SISTEMA CENTRALIZADO DE GESTIÓN DE USUARIOS PARA LA UNIVERSIDAD DEL TOLIMA

FRANCISCO ANDRADE NAVARRO
HERNAN DARIO MENDIETA

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA
ESCUELA DE CIENCIAS BASICAS, TECNOLOGIA E INGENIERIA
ESPECIALIZACIÓN EN SEGURIDAD INFORMATICA
CEAD IBAGUÉ
2015

SISTEMA CENTRALIZADO DE GESTIÓN DE USUARIOS PARA LA UNIVERSIDAD DEL TOLIMA

FRANCISCO ANDRADE NAVARRO
HERNAN DARIO MENDIETA

Trabajo de grado para optar el título de Especialistas en Seguridad Informática

DIRECTOR:
ING. MARTÍN CAMILO CANCELADO RUIZ

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA
ESCUELA DE CIENCIAS BASICAS, TECNOLOGIA E INGENIERIA
ESPECIALIZACIÓN EN SEGURIDAD INFORMATICA
CEAD IBAGUÉ
2015

Nota de Aceptación

_____

_____

_____

_____

_____

Firma del Presidente del Jurado

_____

Firma del Jurado

_____

_____

Firma del Jurado

Ibagué. Septiembre de 2015

DEDICATORIA

Dedicamos este logro profesional a Dios que nos brinda la fortaleza y a nuestras familias, esposas e hijos, quienes nos han dado su apoyo, cariño y comprensión incondicional para el alcance de nuestras metas.

AGRADECIMIENTOS

Agradecemos a Dios por permitirnos lograr nuestros objetivos y alcanzar una formación más en nuestras metas profesionales.

Sinceros agradecimientos a las directivas de la Universidad del Tolima y funcionarios de la Oficina de Gestión Tecnológica por brindarnos su atención y su apoyo para alcanzar los objetivos propuestos.

A la Universidad Nacional Abierta y a Distancia UNAD, por permitirnos enriquecer nuestros conocimientos, por facilitar los mecanismos para que la educación llegue a más personas y especialmente por permitirnos la ejecución de este proyecto como opción de grado.

Ing. Hernan Dario Mendieta Ing. Francisco Andrade Navarro

CONTENIDO

pag.

LISTA DE TABLAS

pag.

LISTA DE FIGURAS

pag.

LISTA DE ANEXOS

INTRODUCCIÓN

Hoy en día el continuo avance de los desarrollos tecnológicos y movilidad, ha logrado que la información sea más accesible a los usuarios, requiriendo de mecanismos de autenticación diferentes haciendo que sea más complejo dicho proceso, ya que se deben de contar con una serie de credenciales (usuarios y contraseñas) diferentes para acceder a dichos servicios. Es por ello la importancia de un modelo que facilita el intercambio de credenciales mediante protocolos y estándares que faciliten la integración entre plataformas, mediante el desarrollo de herramientas y programas que logren asegurar la información gestionando permisos de accesos, derechos, manipulación o consulta de esta, teniendo en cuenta su grado de importancia y el nivel de protección que se le pueda llegar a brindar.

Hoy en día existen una cantidad de tecnologías emergentes, que resaltan la importancia y necesidad de los usuarios, desarrolladores y administradores que requieren la comodidad de identificarse una sola vez y mantener la sesión válida para el resto de aplicaciones de las cuales hacen uso y el creciente aumento de la demanda de información y continuó acceso a ella, han acarreado el aumento de distintas plataformas informáticas que permitan el acceso, generando grandes cargas administrativas a los administradores del sistema, pues se debe asegurar confidencialidad, integridad, disponibilidad y autenticidad de la información, mediante controles y niveles de acceso, llegando a tener grandes cantidades de usuarios en cada una de las plataformas.

El presente proyecto tiene como finalidad la construcción de un sistema de administración centralizada de usuarios basándose en el modelo Single Sing On (SSO), que permite el aseguramiento de credenciales de usuario y el acceso a las distintas plataformas que tiene la universidad del Tolima, dependiendo de los roles de los usuarios y los niveles de acceso a la información de manera controlada.

1. TITULO DEL PROYECTO

SISTEMA CENTRALIZADO DE GESTIÓN DE USUARIOS PARA LA UNIVERSIDAD DEL TOLIMA.

2. PLANTEAMIENTO DEL PROBLEMA

La Universidad del Tolima es una institución de educación superior de carácter público que fomenta el desarrollo de capacidades humanas para una formación integral permanente mediante la búsqueda incesante del saber, la producción, la apropiación y la divulgación del conocimiento en los diversos campos de la ciencia, arte y cultura. Comprendida con el bienestar social, ambiental y desarrollo sustentable de la región.

Para consolidarse como una de las universidades más importantes de Colombia reconocida por su excelencia académica, compromiso social y acreditación institucional de alta calidad, ha profundizado en el uso de las tecnologías de la información y comunicación, implementando diferentes aplicaciones al servicio de la comunidad académico / administrativo que facilitan la interacción y acceso a la información[1].

Para el uso de estas aplicaciones requiere que los usuarios se autentiquen, es decir, que demuestre de algún modo que es quien dice ser, ya que estas aplicaciones también almacenan datos personales.

Esto ha traído consigo inconvenientes porque al tratarse de aplicaciones independientes entre sí, cada una de ellas utiliza su propio sistema de autenticación y gestión de datos personales, siendo esto para el usuario cada vez más grave a medida que el número de sistemas que utiliza crese, teniendo que recordar diferentes tipos de usuario y contraseñas, y autenticarse en cada uno de ellos cada vez que desea usarlos. Además de tener que mantener actualizada la información personal en cada uno de ellos.

