemas del servidor OpenLDAP.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXXIV.
PRUEBA UNITARIA PARA CREAR NUEVOS GRUPOS.

Número de caso de prueba unitaria: 15
Caso de prueba: Crear nuevos grupos.
Objetivo de la prueba: Verificar el correcto registro de un nuevo grupo(s) en el servidor OpenLdap.
Condiciones: - Tener levantado el servidor OpenLdap. - Tener levantado el servidor local de aplicaciones.

- Estar registrado en el servidor OpenLdap, como administrador.
Datos de entrada: Nombre del grupo, nombres de los subgrupos (opcional), nombre del sistema y descripción.
Salida esperada: Notificación del registro de un nuevo grupo en el servidor.
Salida obtenida: Notificación del registro de un nuevo grupo en el servidor.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXXV.
PRUEBA UNITARIA PARA ELIMINAR GRUPOS.

Número de caso de prueba unitaria: 16
Caso de prueba: Eliminar Grupos.
Objetivo de la prueba: Verificar la correcta eliminación de un grupo en el servidor OpenLdap.
Condiciones: - Tener levantado el servidor OpenLDAP. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor Open, como administrador. - Tener registrado como mínimo un grupo en servidor OpenLDAP.
Datos de entrada: Nombre del grupo.
Salida esperada: Notificación de la eliminación de un grupo.
Salida obtenida: Notificación de la eliminación de un grupo.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXXVI.
PRUEBA UNITARIA PARA RESTABLECER LA CONTRASEÑA.

Número de caso de prueba unitaria: 17
Caso de prueba: Restablecer contraseña.
Objetivo de la prueba: Restablecer la contraseña de un usuario.
Condiciones: - Tener levantado el servidor OpenLDAP. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLDAP, como administrador. - Tener una dirección de correo valida.
Datos de entrada: Cédula y Usuario.
Salida esperada: Notificación de las instrucciones para reestablecer la contraseña.
Salida obtenida: Notificación de las instrucciones para reestablecer la contraseña.

Evaluación: Prueba unitaria exitosa.

TABLA A13. CXXVII.
PRUEBA UNITARIA PARA LA CREACIÓN DE NUEVOS SISTEMAS.

Número de caso de prueba unitaria: 18
Caso de prueba: Crear nuevos sistemas.
Objetivo de la prueba: Verificar el correcto registro de un nuevo sistema en el servidor OpenLDAP.
Condiciones: - Tener levantado el servidor OpenLDAP. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLdap, como administrador.
Datos de entrada: Nombre del sistema, subsistemas y descripción.
Salida esperada: Notificación del registro de un nuevo sistema.
Salida obtenida: Notificación del registro de un nuevo sistema.
Evaluación: Prueba unitaria exitosa.

TABLA A15. CXXVIII.
PRUEBA UNITARIA PARA LA ELIMINACIÓN DE SISTEMAS.

Número de caso de prueba unitaria unitaria: 19
Caso de prueba: Eliminar sistemas.
Objetivo de la prueba: Verificar la correcta eliminación de un sistema(s) en el servidor OpenLdap.
Condiciones: - Tener levantado el servidor OpenLDAP. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLDAP, como administrador. - Tener como mínimo un sistema registrado en el servidor OpenLDAP.
Datos de entrada: Nombre del sistema, subsistemas y descripción.
Salida esperada: Notificación del registro de un nuevo sistema.
Salida obtenida: Notificación del registro de un nuevo sistema.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXXIX.
PRUEBA UNITARIA PARA LA GENERACIÓN DE REPORTES.

Número de caso de prueba unitaria: 20
Caso de prueba: Generar reportes

Objetivo de la prueba: Verificar la correcta creación e exportación de un archivo .csv
Condiciones: - Tener levantado el servidor OpenLDAP. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLDAP, como administrador.
Datos de entrada: Archivo .csv
Salida esperada: Notificación del reporte creado con éxito.
Salida obtenida: Notificación del reporte creado con éxito.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXXX.
PRUEBA UNITARIA PARA ELIMINAR TODOS LOS REGISTROS DE UN GRUPO DEL SERVIDOR OPENLDAP.

Número de caso de prueba unitaria: 21
Caso de prueba: Eliminar todos los registros.
Objetivo de la prueba: Verificar la correcta eliminación de todos los usuarios de un grupo del servidor OpenLdap.
Condiciones: - Tener levantado el servidor OpenLdap. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLdap, como administrador.
Datos de entrada: Nombre de un grupo.
Salida esperada: Notificación de la eliminación de todos los usuarios con éxito.
Salida obtenida: Notificación de la eliminación de todos los usuarios con éxito.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXXXI.
PRUEBA UNITARIA PARA CREAR NUEVOS ADMINISTRADORES.

Número de caso de prueba unitaria: 22
Caso de prueba: Crear nuevos administradores.
Objetivo de la prueba: Verificar la correcta creación de nuevos administradores de lectura del servidor OpenLDAP.
Condiciones: - Tener levantado el servidor OpenLDAP. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLDAP, como administrador.

Datos de entrada: Usuario, Contraseña y descripción.
Salida esperada: Notificación de la creación de un nuevo usuario administrador de lectura con éxito.
Salida obtenida: Notificación de la creación de un nuevo usuario administrador de lectura con éxito.
Evaluación: Prueba unitaria exitosa.

TABLA A13. CXXXII.
PRUEBA UNITARIA PARA VISUALIZAR USUARIOS VINCULADOS.

Número de caso de prueba unitaria: 23
Caso de prueba: Visualizar usuarios vinculados.
Objetivo de la prueba: Verificar la correcta visualización de los usuarios vinculados a un sistema del servidor OpenLDAP.
Condiciones: - Tener levantado el servidor OpenLDAP. - Tener levantado el servidor local de aplicaciones. - Estar registrado en el servidor OpenLDAP, como administrador. - Tener como mínimo un sistema registrado en el servidor OpenLDAP. - Tener como mínimo un usuario vinculado en un sistema del servidor OpenLDAP.
Datos de entrada: Nombre de un sistema.
Salida esperada: Interfaz con la descripción del sistema y descripción de los usuarios vinculados.
Salida obtenida: Interfaz con la descripción del sistema y descripción de los usuarios vinculados.
Evaluación: Prueba unitaria exitosa.

3.2. INTEGRACIÓN CONTINÚA.

En este punto se hace mención que se deben hacer integraciones cada poca hora o en lo posible no tardar más de un día entre una y otra integración, entre más se tarda en encontrar un problema, resultará más costoso resolverlo, e integrar frecuentemente evita problemas como el trabajar sobre una clase o método obsoleto. En cuanto a esta parte no existió problema en realizarlo, debido a que se lleva un control de versiones utilizando el software Jenkins, el cual nos permite integrar nuestro trabajo frecuentemente, para detectar errores lo más pronto posible al realizar las pruebas unitarias para cada una de ellas y así poder aprobarlas.

3.3. NO TRABAJAR HORAS EXTRAS.

Plantearse trabajar horas extras después de una jornada completa de desarrollo sugiere más una pérdida de tiempo que una recuperación de los atrasos del proyecto. En el caso de este proyecto no se trabajaron horas extras debido a que se tuvo la precaución de plantear plazos convenientes en las entregas con el fin de considerar cualquier posible problema que surgiera en la implementación e integración, debido a la programación en parejas que se optó como unos de los objetivos principales en nuestra metodología.

4. FASE 4: Pruebas

4.1. IMPLEMENTACIÓN

Una vez culminado el desarrollo del sistema Web se lo implementó, en un servidor local para pruebas y para su validación final, verificando su funcionamiento y realizando las respectivas explicaciones a nuestro director del tema de titulación Ing. Edison Coronel, y personal de la UTI quienes serán los encargados del manejo de el mismo. Llevando a cabo con el cumplimento de esta fase del proyecto.

4.2. PRUEBAS DE ACEPTACIÓN

Las pruebas de aceptación, son diseñadas en base a las historias de usuario, es una prueba formal conducida para determinar si un sistema satisface los criterios de aceptación y permite al cliente determinar si acepta el sistema. Para lo cual se elaboró las siguientes tablas de aceptación (TABLAS A15. CXXXIII– A15. CXXXIII).

TABLA A13. CXXXIV.
PRUEBA DE ACEPTACIÓN SOBRE LA SELECCIÓN DEL SISTEMA SAC CORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 1	**Número de historia de usuario:** 1
Caso de prueba: Selección del sistema SAC	
Nombre de caso de prueba: Selección del sistema SAC correctamente.	
Descripción: Comprobará que el administrador o usuario seleccione correctamente el sistema SAC.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local.	

Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC
Resultado esperado: Se dirige a la interfaz de login del sistema SAC
Evaluación: La evaluación fue aprobada, la selección del sistema SAC fue correcto.

TABLA A13. CXXXV.
PRUEBA DE ACEPTACIÓN SOBRE LA SELECCIÓN DEL SISTEMA SAC INCORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 2	**Número de historia de usuario:** 1
Caso de prueba: Selección del sistema SAC	
Nombre de caso de prueba: Selección del sistema SAC incorrectamente.	
Descripción: Comprobará que si el administrador o usuario no selecciona correctamente el sistema SAC, no podrán acceder al login del mismo.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar incorrectamente el sistema SAC.	
Resultado esperado: No se da acceso a la interfaz de login.	
Evaluación: La evaluación fue aprobada, el redireccionamiento al login del sistema no se realizó.	

TABLA A15. CXXXVI.
PRUEBA DE ACEPTACIÓN SOBRE EL ACCESO AL MÓDULO DE ADMINISTRACIÓN CORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 3	**Número de historia de usuario:** 2
Caso de prueba: Acceso al módulo de administración	
Nombre de caso de prueba: Acceso correcto al módulo de administración.	
Descripción: Comprobará que el administrador pueda acceder al módulo de administración correctamente.	
Condiciones de ejecución: El sistema y OpenLDAP deberán estar levantados en un servidor local.	

Estar registrado en el directorio centralizado LDAP.
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña).
Resultado esperado: Interfaz del módulo de administración.
Evaluación: La evaluación fue aprobada, el acceso al módulo de administración se ejecutó correctamente.

TABLA A13. CXXXVII.
PRUEBA DE ACEPTACIÓN SOBRE EL ACCESO AL MÓDULO DE ADMINISTRACIÓN INCORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 4	**Número de historia de usuario:** 2
Caso de prueba: Acceso al módulo de administración	
Nombre de caso de prueba: Acceso incorrecto al módulo de administración.	
Descripción: Comprobar que si el administrador no ingresa correctamente sus credenciales no podrá acceder al módulo de administración.	
Condiciones de ejecución: - El sistema y OpenLDAP deberán estar levantados en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar incorrectamente las credenciales (Usuario y contraseña).	
Resultado esperado: Notificación de datos mal ingresados.	
Evaluación: La evaluación fue aprobada, el acceso al módulo de administración fue incorrecto.	

TABLA A13. CXXXVIII.
PRUEBA DE ACEPTACIÓN SOBRE EL ACCESO AL MÓDULO DEL USUARIO CORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 5	**Número de historia de usuario:** 3
Caso de prueba: Acceso al módulo del usuario.	
Nombre de caso de prueba: Acceso correcto al módulo del usuario.	

Descripción: Comprobar que el usuario pueda acceder al módulo de usuario correctamente.
Condiciones de ejecución: - El sistema y OpenLDAP deberán estar levantados en un servidor local. - Estar registrado en el directorio centralizado LDAP.
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña).
Resultado esperado: Interfaz del módulo de usuario.
Evaluación: La evaluación fue aprobada, el acceso al módulo del usuario se ejecutó correctamente.

TABLA A15. CXXXIX.
PRUEBA DE ACEPTACIÓN SOBRE EL ACCESO AL MÓDULO DEL USUARIO INCORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 6	**Número de historia de usuario:** 3
Caso de prueba: Acceso al módulo del usuario.	
Nombre de caso de prueba: Acceso incorrecto al módulo del usuario.	
Descripción: Comprobar que si el usuario no ingresa correctamente sus credenciales no podrá tener acceso al módulo de información.	
Condiciones de ejecución: - El sistema y OpenLDAP deberán estar levantados en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: a) Ingresar a la interfaz de acceso principal. b) Seleccionar el sistema SAC. c) Ingresar incorrectamente las credenciales (Usuario y contraseña).	
Resultado esperado: Notificación de datos mal ingresados.	
Evaluación: La evaluación fue aprobada, el acceso al módulo del usuario fue incorrecto.	

TABLA A13. CXL.
PRUEBA DE ACEPTACIÓN PARA VISUALIZAR LA INFORMACIÓN DE CADA USUARIO CORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 7	**Número de historia de usuario:** 4
Caso de prueba: Visualizar la información de cada usuario.	
Nombre de caso de prueba: Visualizar la información del usuario correctamente.	
Descripción: Comprobar que el administrador pueda visualizar la información de los usuarios correctamente.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Seleccionar una sección de los grupos de usuarios (Estudiantes, Docentes, Servidores, Trabajadores, Otros).	
Resultado esperado: Tabla con la información de los usuarios registrados en un grupo.	
Evaluación: La evaluación fue aprobada, la visualización de la información de los usuarios se ejecutó correctamente.	

TABLA A13. CXLI.
PRUEBA DE ACEPTACIÓN PARA VISUALIZAR LA INFORMACIÓN DE CADA USUARIO INCORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 8	**Número de historia de usuario:** 4
Caso de prueba: Visualizar la información de cada usuario.	
Nombre de caso de prueba: No visualizar la información de los usuarios.	
Descripción: Comprobar que si el administrador no selecciona ninguna sección de los grupos de usuarios no podrá visualizar su información.	
Condiciones de ejecución:	

- El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Seleccionar una sección que no sea (Estudiantes, Docentes, Servidores, Trabajadores, Otros).
Resultado esperado: Ningún registro de usuarios.
Evaluación: La evaluación fue aprobada, la visualización de los usuarios no fue posible.

TABLA A13. CXLII.
PRUEBA DE ACEPTACIÓN PARA EL CAMBIO DE CONTRASEÑA CORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 9	**Número de historia de usuario:** 5
Caso de prueba: Cambio de contraseña.	
Nombre de caso de prueba: Cambio de contraseña correctamente.	
Descripción: Comprobar que el usuario pueda cambiar su contraseña correctamente.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Ingresar la contraseña actual correcta, e ingresar las nuevas contraseñas iguales. 5. Dar clic en Actualizar contraseña.	
Resultado esperado: Notificación con la confirmación del cambio de contraseña con éxito.	
Evaluación: La evaluación fue aprobada, el cambio de contraseña se ejecutó correctamente.	

TABLA A13. CXLIII.
PRUEBA DE ACEPTACIÓN PARA EL CAMBIO DE CONTRASEÑA INCORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 10	**Número de historia de usuario:** 5
Caso de prueba: Cambio de contraseña.	
Nombre de caso de prueba: Cambio de contraseña incorrecto.	
Descripción: Comprobar que si el usuario no ingresa correctamente las contraseñas no podrá cambiar su contraseña.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Ingresar la contraseña actual correcta, e ingresar las nuevas contraseñas desiguales. 5. Dar clic en Actualizar contraseña.	
Resultado esperado: Notificación indicando que las contraseñas ingresadas no coinciden.	
Evaluación: La evaluación fue aprobada, el cambio de contraseña no se realizó con éxito.	

TABLA A13. CXLIV.
PRUEBA DE ACEPTACIÓN PARA EL CIERRE DE SESIÓN CORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 11	**Número de historia de usuario:** 6
Caso de prueba: Cerrar sesión	
Nombre de caso de prueba: Cerrar sesión del sistema correctamente.	
Descripción: Comprobar que el administrador o usuario puedan cerrar sesión del sistema correctamente.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	

Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Seleccionar la sección Cerrar Sesión.
Resultado esperado: Se dirige a la interfaz de login del sistema SAC.
Evaluación: La evaluación fue aprobada, el redireccionamiento al login del sistema se realizó con éxito.

TABLA A13. CXLV.
PRUEBA DE ACEPTACIÓN PARA EL CIERRE DE SESIÓN INCORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 12	**Número de historia de usuario:** 6
Caso de prueba: Cerrar sesión	
Nombre de caso de prueba: Cerrar sesión del sistema incorrectamente.	
Descripción: Comprobar que si el administrador o usuario no seleccionan bien la sección no se cerrara la sesión del sistema.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Seleccionar una sección que no sea Cerrar Sesión.	
Resultado esperado: No se cerrará sesión del sistema.	
Evaluación: La evaluación fue aprobada, el sistema no destruyo la sesión.	

TABLA A13. CXLVI.
PRUEBA DE ACEPTACIÓN PARA AÑADIR UN NUEVO USUARIO CORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 13	**Número de historia de usuario:** 7
Caso de prueba: Añadir nuevo usuario	
Nombre de caso de prueba: Añadir un usuario correctamente.	
Descripción: Comprobar que el administrador pueda ingresar a un nuevo usuario al directorio centralizado LDAP.	

Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en el botón Añadir Nuevo 5. Ingresar toda la información obligatoria de un usuario 6. Hacer click en guardar
Resultado esperado: Notificación con la confirmación de la creación de un nuevo usuario.
Evaluación: La evaluación fue aprobada, el nuevo usuario se creó con éxito.

TABLA A13. CXLVII.
PRUEBA DE ACEPTACIÓN PARA AÑADIR UN NUEVO USUARIO INCORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 14	**Número de historia de usuario:** 7
Caso de prueba: Añadir nuevo usuario	
Nombre de caso de prueba: Añadir un usuario incorrectamente.	
Descripción: Comprobar que el administrador no pueda ingresar a un nuevo usuario al directorio centralizado LDAP.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en el botón Añadir Nuevo 5. No ingresar toda la información obligatoria de un usuario 6. Hacer click en guardar	
Resultado esperado: Notificación indicando que faltan ingresar datos.	
Evaluación: La evaluación fue aprobada, el nuevo usuario no se creó con éxito.	

TABLA A13. CXLVIII.
PRUEBA DE ACEPTACIÓN PARA ACTUALIZAR LA INFORMACIÓN DE USUARIOS CORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 15	**Número de historia de usuario:** 8
Caso de prueba: Actualizar información de usuarios	
Nombre de caso de prueba: Actualizar la información de un usuario correctamente.	
Descripción: Comprobar que el administrador pueda actualizar la información de un usuario en el directorio centralizado LDAP.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en el botón Editar 5. Modificar cualquier campo de los datos del usuario 6. Hacer click en Actualizar	
Resultado esperado: Notificación con la confirmación de la actualización de datos de un usuario.	
Evaluación: La evaluación fue aprobada, los datos se actualizaron con éxito.	

TABLA A13. CXLIX.
PRUEBA DE ACEPTACIÓN PARA ACTUALIZAR LA INFORMACIÓN DE USUARIOS INCORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 16	**Número de historia de usuario:** 8
Caso de prueba: Actualizar información de usuarios	
Nombre de caso de prueba: Actualizar la información de un usuario incorrectamente.	
Descripción: Comprobar que el administrador no pueda actualizar la información de un usuario en el directorio centralizado LDAP.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña).	

4. Hacer clic en el botón Editar 5. No modificar ningún campo de los datos del usuario 6. Hacer click en Actualizar
Resultado esperado: Notificación con la confirmación de la actualización de datos de un usuario.
Evaluación: La evaluación fue aprobada, los datos no se actualizaron con éxito.

TABLA A13. CL.
PRUEBA DE ACEPTACIÓN PARA LA ELIMINACIÓN DE UN USUARIO CORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 17	**Número de historia de usuario:** 9
Caso de prueba: Eliminar usuarios	
Nombre de caso de prueba: Eliminar la información de un usuario correctamente.	
Descripción: Comprobar que el administrador pueda eliminar la información de un usuario en el directorio centralizado LDAP.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer click en el botón Eliminar	
Resultado esperado: Notificación con la confirmación de la eliminación de los datos de un usuario.	
Evaluación: La evaluación fue aprobada, los datos se eliminaron con éxito.	

TABLA A13. CLI.
PRUEBA DE ACEPTACIÓN PARA LA ELIMINACIÓN DE UN USUARIO INCORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 18	**Número de historia de usuario:** 9
Caso de prueba: Eliminar usuarios	
Nombre de caso de prueba: Eliminar la información de un usuario incorrectamente.	
Descripción: Comprobar que el administrador no pueda eliminar la información de un usuario en el directorio centralizado LDAP.	

Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. No hacer clic en el botón Eliminar
Resultado esperado: Ninguna notificación ya que el administrador no a optado por eliminar a un usuario.
Evaluación: La evaluación fue aprobada, los datos no se eliminaron.

TABLA A13. CLII.
PRUEBA DE ACEPTACIÓN PARA BUSCAR USUARIOS CORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 19	**Número de historia de usuario:** 10
Caso de prueba: Buscar Usuario	
Nombre de caso de prueba: Buscar la información de un usuario que exista en el directorio centralizado.	
Descripción: Comprobar que el administrador pueda buscar la información de un usuario en el directorio centralizado LDAP.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Ingresar el nombre de un usuario en el campo Buscar	
Resultado esperado: Tabla con la información del usuario ingresado anteriormente.	
Evaluación: La evaluación fue aprobada, los datos del usuario se presentaron con éxito.	

TABLA A13. CLIII.
PRUEBA DE ACEPTACIÓN PARA BUSCAR USUARIOS INCORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 20	**Número de historia de usuario:** 10
Caso de prueba: Buscar Usuario	

Nombre de caso de prueba: Buscar la información de un usuario que no exista en el directorio centralizado.
Descripción: Comprobar que el administrador no pueda buscar la información de un usuario en el directorio centralizado LDAP.
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Ingresar el nombre de un usuario que no exista en el campo Buscar
Resultado esperado: Tabla vacía
Evaluación: La evaluación fue aprobada, la tabla no presento ningún registro.

TABLA A13. CLIV.
PRUEBA DE ACEPTACIÓN PARA LA CARGA DE UN CONJUNTO DE USUARIOS CORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 21	**Número de historia de usuario:** 11
Caso de prueba: Cargar Usuarios (.csv)	
Nombre de caso de prueba: Cargar la información de un archivo .csv en el directorio centralizado.	
Descripción: Comprobar que el administrador pueda cargar la información de usuarios contenida en un archivo .csv en el directorio centralizado LDAP.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Archivo de usuarios .csv	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Seleccionar la sección Cargar 5. Seleccionar un archivo .csv con la estructura definida en la modal 6. Hacer clic en Cargar Usuarios	
Resultado esperado: Notificación con la confirmación de la cantidad de usuarios que se han registrado.	
Evaluación: La evaluación fue aprobada, los usuarios se registraron con éxito.	