[1] Universidad del Tolima. (13 de Agosto de 2013). *Principios y Valores*. Recuperado el 18 de Octubre de 2014, de http://www.ut.edu.co/administrativos/index.php/inti/quienes-somos/principios-y-valores

Esto también ha ocasionado grandes cargas administrativas al tener que depurar la información de los usuarios contantemente para evitar duplicidad de la información o incoherencias de la información, como también fallos en la seguridad por prácticas que comprometen los sistemas como son el uso de contraseñas inseguras, contraseñas guardadas en papeles, documentos de texto, reutilización de contraseñas en diferentes sistemas.

3. FORMULACIÓN DEL PROBLEMA

¿Cómo puedo centralizar y sincronizar la gestión de usuarios de las distintas aplicaciones de la Universidad del Tolima?

Si la Universidad del Tolima no implementa mecanismos de seguridad y políticas en el la administración de los sistemas de información que permitan consistencia en la calidad de la información, conllevara a una baja de credibilidad ante sus usuarios debido a la perdida de la integridad de los datos en sus sistemas de información.

4. JUSTIFICACIÓN

El presente estudio mejorara la seguridad de las aplicaciones, asegurando la privacidad, disponibilidad e integridad de la información permitiendo una mejor experiencia funcional de los usuarios al realizar uso de las aplicaciones implementadas por la Universidad del Tolima.

El estudio busca establecer y minimizar los riesgos de seguridad, del acceso los múltiples sistemas de información, centralizando la administración de credenciales de acceso de los usuarios, eliminado la creación de usuarios por cada aplicación o sistema de información, estableciendo los diferentes niveles seguridad de acuerdo a los perfiles de cada usuario, permitiendo hacer uso de los sistemas sobre los cuales tiene privilegios de acuerdo al nivel de ejecución y criticidad de la información, haciendo énfasis sobre a qué tipo de información tiene derecho de lectura, modificación ejecución o borrado.

Por otra parte se puede utilizar los resultados obtenidos de este estudio para facilitar la toma de dicciones sobré implementaciones de sistemas centralizados de gestión de usuarios, que otras universidades tanto públicas como privadas podrían tomar como modelo a seguir para resolver los problemas del múltiples cuentas de usuarios.

5. OBJETIVOS DEL PROYECTO

General:

Centralizar la gestión de usuarios, permitiendo la sincronización de la información de datos personales en un único sistema de autenticación para las distintas aplicaciones con que cuenta la Universidad del Tolima.

Específicos:

- Realizar un estudio de los distintos sistemas de autenticación centralizados (SSO) existentes, basándose en el performa, modularidad, escalabilidad, seguridad y facilidad de uso.
- Evaluar los requerimientos y necesidades de la gestión de usuarios de la Universidad del Tolima.
- Evaluar las políticas implementadas en el manejo de accesos y autenticaciones en las aplicaciones de la Universidad del Tolima.
- Establecer los requerimientos para la centralización de la gestión de usuarios.

6. MARCO REFERENCIAL

6.1. Marco teórico

El concepto de Single Sing-On[2] se refiere al acceso a múltiples recursos por medio de un único proceso. Gran cantidad de las arquitecturas implementadas en diferentes organizaciones han sido diseñadas con el objetivo de dar acceso a los usuarios a múltiples servicios Web y/o aplicaciones. En la mayoría de los casos se encuentra que cada uno de los servicios o aplicaciones cuenta con su propio componente de seguridad, lo cual generalmente compromete la seguridad de todo el sistema. El principal objetivo de una arquitectura que implementa Single Sign-On es transferir la funcionalidad y complejidad de todos los componentes de seguridad a uno solo (SSO), otro de los beneficios es que los usuarios deben hacer el proceso de ingreso una sola vez, a pesar de que continúan interactuando con múltiples componentes de seguridad en el sistema.

Single Sing-On no necesariamente se refiere a una sincronización de claves, ya que en este caso todas las aplicaciones y servicios funcionan con una misma clave, en el uso de este tipo de aplicaciones no quiere decir que el sistema se esté fortaleciendo al contrario se están debilitando, dado que cuando todas las aplicaciones o servicios utilizan una misma clave se está generando un riesgo, ya que si un intruso logra conseguir la clave de una de las aplicaciones o servicios tendrá inmediatamente acceso a todas ellas.

[2] InterGraphic Desing. (15 de 06| de 2012). Obtenido de Single Sign On: Un solo login, múltiples accesos: http://www.intergraphicdesigns.com/blog/2012/06/15/single-sign-on-un-solo-login-multiples-accesos/

Esta debilidad del sistema se soluciona sometiendo al usuario a un proceso de validación fuerte en el momento de hacer la autenticación haciendo que la arquitectura aumente el sistema de seguridad. Para una implementación adecuada de un sistema SSO se debe contar con un agente que se encargara de almacenar en una base de datos o directorio protegidos las claves, este tipo de autenticación fuerte se refiere al proceso de autenticación en sistemas que requieren múltiples factores para realizar la identificación del usuario, los cuales utilizan tecnología avanzada como contraseñas dinámicas o certificados digitales.

Existen diferentes tipos de arquitecturas para implementación de SSO, con características que las hacen más apropiadas dependiendo de las necesidades particulares de cada organización, dependiendo de los recursos disponibles, diseños de software, tecnologías de aplicaciones y / o programas existentes que permitan la compactibilidad del sistema único de autenticación.