TABLA A13. CLV.
PRUEBA DE ACEPTACIÓN PARA LA CARGA DE UN CONJUNTO DE USUARIOS INCORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 22	**Número de historia de usuario:** 11
Caso de prueba: Cargar Usuarios (.csv)	
Nombre de caso de prueba: Cargar la información de un archivo .csv mal estructurado en el directorio centralizado.	
Descripción: Comprobar que el administrador no pueda cargar la información de usuarios contenida en un archivo .csv que tiene la estructura mal definida en el directorio centralizado LDAP.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Archivo de usuarios .csv	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Seleccionar la sección Cargar 5. Seleccionar un archivo .csv con una estructura diferente a la presentada en la modal 6. Hacer clic en Cargar Usuarios	
Resultado esperado: Notificación indicando que la estructura está mal definida.	
Evaluación: La evaluación fue aprobada, los usuarios no se registraron con éxito.	

TABLA A15. CLVI.
PRUEBA DE ACEPTACIÓN PARA VINCULAR UN USUARIO CORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 23	**Número de historia de usuario:** 12
Caso de prueba: Vincular Usuario	
Nombre de caso de prueba: Vincular a un usuario a uno o varios sistemas.	
Descripción: Comprobar que el administrador pueda vincular a un usuario a uno o varios sistemas registrados en el directorio centralizado.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Mínimo un sistema registrado. - Mínimo un usuario registrado.	

Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en el botón Vincular 5. Seleccionar uno o varios sistemas 6. Hacer clic en Vincular
Resultado esperado: Notificación con la confirmación de la vinculación del usuario.
Evaluación: La evaluación fue aprobada, el usuario fue vinculado con éxito.

TABLA A13. CLVII.
PRUEBA DE ACEPTACIÓN PARA VINCULAR UN USUARIO INCORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 24	**Número de historia de usuario:** 12
Caso de prueba: Vincular Usuario	
Nombre de caso de prueba: Vincular a un usuario en un sistema en el que ya se encuentra vinculado.	
Descripción: Comprobar que el administrador no pueda vincular a un usuario cuando este ya se encuentre vinculado anteriormente.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Mínimo un sistema registrado. - Mínimo un usuario registrado.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en el botón Vincular 5. Buscar un sistema en el que el usuario ya se encuentra. 6. Hacer clic en Vincular	
Resultado esperado: Notificación indicando de que el usuario no puede vincularse.	
Evaluación: La evaluación fue aprobada, el usuario no fue vinculado con éxito.	

TABLA A13. CLVIII.
PRUEBA DE ACEPTACIÓN PARA VINCULAR VARIOS USUARIOS CORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 25	**Número de historia de usuario:** 13

Caso de prueba: Vincular varios usuarios
Nombre de caso de prueba: Vincular varios usuarios a uno o varios sistemas.
Descripción: Comprobar que el administrador pueda vincular uno o varios usuarios a uno o varios sistemas registrados en el directorio centralizado.
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Mínimo un sistema registrado. - Mínimo un usuario registrado.
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en el botón Vincular Usuarios 5. Seleccionar uno o varios usuarios 6. Seleccionar uno o varios sistemas 7. Hacer clic en Vincular Usuarios
Resultado esperado: Notificación con la confirmación de la vinculación de los usuarios.
Evaluación: La evaluación fue aprobada, los usuarios fueron vinculados con éxito.

TABLA A13. CLIX.
PRUEBA DE ACEPTACIÓN PARA VINCULAR VARIOS USUARIOS INCORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 26	**Número de historia de usuario:** 13
Caso de prueba: Vincular varios usuarios	
Nombre de caso de prueba: Vincular varios usuarios a uno o varios sistemas en los que ya se encuentren vinculados.	
Descripción: Comprobar que el administrador no pueda vincular a varios usuarios cuando estos ya se encuentren vinculados.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Mínimo un sistema registrado. - Mínimo un usuario registrado.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en el botón Vincular Usuarios 5. Seleccionar uno o varios usuarios	

6. Seleccionar uno o varios sistemas 7. Hacer clic en Vincular Usuarios
Resultado esperado: Notificación indicando de que los usuarios no pueden vincularse
Evaluación: La evaluación fue aprobada, los usuarios no fueron vinculados con éxito.

TABLA A13. CLX.
PRUEBA DE ACEPTACIÓN PARA VISUALIZAR LA INFORMACIÓN DE LOS GRUPOS Y SISTEMAS CORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 27	**Número de historia de usuario:** 14
Caso de prueba: Visualizar información de grupos y sistemas	
Nombre de caso de prueba: Visualizar la información de todos los grupos y sistemas registrados en el directorio centralizado.	
Descripción: Comprobar que el administrador pueda visualizar la información de todos los grupos y sistemas registrados en el directorio centralizado.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Mínimo un sistema registrado. - Mínimo un grupo registrado.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en la sección Grupos	
Resultado esperado: Interfaz con toda la información de los grupos y sistemas.	
Evaluación: La evaluación fue aprobada, se puede visualizar a todos los grupos y sistemas.	

TABLA A13. CLXI.
PRUEBA DE ACEPTACIÓN PARA VISUALIZAR LA INFORMACIÓN DE LOS GRUPOS Y SISTEMAS INCORRECTAMENTE.

Prueba de Aceptación	
Número de caso de prueba: 28	**Número de historia de usuario:** 14
Caso de prueba: Visualizar información de grupos y sistemas	
Nombre de caso de prueba: No visualizar la información de los grupos y sistemas registrados en el directorio centralizado.	
Descripción: Comprobar que el administrador no pueda visualizar la información de todos los grupos y sistemas.	

Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Mínimo un sistema registrado. - Mínimo un grupo registrado.
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en cualquier sección menos la de (Grupos)
Resultado esperado: Ningún grupo, ni sistema.
Evaluación: La evaluación fue aprobada, porque no se puedo observar los grupos y sistemas.

TABLA A13. CLXII.
PRUEBA DE ACEPTACIÓN PARA CREAR NUEVOS GRUPOS.

Prueba de Aceptación	
Número de caso de prueba: 29	**Número de historia de usuario:** 15
Caso de prueba: Crear nuevos grupos	
Nombre de caso de prueba: Crear nuevos grupos en el directorio centralizado.	
Descripción: Comprobar que el administrador pueda crear nuevos grupos en el directorio centralizado.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en la sección Grupos 5. Hacer clic en el botón Nuevo Grupo 6. Llenar la información solicitada en el formulario 7. Hacer clic en el botón insertar	
Resultado esperado: Notificación indicando la creación de un nuevo grupo.	
Evaluación: La evaluación fue aprobada, porque se pudo crear un nuevo grupo.	

TABLA A13. CLXIII.
PRUEBA DE ACEPTACIÓN PARA NO CREAR NUEVOS GRUPOS.

Prueba de Aceptación	
Número de caso de prueba: 30	**Número de historia de usuario:** 15

Caso de prueba: Crear nuevos grupos
Nombre de caso de prueba: No crear nuevos grupos en el directorio centralizado.
Descripción: Comprobar que el administrador no pueda crear nuevos grupos en el directorio centralizado.
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en la sección Grupos 5. Hacer clic en el botón Nuevo Grupo 6. No llenar la información solicitada en el formulario 7. Hacer clic en el botón insertar
Resultado esperado: Notificación indicando que no se pudo crear un nuevo grupo.
Evaluación: La evaluación fue aprobada, porque si el formulario no es llenado no se podrá crear un nuevo grupo.

TABLA A13. CLXIV.
PRUEBA DE ACEPTACIÓN PARA ELIMINAR GRUPOS.

Prueba de Aceptación	
Número de caso de prueba: 31	**Número de historia de usuario:** 16
Caso de prueba: Eliminar Grupos	
Nombre de caso de prueba: Eliminar grupos en el directorio centralizado.	
Descripción: Comprobar que el administrador pueda eliminar grupos en el directorio centralizado.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Mínimo un grupo registrado	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en la sección Grupos 5. Hacer clic en el botón Eliminar Grupo 6. Seleccionar uno o varios Grupos 7. Hacer clic en el botón eliminar	
Resultado esperado: Notificación indicando que los grupos fueron eliminados con éxito.	

Evaluación: La evaluación fue aprobada, porque se eliminaron los grupos seleccionados.

TABLA A13. CLXV.
PRUEBA DE ACEPTACIÓN PARA NO ELIMINAR GRUPOS.

Prueba de Aceptación	
Número de caso de prueba: 32	**Número de historia de usuario:** 16
Caso de prueba: Eliminar Grupos	
Nombre de caso de prueba: No eliminar grupos en el directorio centralizado.	
Descripción: Comprobar que el administrador no pueda eliminar grupos en el directorio centralizado.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Mínimo un grupo registrado	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en la sección Grupos 5. Hacer clic en el botón Eliminar Grupo 6. No seleccionar ningún Grupo 7. Hacer clic en el botón eliminar	
Resultado esperado: Notificación indicando que no se a eliminado ningún grupo.	
Evaluación: La evaluación fue aprobada, porque si no se selecciona un grupo este no se elimina.	

TABLA A13. CLXVI.
PRUEBA DE ACEPTACIÓN PARA RESTABLECER CONTRASEÑA.

Prueba de Aceptación	
Número de caso de prueba: 33	**Número de historia de usuario:** 17
Caso de prueba: Restablecer Contraseña	
Nombre de caso de prueba: Restablecer la contraseña de un usuario.	
Descripción: Comprobar que el usuario pueda restablecer su contraseña de acceso.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Conocer el correo electrónico registrado bajo esa cuenta de usuario.	

Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Hacer clic en el botón Recuperar Contraseña 4. Llenar el formulario (Cédula y Usuario) 5. Hacer clic en el botón Recuperar Contraseña
Resultado esperado: Notificación indicando que las indicaciones se le han enviado a su cuenta de correo electrónica.
Evaluación: La evaluación fue aprobada, porque las indicaciones fueron enviadas al usuario que solicito el restablecimiento de contraseña.

TABLA A15. CLXVII.
PRUEBA DE ACEPTACIÓN PARA NO RESTABLECER CONTRASEÑA.

Prueba de Aceptación	
Número de caso de prueba: 34	**Número de historia de usuario:** 17
Caso de prueba: Restablecer Contraseña	
Nombre de caso de prueba: No restablecer la contraseña de un usuario.	
Descripción: Comprobar que el usuario no pueda restablecer su contraseña de acceso.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Conocer el correo electrónico registrado bajo esa cuenta de usuario.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Hacer clic en el botón Recuperar Contraseña 4. No llenar el formulario (Cédula y Usuario) 5. Hacer clic en el botón Recuperar Contraseña	
Resultado esperado: Notificación indicando que el usuario es incorrecto.	
Evaluación: La evaluación fue aprobada, porque no se envió las indicaciones al correo electrónico ya que no se proporciona las credenciales para saber la identidad de un usuario.	

TABLA A13. CLXVIII.
PRUEBA DE ACEPTACIÓN PARA CREAR NUEVOS SISTEMAS.

Prueba de Aceptación	
Número de caso de prueba: 35	**Número de historia de usuario:** 18
Caso de prueba: Crear nuevos sistemas	
Nombre de caso de prueba: Crear nuevos sistemas en el directorio centralizado.	

Descripción: Comprobar que el administrador pueda crear nuevos sistemas en el directorio centralizado.
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en la sección Grupos 5. Hacer clic en el botón Nuevo Sistema 6. Llenar la información solicitada en el formulario 7. Hacer clic en el botón insertar
Resultado esperado: Notificación indicando la creación de un nuevo sistema.
Evaluación: La evaluación fue aprobada, porque se pudo crear un nuevo sistema.

TABLA A13. CLXIX.
PRUEBA DE ACEPTACIÓN PARA NO CREAR NUEVOS SISTEMAS.

Prueba de Aceptación	
Número de caso de prueba: 36	**Número de historia de usuario:** 18
Caso de prueba: Crear nuevos sistemas	
Nombre de caso de prueba: No crear nuevos sistemas en el directorio centralizado.	
Descripción: Comprobar que el administrador no pueda crear nuevos grupos en el directorio centralizado.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLdap deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en la sección Grupos 5. Hacer clic en el botón Nuevo Sistema 6. No llenar la información solicitada en el formulario 7. Hacer clic en el botón insertar	
Resultado esperado: Notificación indicando que no se pudo crear un nuevo sistema.	
Evaluación: La evaluación fue aprobada, porque si el formulario no es llenado no se podrá crear un nuevo sistema.	

TABLA A13. CLXX.
PRUEBA DE ACEPTACIÓN PARA ELIMINAR SISTEMAS.

Prueba de Aceptación	
Número de caso de prueba: 37	**Número de historia de usuario:** 19
Caso de prueba: Eliminar Sistemas	
Nombre de caso de prueba: Eliminar sistemas en el directorio centralizado.	
Descripción: Comprobar que el administrador pueda eliminar sistemas en el directorio centralizado.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Mínimo un grupo registrado - Mínimo un sistema registrado	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en la sección Grupos 5. Hacer clic en el botón Eliminar Sistema 6. Seleccionar uno o varios Sistemas 7. Hacer clic en el botón eliminar	
Resultado esperado: Notificación indicando que los sistemas fueron eliminados con éxito.	
Evaluación: La evaluación fue aprobada, porque se eliminaron los sistemas seleccionados.	

TABLA A13. CLXXI.
PRUEBA DE ACEPTACIÓN PARA NO ELIMINAR SISTEMAS.

Prueba de Aceptación	
Número de caso de prueba: 38	**Número de historia de usuario:** 19
Caso de prueba: Eliminar Sistemas	
Nombre de caso de prueba: No eliminar Sistemas en el directorio centralizado.	
Descripción: Comprobar que el administrador no pueda eliminar sistemas en el directorio centralizado.	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Mínimo un grupo registrado - Mínimo un sistema registrado	
Entrada: 1. Ingresar a la interfaz de acceso principal.	

2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en la sección Grupos 5. Hacer clic en el botón Eliminar Sistema 6. No seleccionar ningún Sistema 7. Hacer clic en el botón eliminar
Resultado esperado: Notificación indicando que no se a eliminado ningún sistema
Evaluación: La evaluación fue aprobada, porque si no se selecciona un sistema este no se elimina.

TABLA A13. CLXXII.
PRUEBA DE ACEPTACIÓN PARA GENERAR REPORTES.

Prueba de Aceptación	
Número de caso de prueba: 39	**Número de historia de usuario: 20**
Caso de prueba: Generar Reportes	
Nombre de caso de prueba: Generar un reporte de usuarios	
Descripción: Comprobar que el administrador pueda generar un reporte de usuarios de un grupo seleccionado	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Mínimo un usuario registrado	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en el botón Generar Reporte 5. Seleccionar un grupo (Estudiantes, docentes, Servidores, Trabajadores u Otros). 6. Hacer clic en el botón Seleccionar	
Resultado esperado: Notificación indicando que el reporte se a generado con éxito.	
Evaluación: La evaluación fue aprobada, porque el reporte se generó y se descargo automáticamente.	

TABLA A13. CLXXIII.
PRUEBA DE ACEPTACIÓN PARA GENERAR UN REPORTE INCORRECTO.

Prueba de Aceptación	
Número de caso de prueba: 40	**Número de historia de usuario: 20**
Caso de prueba: Generar Reportes	
Nombre de caso de prueba: Generar un reporte de usuarios incorrecto	

Descripción: Comprobar que el administrador no pueda generar un reporte de usuarios de un grupo
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Mínimo un usuario registrado
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en el botón Generar Reporte 5. Seleccionar un grupo (Estudiantes, docentes, Servidores, Trabajadores u Otros). 8. Hacer clic en el botón Cancelar
Resultado esperado: Interfaz principal del sistema SAC.
Evaluación: La evaluación fue aprobada, porque el reporte no se generó debido a que se opto por cancelar el proceso

TABLA A13. CLXXIV.
PRUEBA DE ACEPTACIÓN PARA ELIMINAR TODOS LOS REGISTROS DE UN GRUPO.

Prueba de Aceptación	
Número de caso de prueba: 41	**Número de historia de usuario: 21**
Caso de prueba: Eliminar todos los registros	
Nombre de caso de prueba: Eliminar todos los registros de un grupo seleccionado	
Descripción: Comprobar que el administrador pueda eliminar todos los registros de un grupo seleccionado	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en el botón Eliminar Todo 5. Seleccionar un grupo (Estudiantes, docentes, Servidores, Trabajadores u Otros). 6. Hacer clic en el botón Seleccionar 7. Hacer clic en el botón Aceptar	
Resultado esperado: Notificación indicando que todos los usuarios del grupo seleccionado fueron eliminados con éxito.	
Evaluación: La evaluación fue aprobada, porque se eliminaron todos los registros de un grupo.	

TABLA A13. CLXXV.
PRUEBA DE ACEPTACIÓN PARA NO ELIMINAR TODOS LOS REGISTROS DE UN GRUPO.

Prueba de Aceptación	
Número de caso de prueba: 42	**Número de historia de usuario:** 21
Caso de prueba: Eliminar todos los registros	
Nombre de caso de prueba: No eliminar todos los registros de un grupo	
Descripción: Comprobar que el administrador no pueda eliminar todos los registros de un grupo seleccionado	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en el botón Eliminar Todo 5. Seleccionar un grupo (Estudiantes, docentes, Servidores, Trabajadores u Otros). 6. Hacer clic en el botón Seleccionar 9. Hacer clic en el botón Cancelar	
Resultado esperado: Interfaz principal del sistema SAC.	
Evaluación: La evaluación fue aprobada, porque no se eliminaron los registros de un grupo, debido a que se cancelo el proceso.	

TABLA A13. CLXXVI.
PRUEBA DE ACEPTACIÓN PARA CREAR NUEVOS ADMINISTRADORES DE LECTURA.

Prueba de Aceptación	
Número de caso de prueba: 43	**Número de historia de usuario: 22**
Caso de prueba: Crear nuevos administradores de lectura	
Nombre de caso de prueba: Crear nuevos administradores con acceso solo de lectura al sistema SAC	
Descripción: Comprobar que el administrador pueda crear nuevos administradores con permisos de lectura para SAC	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal.	

2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en la sección Administrador 5. Llenar el formulario solicitado 10. Hacer clic en el botón Crear
Resultado esperado: Notificación indicando que el administrador solo de lectura fue creado con éxito
Evaluación: La evaluación fue aprobada, porque crea el nuevo administrador de lectura en el directorio centralizado.

TABLA A13. CLXXVII.
PRUEBA DE ACEPTACIÓN PARA NO CREAR NUEVOS ADMINISTRADORES DE LECTURA.

Prueba de Aceptación	
Número de caso de prueba: 44	**Número de historia de usuario: 22**
Caso de prueba: Crear nuevos administradores de lectura	
Nombre de caso de prueba: No crear nuevos administradores con acceso solo de lectura al sistema SAC	
Descripción: Comprobar que el administrador no pueda crear nuevos administradores con permisos de lectura para SAC	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP.	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en la sección Administrador 5. No llenar el formulario solicitado 11. Hacer clic en el botón Crear	
Resultado esperado: Notificación indicando que el administrador solo de lectura no pudo ser creado con éxito	
Evaluación: La evaluación fue aprobada, porque no se crea el nuevo administrador de lectura en el directorio centralizado, debido a que no se llena el formulario.	

TABLA A13. CLXXVIII.
PRUEBA DE ACEPTACIÓN PARA VISUALIZAR LA INFORMACIÓN DE UN GRUPO DE USUARIOS DE UN SISTEMA

Prueba de Aceptación	
Número de caso de prueba: 45	**Número de historia de usuario: 23**
Caso de prueba: Visualizar Usuarios Vinculados	

Nombre de caso de prueba: Visualizar la información de un grupo de usuarios vinculados a un sistema
Descripción: Comprobar que el administrador pueda visualizar la información de los usuarios que se encuentran dentro de un sistema
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Mínimo un grupo registrado - Mínimo un sistema registrado - Mínimo un usuario registrado
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en la sección Grupos 5. Hacer clic en el nombre del sistema que se quiere visualizar
Resultado esperado: Interfaz con la información de los usuarios
Evaluación: La evaluación fue aprobada, porque se puede visualizar la lista con los usuarios vinculados al sistema seleccionado

TABLA A13. CLXXIX.
PRUEBA DE ACEPTACIÓN PARA NO VISUALIZAR LA INFORMACIÓN DE UN GRUPO DE USUARIOS DE UN SISTEMA

Prueba de Aceptación	
Número de caso de prueba: 46	**Número de historia de usuario: 23**
Caso de prueba: Visualizar Usuarios Vinculados	
Nombre de caso de prueba: No visualizar la información de un grupo de usuarios vinculados a un sistema	
Descripción: Comprobar que el administrador no pueda visualizar la información de los usuarios que no se encuentran dentro de un sistema	
Condiciones de ejecución: - El sistema deberá estar levantado en un servidor local. - El servidor OpenLDAP deberá estar levantado en un servidor local. - Estar registrado en el directorio centralizado LDAP. - Mínimo un grupo registrado - Mínimo un sistema registrado	
Entrada: 1. Ingresar a la interfaz de acceso principal. 2. Seleccionar el sistema SAC. 3. Ingresar correctamente las credenciales (Usuario y contraseña). 4. Hacer clic en la sección Grupos 5. Hacer clic en el nombre del sistema que no tenga usuarios vinculados	
Resultado esperado: Interfaz vacía	

Evaluación: La evaluación fue aprobada, porque se puede visualizar la interfaz vacía ya que no existen usuarios vinculados

Anexo 14. Configuración y personalización del Sistema Jasig CAS.

A continuación, se darán a conocer los principales métodos de conexión, autenticación y personalización del Sistema Jasig CAS con el servidor OpenLDAP, la codificación completa del sistema se encuentra disponible en la siguiente dirección https://github.com/Antonio-Cis/Sistema-Jasig-CAS.git

Paso 1

Descargamos una versión estable de Jasig CAS en nuestro caso se utilizó la versión 4.0.1 disponible en el siguiente enlace:

https://www.apereo.org/projects/cas/download-cas

Paso 2

Descomprimimos el archivo **cas-4.0.1.zip** que descargamos y lo guardamos en cualquier lugar y con cualquier nombre (en nuestro caso tendrá el mismo nombre), con el siguiente comando.

unzip cas-4.0.1.zip

Paso 3

Otorgamos los permisos necesarios para el acceso al archivo con el siguiente comando.

sudo chmod 777 cas-4.0.1

Paso 4

Ingresamos al archivo **cas.properties** que se encuentra en **cas-4.0.1/cas-server-Webapp/src/main/Webapp/WEB-INF** para establecer las variables que se conectaran con el servidor OpenLDAP.