Las arquitecturas de SSO se componen de tres componentes básicos:

- Interface: Es la manera en que el sistema de SSO interactúa con el usuario y las distintas aplicaciones de software a las que se quieren tener acceso por medio del sistema SSO.
- Administración: Modulo que permite configurar, mantener, monitorear y gestionar los procesos de SSO.
- Credenciales: Cada software o aplicación a la que se requiere acceder desde el sistema SSO, requiere de información confidencial como nombres de usuario, contraseñas, identificadores de usuario, etc. que al estar agrupados reciben el nombre de credenciales. Las credenciales son almacenadas de manera segura y protegida para que el único que pueda acceder a ellas sea los agentes del sistema SSO.

Las arquitecturas de los sistemas SSO son evaluadas y calcificas según las características de cómo ha sido usadas los tres componentes básicos que deben tener, nombrados anteriormente.

6.1.1 Arquitectura Password Vault

Se trata de la más básica de un sistema SSO en el uso de credenciales. Esta arquitectura se compone de tres elementos ubicados en el cliente, por lo cual es esta desde allí donde se acceden a las aplicaciones, para lo cual se deben almacenar las credenciales correspondientes para que puedan ser suministradas a las aplicaciones cuando sea necesario[3].

Figura 1 Arquitectura Password Vault

Fuente: http://www.criptored.upm.es/guiateoria/gt_m142j.htm

Mediante esta arquitectura las funciones administrativas son limitadas, ya que la administración se debe realizar en cada una de las estaciones de trabajo por lo cual generalmente termina quedando a cargo de los usuarios. No se puede realizará actualizaciones masivas de usuarios, se requiere realizarla equipo por equipo. El nivel de trasparencia para el usuario es bajo ya que el usuario tiene que intervenir en los procesos de configuración y administración del proceso de ingreso. El almacenamiento de credenciales es local lo cual impide la movilidad de los usuarios para el ingreso a las aplicaciones desde otros puntos de acceso.

[3] Iván M. Caballero, J. J. (07 de 2003). *Universidad de los Andes.* Obtenido de http://www.criptored.upm.es/guiateoria/gt_m142j.htm

6.1.2 Arquitectura Centralizada con Almacenamiento local de Credenciales

Este modelo brinda la administración de manera centralizada, donde solo el agente SSO y las credenciales permanecen en el cliente favoreciendo el control y monitoreo del proceso de ingreso, eliminado la necesidad de configurar el SSO en cada uno de los clientes. Para esta configuración se requiere de un servidor de aplicaciones y un servidor de para efectuar la administración[4].

Figura 2 Arquitectura centralizada con almacenamiento local de credenciales

Fuente: http://www.criptored.upm.es/guiateoria/gt_m142j.htm

[4] Iván M. Caballero, J. J. (07 de 2003). *Universidad de los Andes.* Obtenido de http://www.criptored.upm.es/guiateoria/gt_m142j.htm

6.1.3 Arquitectura de administración y almacenamiento de credenciales centralizadas

Este modelo permite centralizar los controles de acceso de los proceso de SSO, en este modelo el servidor central se encarga de realizar la administración y de almacenar y proveer la credenciales de acceso**5**.

Figura 3 Arquitectura de administración y almacenamiento de credenciales centralizadas

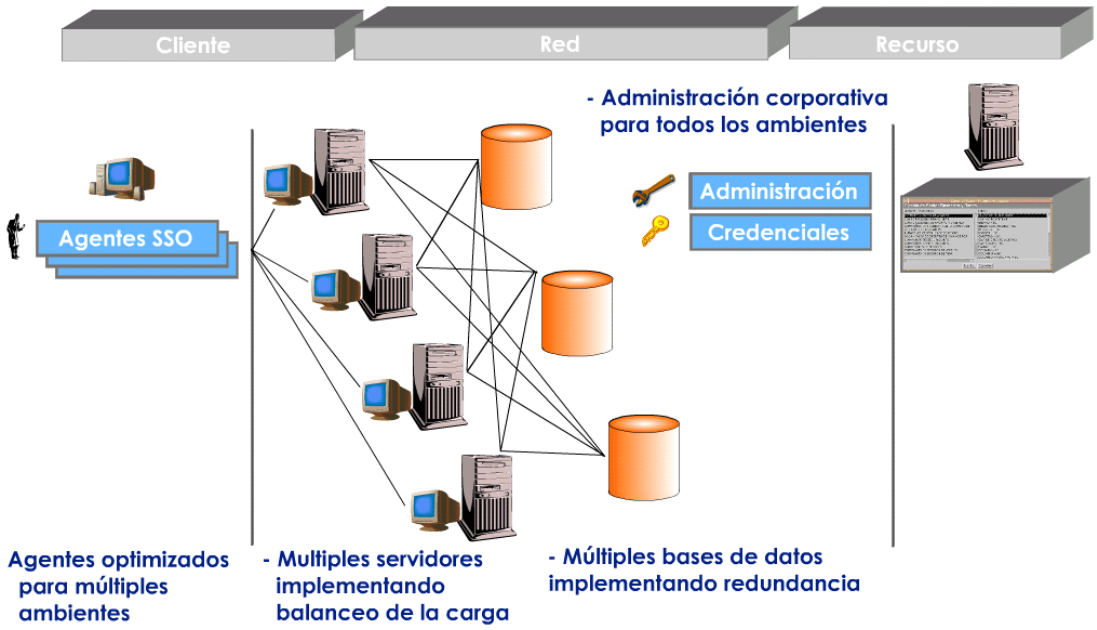
Fuente: http://www.criptored.upm.es/guiateoria/gt_m142j.htm

[5] Iván M. Caballero, J. J. (07 de 2003). *Universidad de los Andes.* Obtenido de http://www.criptored.upm.es/guiateoria/gt_m142j.htm

6.1.4 Arquitectura totalmente distribuida

Esta arquitectura se caracteriza por separar el servidor de la base de datos, lo cual la hace totalmente modular, estableciendo múltiples bases de datos y balanceo de caga entre ellas. La base de dato se encuentran sincronizadas para lograr redundancia entre ellas, el servidor de aplicaciones es independiente y la administración cuanta con una interfaz diferente[6].