A continuación, se detallan el valor de las variables:

- **server.name =** dirección del servidor CAS con su puerto
- **server.prefix =** concatenación de la variable anterior y el prefijo o alias para el servicio CAS, luego del puerto.
- **ldap.url =** ingresamos el nombre o dirección IP del servidor OpenLDAP, seguido de su puerto. En esta variable ingresamos de la siguiente manera “ldap://midominio:puerto”.
- **cas.securityContext.status.allowedSubnet =** establecemos la dirección IP de nuestro dominio.

```
server.name=https://localhost:8443
server.prefix=${server.name}/cas
ldap.url=ldap://ldapserver:389
# IP address or CIDR subnet allowed to access the /status URI of CAS that exposes health check information
cas.securityContext.status.allowedSubnet=127.0.0.1
```

Figura A14. 1. Inicialización de variables para la conexión con el servidor OpenLDAP.

Paso 5

Ingresamos al archivo **deployerConfigContext.xml** que se encuentra en **cas-4.0.1/cas-server-Webapp/src/main/Webapp/WEB-INF** para crear los métodos de autenticación y conexión con el servidor OpenLDAP.
A continuación, se describen los métodos para la conexión y autenticación. Los métodos, que se muestran en la Figura A14. 2 son de carácter obligatorio para el correcto funcionamiento del sistema.
El primer método nos sirve para establecer la url de conexión con nuestro servidor OpenLDAP, definido en el paso anterior.

El segundo método nos sirve para establecer la ruta y contraseña del administrador del servidor OpenLDAP y poder tener acceso al mismo para realizar los procesos de autenticación con la información almacenada.

En el tercer método definimos la clase Auth que nos sirve para la autenticación mediante un identificador de usuario y clave asociada, así mismo se define el dominio o ruta donde se realizará la autenticación de los usuarios que se encuentran en el servidor OpenLDAP. A continuación, se detallan varias variables:

- **subtreeSearch =** la definimos en **true**, para poder realizar búsquedas.
- **allowMultipleDns =** la definimos en **true**, para poder realizar búsquedas en subgrupos o sub-unidades organizacionales que se encuentren luego de la ruta principal.
- **userFilter =** establecemos el atributo **uid** como el identificador principal de cada usuario para que la información ingresada en ese campo tome ese valor y los verifique su información en el servidor OpenLDAP. Por ejemplo “uid=usuario,ou=subgrupo,ou=grupo,dc=mi,dc=dominio”.

```
<bean id="connectionConfig" class="org.ldaptive.ConnectionConfig"
          p:ldapUrl="${ldap.url}"
          p:connectTimeout="3000"
          p:useStartTLS="false"
          p:connectionInitializer-ref="bindConnectionInitializer"/>

<bean id="bindConnectionInitializer" class="org.ldaptive.BindConnectionInitializer"
      p:bindDn="cn=admin,dc=cas,dc=com">
      <property name="bindCredential">
         <bean class="org.ldaptive.Credential"  c:password="ldap1234" />
      </property>
</bean>

<bean id="pooledSearchDnResolver" class="org.ldaptive.auth.PooledSearchDnResolver"
        p:baseDn="ou=personal,dc=cas,dc=com"
        p:subtreeSearch="true"
        p:allowMultipleDns="true"
        p:connectionFactory-ref="pooledConnectionFactory"
        p:userFilter="uid={user}" />
```

Figura A14. 2. Métodos de conexión y autenticación con el servidor OpenLDAP

Paso 6

Para la personalización de la interfaz principal del Sistema Jasig CAS, se debe configurar documentos con las siguientes extensiones (.css, .jsp, .xml); donde describiremos los principales y el resto de la codificación se encuentra en el repositorio indicado en el primer paso.

Ingresamos al archivo **Web.xml** que se encuentra en **cas-4.0.1/cas-server-Webapp/src/main/Webapp/WEB-INF** para definir por defecto el idioma español en la interfaz principal del Sistema Jasig CAS y anular la selección del resto de idiomas que nos brinda Jasig CAS.

En la siguiente línea de código, en la variable **defaultLocale** definimos "**es**" estableciendo el idioma por defecto español.

```
<!-- Locale Resolver -->
<bean id="localeResolver" class="org.springframework.web.servlet.i18n.CookieLocaleResolver" p:defaultLocale="es" />
```

Figura A14. 3. Definir idioma por defecto español en la interfaz principal del Sistema Jasig CAS

Paso 7

Ingresamos al archivo **casLoginView.jsp** que se encuentra en **cas-4.0.1/cas-server-Webapp/src/main/Webapp/WEB-INF/view/jsp/default/ui** para agregar dos botones que nos dirigirán a los métodos cambio y recuperación de contraseña los cuales interactúan directamente con el servidor OpenLDAP y serán los únicos aprobados para este proceso. Estos métodos se encuentran definidos en el (Anexo 12. Desarrollo de un Web Service con métodos administrativos y de autenticación).

En la (Figura A14. 4), se detalla el código que se debe agregar para llamar estos métodos:

```
<section class="row btn-row">
  <input type="hidden" name="lt" value="${loginTicket}" />
  <input type="hidden" name="execution" value="${flowExecutionKey}" />
  <input type="hidden" name="_eventId" value="submit" />

  <input class="btn-submit" name="submit" accesskey="l" value="<spring:message code="screen.welcome.button.login" />" tabindex="4"
  type="submit" />
  <input class="btn-reset" name="reset" accesskey="c" value="<spring:message code="screen.welcome.button.clear" />" tabindex="5" type
  ="reset" /><br></br>
  <a class="text-secondary" href="https://sac.unl.edu.ec/index.php" style="font-weight: bold;">Cambiar contrase&ntilde;a</a>
  <a class="text-secondary" href="https://sac.unl.edu.ec/recuperarContrasenia.php" style="font-weight: bold;">Recuperar contrase&
  ntilde;a</a>
</section>
```

Figura A14. 4. Agregar botones en la interfaz principal del Sistema Jasig CAS.

Paso 8

Abrimos una terminal y nos ubicamos en la siguiente ruta **cas-4.0.1/cas-server-Webapp/** con el siguiente comando:

cd cas-4.0.1/cas-server-Webapp/

Paso 9

Ejecutamos la superposición de maven para evitar códigos repetitivos que utilizara el archivo **pom.xml** del archivo **cas-server-Webapp** para crear el archivo final **cas.war** que se lo va a implementar en un contenedor de servlets en nuestro caso **Apache Tomcat.**

mvn clean package

```
[INFO] Packaging webapp
[INFO] Assembling webapp [cas-server-webapp] in [/home/manolo/Descargas/cas-4.0 (1).1/cas-serve
r-webapp/target/cas-server-webapp-4.0.1]
[INFO] Processing war project
[INFO] Copying webapp webResources [/home/manolo/Descargas/cas-4.0 (1).1/cas-server-webapp/src/
main/webapp/WEB-INF] to [/home/manolo/Descargas/cas-4.0 (1).1/cas-server-webapp/target/cas-serv
er-webapp-4.0.1]
[INFO] Copying webapp resources [/home/manolo/Descargas/cas-4.0 (1).1/cas-server-webapp/src/mai
n/webapp]
[INFO] Webapp assembled in [654 msecs]
[INFO] Building war: /home/manolo/Descargas/cas-4.0 (1).1/cas-server-webapp/target/cas.war
[INFO] Packaging classes
[INFO] Building jar: /home/manolo/Descargas/cas-4.0 (1).1/cas-server-webapp/target/cas-classes.
jar
[INFO]
[INFO] --- maven-site-plugin:3.1.r1174614:attach-descriptor (attach-descriptor) @ cas-server-we
bapp ---
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 23.426 s
[INFO] Finished at: 2019-01-27T20:37:41-05:00
[INFO] ------------------------------------------------------------------------
```

Figura A14. 5. Compilación y resultado de los cambios para el Sistema Jasig CAS.

Paso 11

Una vez finalizado la ejecución de maven ya tendremos nuestro Servicio CAS listo para levantarlo en cualquier servidor, mediante el uso de su interfaz principal del Sistema Jasig CAS.

Anexo 15. Instalación Apache Tomcat y levantamiento de Jasig CAS

Requerimientos:

- Apache Tomcat.
- cas.war Resultado del (Anexo 14. Configuración y personalización del Sistema Jasig CAS.).
- Certificado de seguridad SSL (Herramienta Keytool de Java).
- Conexión a internet

Paso 1

Comprobamos e instalamos las actualizaciones más recientes a nuestro sistema con el siguiente comando.

apt-get update && apt-get upgrade

Paso 2

Descargamos una versión estable de Apache Tomcat en nuestro caso se utilizó la versión 8.5.34 disponible en el siguiente enlace:

https://tomcat.apache.org/download-80.cgi

Paso 3

Descomprimimos el archivo apache-tomcat-8.5.34.zip que descargamos y lo guardamos con cualquier nombre en la ruta de nuestra preferencia (en nuestro caso tiene el mismo nombre), con el siguiente comando:

unzip apache-tomcat-8.5.34.zip

Paso 4

Otorgamos los permisos necesarios para el acceso al archivo con el siguiente comando.

sudo chmod 777 apache-tomcat-8.5.34

Paso 5

Para iniciar el servidor Apache Tomcat abrimos una terminal e ingresamos a la siguiente ruta *apache-tomcat-8.5.34/bin/* y ejecutamos el archivo *startup.sh* con los siguientes comandos.

cd apache-tomcat-8.5.34/bin/

sh startup.sh

Paso 6

Para comprobar que Apache Tomcat se levantó correctamente ingresamos a nuestro navegador e ingresamos a la siguiente dirección:

http://localhost:8080/

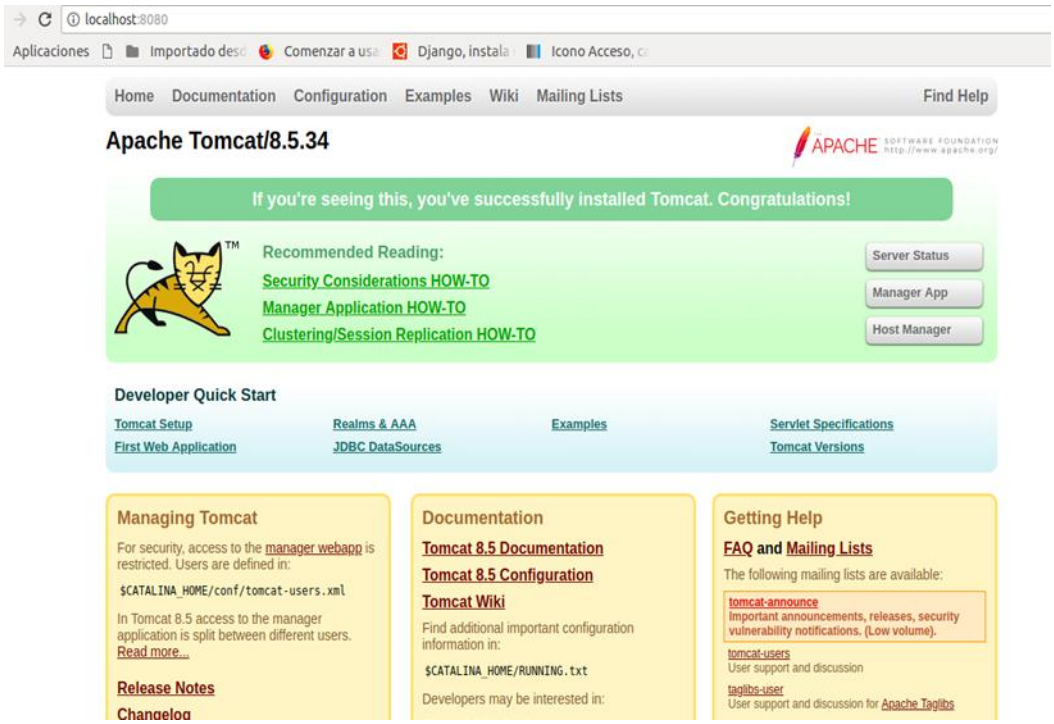


Figura A15. 1. Interfaz principal de Apache Tomcat

Paso 7

Para acceder al panel de administración de Apache tomcat ingresamos a la siguiente ruta *apache-tomcat-8.5.34/conf/* abrimos una terminal y modificamos el archivo *tomcat-users.xml* y añadimos lo siguiente:

sudo gedit tomcat-user.xml

Username: Definimos un nombre de usuario

Password: Definimos una contraseña de usuario

Paso 8

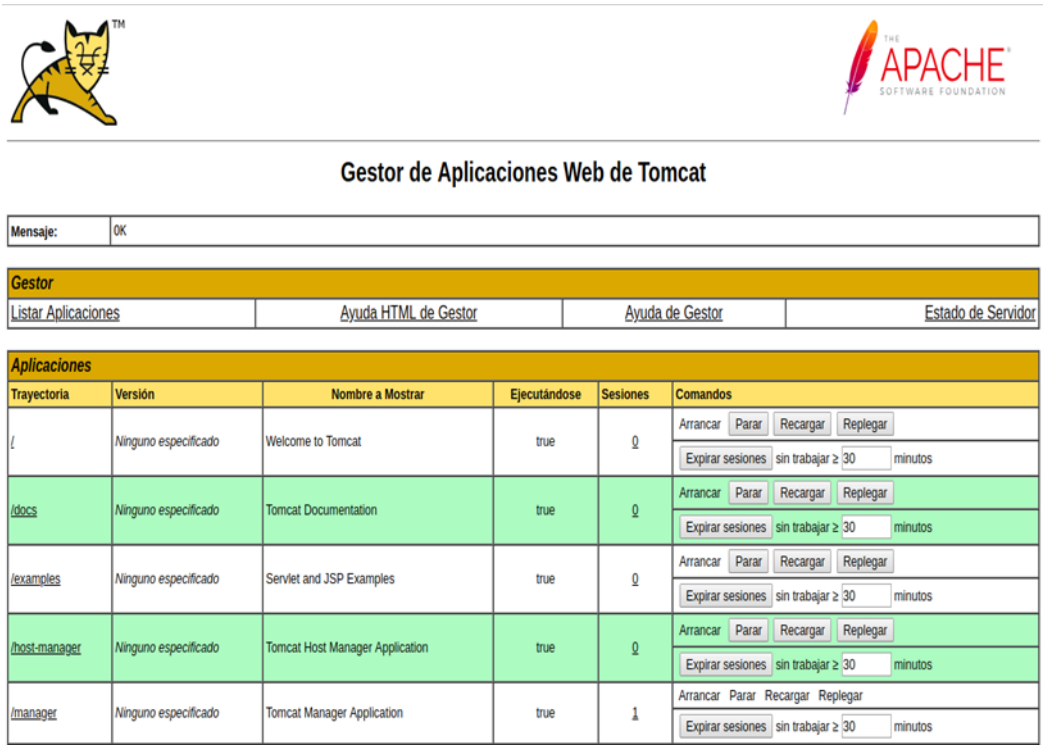
Para guardar los cambios reiniciamos el servidor Apache Tomcat con los siguientes comandos:

sh shutdown.sh

sh startup.sh

Paso 9

Comprobamos si podemos ingresar al panel de administración de Apache Tomcat con el usuario y password que definimos en el paso 7, en nuestro navegador en la opción Manager App.



Gestor de Aplicaciones Web de Tomcat

Mensaje:	OK

Gestor			
Listar Aplicaciones	Ayuda HTML de Gestor	Ayuda de Gestor	Estado de Servidor

Aplicaciones					
Trayectoria	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	*Ninguno especificado*	Welcome to Tomcat	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/docs	*Ninguno especificado*	Tomcat Documentation	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/examples	*Ninguno especificado*	Servlet and JSP Examples	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/host-manager	*Ninguno especificado*	Tomcat Host Manager Application	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/manager	*Ninguno especificado*	Tomcat Manager Application	true	1	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos

Figura A15. 2. Gestor de Aplicaciones Web de Tomcat

Si todo salió bien debería presentarnos una imagen con las rutas, versiones, nombre, estado y comandos para cada aplicación del servidor.

Paso 10

Para que nuestro Jasig CAS funcione sin problemas necesita ejecutarse en un ambiente de comunicación seguro para lo cual creamos y añadimos un certificado de seguridad SSL a nuestro Apache Tomcat con el programa de utilidad keytool que nos ofrece JAVA.

Paso 11

Creamos un certificado de seguridad *SSL* ejecutando el siguiente comando en una terminal para crear un par de claves (Pública - Privada):

```
sudo keytool -genkey -alias apachetomcat -keyalg RSA
-keystore /var/local/apache-tomcat-8.5.34
```

En donde:

genkey: Es la petición para crear una nueva clave.

alias: Es el nombre con el que haremos referencia al par de claves creados.

keyalg: Es el algoritmo que se utiliza para generar la clave.

keystore: Es la ruta para guardad las claves.

```
manolo@ldap2:~$ sudo keytool -genkey -alias apachetomcat -keyalg RSA -keystore /
var/local/apache-tomcat-8.5.34/clave
Introduzca la contraseña del almacén de claves:
Volver a escribir la contraseña nueva:
¿Cuáles son su nombre y su apellido?
  [Unknown]:  ApacheTomcat
¿Cuál es el nombre de su unidad de organización?
  [Unknown]:  ApacheTomcat
¿Cuál es el nombre de su organización?
  [Unknown]:  ApacheTomcat
¿Cuál es el nombre de su ciudad o localidad?
  [Unknown]:  Loja
¿Cuál es el nombre de su estado o provincia?
  [Unknown]:  Loja
¿Cuál es el código de país de dos letras de la unidad?
  [Unknown]:  EC
¿Es correcto CN=ApacheTomcat, OU=ApacheTomcat, O=ApacheTomcat, L=Loja, ST=Loja,
C=EC?
  [no]:  si

Introduzca la contraseña de clave para <apachetomcat>
        (INTRO si es la misma contraseña que la del almacén de claves):

Warning:
El almacén de claves JKS utiliza un formato propietario. Se recomienda migrar a
PKCS12, que es un formato estándar del sector que utiliza "keytool -importkeysto
re -srckeystore /var/local/apache-tomcat-8.5.34/clave -destkeystore /var/local/a
pache-tomcat-8.5.34/clave -deststoretype pkcs12".
```

Figura A15. 3. Creación de Certificado de Seguridad con Keytool

Paso 12

Si nos presenta un Warning no quiere decir que este mal, únicamente nos recomienda migrar a **PKCS12** si deseamos cambiarlo ejecutamos el siguiente comando caso contrario pasar al paso 13:

sudo keytool -importkeystore -srckeystore /var/local/apache-tomcat-8.5.34/clave -destkeystore /var/local/apache-tomcat-8.5.34/clave -deststoretype pkcs12

```
manolo@ldap2:~$ sudo keytool -importkeystore -srckeystore /var/local/apache-tomc
at-8.5.34/clave -destkeystore /var/local/apache-tomcat-8.5.34/clave -deststorety
pe pkcs12
Introduzca la contraseña de almacén de claves de origen:
La entrada del alias apachetomcat se ha importado correctamente.
Comando de importación completado: 1 entradas importadas correctamente, 0 entrad
as incorrectas o canceladas

Warning:
Se ha migrado "/var/local/apache-tomcat-8.5.34/clave" a Non JKS/JCEKS. Se ha rea
lizado la copia de seguridad del almacén de claves JKS como "/var/local/apache-t
omcat-8.5.34/clave.old2".
```

Figura5. 4. Migrar clave a PKCS12

Paso 13

Revisamos la ruta de destino que especificamos en el paso 11 para ver si la clave ya se encuentra creada.

Paso 14

Otorgamos los permisos necesarios para acceder a la clave con el siguiente comando:

sudo chmod 777 clave

Paso 15

Para agregar la clave al servidor Apache tomcat ingresamos a la siguiente ruta **apache-tomcat-8.5.34/conf/** abrimos una terminal y modificamos el archivo **server.xml** y añadimos lo siguiente:

cd ruta_establecida/apache-tomcat-8.5.34/conf/

sudo gedit server.xml

```
<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"
           maxThreads="150" SSLEnabled="true" keystoreFile="/var/local/clave" keystorePass="manolo1994">
  <!-- <SSLHostConfig>
       <Certificate certificateKeystoreFile="conf/localhost-rsa.jks"
                    type="RSA" />
   </SSLHostConfig> -->
</Connector>
```

Figura A15. 5. Clave agregada al servidor Apache Tomcat

En donde:

Port: Es el puerto seguro HTTPS.

SSLEnable: true para el uso del certificado de seguridad.

keystoreFile: Indica la ruta donde guardamos el certificado.

keystorePass: Es la contraseña que le pusimos al certificado.

Paso 16

Para guardar los cambios reiniciamos el servidor Apache Tomcat con los siguientes comandos:

sh shutdown.sh

sh startup.sh

Paso 17

Para comprobar que Apache Tomcat se levantó correctamente con el certificado de seguridad en el puerto seguro 8443 ingresamos a nuestro navegador e ingresamos a la siguiente dirección:

https://localhost:8443

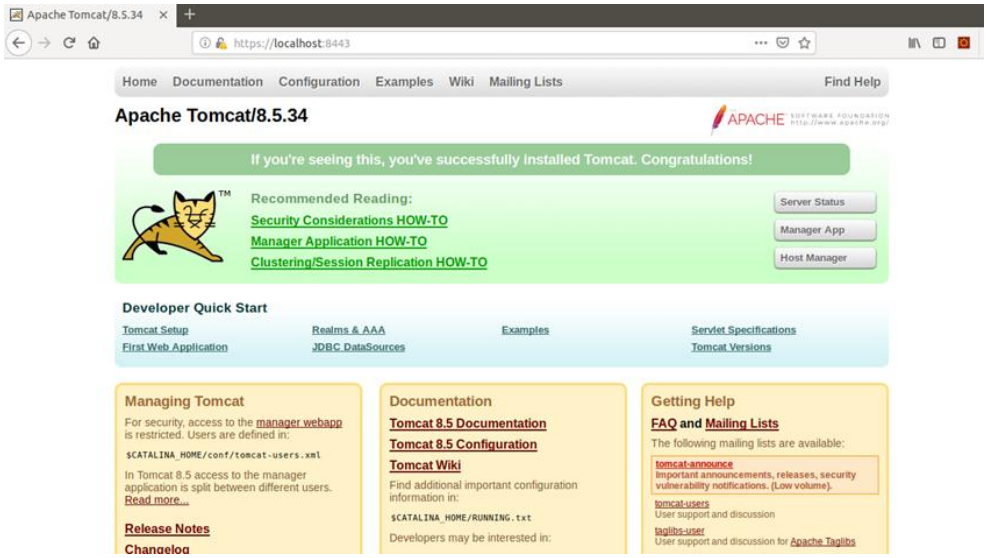


Figura A15. 6. Interfaz Principal de Apache Tomcat con Certificado de Seguridad Autofirmado

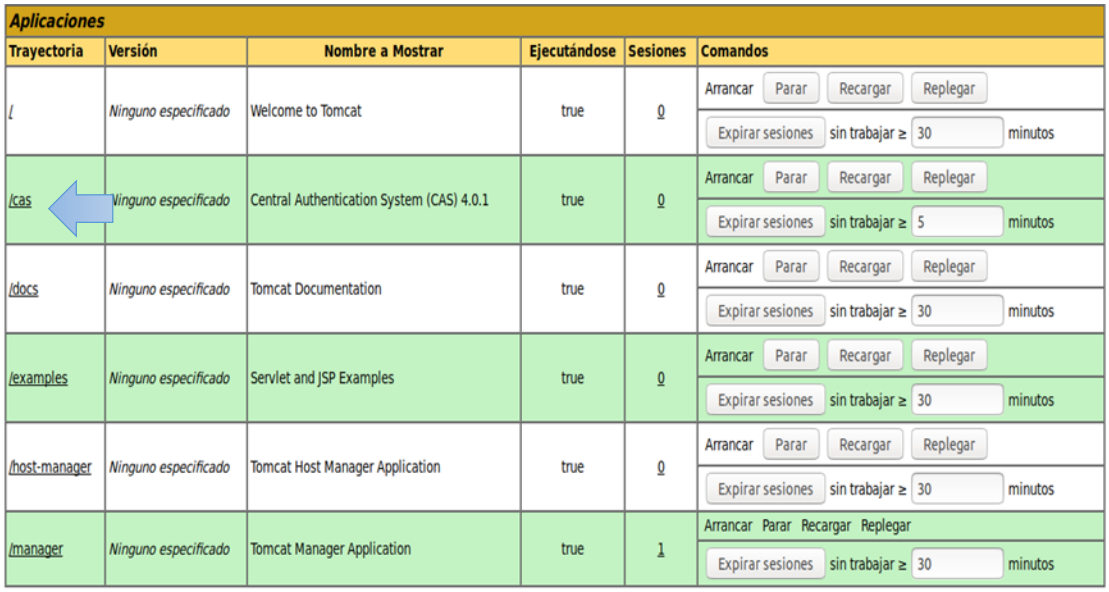
Paso 18

Para levantar nuestro Jasig CAS (cas.war), que lo creamos en el (Anexo 14. Configuración y personalización del Sistema Jasig CAS.) ingresamos a Manager App con las credenciales que ingresamos en el paso 7 y nos ubicamos en la sección Desplegar.