Figura 4 Arquitectura totalmente distribuida



Fuente: http://www.criptored.upm.es/guiateoria/gt_m142j.htm

[6] Iván M. Caballero, J. J. (07 de 2003). *Universidad de los Andes.* Obtenido de http://www.criptored.upm.es/guiateoria/gt_m142j.htm

6.1.5 Administración y almacenamiento de credenciales centralizada garantizando la alta disponibilidad y redundancia.

Este modelo se basa en administrar y almacenar la credenciales de manera centralizada, con la configuración de un clúster HA el cual permite tener servidores duplicados con las mismas configuraciones e información, sincronizados para que todos tengan la misma información y caso de caída del servidor principal el segundo asuma el rol del servidor principal de manera automática[7].

Figura 5 Administración y almacenamiento de credenciales centralizada

Fuente: http://www.criptored.upm.es/guiateoria/gt_m142j.htm

[7] Iván M. Caballero, J. J. (07 de 2003). *Universidad de los Andes.* Obtenido de http://www.criptored.upm.es/guiateoria/gt_m142j.htm

6.2 Marco Contextual

La universidad del Tolima cuenta con una serie de sistemas de información debido al constante crecimiento de su infraestructura académica y administrativa generando grandes cambios a nivel de necesidades informáticas, muchas de ellas se han desarrollado para satisfacer necesidades momentáneas generando así una serie de sistemas distribuidos que en su mayoría de casos se hace necesario la generación de diferentes usuarios con sus respectivas claves de seguridad haciendo que el proceso de autenticación sea más dispendioso para el usuario, esto conlleva a realizar la implementación de un sistema que nos permita de forma centralizada generar un solo usuario y contraseña para dar acceso a los diferentes sistemas de información permitiendo tener una sola validación con acceso a todos los sistemas académicos y administrativos, es en esta parte que una herramienta como Single Sing-On se hace indispensable para dicho trabajo así como lo ha implementado diferentes universidades y organizaciones obteniendo resultados positivos.

Un ejemplo es el caso de la universidad de Sevilla que opto por la utilización de este sistema simplificando el acceso de los usuarios a las diferentes aplicaciones la comodidad de identificarse una sola vez y hacer uso de ellas es una forma de simplificar las cosas, mientras que para los desarrolladores es una manera de simplificar la lógica de sus aplicaciones al poder delegar la tarea de autenticación a un sistema totalmente independiente de las mismas, como lo es en el caso de Google Apss que con una sola cuenta un usuario puede acceder a (gmail,google calendar, google maps, google play, youtube, google docs) que son subdominios de google.com y otros son

dominios completamente aparte y que se encuentran posiblemente en múltiples servidores[8].

[8] InterGraphic Desing. (15 de 06| de 2012). Obtenido de Single Sign On: Un solo login, múltiples accesos: http://www.intergraphicdesigns.com/blog/2012/06/15/single-sign-on-un-solo-login-multiples-accesos/

6.3 **Marco legal**

El marco legal sobre el cual se respalda el proyecto de investigación, es el que rige en la legislación colombiana, específicamente en relación con la seguridad de la información y protección de los datos personales.

Ley 1581 de 2012: La ley de protección de datos personales – Ley 1581 de 2012 – es una ley que complementa la regulación vigente para la protección del derecho fundamental que tienen todas las personas naturales a autorizar la información personal que es almacenada en bases de datos o archivos, así como su posterior actualización y rectificación. Esta ley se aplica a las bases de datos o archivos que contengan datos personales de personas naturales[9].

Ley 1266 de 2008: Por la cual se dictan las disposiciones generales del hábeas data y se regula el manejo de la información contenida en bases de datos personales, en especial la financiera, crediticia, comercial, de servicios y la proveniente de terceros países y se dictan otras disposiciones[10].

Ley 1273 de 2009: (La cual añade dos nuevos capítulos al Código Penal) Capítulo Primero: De los atentados contra la confidencialidad, la integridad y la disponibilidad de los datos y de los sistemas informáticos[11].

---

[9] bogota, A. m. (18 de 10 de 2012). *http://www.alcaldiabogota.gov.co/*. Obtenido de http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=49981

[10] bogotá, A. m. (31 de 12 de 2008). *http://www.alcaldiabogota.gov.co/*. Obtenido de http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=34488

[11] Bogotá, A. M. (05 de 01 de 2009). *http://www.alcaldiabogota.gov.co/*. Obtenido de http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=34492

Artículo 269A: Acceso abusivo a un sistema informático
Artículo 269B: Obstaculización ilegítima de sistema informático o red de telecomunicación.
Artículo 269C: Interceptación de datos informáticos.
Artículo 269D: Daño Informático
Artículo 269E: Uso de software malicioso
Artículo 269F: Violación de datos personales.
Artículo 269G: Suplantación de sitios web para capturar datos personales.