Ingresamos en examinar y seleccionamos el archivo cas.war que ya se encuentra ceado y le damos clic en desplegar.

Paso 19

Esperamos unos segundos a que cargue el archivo cas.war y nos ubicamos en Aplicaciones para revisar si el archivo ya se encuentra cargado.



Aplicaciones

Trayectoria	Versión	Nombre a Mostrar	Ejecutándose	Sesiones	Comandos
/	Ninguno especificado	Welcome to Tomcat	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/cas	Ninguno especificado	Central Authentication System (CAS) 4.0.1	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 5 minutos
/docs	Ninguno especificado	Tomcat Documentation	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/examples	Ninguno especificado	Servlet and JSP Examples	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/host-manager	Ninguno especificado	Tomcat Host Manager Application	true	0	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos
/manager	Ninguno especificado	Tomcat Manager Application	true	1	Arrancar Parar Recargar Replegar Expirar sesiones sin trabajar ≥ 30 minutos

Figura A15. 7. Gestor Apache Tomcat con archivo cas.war generado

Paso 20

Para verificar si se levantó correctamente *Jasig CAS* le damos clic en */cas* o ingresamos a la siguiente dirección:

https://localhost:8443/cas/

Figura A15. 8. Interfaz principal y personalizada del Sistema Jasig CAS

Paso 21

Para comprobar si está funcionando correctamente podemos ingresar las credenciales que se encuentran almacenadas en nuestro servidor OpenLDAP.

Se nos presentará una ventana de inicio de sesión exitos.

Figura A15. 9. Interfaz de Inicio de Sesión Exitoso en el Sistema Jasig CAS

Anexo 16. Ciclo DevOps

A. Evaluar el Servicio de Autenticación Central desarrollado a través de los DevOps.

Para evaluar el prototipo final se hace uso del ciclo de vida de los DevOps lo que nos permite organizar toda la información en cada una de sus fases para lograr un prototipo funcional, el ciclo DevOps consta de 6 fases las cuales se las desarrolla a continuación:

1. Requisitos

El enfoque DevOps es utilizado por la mayoría de las aplicaciones Web, por ende, no se tiene un informe exacto de la cantidad en utilizar este enfoque lo que quiere decir que se hizo uso de un muestreo no probabilístico por conveniencia, seleccionando 13 aplicaciones Web que utilicen este enfoque para realizar una comparación de los mismos, los cuales permitan: el uso del sistema SSO, Protocolo de Autenticación CAS, sean de licencia libre y utilicen el sistema de directorio centralizado LDAP. Teniendo en cuenta estás 4 características como las principales, las cuales se obtuvieron en la (Objetivo 1: Analizar la autenticación del protocolo CAS, como mecanismo centralizado), y las características restantes definidas como son: lenguaje de desarrollo, para verificar el soporte de este con el Protocolo CAS y el enfoque DevOps, para el desarrollo y operaciones del sistema.

1.1. Evaluación del desempeño

Para llevar a cabo esta evaluación del desempeño se requieren dos etapas:

- El primero, es donde se van asignando puntos por la presencia de ciertas características en el desempeño en el producto.
- El segundo, es en el que se describe a detalle cómo se ubicará a un producto en cada uno de los niveles de desempeño.

TABLA A16. I.

CALIFICACIÓN DE LA EVALUACIÓN DE DESEMPEÑO

Pesos (Puntos)	Descripción
0	No cumple o no tiene implementado este proceso específico.
1	Cumple o tiene implementado el proceso indicado.

1.2. Revisión de Registros

Para llevar a cabo este método, se procedió a realizar una revisión de información mediante páginas oficiales de todas las aplicaciones Web, que se tomaron en cuenta para el proceso de selección final.

1.3. Identificación de las características de los sistemas Web

- **Lenguaje de desarrollo:** Hace referencia al lenguaje de programación en el que se encuentran desarrollados los sistemas Web, es una característica muy importante para poder determinar si el protocolo CAS tiene soporte para dichos lenguajes.
    - **PHP:** Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo Web y que puede ser incrustado en html [56].
    - **Python:** Es un lenguaje de programación de alto nivel, interpretativo, imperativo, orientado a objetos, funcional, de tipeado dinámico y fuerte [57].
    - **Java:** Java es un lenguaje sencillo, orientado a objetos distribuido, interpretado, robusto, securizado, independiente de las arquitecturas, portable, eficaz, multihilo y dinámico [58].
    - **Otros:** Como por ejemplo Ruby, .Net
- **Enfoque DevOps:** El enfoque DevOps nos permite eliminar la interdependencia del desarrollo de software y las operaciones de las tecnologías de la información, en otras palabras, el uso del enfoque DevOps establece la integración de los desarrolladores de software y administradores de sistemas.
- **Protocolo CAS:** CAS es un protocolo de inicio de sesión único para la Web. Su propósito es permitir que un usuario acceda a múltiples aplicaciones mientras proporciona sus credenciales (como identificación de usuario y contraseña) solo una vez.
- **Sistema Single Sign On SSO:** Permite a un usuario acceder a múltiples servicios, o sistemas de aplicación después de ser autenticado solo una vez. El proceso requiere que el usuario inicie sesión por medio de un portal solo una vez al comienzo, y luego durante la sesión el sistema SSO le proporciona el acceso a los diferentes servicios, recurso o aplicaciones del sistema que se encuentran asignados.

- **OpenSource:** o código abierto, es la expresión con la que se conoce al software distribuido y desarrollado libremente. Es un movimiento más pragmático, se enfoca más en los beneficios prácticos como acceso al código fuente que en aspectos éticos o de libertad que son tan relevantes en el Software Libre.

1.4. Selección de los Sistemas Web

Según los resultados obtenidos mediante el análisis de las características establecidas para definir cada aplicación Web se establecen los siguientes puntos, descritos en la TABLA A16. II.

TABLA A16. II.
TABLA COMPARATIVA DE LAS APLICACIONES WEB

			Aplicaciones Web												
			Moodle	GitLab	CodeShip	WordPress	OpenMaker	ElasticBox	Drupal	Jasmine	Jenkins	Dropbox	Heroku	Azure	AWS
Características del Sistema Web	Lenguaje de desarrollo	PHP	1	0	0	1	0	0	1	0	0	0	0	0	0
		Python	0	0	0	0	0	0	0	0	0	1	0	0	0
		Java	0	0	0	0	0	0	0	0	1	0	0	0	0
		Otros(Ruby, .Net)	0	1	1	0	1	1	0	1	0	0	1	1	1
	Enfoque DevOps		1	1	1	1	1	1	1	1	1	1	1	1	1
	Protocolo de Autenticación CAS		1	1	0	1	0	0	1	0	1	0	0	0	0
	Single Sign On SSO		1	1	0	1	0	0	1	0	1	1	1	1	1
	OpenSource		1	1	0	1	0	0	1	0	1	0	0	0	0
	Sistema de directorio LDAP		1	1	1	1	0	1	1	1	1	1	1	1	1
Resultado Final			6	6	3	6	2	3	6	3	6	4	4	4	4

MOODLE

Se puede determinar que Moodle de entre todas las características establecidas, cumple con cada una de ellas, haciendo uso de las características de mayor importancia como son: el Protocolo de Autenticación CAS, licencia libre y el directorio centralizado LDAP como se puede observar en la TABLA A16. II. por lo que se lo considera como un Sistema Web apto para la integración con el Servicio de Autenticación Central.

GITLAB

Se puede determinar que GitLab de entre todas las características establecidas, cumple con cada una de ellas, haciendo uso de las características de mayor importancia como son: el Protocolo de Autenticación CAS, licencia libre y el directorio centralizado LDAP como se puede observar en la TABLA A16. II. por lo que se lo considera como un Sistema Web apto para la integración con el Servicio de Autenticación Central.

CODESHIP

Se puede determinar que CodeShip de entre todas las características establecidas, cumple con 3 de ellas, haciendo uso solamente de una de las características de mayor importancia como es: el directorio centralizado LDAP, faltando por cumplir el Protocolo de Autenticación CAS y licencia libre, como se puede observar en la TABLA A16. II. por lo que se lo considera como un Sistema Web no apto para la integración con el Servicio de Autenticación Central.

WORDPRESS

Se puede determinar que WordPress de entre todas las características establecidas, cumple con cada una de ellas, haciendo uso de las características de mayor importancia como son: el Protocolo de Autenticación CAS, licencia libre y el directorio centralizado LDAP como se puede observar en la TABLA A16. II. por lo que se lo considera como un Sistema Web apto para la integración con el Servicio de Autenticación Central.

OPENMAKER

Se puede determinar que OpenMaker de entre todas las características establecidas, cumple con 2 de ellas, sin hacer uso de ninguna de las características de mayor importancia como es: el Protocolo de Autenticación CAS, licencia libre y el directorio centralizado LDAP, como se puede observar en la TABLA A16. II. por lo que se lo considera como un Sistema Web no apto para la integración con el Servicio de Autenticación Central.

ELASTICBOX

Se puede determinar que ElasticBox de entre todas las características establecidas, cumple con 3 de ellas, haciendo uso solamente de una de las características de mayor

importancia como es: el directorio centralizado LDAP, faltando por cumplir el Protocolo de Autenticación CAS y licencia libre, como se puede observar en la TABLA A16. II. por lo que se lo considera como un Sistema Web no apto para la integración con el Servicio de Autenticación Central.

DRUPAL

Se puede determinar que Drupal de entre todas las características establecidas, cumple con cada una de ellas, haciendo uso de las características de mayor importancia como son: el Protocolo de Autenticación CAS, licencia libre y el directorio centralizado LDAP como se puede observar en la TABLA A16. II. por lo que se lo considera como un Sistema Web apto para la integración con el Servicio de Autenticación Central.

JASMINE

Se puede determinar que Jasmine de entre todas las características establecidas, cumple con 3 de ellas, haciendo uso solamente de una de las características de mayor importancia como es: el directorio centralizado LDAP, faltando por cumplir el Protocolo de Autenticación CAS y licencia libre, como se puede observar en la TABLA A16. II. por lo que se lo considera como un Sistema Web no apto para la integración con el Servicio de Autenticación Central.

JENKINS

Se puede determinar que Jenkins de entre todas las características establecidas, cumple con cada una de ellas, haciendo uso de las características de mayor importancia como son: el Protocolo de Autenticación CAS, licencia libre y el directorio centralizado LDAP como se puede observar en la TABLA A16. II. por lo que se lo considera como un Sistema Web apto para la integración con el Servicio de Autenticación Central.

DROPBOX

Se puede determinar que Dropbox de entre todas las características establecidas, cumple con 3 de ellas, haciendo uso de 2 de las características de mayor importancia como es: licencia libre y el directorio centralizado LDAP, faltando por cumplir el Protocolo de Autenticación CAS, como se puede observar en la TABLA A16. II. por lo que se lo considera como un Sistema Web no apto para la integración con el Servicio de Autenticación Central.

HEROKU

Se puede determinar que Heroku de entre todas las características establecidas, cumple con 3 de ellas, haciendo uso solamente de una de las características de mayor importancia como es: el directorio centralizado LDAP, faltando por cumplir el Protocolo de Autenticación CAS y licencia libre, como se puede observar en la TABLA A16. II. por lo que se lo considera como un Sistema Web no apto para la integración con el Servicio de Autenticación Central.

AZURE

Se puede determinar que Azure de entre todas las características establecidas, cumple con 3 de ellas, haciendo uso solamente de una de las características de mayor importancia como es: el directorio centralizado LDAP, faltando por cumplir el Protocolo de Autenticación CAS y licencia libre, como se puede observar en la TABLA A16. II. por lo que se lo considera como un Sistema Web no apto para la integración con el Servicio de Autenticación Central.

AWS

Se puede determinar que AWS de entre todas las características establecidas, cumple con 3 de ellas, haciendo uso solamente de una de las características de mayor importancia como es: el directorio centralizado LDAP, faltando por cumplir el Protocolo de Autenticación CAS y licencia libre, como se puede observar en la TABLA A16. II. por lo que se lo considera como un Sistema Web no apto para la integración con el Servicio de Autenticación Central.

Figura A16. 1. Resultado de la tabla comparativa de las Aplicaciones Web

Mediante el análisis de todas las Aplicaciones Web, podemos concluir las aplicaciones que cumplen con todas las características definidas en el punto anterior son: Moodle, GitLab, WordPres, Drupal y Jenkins, todos estos obteniendo un puntaje de 6pts. Por lo que se los considera como sistemas aptos para la integración con el Sistema de Gestión único CAS (SiGUCAS), en la elaboración del tema de titulación TT.

2. Desarrollo

Descripción de los sistemas, servidores y protocolos a utilizar:

- SAC, es un sistema Web desarrollado en el lenguaje de programación PHP, donde el administrador se conecta directamente con el servidor OpenLdap el cual es un directorio centralizado, este nos permite controlar todas sus funciones para la gestión de los usuarios, grupos. Para los usuarios, que no tiene el perfil de administrador les permite cambiar y recuperar su contraseña, mediante el Sistema Jasig CAS.
- Moodle, es una plataforma de aprendizaje diseñada para educadores, administradores y estudiantes. Proporcionándoles un sistema integrado único, robusto y seguro para crear ambientes de aprendizaje personalizados, ideal para la integración con el protocolo CAS gracias a que tiene implementado un plugin que permite el inicio de sesión único (SSO).
- Drupal, es un sistema de gestión de contenidos (CMS) que se utiliza para crear sitios Web dinámicos y con gran variedad de funcionalidades de software libre, que cuenta con una amplia y activa comunidad de usuarios y desarrolladores que colaboran conjuntamente en su mejora y ampliación, ideal para la integración con el protocolo CAS gracias a que tiene implementado un módulo que permite el inicio de sesión único (SSO.
- Wordpress, sistema de gestión de contenidos (CMS) que permite crear y mantener un blog u otro tipo de Web, ideal para la integración con el protocolo CAS gracias a que tiene implementado un plugin que permite el inicio de sesión único (SSO).
- Jenkins, es un software de integración continúa escrito en Java. Se trata de una herramienta muy poderosa con la que podremos realizar pruebas de una aplicación para así detectar fallos cuanto antes, ideal para la integración con el protocolo CAS gracias a que tiene implementado un plugin que permite el inicio de sesión único (SSO).
- GitLab, es un servicio que ofrece alojamiento de repositorios con varias funciones de seguimientos de problemas, entre otras características extra,

ideal para la integración con el protocolo CAS gracias a que tiene implementado un plugin que permite el inicio de sesión único (SSO).

3. Construcción

3.1. Diseño del prototipo para el servicio de Autenticación Central de Usuarios en Aplicaciones Web.

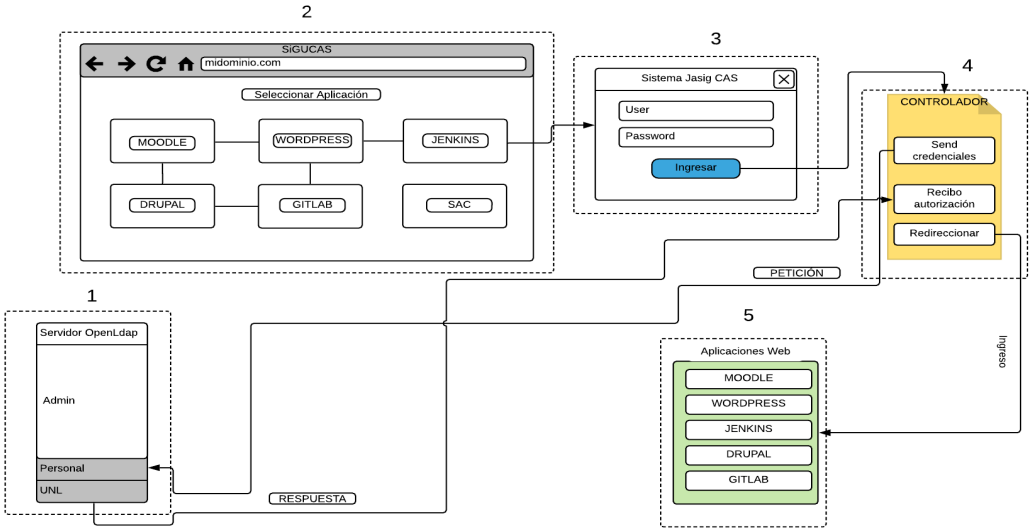


Figura A16. 2. Diagrama del Sistema SiGUCAS

En el diagrama propuesto nos permite hacer uso del Servicio de Autenticación Central CAS, utilizando el Sistema Jasig CAS, para el acceso a distintos sistemas y el uso de un directorio centralizado donde se almacenará la información de cada usuario que utilice los sistemas integrados. A continuación, se detallan cada uno de los módulos presentados en la (Figura A16. 2).

Interfaz principal sistema SiGUCAS

Se desarrolló una interfaz gráfica que contiene el acceso a los sistemas que se integraron con el protocolo CAS y el sistema de administración SAC, que el usuario maneja habitualmente. Fue desarrollada en el lenguaje de etiquetado HTML y el framework BootStrap los cuales tienen gran impacto en el desarrollo de páginas.

El objetivo de esta interfaz es permitirle al usuario redireccionarse al login centralizado de los sistemas o al login de administración de SAC.

Login Jasig CAS

Se configuró el login centralizado Jasig CAS para los sistemas seleccionados, el cual es el más utilizado para la integración de inicio de sesión único del Servicio de Autenticación Central. Donde le permite al usuario ingresar sus credenciales de autenticación (usuario y contraseña).

El objetivo de este login es enviar las credenciales ingresadas por el usuario a un controlador configurado para hacer uso del servidor OpenLdap y permita validar si la información es correcta y dar acceso al sistema antes seleccionado.

Controlador

Se implementó todos los métodos necesarios para la autenticación y gestión del servidor OpenLDAP, los mismos que se encuentran desarrollados como Servicio Web para ser utilizados por otros sistemas o aplicaciones Web, los cuales de encuentran desarrollados en el Lenguaje de Programación PHP.

El objetivo del Servicio Web es contener los metodos administrativos para el servidor OpenLDAP, para que otras aplicaciones os sitemas web puedan utilizarlos sin la necesidad de programacion extra.

Servidor OpenLDAP

Se implementó y se configuró para el almacenamiento de las cuentas de los usuarios que van hacer uso de las diferentes aplicaciones Web.

El objetivo principal es contener las credenciales de acceso para que el controlador descrito en el punto anterior pueda autenticar y gestionar a los mismos.

Sistema Seleccionado

Es el sistema que el usuario selecciona en la interfaz principal, el cual solicita una petición de acceso al sistema Jasig CAS, si las credenciales son ingresadas correctamente se generará un ticket seguro que el sistema a solicitado, permitiendo al usuario acceder a los recursos del sistema antes seleccionado.

4. Pruebas

4.1. Características que se probarán

- Fluidez de datos en la consulta
- Inicio de sesión único
- Cierre de sesión único
- Interoperabilidad de los sistemas seleccionados
- Interfaz del sistema SiGUCAS
- Administración del servidor OpenLDAP a través de SAC

4.2. Características que no deben probarse

- Errores relacionados con el tiempo.
- Condiciones de error no detectadas.
- Condiciones especiales de los datos.
- Invalidez de la información mostrada por pantalla.
- Fallos de configuración/compatibilidad con software
- Incapacidad de soportar el volumen de carga o fallos hard.

4.3. Enfoque

- **Prueba Funcional:** Esta prueba busca que SiGUCAS funcionen correctamente, se busca fallos en la ejecución de la prueba de concepto.
- **Prueba Basada en Escenarios:** Consiste en validar la aceptabilidad de SiGUCAS.

4.4. Criterios de aprobación / rechazo de elementos

Pruebas funcionales: Se espera un porcentaje mayor al 70% de efectividad de la interoperabilidad de los sistemas al momento de iniciar sesión.

4.5. Necesidades ambientales

Software y Hardware:

- Sistema operativo Windows, IOS o Linux.
- Acceso a la red interna
- Un computador con requerimiento mínimo de un microprocesador 486
- Un smartphone con sistema operativo Android 4.4 o superior

Documentación

- Absoluta comodidad, tranquilidad.
- Hoja de respuestas o aceptabilidad

4.6. Responsabilidades

- Pruebas de Documentación: Directo del TT.
- Pruebas de funcionalidad y aceptabilidad: Personal de la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja y Directo del TT.
- Pruebas de Software: Personal de la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja y autores del TT.

4.7. Resultados de la prueba

- Informe de aceptabilidad de SiGUCAS.
- Recomendaciones y observaciones de la población que utilizo SiGUCAS.