Capitulo Segundo: De los atentados informáticos y otras infracciones. Como se puede ver en el primer capítulo, esta Ley está muy ligada a la ISO27000, lo cual coloca al País a la vanguardia en legislación de seguridad de la información, abriendo así la posibilidad de nuevas entradas con este tema. Protección de Datos personales En un sentido amplio, delito informático es todo aquel que implique la utilización de cualquier medio de tecnología informática. Delitos contra la intimidad, en el que se produce un tratamiento ilegal de los datos de carácter personal. Relativos al contenido, es decir a la difusión de contenidos ilegales en la Red; delitos económicos, relacionados con el acceso autorizado a sistemas informáticos para llevar a cabo fraude, sabotaje o falsificación, suplantación de entidades bancarias, delitos contra la propiedad intelectual vinculados con la protección de programas de ordenador, bases de datos y derechos de autor.

De los atentados informáticos y otras infracciones
Artículo 269I: Hurto por medios informáticos y semejantes.
Artículo 269J: Transferencia no consentida de activos.

7. DISEÑO METODOLÓGICO PRELIMINAR

La propuesta metodología del presente estudio se basa en la construcción de un método de simplificación de acceso a plataformas y servicios prestados por la Universidad del Tolima de una forma controlada y simplificada:

Se elaborará un perfil de la organización que busca caracterizar los principales problemas que se han generado por el constante crecimiento de la infraestructura física y tecnológica, que comprenden estados académicos y administrativos. La información requerida para este análisis será recolectada mediante la encuesta estructurada para el estudio técnico de los diferentes servicios ofrecidos por la universidad del Tolima.

Se evaluaran todos los sistemas de información en cuanto a tecnología usada (leguaje de programación) método de autenticación, compatibilidades de sistemas, estructura de nivel de permisos de acceso y orígenes de datos de credenciales de autenticación, mediante listas de chequeo.

Se estandarizaran los datos de las credenciales de autenticación estableciendo la compatibilidad con cada uno de los sistemas de información en cuanto a la información requerida, protocolos de comunicación y seguridad.

Se evaluaran las políticas de seguridad existentes y su nivel a aplicabilidad y cumplimiento en los procesos de autenticación en los sistemas de información con que cuenta la Universidad del Tolima.

Se realizaran reuniones con los directivos para presentar y socializar la importancia del proyecto en cuanto adoptar una cultura innovadora y profundizar

la estrategia asociativa para desarrollar procesos de trabajo enmarcados en la seguridad de la información y la protección de datos personales.

De acuerdo a la información recolecta se establecerá la herramienta SSO más adecuada compatible con las diferentes tecnologías de los sistemas de información de la Universidad del Tolima, estableciendo los diferentes orígenes de datos de autenticación, tipos de seguridad e encriptación, siendo lo menos invasivo posible en la modificación del modelo de autenticación de cada uno de los sistemas de información.

7.1 Sistemas de autenticación centralizados (SSO) existentes

Existen múltiples soluciones de sistemas de autenticación SSO de código abierto, basados en estándares, con distintas características de diseño y compatibilidades con distintas plataformas web, los más reconocidos y utilizados son[12]:

7.1.1 JOSSO

- JOSSO (Sign-On abierto de Java solo) es una fuente abierta J2EE e infraestructura del resorte SSO estado dirigida para proporcionar una solución para la autentificación y la autorización neutrales centralizadas de usuario de la plataforma.
- J2EE y dominio cruzado Sign-On transparente del resorte solo/organización cruzada.
- Marco enchufable para permitir la puesta en práctica de los componentes de encargo de la identidad usar el resorte o el envase incorporado del COI.
- Funcionamientos en el Tomcat de Apache.
- Funcionamientos en el servidor de aplicaciones de JBoss.

[12] Oyuela, S. G. (31 de 05 de 2014). *Salpicadero*. Obtenido de http://www.josso.org/confluence/display/JOSSO1/JOSSO+-+Java+Open+Single+Sign-On+Project+Home

- Funcionamientos en BEA WebLogic 9 y el servidor de aplicaciones de WebLogic 10.
- Funcionamientos en el servidor de aplicaciones de Apache Gerónimo Ayuda nativa del httpd 2.x de Apache que permite así SSO transparente con usos del rubí, de PHP, del pitón, del Perl, del etc.
- Integra con la seguridad del resorte para permitir la autorización de grano fino.
- Proporciona la información de la identidad a las aplicaciones web y a EJBs con el Servlet estándar y la seguridad API de EJB respectivamente.
- Autentificación fuerte de las ayudas usar certificados de cliente X.509.
- Ayuda de DAP para almacenar la información y las credenciales de usuario.
- Ayuda de base de datos para almacenar la información y las credenciales de usuario.
- Cliente API para el PHP. Esto permite construir usos SSO-permitidos del PHP.
- Cliente API para Microsoft ASP. Esto permite construir usos SSO-permitidos del ASP
- Compatibilidad con el envase de Apache Pluto Portlet
- El estándar basó: JAAS, Web Services/SOAP, EJB, puntales, Servlet/JSP, J2EE.
- Java 100%

7.1.2 WSO2 Identity Server Proveedor SSO

WSO2. Es la empresa de código abierto para Arquitectura Orientada a Servicios (SOA) fundada por los pioneros de la Fundación Apache Software comunidad de servicios Web.