4.8. Tareas de la prueba

Tareas del Sistema SiGUCAS						
Nro.	Tarea	Aprobación		Calificación (1 - 5)	Recomendaciones	Observaciones
		SI	NO			
1	El administrador del Sistema SAC, le creó una cuenta para el acceso a las aplicaciones Web.					
2	Verifique en su correo electrónico, la creación de una cuenta y credenciales de acceso.					
3	Ingrese a la interfaz principal del Sistema SiGUCAS.					
4	Seleccione el sistema SAC, para realizar el cambio de su contraseña (Opcional).					
5	Seleccione un sistema (Drupal, Moodle, Wordpress, Jenkins, GitLab y Jasig CAS), para verificar el acceso al login principal.					
6	Inicie sesión con las credenciales enviadas a su correo electrónico, para ingresar al sistema seleccionado.					
7	Verifique el inicio de sesión en el sistema Moodle, sin ingresar las credenciales.					
8	Verifique el inicio de sesión en el sistema Drupal, sin ingresar las credenciales.					
9	Verifique el inicio de sesión en el sistema WordPress, sin ingresar las credenciales.					
10	Verifique el inicio de sesión en el sistema Jenkins, sin ingresar las credenciales.					

11	Verifique el inicio de sesión en el sistema GitLab, sin ingresar las credenciales.					
12	Verifique el inicio de sesión en el sistema de acceso Jasig CAS, sin ingresar las credenciales.					
13	¿Se le creó una cuenta, en cada uno de los sistemas (Drupal, Moodle, Wordpress, Jenkins y GitLab), al momento de iniciar sesión?					
14	Ingrese con credenciales de administrador al sistema Jenkins, mediante Jasig CAS.					
15	Ingrese con credenciales de administrador al sistema WordPress, mediante Jasig CAS.					
16	Ingrese con credenciales de administrador al sistema Moodle, mediante Jasig CAS.					
17	Ingrese con credenciales de administrador al sistema Drupal mediante su propia interfaz de autenticación, esto debido a la arquitectura del sistema.					
18	Ingrese con credenciales de administrador al sistema GitLab mediante su propia interfaz de autenticación, esto debido a la arquitectura del sistema.					
19	Cierre sesión en uno de los sistemas (Drupal, Moodle, Wordpress, Jenkins, GitLab y Acceso Jasig CAS).					

20	Verifique el cierre de sesión único en Jasig CAS					
21	Cierre sesión en el sistema Jenkins.					
22	Cierre sesión en el sistema Drupal.					
23	Cierre sesión en el sistema Moodle.					
24	Cierre sesión en el sistema WordPress.					
25	Cierre sesión en el sistema GitLab.					

4.9. Roles

TABLA A16. III.
ROLES PARA LAS PRUEBAS DEL TT

Roles:	Asignado A:	Total
Programadores:	- Wilmer Antonio Aguilar Soto - Manuel Stalin Armijos Ordóñez	2
Pruebas de Documentación:	- Director del proyecto de titulación	1
Pruebas de Funcionalidad y Aceptación:	- Director del proyecto de titulación - Personal de la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja.	7
Pruebas de Software:	- Director del proyecto de titulación - Personal de la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja.	7

4.10. Resultados de las Tareas de SiGUCAS

Los resultados de las tareas de SiGUCAS, se los obtuvo mediante una encuesta de aceptación y funcionalidad las cuales se dettalan más adelante.

Funcionales: Para validar la correcta funcionalidad de SIGUCAS, se empleó un método de calificación que va desde 1 a 5, como se describe en la TABLA A16. IV. donde se puede especificar si el sistema es aceptable.

TABLA A16. IV.
MÉTODO DE CALIFICACIÓN PARA LAS PRUEBAS FUNCIONALES DE SIGUCAS

CALIFICACIÓN	VALOR
MALO	1
REGULAR	2
BUENO	3
MUY BUENO	4
EXCELENTE	5

Las tareas definidas para el ambiente de pruebas se establecieron en 25 preguntas, de las cuales solo una de ellas obtuvo una calificación de 4 y las restantes obtuvieron

una calificación de 5 que se puede observar en la TABLA A16. V. que describe la calificación de los 7 responsables encargados de la misma.

TABLA A16. V.
RESULTADO DE LA CALIFICACIÓN DEL SIGUCAS

	Calificación							
	Ing. Jhon Calderon	Ing. Danny Muñoz	Ing. Pablo Ramón	Ing. José Martínez	Ing. Lissette López	Ing. Carlos Riofrío	Ing. Edison Coronel	Total
Pregunta 1	5	5	5	5	5	5	5	35
Pregunta 2	5	5	5	5	5	5	5	35
Pregunta 3	5	5	5	5	5	5	5	35
Pregunta 4	5	5	5	5	5	5	5	35
Pregunta 5	5	5	5	5	5	5	5	35
Pregunta 6	5	5	5	5	5	5	5	35
Pregunta 7	5	5	5	5	5	5	5	35
Pregunta 8	5	5	5	5	5	5	5	35
Pregunta 9	5	5	5	5	5	5	5	35
Pregunta 10	5	5	5	5	5	5	5	35
Pregunta 11	5	5	5	5	5	5	5	35
Pregunta 12	4	5	5	5	5	5	5	34
Pregunta 13	5	5	5	5	5	5	5	35
Pregunta 14	5	5	5	5	5	5	5	35
Pregunta 15	5	5	5	5	5	5	5	35
Pregunta 16	5	5	5	5	5	5	5	35
Pregunta 17	5	5	5	5	5	5	5	35
Pregunta 18	5	5	5	5	5	5	5	35
Pregunta 19	5	5	5	5	5	5	5	35
Pregunta 20	5	5	5	5	5	5	5	35
Pregunta 21	5	5	5	5	5	5	5	35
Pregunta 22	5	5	5	5	5	5	5	35
Pregunta 23	5	5	5	5	5	5	5	35
Pregunta 24	5	5	5	5	5	5	5	35
Pregunta 25	5	5	5	5	5	5	5	35

Aceptación: Para validar si SIGUCAS es aceptable, se empleó un método de calificación de 0 y 1, como se describe en la TABLA A16. VI. donde se puede especificar si el sistema es aceptable, es decir; cumple con la tarea especificada, o no aceptable, es decir; no cumple o no realiza una tarea especificada.

TABLA A16. VI.
MÉTODO DE CALIFICACIÓN PARA LAS PRUEBAS DE ACEPTACIÓN DE SIGUCAS

MÉTODO DE APROBACIÓN DEL SISTEMA SiGUCAS.	
CALIFICACIÓN	**VALOR**
SI	1
NO	0

TABLA A16. VII.
RESULTADOS DE ACEPTACIÓN DE SIGUCAS

	Aprobación						
	Ing. Jhon Calderon	Ing. Danny Muñoz	Ing. Pablo Ramón	Ing. José Martínez	Ing. Lissette López	Ing. Carlos Riofrío	Ing. Edison Coronel
Pregunta 1	Si	Si	Si	Si	Si	Si	Si
Pregunta 2	Si	Si	Si	Si	Si	Si	Si
Pregunta 3	Si	Si	Si	Si	Si	Si	Si
Pregunta 4	Si	Si	Si	Si	Si	Si	Si
Pregunta 5	Si	Si	Si	Si	Si	Si	Si
Pregunta 6	Si	Si	Si	Si	Si	Si	Si
Pregunta 7	Si	Si	Si	Si	Si	Si	Si
Pregunta 8	Si	Si	Si	Si	Si	Si	Si
Pregunta 9	Si	Si	Si	Si	Si	Si	Si
Pregunta 10	Si	Si	Si	Si	Si	Si	Si
Pregunta 11	Si	Si	Si	Si	Si	Si	Si
Pregunta 12	Si	Si	Si	Si	Si	Si	Si
Pregunta 13	Si	Si	Si	Si	Si	Si	Si
Pregunta 14	Si	Si	Si	Si	Si	Si	Si
Pregunta 15	Si	Si	Si	Si	Si	Si	Si
Pregunta 16	Si	Si	Si	Si	Si	Si	Si
Pregunta 17	Si	Si	Si	Si	Si	Si	Si
Pregunta 18	Si	Si	Si	Si	Si	Si	Si
Pregunta 19	Si	Si	Si	Si	Si	Si	Si

Pregunta 20	Si	Si	Si	Si	Si	Si	Si
Pregunta 21	Si	Si	Si	Si	Si	Si	Si
Pregunta 22	Si	Si	Si	Si	Si	Si	Si
Pregunta 23	Si	Si	Si	Si	Si	Si	Si
Pregunta 24	Si	Si	Si	Si	Si	Si	Si
Pregunta 25	Si	Si	Si	Si	Si	Si	Si
Total	25	25	25	25	25	25	25

Las tareas definidas para el ambiente de pruebas se establecieron en 25 preguntas, de las cuales todas y cada una de ellas fueron aprobadas correctamente como se puede observar en la TABLA A16. VII.

4.11. Aprobaciones

El prototipo propuesto en el presente TT, fue aprobado por personal de la UTI de la UNL, evidenciando los participantes en el (Anexo 2. Listado de participantes para las pruebas del presente TT.), siguiendo el formato definido en la TABLA A16.VIII.

TABLA A16. VIII.
PARTICIPANTES DEL AMBIENTE DE PRUEBAS EN EL PRESENTE TT.

Población:	**Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja**			
Nombre:	Antonio Aguilar, Manuel Armijos	**Rol:**	Supervisores del presente TT	**FIRMA**
Nombre:	Datos del Usuario	**Rol:**	Administrador del sistema SiGUCAS	
Nombre:	Datos del Usuario	**Rol:**	Usuario Común	
Nombre:	Datos del Usuario	**Rol:**	Usuario Común	

5. Despliegue

El presente TT fue implementado y validado en la UTI de la UNL.

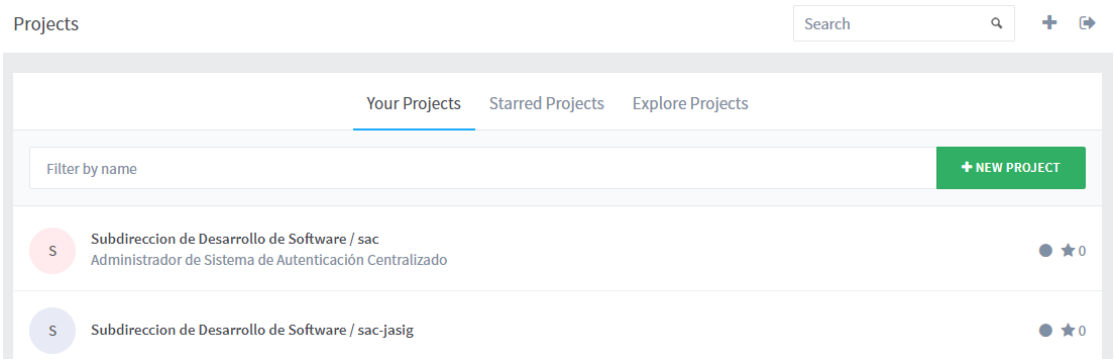


Figura A16. 6. Despliegue de SAC y SiGUCAS

6. Monitoreo

Con la etapa de monitoreo del ciclo DevOps, nos permitió la corrección de errores presentadas por la comunidad universitaria, a través de correos electrónicos, visitas técnicas y medios de comunicación digitales, como podemos observar en la Figura 207. la cual nos indica las necesidades que se tomo en cuenta para la correcta implantación del presente TT en la UNL.

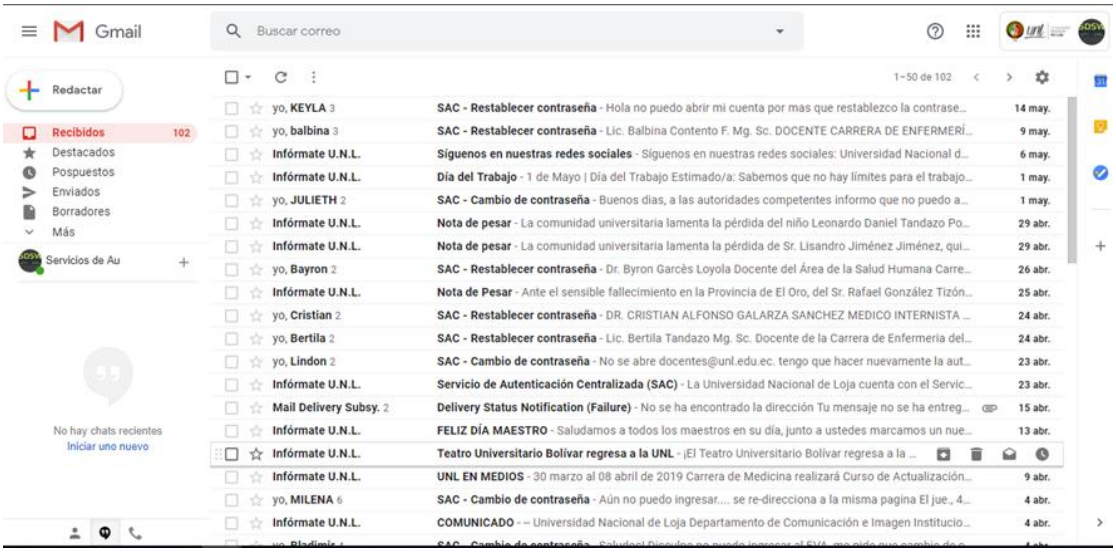


Figura A16. 722. Monitoreo de la implantación del presente TT en la UNL.

Anexo 17. Configuración de la Aplicación Web Moodle

a) Instalación de Moodle y su base de datos

Paso 1

Descargamos una versión estable de Moodle en nuestro caso se utilizó la última versión disponible en el siguiente enlace:

https://download.moodle.org/

Paso 2

Descomprimimos el **archivomoodle-3.x.x.tgz** que descargamos y lo copiamos con cualquier nombre en la ruta de nuestro servidor local de aplicaciones en nuestro caso en la carpeta **htdocs** del servidor Xampp.

Debemos encontrarnos en la ruta donde se descargó nuestro archivo.

sudo rm archivomoodle-3.x.x moodle

sudo cp –a moodle /opt/lampp/htdocs

Paso 3

Otorgamos los permisos necesarios para el acceso al archivo con el siguiente comando.

sudo chmod 777 /opt/lampp/htdocs/moodle

Paso 4

Ingresamos a nuestro navegador y colocamos la ruta del servidor local seguido del nombre de nuestro archivo moodle posteriormente se presentará la pantalla de instalación para lo cual seleccionaremos el idioma español.

http://localhost/moodle

Paso 5

Confirmamos la ruta de acceso Web a moodle, la ruta del directorio del código y la ruta de directorio de datos donde moodle podrá guardar archivos subidos, las mismas que ya están definidas por defecto.

Figura A17. 1. Rutas para los directorios de datos de Moodle

Si se nos presenta algún error, debemos definir otra ruta o dar permisos a la carpeta que nos indica, por ejemplo:

sudo chmod 777 –R /opt/lampp/moodledata

Paso 6

Seleccionamos la base de datos MariaDB, soportada por Moodle que use manejadores estándar mysqli.

Paso 7

Mediante **localhost phpmyadmin**, el cual nos brinda una interfaz gráfica creamos una base de datos **mysql** con el nombre: **moodle**.

Paso 8

Los campos que se presentan en la pantalla los configuramos de la siguiente manera:
Servidor de la base de datos: localhost si es un servidor local.
Nombre de la base de datos: moodle que definimos en el paso 7.
Usuario de la base de datos: root si es el propietario.
Contraseña de la base de datos: Si se creó una contraseña ingresarla caso contrario dejar en blanco.
Prefijo de tablas: Mdl_ por defecto.

Paso 9

Aceptamos los términos y condiciones y continuamos con la instalación.

Paso 10

Se nos presenta una ventana con la comprobación de los requisitos necesarios para continuar con la instalación, le damos en continuar.

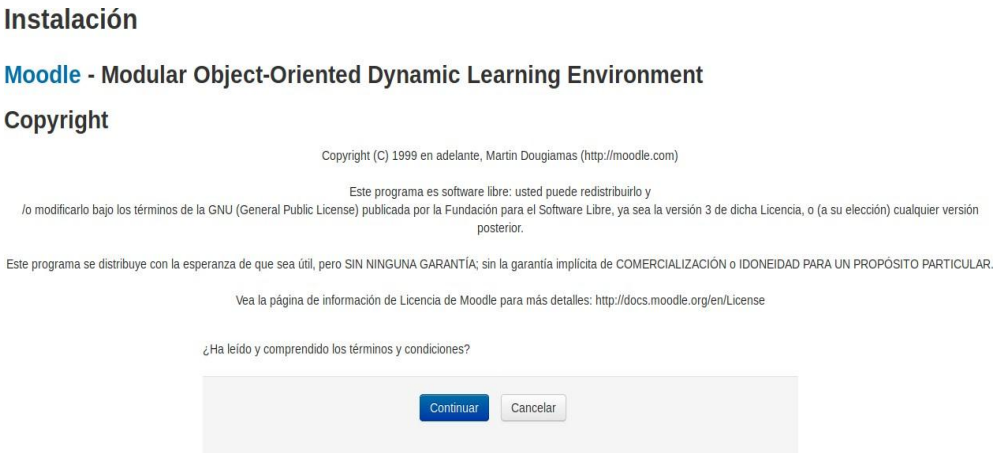


Figura A17. 2. Comprobación de requisitos necesarios en Moodle

Paso 11

Esperamos un momento y se nos presentara una ventana para Configurar una cuenta de administrador en la cual deberemos colocar un nombre de usuario, contraseña, nombre, apellido, correo y ciudad.

Paso 12

Ingresamos la información de la página principal como el nombre completo del sitio, un nombre corto y una descripción.

Hemos terminado con la instalación se nos presentará la página principal de administración.

b) Instalación del plugin de CAS en moodle

PASO 1

Como nos encontramos con perfil de administrador nos dirigimos a **Administración del sitio>>Extensiones>>Autenticación>>Gestionar la Autenticación** y damos clics en el icono del ojo que se encuentra frente al servidor CAS (SSO).

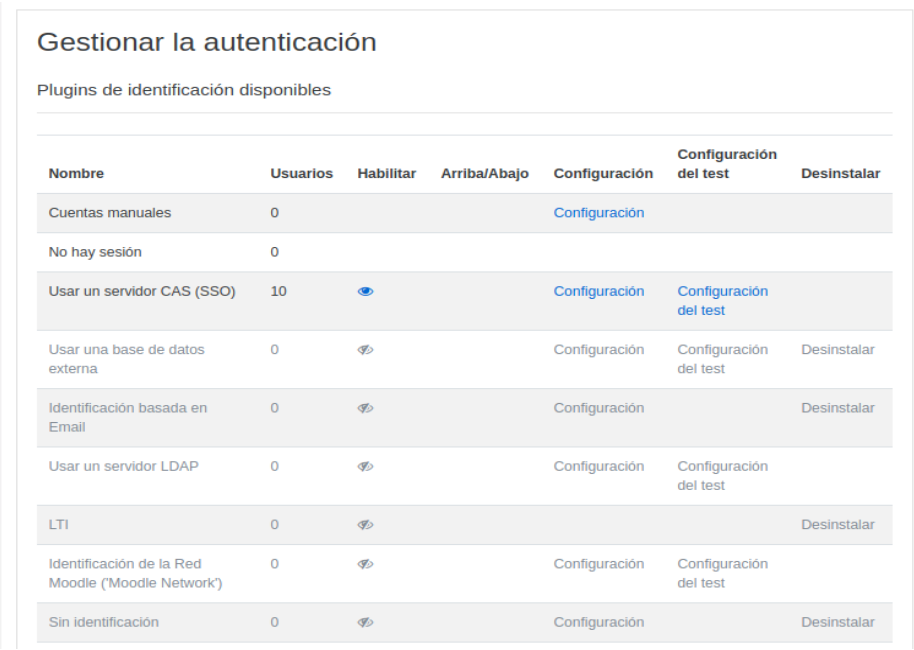


Figura A17. 3. Seleccionar Autenticación mediante CAS (SSO).

PASO 2

Nos dirigimos a **Administración del sitio>>Extensiones>>Autenticación>>Usar un servidor CAS (SSO)**, para proceder con la configuración del protocolo CAS.

Se configuran los siguientes parámetros necesarios para conectar Moodle con el protocolo CAS.

- **Nombre del Host:** Ruta del servidor CAS -> *https://localhost*
- **URI Base:** Instancia del servidor CAS -> *https://localhost/cas/*
- **Puerto:** Puerto seguro en donde se ejecuta CAS -> 8443
- **Versión CAS:** Versión utilizada del servidor CAS -> *CAS 2.0*
- **Idioma:** Idioma de preferencia -> *Español*
- **Opciones de salida del CAS:** Activamos Si para permitir el cierre de sesíon único.
- **Multi-identificación:** Seleccionamos que NO para dejar esteblecido que el único medio de acceso a Moodle será por medio del protocolo CAS.

Figura A17. 8. Ruta del Servicio CAS para el inicio de sesión único en Moodle

Una vez realizados los cambios los Guardamos luego salimos de Moodle y volvemos a ingresar nos aparecerá la página de inicio de sesión del sistema Jasig CAS.

PASO 3

Adicionalmente se puede hacer uso de un servidor LDAP para el mapeo de datos respectivamente a las credenciales de acceso.

Se configuran los siguientes parámetros necesarios para conectar Moodle con el protocolo CAS.

- **URI del host:** Ruta del servidor LDAP-> *ldap://ldapserver.cas.com*
- **Versión:** Versión utilizada del servidor LDAP -> *versión 3*
- **Usar TLS***: Activar si se cuenta con seguridad TLS.*
- **Codificación LDAP:** *Por defecto utf-8*
- **Nombre distinguido:** *Nombre del administrador de LDAP.*

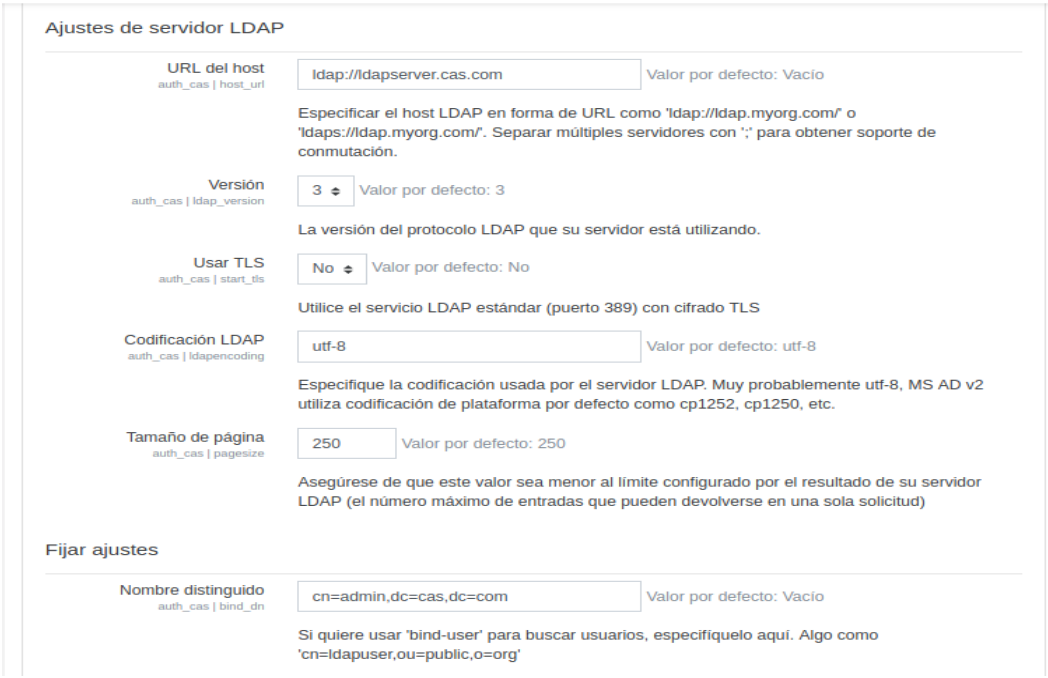


Figura A17. 9. Ruta del Servidor OpenLDAP para el Mapeo de datos en Moodle

Paso 4

Cerramos sesión e ingresamos de nuevo a la dirección que levantamos WordPress, debe presentarnos la interfaz de acceso de Jasig CAS; donde ingresamos un usuario que ya se encuentre registrado en nuestro servidor OpenLDAP y damos click a “LOGIN”.

Anexo 18. Configuración de la Aplicación Web GitLab

a) Instalación de Gitlab

Paso 1

Actualizamos nuestra lista de paquetes y nuestro sistema con los siguientes comandos.

sudo apt-get update && upgrade

Paso 2

Instalamos las dependencias necesarias que el sistema Gitlab necesita para su instalación con el siguiente comando.

sudo apt-get install -y curl openssh-server ca-certificates

Paso 3

Opcionalmente si deseamos enviar correos electrónicos de notificación instalamos Postfix con el siguiente comando.

sudo apt-get install -y postfix

Se nos presentará una pantalla descrita a continuación en donde seleccionaremos *Sitio de Internet.*

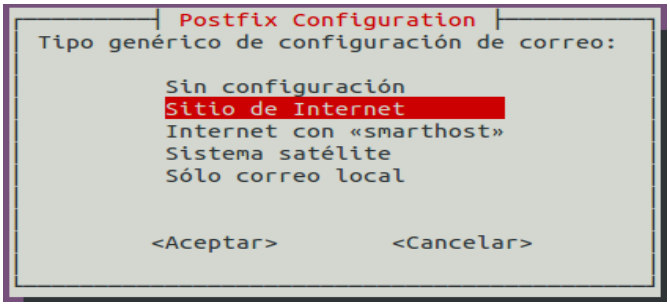


Figura A18. 1. Configuración de Postfix para el envío de notificaciones.

Y para finalizar ingresamos el DNS externo de nuestro servidor de correos electrónicos.

Paso 4

Agregamos e instalamos el repositorio de paquetes de Gitlab con el siguiente comando.

curl https://packages.gitlab.com/install/repositories/gitlab/gitlab-ee/script.deb.sh | sudo bash

Paso 5

Instalamos el paquete Gitlab para lo cual cambiaremos *EXTERNAR_URL* por la URL que deseamos ingresar a nuestro Gitlab, con el siguiente comando.

sudo EXTERNAL_URL="http://gitlab.ejemplo.com" apt-get install gitlab-ee

Si se desea ingresar con https:// se deberá especificar un certificado de seguridad.

Paso 6

Una vez finalizada la instalación del paso anterior ingresaremos a nuestro navegador favorito e ingresaremos la URL especificada en *EXTERNAR_URL.* Se nos presentará una pantalla para el cambio de contraseña del administrador.

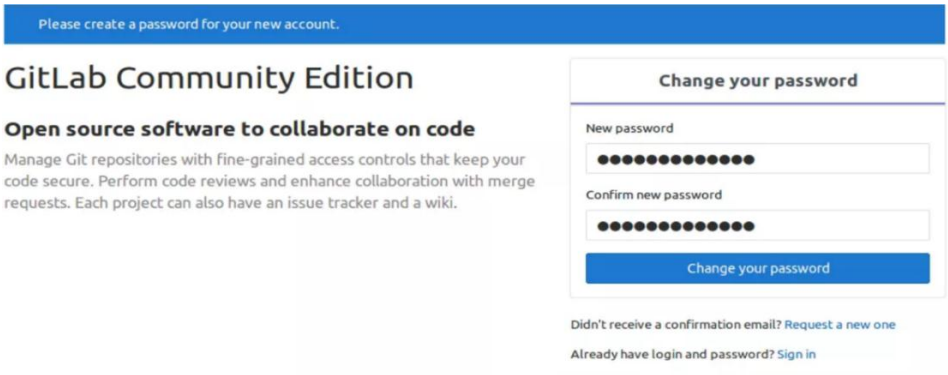


Figura A18. 2. Ventana para el cambio de contraseña del sistema Gitlab.

Paso 7

Una vez cambiada la contraseña se nos presentará una ventana para iniciar sesión en donde por defecto ingresaremos en Usuario root.

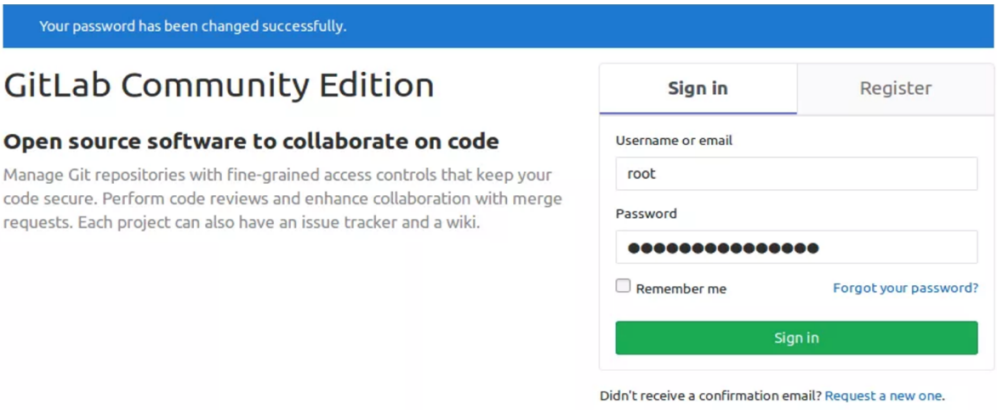


Figura A18. 3. Página principal de inicio de sesión de Gitlab.

Paso 8

Se nos presentará la interfaz principal del administrador.

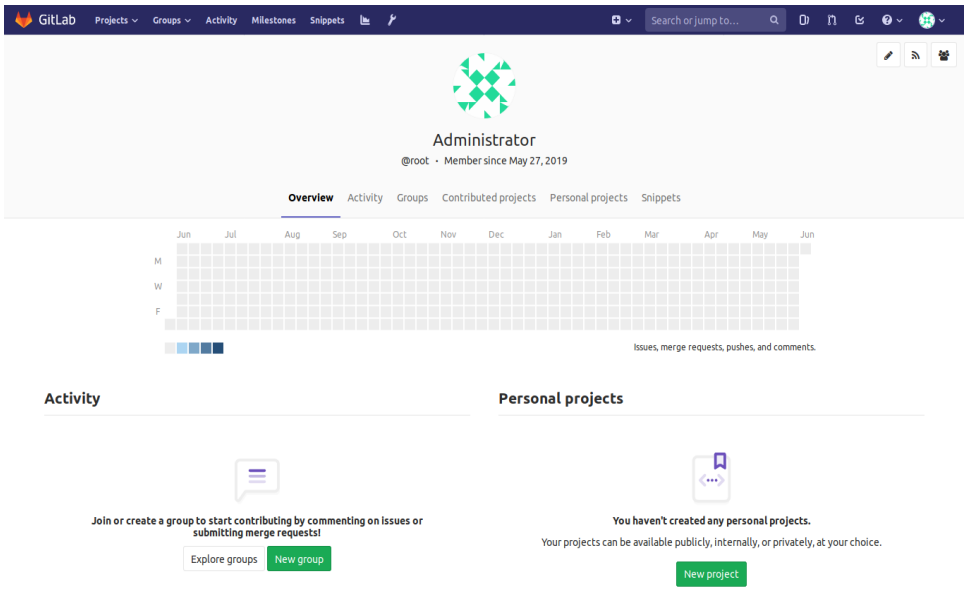


Figura A18. 4. Página principal del administrador del Gitlab.

b) Configuración de gitlab.rb para el protocolo CAS

Paso 1

Editamos el archivo gitlab.rb para configurar la conexión con el protocolo CAS con el siguiente comando.

sudo gedit /etc/gitlab/gitlab.rb

Paso 2

Descomentamos las líneas que se muestran en la Figura siguiente.

```
### OmniAuth Settings
###! Docs: https://docs.gitlab.com/ce/integration/omniauth.html
gitlab_rails['omniauth_enabled'] = true
gitlab_rails['omniauth_allow_single_sign_on'] = true
#gitlab_rails['omniauth_allow_single_sign_out'] = true
# gitlab_rails['omniauth_sync_email_from_provider'] = 'saml'
# gitlab_rails['omniauth_sync_profile_from_provider'] = ['saml']
# gitlab_rails['omniauth_sync_profile_attributes'] = ['email']
# gitlab_rails['omniauth_auto_sign_in_with_provider'] = 'saml'
 gitlab_rails['omniauth_block_auto_created_users'] = false
#gitlab_rails['omniauth_auto_link_ldap_user'] = true
# gitlab_rails['omniauth_auto_link_saml_user'] = false
gitlab_rails['omniauth_external_providers'] = ['cas3']
```

Figura A18. 5. Archivo de configuración para la conexión con el protocolo CAS.

Paso 3

Añadimos las líneas que se muestran en la Figura siguiente.

```
gitlab_rails['omniauth_providers'] = [
  {
        "name"=> "cas3",
        "label"=> "CAS_Server Login",
        "args"=> {
                "url"=> 'https://10.20.5.31:8443',
                "login_url"=> '/cas/login',
                "service_validate_url"=> '/cas/p3/serviceValidate',
                "logout_url"=> '/cas/logout',
                "disable_ssl_verification"=> true
        }
  }
]
```

Figura A18. 6. Código para establecer las variables de conexión con el servidor CAS.

- ***name***: Nombre del protocolo de autenticación, en Gitlab se lo nombra como cas3
- ***label:*** Nombre que aparecerá para link de redireccionamiento a CAS, en este caso se lo nombre *CAS_Server Login*
- ***url:*** Ruta del servidor en donde se encuentra levantado el protocolo CAS con su puerto seguro https://.
- ***login_url:*** URL de inicio de sesión de cas que será /cas/login
- ***service_validate_url:*** URL para validar un ticket generado
- ***logout_url:*** URL para el cierre de sesión de cas que será /cas/logout
- ***disable_ssl*** : Comando para validar un certificado de seguridad autofirmado

Paso 4

Ejecutamos el siguiente comando en la terminal para reconfigurar Gitlab.

sudo gitlab-ctl reconfigure

Paso 5

Ingresamos nuevamente a la URL ingresada el EXTERNAR_URL, donde nos presentará la interfaz de inició de sesión con la opción de autenticación CAS.

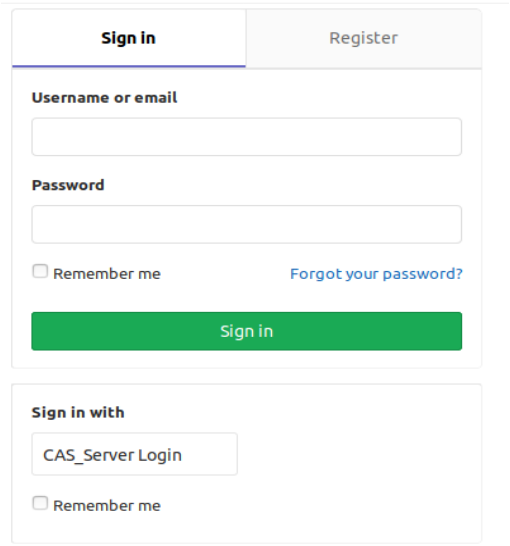


Figura A18. 7. Inicio de sesión del Gitlab, junto con la opción de inicio de sesión con el protocolo CAS.

Haciendo clic en CAS_Server Login ingresará a la página de inicio de sesión del sistema Jasig CAS.

Anexo 19. Configuración de la Aplicación Web WordPress

Instalación de WordPress y su base de datos

Paso 1

Descargamos una versión estable de Wordpress en nuestro caso se utilizó la versión 5.0.2 disponible en el siguiente enlace:

https://es.wordpress.org/download/

Paso 2

Descomprimimos el archivo wordpress-5.x.x-es_MX.tar.gz que descargamos y lo copiamos con cualquier nombre en la ruta de nuestro servidor local de aplicaciones en nuestro caso en la carpeta **htdocs** del servidor Xampp.

sudo rm wordpress-5.x.x-es_MX.tar.gz wordpress

sudo cp –a wordpress /opt/lampp/htdocs/

Paso 3

Otorgamos los permisos necesarios para el acceso al archivo con el siguiente comando.

sudo chmod 777 – R /opt/lampp/htdocs/wordpress

Paso 4

Ingresamos a nuestro navegador y colocamos la ruta del servidor local seguido del nombre de nuestro archivo wordpress,

http://localhost/wordpress

Figura A19. 1. Información para la base de datos de WordPress

Paso 5

Mediante **localhost phpmyadmin**, el cual nos brinda una interfaz gráfica creamos una base de datos **mysql** con el nombre: **wordpress**.

Paso 6

Los campos que se presentan en la pantalla los configuramos de la siguiente manera:

- Nombre de la base de datos: wordpress que definimos en el paso 5.
- Usuario de la base de datos: root si es el propietario.
- Contraseña de la base de datos: Si se creó una contraseña ingresarla caso contrario dejar en blanco.
- Servidor de la base de datos: localhost si es un servidor local.
- Prefijo de tablas: wp_ por defecto.

Paso 7

Creamos el archivo wp-config.php en la siguiente ruta /opt/lampp/htdocs/wordpress/ y copiamos el contenido de la ventana que se presenta a continuación y seleccionamos Ejecutar la instalación:

Figura A19. 2. Contenido del archivo wp-config.php

Paso 8

Establecemos un nombre del sitio, un nombre de usuario, una contraseña y un correo electrónico y seleccionamos Instalar Wordpress.

Si no se presentó ningún inconveniente en la instalación, nos presentara una ventana con el Nombre de usuario y Contraseña. Seleccionamos Acceder:

Paso 9

Para finalizar se nos aparecerá la página principal de acceso a WordPress, ingresamos las credenciales que creamos en paso 8 y seleccionamos Acceder. Dando como resultado a la ventana principal de administración de WordPress.

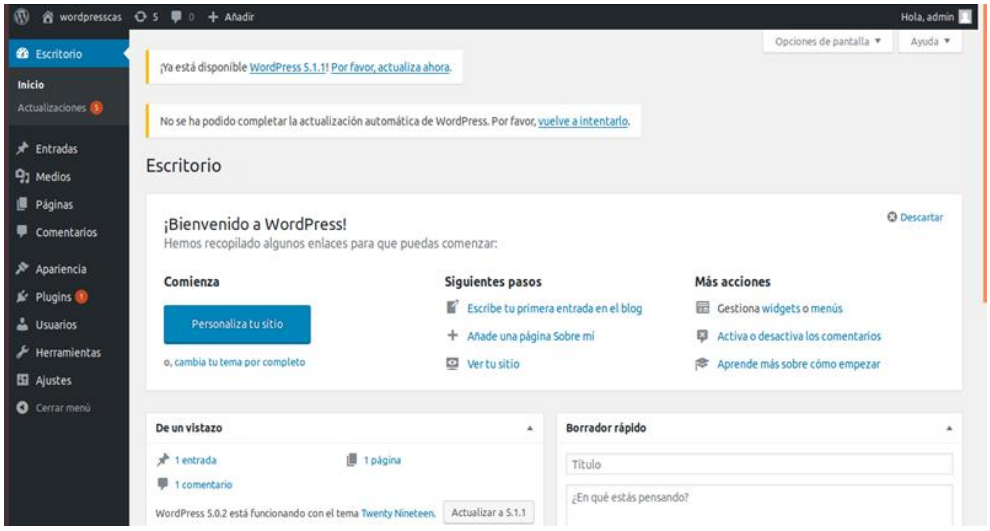


Figura A19. 3. Interfaz principal de WordPress

Instalación del plugin de CAS en WordPress

Paso 1

En la página principal del administrador de WordPres, vamos a configurar el plugin de CAS. Seleccionamos **Plugins>>Añadir Nuevo** y buscamos **WP Cassify** y seleccionamos Instalar ahora.

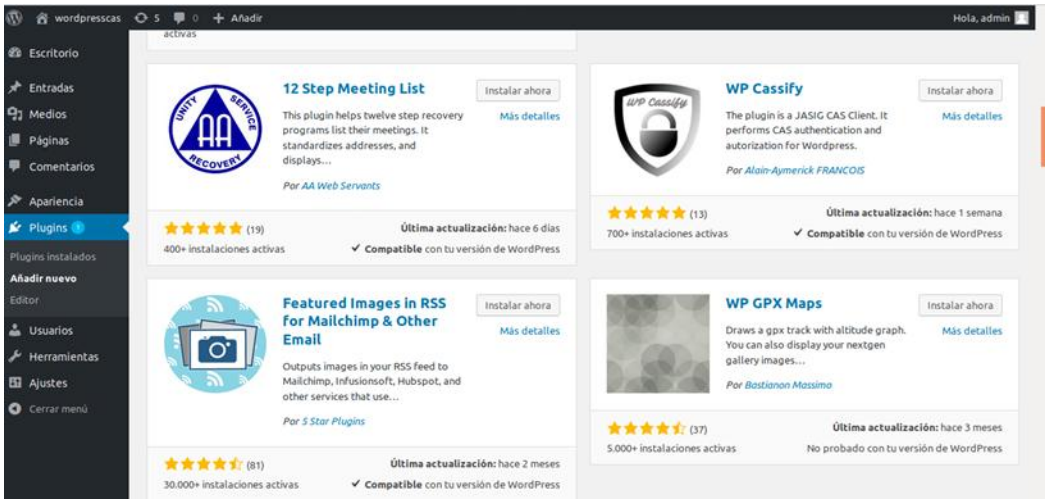


Figura A19. 4. Seleccionar el plugin WP Cassify

Paso 2

Comprobamos si el plugin se instaló correctamente ingresando a **Plugins>>Plugins instalados** y buscamos **WP Cassify**. Le damos click a Activar e ingresamos a la configuración del plugin instalado.

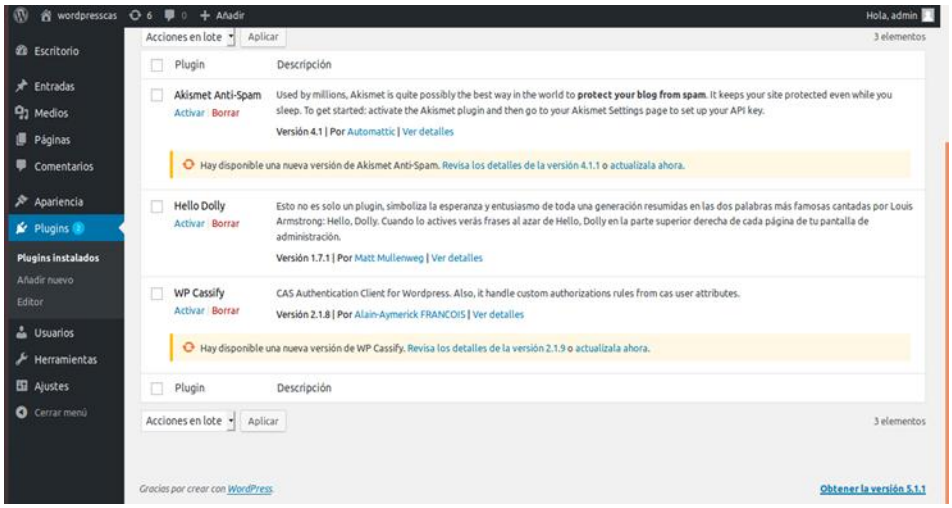


Figura A19. 5. Activación del plugin WP Cassify

Paso 3

Seleccionamos **Ajustes→WP Cassify** e ingresamos la información del servicio CAS para la autenticación centralizada y damos click en “Save options”.

- CAS Server base url: Ruta del servicio CAS, previamente levantado y definido en el (Anexo 15. Instalación Apache Tomcat).
- CAS Version protocol: 3
- Create user if not exist: SI

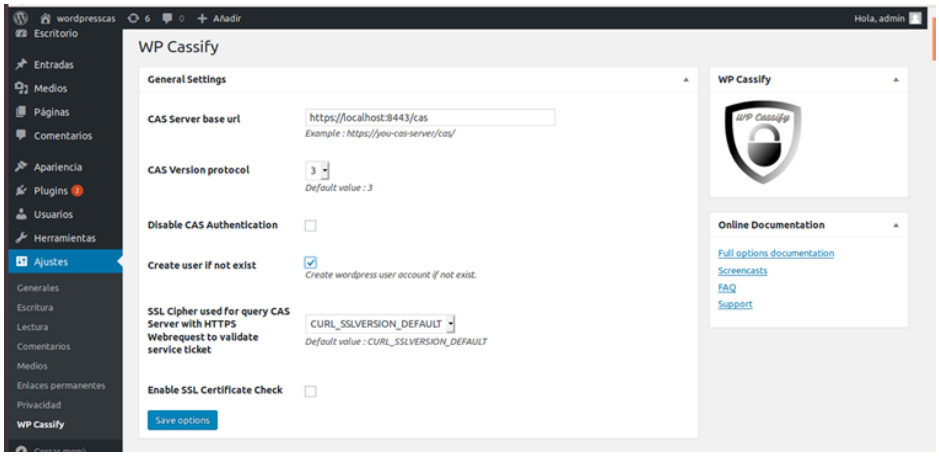


Figura A19. 6. Ruta del Servicio CAS para el inicio de sesión único en WordPress

- Service logout redirect url: Ruta del cierre de sesión del servicio CAS
- Name of the login servlet: Nombre de la ruta de inicio de sesión.
- Name of the logout servlet: Nombre de la ruta de cierre de sesión.