WSO2 Identity Server proporciona seguridad sofisticada y gestión de la identidad de las aplicaciones web empresariales, servicios y APIs. Provee gestión de identidades a partir del acceso a servidores de LDAP para extraer información de usuarios, así como a través del uso de servicios de token de seguridad para su utilización en diversos escenarios de conversación segura entre servicios. Provee facilidades Single Sign On basada en OpenID [13] [14].

[13] EcuRed. (09 de 06 de 2015). *EcuRed*. Obtenido de http://www.ecured.cu/index.php/WSO2
[14] Weerawarana, S., & Fremantle, P. (09 de 06 de 2015). *WSO2*. Obtenido de http://wso2.com/products/identity-server/

Sistema de Identidad y Gestión de usuarios

- API para la integración de la gestión de identidad a cualquier aplicación
- Autenticación de múltiples factores
- Single Sign-On (SSO) a través de OpenID, SAML2 y Kerberos KDC
- SSO puente entre los sistemas en las instalaciones y aplicaciones de la nube
- Asignación de credenciales a través de diferentes protocolos
- Auditoría a través XDAS
- Delegación a través de OAuth 1.0a, OAuth 2.0 y WS-Trust
- Federación a través de OpenID, SAML2 y WS-Trust STS
- La integración con Microsoft SharePoint con el apoyo STS pasivos
- Implementar la seguridad REST con OAuth 2.0 y XACML
- Implementar la seguridad REST con OpenID Conectar
- Trusted Proveedores SAML2 de identidad por el inquilino
- Fuera de la caja de la integración con Google Apps y Salesforce
- Páginas de acceso personalizables para OpenID, OAuth, OpenID Connect, SAML2 y STS pasivos
- Control de acceso basado en roles (RBAC)
- Atributo o acceso basado reclamo control a través de XACML y WS-Trust, OpenID, y gestión de reclamaciones
- Control de acceso de políticas basadas en grano fino a través de XACML
- Auditoría y gestión de derechos Avanzada
- Gestión de Titularidad para cualquier llamada REST o SOAP

7.1.3 CAS

El servicio de autenticación central (CAS) es un proyecto de código abierto el cual consiste en un único inicio de sesión para plataformas Web. Su propósito es permitir al usuario acceder a múltiples aplicaciones, proporcionando sus credenciales una única vez (tales como identificador de usuario y contraseña). El nombre CAS también se refiere a un paquete de software que implementa este tipo de protocolo[15] [16].

[15] CAS. (09 de 06 de 2015). *Single Sign-On para la Web*. Obtenido de http://jasig.github.io/cas/4.0.x/planning/Architecture.html

[16] Wikipedia. (07 de 01 de 2015). *Wikipedia*. Obtenido de http://en.wikipedia.org/wiki/Central_Authentication_Service

Protocolos Soportados

- CAS (versiones 1, 2 y 3)
- SAML 1.1
- OpenID
- OAuth (1.0, 2.0)

7.1.4 SimpleSAMLphp

SimpleSAMLphp es un software de código abierto escrito en php nativo, que implementa tanto un proveedor de identidad como un proveedor de servicio SAML2 completos, además de permitir la integración con numerosos protocolos y frameworks de autenticación y autorización [17].

SimpleSAMLphp también es compatible con otros protocolos y proveedores de identidad como Shibboleth 1.3, A-Select, CAS, OpenID, WS-Federation o OAuth

7.1.5 Shibboleth

Es un proyecto de código abierto desarrollado inicialmente en internet2 que implementa un sistema federado de SSO con intercambio de atributos basados en estándares abiertos, principalmente SAML. Además provee funcionalidad de privacidad extendida que permite al usuario / institución controlar los atributos liberados a cada aplicación.

Este sistema federado provee acceso seguro a través de diferentes dominios de seguridad, preservando la privacidad de los datos de los usuarios, y posibilita la escalabilidad del sistema a través de relaciones de confianza [18] [19].

---

[17] SimpleSAMLphp. (09 de 06 de 2015). *SimpleSAMLphp*. Obtenido de https://simplesamlphp.org/

[18] Barrancos, I. (28 de 12 de 2010). *tecnoquia*. Obtenido de http://tecnoquia.blogspot.com/2010/12/shibboleth-2-gestion-de-identidades.html

[19] Martin, S. (2012). *http://confia.aupa.info/*. Obtenido de http://confia.aupa.info/docs/cursos/2012/noviembre/anexo_software.html

Shibboleth es un producto final que mantiene la compatibilidad. Dispone de IdP, de SP y de DS (servicio de descubrimiento). Soporta el perfil SAML 2.0 Web Browser SSO, Cardspace, perfil Shibboleth, ADFS y SAML 1.1

IdP

- Está escrito en Java y funciona en cualquier contenedor servlet estándar.
- Permite múltiples fuentes de autenticación: Ldap, SQL, Kerberos.
- Los atributos pueden ser recogidos de la fuente o ser generados manualmente. Permitiendo posteriormente realizar la transformación con reglas previamente prefijadas o programadas.