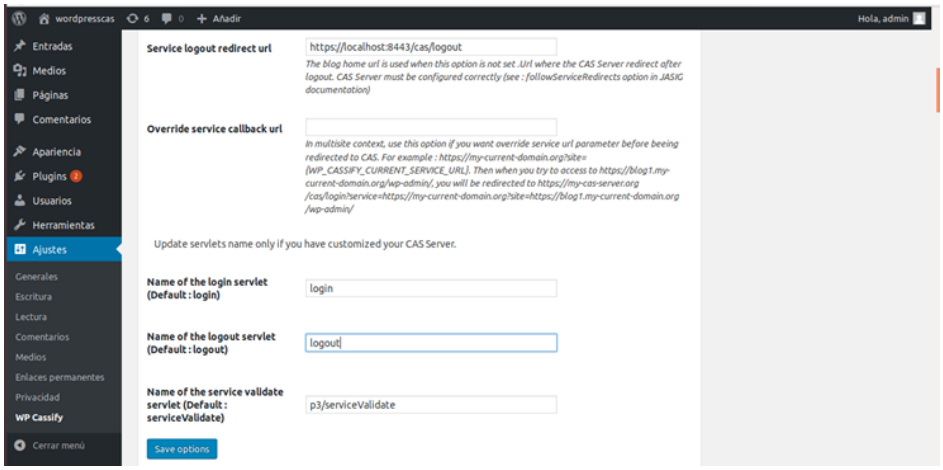


Figura A19. 7. Ruta del cierre de sesión en WordPress

Paso 4

Cerramos sesión e ingresamos de nuevo a la dirección que levantamos WordPress, debe presentarnos la interfaz de acceso de Jasig CAS; donde ingresamos un usuario que ya se encuentre registrado en nuestro servidor OpenLDAP y damos click a “LOGIN”.

Anexo 20. Configuración de la Aplicación Web Drupal

Para la configuración del sistema Web Drupal, es necesario tener instalado correctamente el servidor apache, debido a que utilizaremos su base de datos local. A continuación, se realizan los siguientes pasos para la correcta configuración del mismo:

a) Instalación Drupal y su base de datos

Paso 1

Actualizar la lista de paquetes disponibles y sus versiones, ingresando el siguiente comando en la terminal:

sudo apt-get update

Paso 2

Se debe ingresar a la plataforma mysql, para trabajar sobre la base de datos que utilizaremos para Drupal, ingresar el siguiente comando a la terminal:

mysql -u root –p

En este proceso nos solicita que ingresemos la contraseña del usuario root para acceder a mysql.

Paso 3

En este paso crearemos la base de datos con la cual vamos a trabajar, ingresar el siguiente comando a la terminal:

CREATE DATABASE nombre_base_de_datos;

Donde “nombre_base_de_datos”, es el nombre que nosotros deseamos asignar. Una vez creada nuestra base de datos, ingresamos el siguiente comando a la terminal, donde hacemos énfasis que utilizaremos esa base de datos:

use nombre_base_de_datos;

Donde “nombre_base_de_datos”, es el nombre que nosotros deseamos asiganar.

Luego de esto es necesario crear una cuenta de usuario para la base de datos de drupal, ingresamos el siguiente comando en la terminal:

CREATE USER nombre_usuario@localhost IDENTIFIED BY ‘contraseña’;

Donde “contraseña” es la que nosotros deseemos aplicar al usuario “nombre_usuario”, y que, por supuesto debemos recordar y tener bien presente, para todos los procedimientos que vayamos a realizar con el mismo en la base de datos. Justamente para ello vamos a ejecutar un nuevo comando, que es el que habilita al usuario acceso a todas las características o funcionalidades:

GRANT ALL PRIVILEGES ON nombre_base_de_datos. TO nombre_usuario@localhost;*

Para que se consoliden estos permisos debemos grabarlos, ingresando el siguiente comando en la terminal:

FLUSH PRIVILEGES;

Para terminar este paso, procedemos a salir de la plataforma mysql, ingresando el siguiente comando:

Exit

Si se presenta algún inconveniente al ingresar a **mysql**, realizamos el Paso 3 mediante **localhost phpmyadmin**, el cual nos brinda una interfaz gráfica para realizar el proceso.

Paso 4

En este paso vamos a configurar el sitio Web que manejaremos con Drupal. A continuación, iremos a la página oficial de Drupal, para descargar e instalar drupal en su última versión:

https://www.drupal.org/download

Una vez descargado, procedemos a dirigirnos al sitio donde está el archivo, ingresando el siguiente comando:

cd Descargas/

Una vez en el sitio donde se encuentra nuestro archivo descargado, procedemos a descomprimirlo, ingresando el siguiente comando en la terminal:

sudo tar zxvf drupa-8.6.1.tar.gz

Luego de descomprimir el archivo, se le puede cambiar de nombre al mismo, ingresando el siguiente comando en la terminal:

sudo mv drupal-8.6.1 "nuevo_nombre"

Una vez desempaquetado el fichero y con su nuevo nombre, tendremos que copiar todo el archivo o directorio de Drupal al directorio donde tenemos lo archivos html queusa el servidor Web apache, en nuestro caso es el directorio **/opt/lampp/htdocs/**. Ingresando el siguiente comando:

sudo cp –rf ~/druapl8/ /opt/lampp/htdocs/

Ahora debemos cambiar el propietario de los ficheros para que sean www-data que es el usuario que habitualmente usa Apache, ingresando el siguiente comando:

sudo chown –R www-data:www-data /opt/lampp/htdocs/drupal8

Con este último proceso hemos terminado con la instalación por consola, ahora debemos continuar la instalación mediante vía Web, ingresando al navegador especificando nuestra dirección donde se procedió a levantar Drupal.

b) Instalación del CMS Drupal

Para ello debemos ingresar a nuestro navegador especificando nuestra (dirección IP o localhost) y accederemos a la pantalla de instalación de Drupal.

Nota: Tener en cuenta si utilizamos un puerto en específico.

Paso 1

Seleccionamos el idioma y damos click en "Guardar y continuar".

Paso 2

Seleccionamos el perfil de instalación, se recomienda el estándar ya que es uno de lo más utilizamos y para nuestro proyecto el más acorde a lo que necesitamos.

Paso 3

Verificamos que no exista problemas de requisitos para la correcta instalación de Drupal, caso contario no permitirá continuar con los siguientes pasos de instalación.

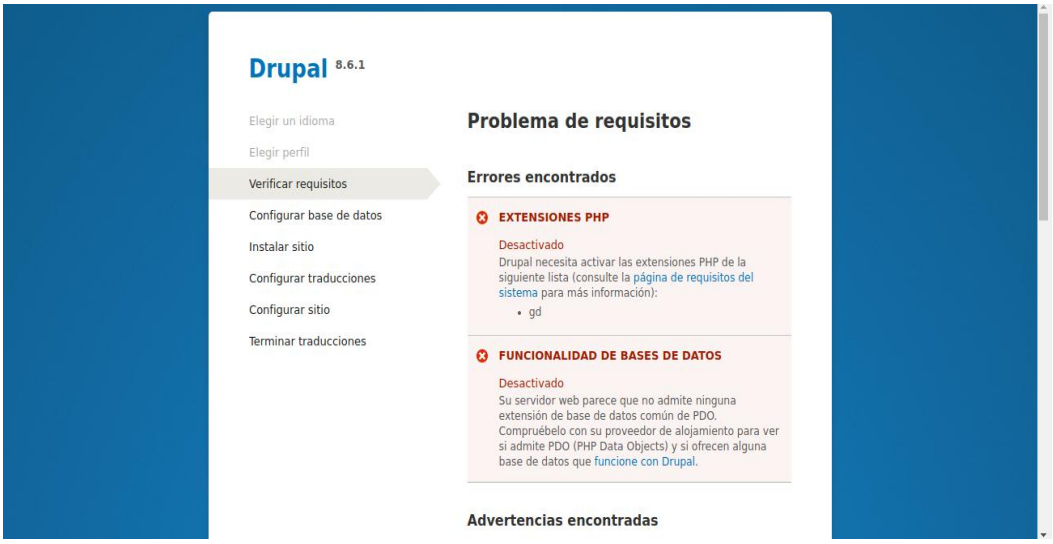


Figura A20. 1. Errores en los requisitos de Drupal

En este proceso de instalación, se presentaron 2 errores de configuración. Para la solución de estos errores, se utilizó los siguientes comandos ingresados en la terminal.

sudo apt-get install php7.2-mbstring

sudo apt-get install php7.2-gd

sudo apt-cache search php-pdo

sudo apt-get install php7.2-mysql

Ingresando los comandos anteriores, se solucionan los inconvenientes presentados y nos permitirá continuar con el proceso de instalación.

Paso 4

Debemos ingresar el nombre de la base de datos, el usuario del mismo y su respectiva contraseña, que se creó en los pasos principales utilizando la plataforma mysql.

Paso 5

Como es un sitio Web, se debe llenar los siguientes campos dependiendo la organización que desee utilizar el sistema Web Drupal.

Nota: En la parte de Cuenta de Mantenimiento del Sitio, se debe asignar un nombre o cuenta, junto con la contraseña que se desee para el administrador del sitio Web.

Paso 6

Es el paso final donde se procede a aceptar la configuración que se ingresó para el sitio Web.

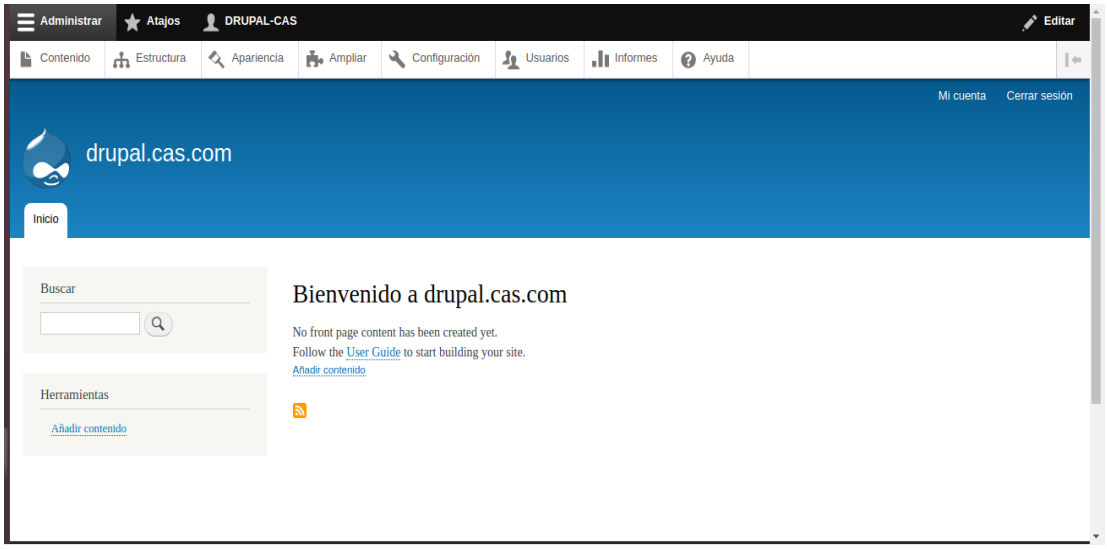


Figura A20. 2. Interfaz principal de Drupal

c) Instalación del plugins CAS en Drupal

Drupal, nos ofrece la facilidad de instalar el módulo CAS y External Authentication mediante una interfaz amigable, a continuación, se detallan los pasos a seguir para la correcta instalación del módulo CAS y External Authentication:

Paso 1

En la página principal del administrador de Drupal, vamos a configurar el plugin de CAS. Seleccionamos Administrar>>Ampliar y seleccionamos Instalar Nuevo Módulo.

Nos dirigimos a la siguiente dirección https://www.drupal.org/project/cas para copiar el enlace del archivo CAS o descargar el mismo, en nuestro caso es la versión 8.X-1.5.tar o .zip. Si descargamos el archivo damos click en Seleccionar archivo y buscamos la descarga en el sitio que se almacena y posteriormente damos click en Instalar.

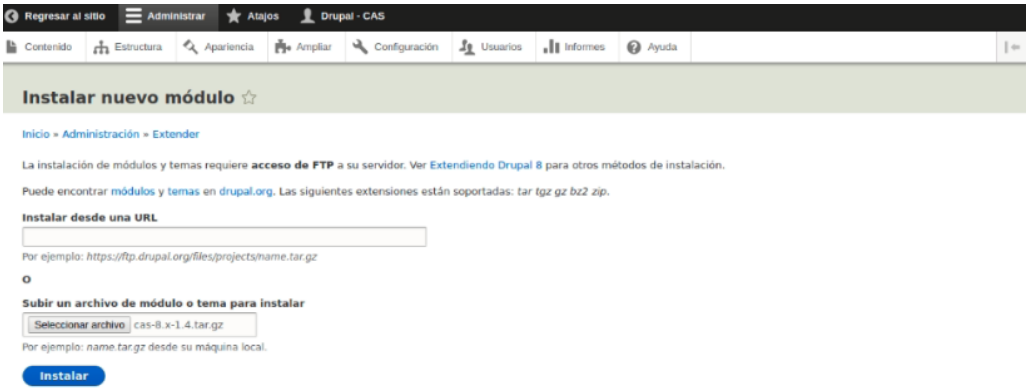


Figura A20. 3. Módulo CAS para Drupal

Para el funcionamiento del plugin CAS en Drupal necesitamos otro plugin secundario por la arquitectura de la Aplicación el cual es ExternalAuth . Nos dirigimos a la siguiente dirección https://www.drupal.org/project/externalauth/releases/8.x-1.1 para copiar el enlace del archivo o descargar el mismo “externalauth 8.x-1.1”. Si descargamos el archivo damos click en Seleccionar archivo y buscamos la descarga en el sitio que se almacena y posteriromente damos click en Instalar

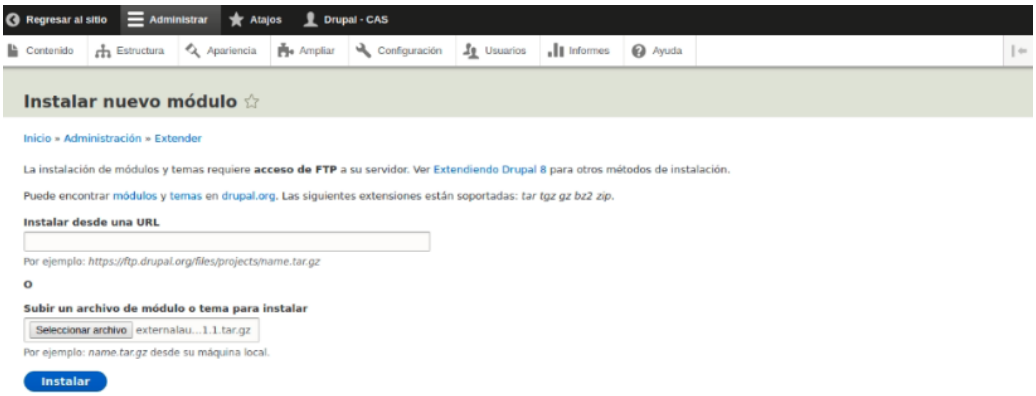


Figura A20. 4. Módulo externalAuth para Drupal

Paso 2

En el buscador ingresamos el nombre del módulo CAS para activarlo. Habilitamos el módulo dando click en el cuadro cerca del nombre y se activará un visto. Así ya podremos hacer uso de este módulo, seleccionándolo para ir a sus configuraciones.

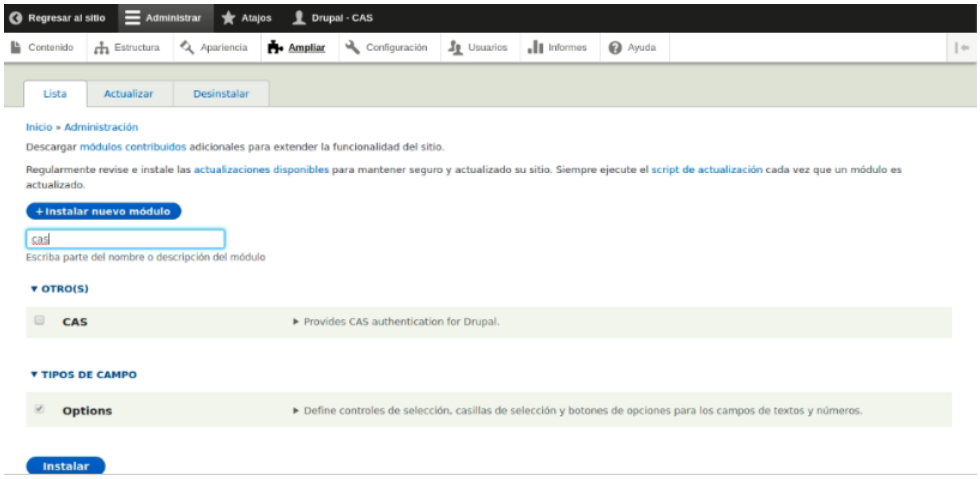


Figura A20. 5. Activación del Módulo CAS para Drupal

Paso 3

Una vez agregado los nuevos plugin que vamos a utilizar, nos dirigimos a **Inicio>>Adminitración>>Configuración>>Usuarios** y seleccionamos CAS para configurar su parte de autenticación.

En este parte procedemos a configurar de la siguiente manera:

- Seleccionamos la versión 3 de nuestro protocolo.
- Asignamos el nombre nuestro servidor CAS o dirección la cual se configuro y definió en el (Anexo 15. Instalación Apache Tomcat).
- Ingresamos el puerto de nuestro servidor CAS, configurado anteriormente.

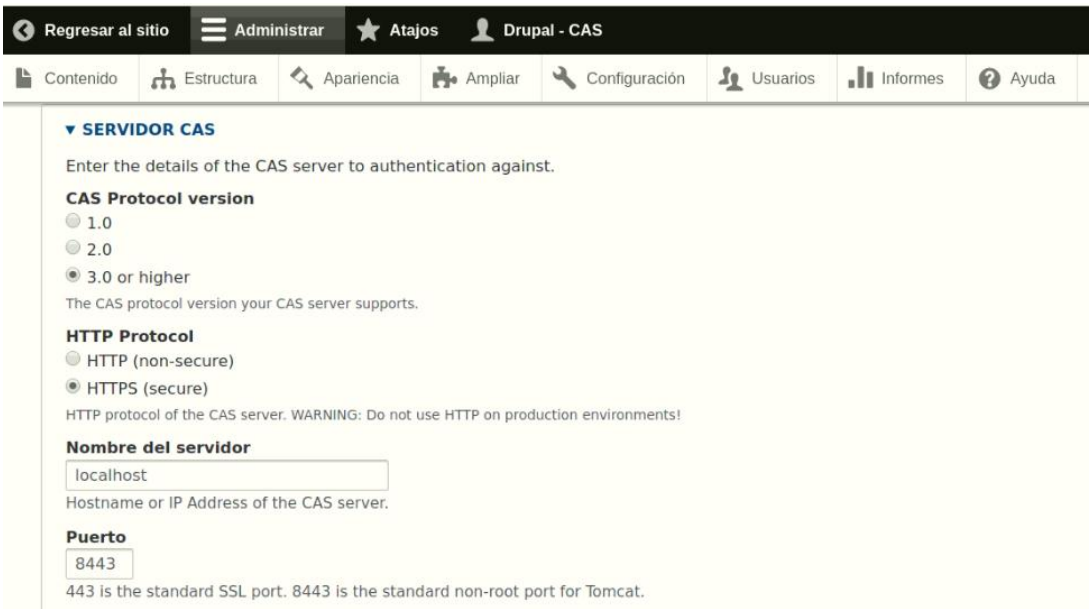


Figura A20. 6. Ruta del Servicio CAS para el inicio de sesión único en Drupal

- Ingresamos la ruta de inicio de sesión de nuestro servidor, en este caso no la ingresamos ya que la dirección de nuestro servidor CAS nos dirige directamente a su login.
- Seleccionamos NO verificar, debido a que el sistema de autenticación única tiene un certificado SSL.
- Habilitamos el enlace para que nos redirija al formulario de inicio de sesión de nuestro servidor CAS.
- Ingresamos el nombre para el enlace.

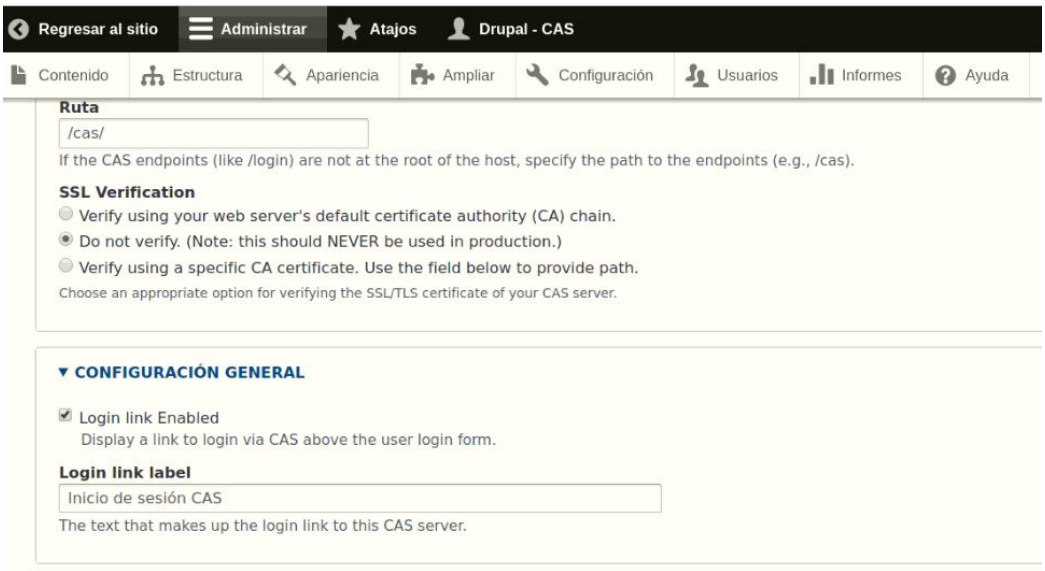


Figura A20. 7. Habilitación del servicio CAS en Drupal

- Habilitamos la creación automática de cuentas locales, en la base de datos de Drupal, debido a que el proceso de Autenticación es con el Protocolo CAS y por ende el mismo con el servidor OpenLDAP, el cual se lo configuro anteriormente. Ver anexo conexión cas y ldap.
- Asignamos la dirección de correo electrónico usando un atributo CAS, por la información que ya tenemos almacenado en ese servidor con su respectivo correo electrónico.
- Evitamos el inicio de sesión normal para los usuarios de CAS, debido a que el proceso de autenticación debe ser únicamente con CAS.
- Restringimos la gestión de contraseñas para Drupal, por lo que se debe trabajar específicamente con CAS.
- Restringimos la gestión de correo electrónico, por ser un proceso inhabilitado para el servidor CAS, el cual lo definimos anteriormente como una política de privacidad.