SP

- Funciona en Apache, IIS y NSAPI. Entornos que pueden ser utilizados detrás de un proxy en Java y otros servidores web.
- Puede funcionar automáticamente o bajo demanda.
- Permite proteger el mismo los recursos o delegar en las aplicaciones para que manejen la autorización. (Clusterizable)

Ventajas

- Facilitar la gestión de múltiples contraseñas en múltiples aplicaciones
- Simplificar la gestión de cuentas de acceso de múltiples aplicaciones
- Preservar la privacidad de los usuarios
- Posibilitar la interacción entre organizaciones y sus usuarios
- Habilitar la posibilidad de que elijamos en la institución donde deseamos autenticarnos
- Permitir que los proveedores de servicios controlen el acceso a sus recursos
- Facilitar la integración rápida y efectiva de servicios de terceros dispares
- Provee un proveedor de identidad (IdP) en Java y un proveedor de servicio (SP) en C++, como módulo del servidor Web Apache
- Está basado en OpenSAML
- Dispone de tres versiones
- 1.3 que implementa SAML v1.1 en el IdP y SP
- 2.0 que implementa SAML v2.0 en el IdP y SP además de soportar SAML v1.1
- 3.0 que implementa SAML v3.0 en el IdP y SP además de soportar SAML v2.0 y SAML v1.1

7.2 Modelo de autenticación SSO propuesto para la Universidad del Tolima

Para dar solución al problema presentado en la Universidad de Tolima según sus requerimientos se plantea un modelo de Federación de Identidad mediante la utilización del software Shibboleth debido soporte, comunidad de desarrollo, facilidad de documentación y a las numerosas aplicaciones que tienen en la actualidad soporte de Shibboleth en las cuales se encuentran varias de las usadas por la Universidad del Tolima, como lo son Moodle y Ezproxy.

7.2.1 Autenticación y Autorización de Identidad

El sistema de Autenticación y Autorización de Identidad (AAI) permite a usuarios de distintas instituciones acceder a recursos compartidos mediante procedimientos de SSO (Single Sign-On). Shibboleth es una implementación libre y de código abierto de la Federación de Identidad basado en SAML (Security Assertion Markup Language), esta implementación se divide en distintos componentes, los cuales son: Proveedor de Identidad, Proveedor de Servicio y el Servicio de Descubrimiento.

El Proveedor de Identidad provee una interfaz web al usuario final, de esta forma ingresa sus credenciales y luego el sistema realiza la verificación de autenticación ante el directorio LDAP. La interfaz web es personalizable para mantener la identidad institucional.

El Proveedor de Servicio es aquel recurso que se desea tener acceso mediante la autenticación. Existen distintas formas de implementar un Proveedor de Servicio, entre los más conocidos es a través de la instalación de un módulo de Apache pero también se conocen plugins de los principales CMS como Drupal, Joomla, Wordpress, entre otros.

El Servicio de Descubrimiento provee también una interfaz web para el usuario final que consta de una lista de Proveedores de Identidad los cuales pueden ser seleccionados para realizar la autenticación, esta información la provee un archivo que contiene la información de los distintos Proveedores de Identidad, a esto se le conoce como el problema de WAY (Where are you from?).

Figura 6 Lista de Aplicaciones que tienen en la actualidad soporte de Shibboleth.

- American Chemical Society
- ArtSTOR
- Atypon
- CSA
- Digitalbrain PLC
- EBSCO Publishing
- Elsevier ScienceDirect
- ExLibris
- H.W. Wilson
- JSTOR
- The Literary Encyclopedia
- Metapress
- NSDL
- OCLC
- Ovid Technologies Inc.
- Project MUSE
- Proquest Information and Learning
- Serials Solutions
- SCRAN
- Schweizerisches Bundesgericht
- Thomson Gale
- Thomson Reuters
- Useful Utilities - EZproxy

- Blackboard
- CLIX
- Fronter
- ILIAS
- INSTRUCT
- Moodle
- OLAT
- Sakai
- WebAssign
- WebCT

- ActiveShareFS 2007 (for SharePoint 2007)
- ActiveShareFS 2010 (for SharePoint 2010)
- Apcoa Parking Permit Management
- Bodington.org
- Box.com Filesharing -enterprise edition supports SAML
- Condor
- Confluence Wiki
- Darwin Streaming Server
- Dokuwiki
- DokuWiki authentication backend
- Drupal
- Druva inSync
- DSpace
- eAcademy
- EPrints Repository
- Fedora Repository
- Filesender (for large file transfers)
- Google Apps/Email
- GridSphere
- GridShib
- GroupGTI TargetConnect -Graduate Careers

Fuente: https://wiki.shibboleth.net/confluence/display/SHIB2/ShibEnabled

- **Seguridad Shibboleth**

Shibboleth se basa en la sensibilidad de la seguridad del acceso a los recursos protegidos, por lo cual utiliza técnicas para proteger el tránsito de los atributos contemplados en el estándar SAML, usando un canal seguro utilizando algoritmos de cifrado SSL y certificados digitales.