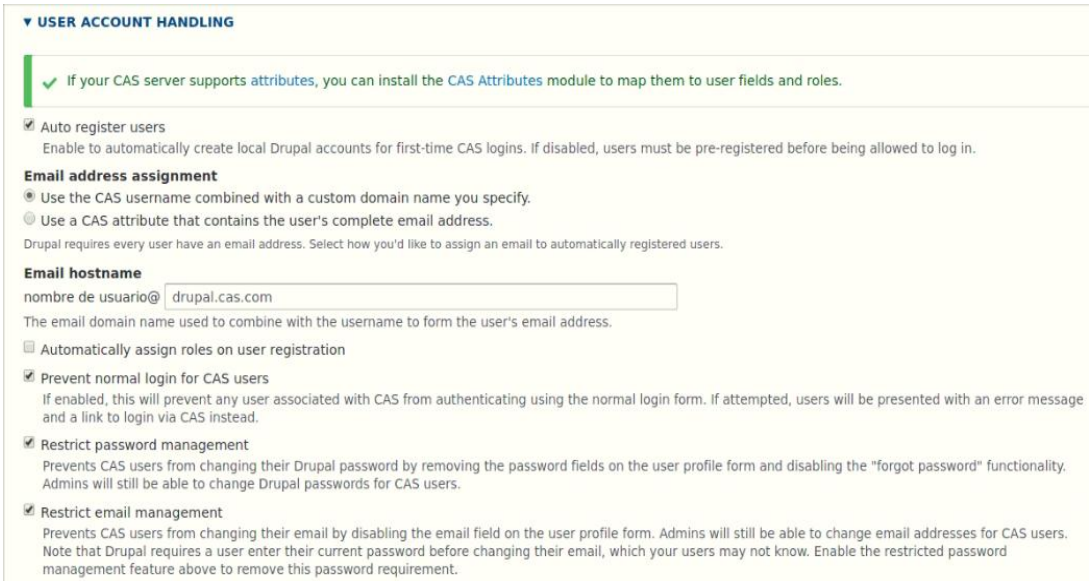


Figura A20. 8. Configuración de cuentas de usuario en Drupal

Paso 4

Cerramos sesión e ingresamos de nuevo a la dirección que levantamos WordPress, debe presentarnos la interfaz de acceso de Jasig CAS; donde ingresamos un usuario que ya se encuentre registrado en nuestro servidor OpenLDAP y damos click a “LOGIN”.

Anexo 21. Configuración de la Aplicación Web Jenkins

A continuación, se realizan los siguientes pasos para la correcta configuración de Jenkins:

a) Instalación Jenkins

Paso 1

Ejecutamos las siguientes instrucciones en línea de comandos:

wget -q -O — https://pkg.jenkins.io/debian/jenkins-ci.org.key | sudo apt-key add -

echo deb https://pkg.jenkins.io/debian-stable binary/ | sudo tee

/etc/apt/sources.list.d/jenkins.list

Paso 2

Actualizar la lista de paquetes disponibles y sus versiones, ingresando el siguiente comando en la terminal:

sudo apt-get update

Paso 3

Procedemos a instalar Jenkins e inicializarlo con los siguientes comandos:

sudo apt-get install jenkins

sudo systemctl start Jenkins

Paso 4

Una vez terminada la instalación, procedemos a cambiar el puerto para ejecutar Jenkins, por defecto utiliza el puerto 8080, el cual para nuestro desarrollo nos generará inconvenientes y habilitaremos otro puerto para su ejecución, ingresamos el siguiente comando:

sudo nano /etc/default/Jenkins

Se nos abrirá un archivo el cual debemos dirigirnos y modificar en la parte donde se encuentre:

HTTP_PORT=8080

Ingresamos el nuevo puerto que deseamos configurar por defecto y eliminamos el anterior.

Paso 5

Una vez terminado el cambio de puerto, procedemos a reiniciar Jenkins, ingresamos el siguiente comando:

service jenkins restart

Paso 6

Debemos ingresar a nuestro navegador especificando nuestra (dirección IP o localhost) junto al puerto que configuramos y accederemos a la pantalla de instalación de Jenkins.

Paso 7

Ingresamos la contraseña que nos pide Jenkins para poder continuar, copiando la ruta que se nos presente en la pantalla principal agregando al principio lo siguiente: sudo cat Nos quedaría este comando para ingresar mediante consola:

sudo cat /ruta_señalada

Figura A21. 1. Contraseña de Administrador de Jenkins

Paso 8

Se nos presentará una contraseña temporal, la cual debemos copiar.

Una vez copiada la contraseña en nuestro portapapeles, lo pegaremos en la administración Web o pantalla principal indicada en el paso anterior.

Paso 9

Instalaremos los plugins más comunes, dando click en la opción izquierda “Install suggested plugins”.

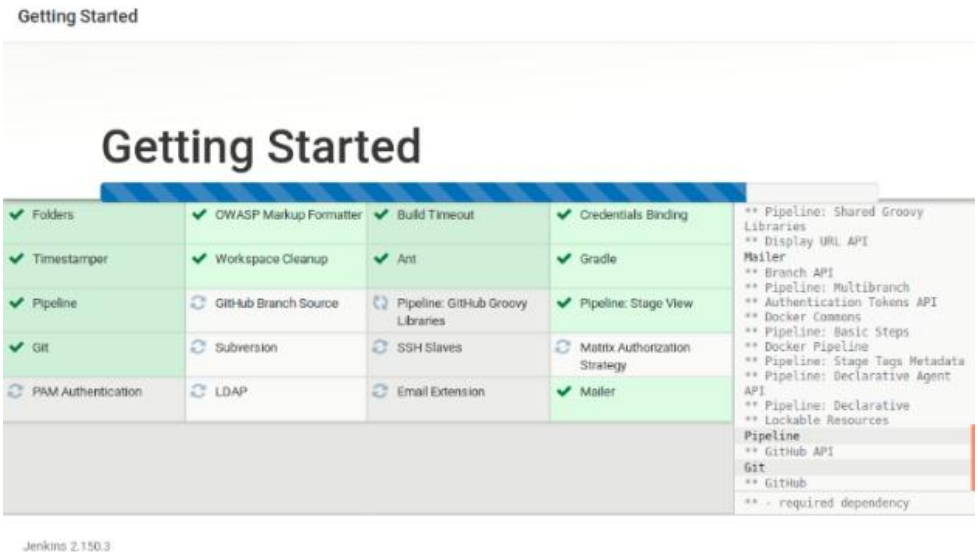


Figura A21. 2. Instalación de plugins por defecto en Jenkins

Paso 10

Posterior nos solicitará un usuario y una contraseña, que usaremos parae poder acceder a la administración de Jenkins:

Paso 11

Agregamos la dirección con la cual vamos a ingresar o inicializar Jenkins cuando deseemos acceder.

Hemos finalizado con la instalación de nuestro Jenkins

Figura A21. 3. Interfaz principal de Jenkins

b) Instalación del plugins CAS en Jenkins

Jenkins, nos ofrece la facilidad de instalar el módulo CAS, mediante una interfaz amigable, a continuación, se detallan los pasos a seguir para la correcta instalación del módulo CAS para Jenkins:

Paso 1

En la página principal del administrador de Jenkins, vamos a configurar el plugin de CAS. Seleccionamos **Administrar Jenkins>>Administrar Plugins** seleccionamosen la parte superiorTodos los Plugins.

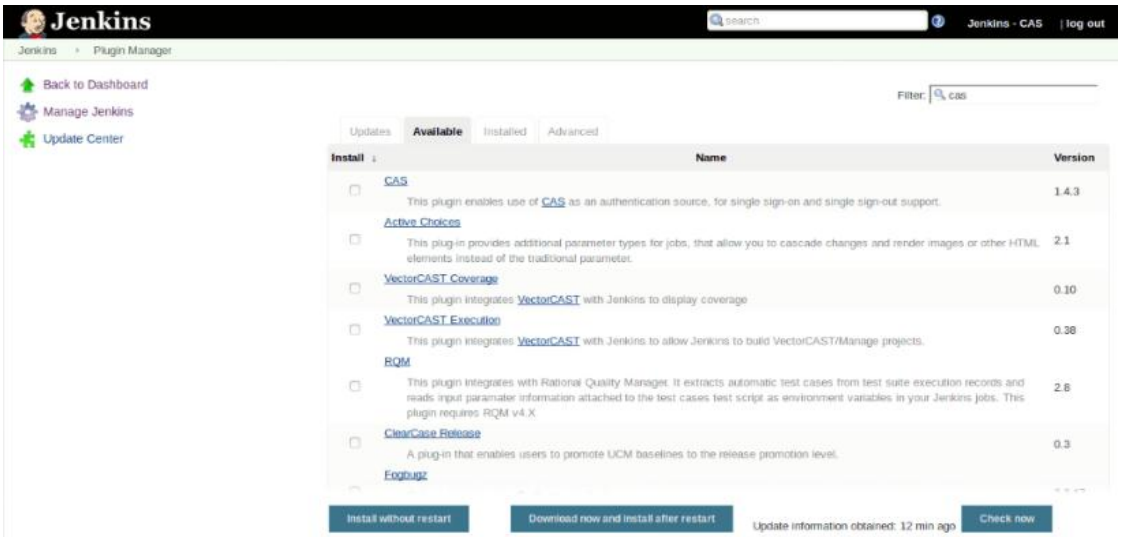


Figura A21. 4. Seleccionar el plugin para el servicio CAS

Paso 2

n el buscador ingresamos el nombre del módulo CAS para activarlo. Habilitamos el módulo dando click en el cuadro cerca del nombre y se activará un visto, posteriormente damos click en Instalar sin reiniciar.

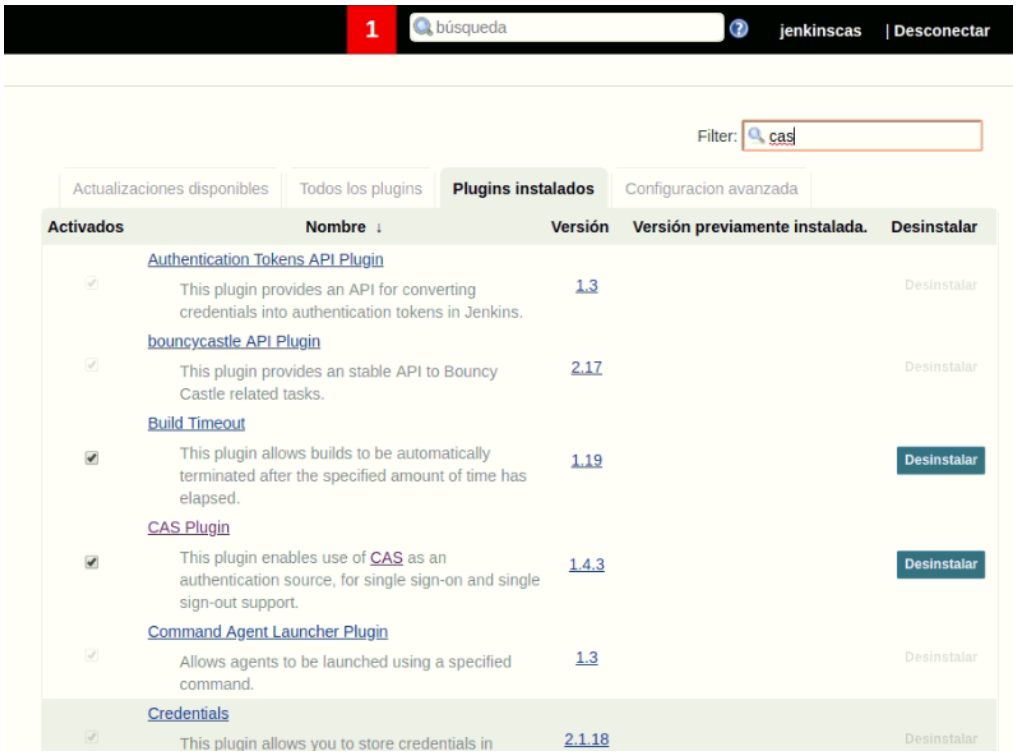


Figura A21. 5. Activación del plugin CAS

Paso 3

Luego de instalar el plugin de CAS, procedemos a su configuración para ello seleccionamos **Administrar Jenkins>>Configuración global de la Seguridad.** y Seleccionamos CAS para configurar de la siguiente manera:

- Asignamos el nombre nuestro servidor CAS o dirección la cual se configuro y definió en el (Anexo 15. Instalación Apache Tomcat). La ruta se la establece completa, incluido su puerto.
- Seleccionamos la versión 3 de nuestro protocolo.
- Opcionalmente, podemos habilitar el REST API de Jenkins.

Figura A21. 6. Ruta del servicio CAS para el inicio de sesión en Jenkins

Paso 4

Jenkins, nos permite configurar la autorización o perfiles de usuario en esta sección, para el presente TT realizamos las configuraciones. De los perfiles de Usuario como se indican en la Figura A21. 7. Antes de ellos debemos seleccionar **Configuración de Seguridad** para completar el proceso.

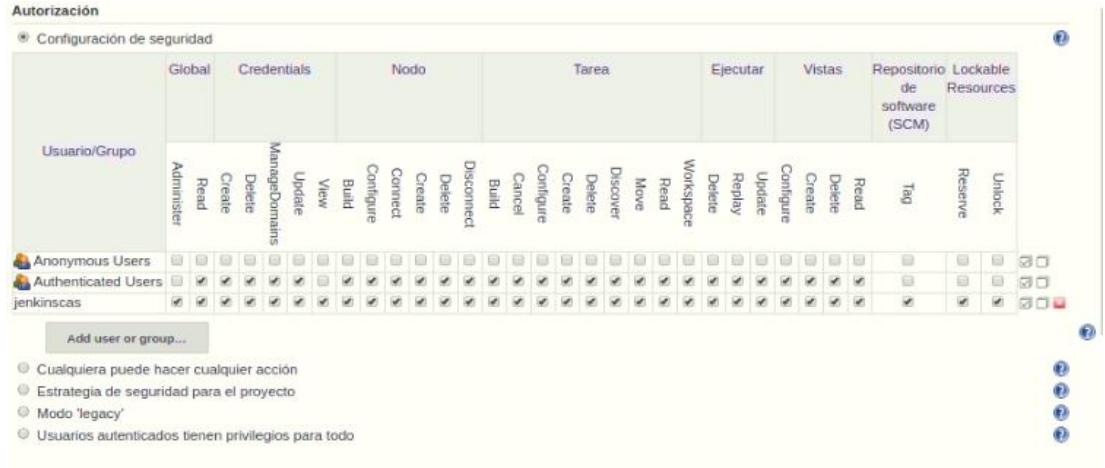


Figura A21. 7. Configuración de la Autorización o Peefiles de Usuario.

Paso 5

Cerramos sesión e ingresamos de nuevo a la dirección que levantamos WordPress, debe presentarnos la interfaz de acceso de Jasig CAS; donde ingresamos un usuario que ya se encuentre registrado en nuestro servidor OpenLDAP y damos click a “LOGIN”.

Anexo 22. Configuración del Cierre de Sesión Único

Las Aplicaciones Web que se seleccionaron manejaban el cierre de sesión individual, por lo cual se procedió a realizar la configuración para un cierre de sesión único entre todas. A continuación, se indica el procedimiento para su habilitación paras la cual se la puede hacer de dos maneras, para el desarrollo del presente TT se utilizarán las dos formas:

1. Ingresamos al archivo **cas.properties** que se encuentra en **cas-4.0.1/cas-server-Webapp/src/main/Webapp/WEB-INF** para establecer la variable que se habilitará el cierre de sisón único.

```
# CAS Logout Behavior
# WEB-INF/cas-servlet.xml
#
# Specify whether CAS should redirect to the specified service parameter on /logout requests
cas.logout.followServiceRedirects=true
```

Figura A22. 1. Archivo para habilitar el cierre de sesión único.

2. Ingresamos al archivo **cas-servlet.xml** que se encuentra en **cas-4.0.1/cas-server-Webapp/src/main/Webapp/WEB-INF** para establecer la variable que se habilitará el cierre de sisón único.

```
<bean id="logoutAction" class="org.jasig.cas.web.flow.LogoutAction"
      p:servicesManager-ref="servicesManager"
      p:followServiceRedirects="${cas.logout.followServiceRedirects:true}"/>
```

Figura A22. 2. Archivo para establecer la variable para el cierre de sesión único

Para cualquiera de las dos formas, se utiliza la misma variable, en la cual la definimos de a siguiente manera:

- **cas.logout.followServiceRedirects =** lo establecemos en **true** para habilitar el cierre de sesión único o la petición que enviará CAS indicando que el ticket de cual se esta haciendo uso ya no es válido.

GitLab

Por la arquitectura de la Aplicación GitLab, al establecer la configuración inicial de uso del protocolo CAS, ya activa el cierre de sesión único.

WordPress

Por la arquitectura de la Aplicación WordPress, no permite configurar el cierre de sesión único, ya que activa internamente un inicio de sesión y solamente se lo puede destruir directamente desde el mismo sistema.

Drupal

Nos dirigimos a **Inicio>>Adminitración>>Configuración>>Usuarios** y seleccionamos el servidor CAS para configurar su parte de cierre de sesión único de la siguiente manera:

- Habilitamos el cierre de sesión de Drupal
- En cerrar sesión destino, agregamos la ruta donde se nos redirecciona cuando cerremos sesión.
- Habilitamos el cierre de sesión único para manejarlo junto al inicio de sesión único.

Figura A22. 3. Configurar Cierre de Sesión Único en Drupal

Jenkins

Nos dirigimos a **Inicio>>Adminitración>>Configuración>>Usuarios** y seleccionamos el servidor CAS para configurar su parte de cierre de sesión único de la siguiente manera:

- Habilitamos el Single Sign-Out (en español, Cierre de Sesión Único) para manejarlo junto al inicio de sesión único.

Figura A22. 4. Configurar Cierre de Sesión Único en Jenkins

Moodle

Por la arquitectura de la Aplicación Moodle, no permite configurar el cierre de sesión único, ya que activa internamente un inicio de sesión y solamente se lo puede destruir directamente desde el mismo sistema.

De las aplicacines utilizadas en el presente TT, Jenkins y Gitlab son las únicas que permiten el cierre de sesión único, es decir si cierro sesión en otra de las aplicaciones o sistemas web intengrados con el protocolo CAS, estas aceptarán el aviso que el ticket ya no es válido y cancelarán la sesión.

Anexo 23. Plan de Trabajo y Cambios para el Sistema SAC con personal de la UTI.

UNIVERSIDAD NACIONAL DE LOJA

INF-CIS-UNL

Plan de Trabajo en la Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja	
Tareas	**Fecha**
Propuesta de la estructura jerárquica para el servidor (ldap.unl.edu.ec) y sistema SiGUCAS	26/12/2018 - 26/12/2018
Rediseño de la estructura antes propuesta, en el servidor de la UNL	27/12/2018 - 28/12/2018
Implementación del sistema SAC con el servidor (ldap.unl.edu.ec), utilizando datos de prueba.	03/01/2019 - 15/01/2019
Corregir los errores presentados en la validación del sistema SAC.	66/01/2019 - 22/01/2019
Dar a conocer, el funcionamiento del sistema SAC al Director de la UTI.	23/01/2019 - 23/01/2019
Realizar los cambios y agregar los módulos, que solicito el cliente (UTI).	24/01/2019 - 08/02/2019
Pruebas de funcionalidad por parte de los desarrolladores del sistema SAC	12/02/2019 - 15/02/2019
Presentar el funcionamiento final del sistema SAC al Director de la UTI.	18/02/2019 - 18/02/2019
Carga masiva de usuarios, en el servidor (ldap.unl.edu.ec)	25/02/2018 - 27/02/2019
Convocatoria a la presentación final del sistema SAC y del tema de titulación **"Desarrollo de un Prototipo para el Servicio de Autenticación Central de Usuarios en Aplicaciones Web".**	08/03/2019 - 08/03/2019
Pruebas del sistema SAC por parte del personal de la UTI	12/03/2019 - 12/03/2019
Pruebas del Sistema SiGUCAS por parte del personal de la UTI.	12/03/2019 - 12/03/2019

UNIVERSIDAD NACIONAL DE LOJA

INF-CIS-UNL

Loja, 12 de marzo del 2019

Unidad de Telecomunicaciones e Información de la Universidad Nacional de Loja
Ciudad. –

De mi consideración:

Yo, **Wilmer Antonio Aguilar Soto**, con cédula No. 1900481878 y **Manuel Stalin Armijos Ordoñez**, con cédula No.1105593238 egresados de la Carrera de Ingeniería en Sistemas, por medio de la presente solicitamos la aceptación de la lista de cambios sugeridos, en la implementación del Servicio de Autenticación Central (SAC), con el correcto funcionamiento administrativo del servidor (ldap.unl.edu.ec), para el personal de la Universidad Nacional de Loja. Los cuales se los realizó entre el 26 de diciembre del 2018 al 27 de febrero del 2019

Adjunto a la presente, el listado de cambios solicitados.

Atentamente:

Wilmer Antonio Aguilar Soto
1900481878

Manuel Stalin Armijos Ordoñez
1105593238

Aprobación del Listado de cambios:

UNIVERSIDAD NACIONAL DE LOJA
RECIBIDO
Fecha: 21-3-2019
Hora: 12:12
Responsable:

Ing. Jhon Calderón
Director de la Unidad de Telecomunicaciones e Información

Ing. Juan Pablo Ramón

Ing. Danny Muñoz

Cambios Sistema SAC					
Detalle	Ambiente	Aprobación		Recomendaciones	Observaciones
		SI	NO		
Carga masiva de usuarios, se utiliza un formato (.csv)	Pedido del cliente (Director UTI).				Se indica el formato, que necesito el archivo para subir los usuarios.
Método de restablecer contraseña, enviando un link encriptando las variables, al correo electrónico.	Reunión, personal de la UTI				Se utiliza la librería JWT (Autenticación basada en Tokens).
Enviar al correo electrónico las credenciales del usuario para su acceso, el cual se lo agrego mediante el sistema SAC.	Reunión, personal de la UTI				
Método para crear administradores de lectura, en el servidor ldap.unl.edu.ec	Pedido del cliente (Director UTI).				
Generar un reporte de todos los usuarios, que se encuentran en el servidor ldap.unl.edu.ec	Reunión, personal de la UTI				Se debe seleccionar el grupo, para generar el listado de los integrantes.
Eliminar todos los usuarios, que se encuentran en el servidor ldap.unl.edu.ec	Pedido del cliente (Director UTI).				Se debe seleccionar el grupo, para eliminar todos los integrantes.
Validación cédula del usuario	Pedido del cliente (Director UTI).				
Presentar información, de los usuarios vinculados a un sistema específico.	Reunión, personal de la UTI				
Imprimir información de los usuarios vinculados a un sistema específico.	Adicional de los desarrolladores				
Mejorar el tiempo de respuesta, al presentar el listado de los usuarios que se encuentra en el servidor ldap.unl.edu.ec	Reunión, personal de la UTI			Utilizar datatable(), que es una extensión de JQuery.	

Anexo 24. Licencia Creative Commons