- **Arquitectura de Shibboleth y sus Componentes.**

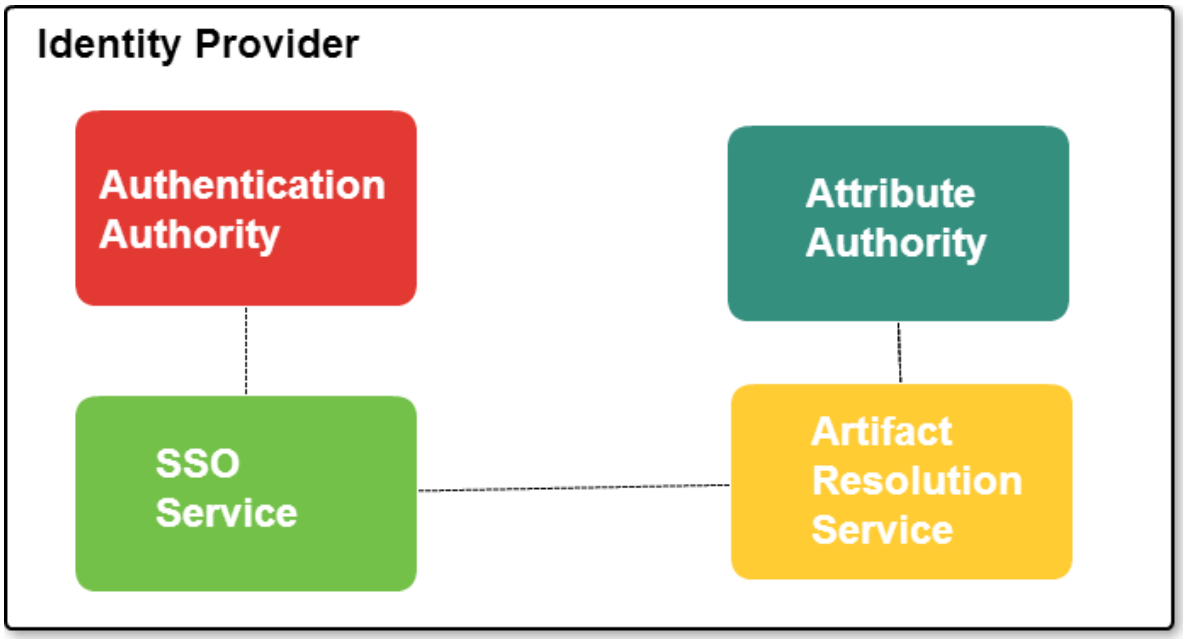
Shibboleth utiliza el estándar SAML como base de autenticación federada, se divide en componentes funcionales que son fundamentales:

- **Proveedor de Identidad o Identity Provider**

El Proveedor de Identidad provee una interfaz web al usuario final, de esta forma ingresa sus credenciales y luego el sistema realiza la verificación de autenticación ante el directorio LDAP. La interfaz web es personalizable para mantener la identidad institucional.

Los componentes del IdP se describen a continuación.

Figura 7 Identity Provider



Fuente: https://dev.e-taxonomy.eu/trac/wiki/ShibbolethProxy

- **Authentication Authority**
  Se encarga de expedir afirmaciones de autentificación. Interacciona con el Servicio Single-Sign On.

- **Single Sign-On Service**
  Constituye el primer punto de contacto del IdP. Inicia el proceso de autentificación y, en último lugar, redirige al cliente al Proveedor de servicios. Este componente no se definía en la versión 1.1 del estándar SAML, siendo uno de los aspectos en los que Shibboleth realiza procesos mejor que el estándar.

- **Artifact Resolution Service**
  En determinados situaciones, el Proveedor de Identidad devuelve un SAML artifact de nuevo al Proveedor de Servicios, en lugar de dar la afirmación propiamente dicha. El SP envía entonces dicho artifact al Artifact Resolution Service, utilizando algún canal alternativo de comunicación. Como respuesta, el Proveedor de Identidad le devuelve la aserción de autentificación requerida.

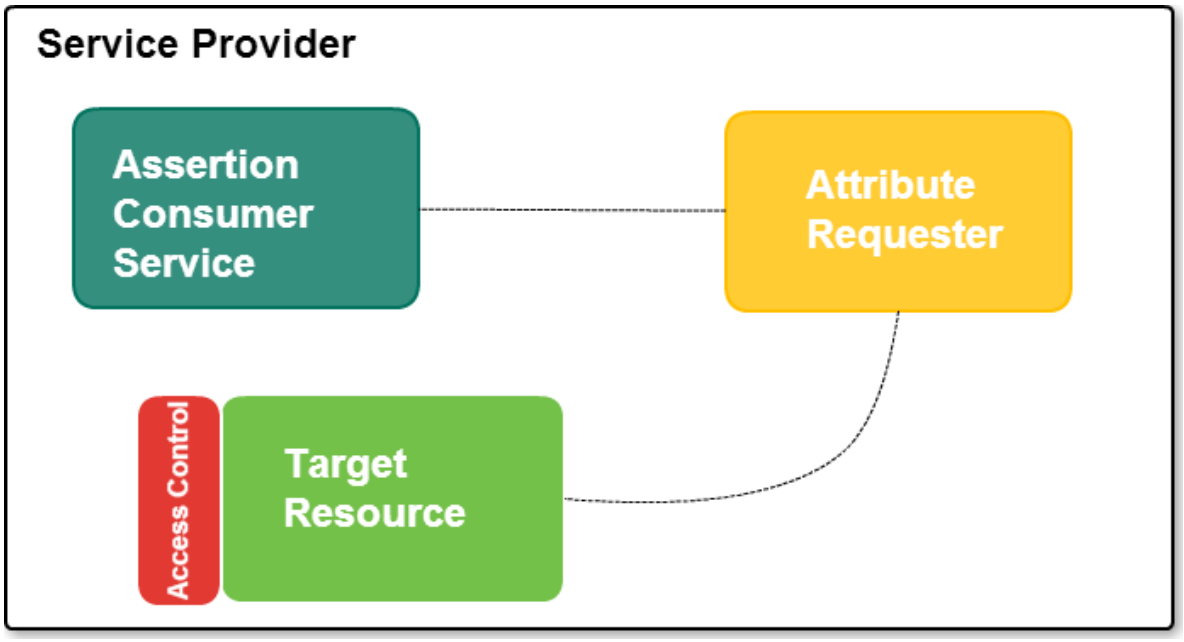
- **Attribute Authority**
  Se encarga de procesar peticiones de atributos (“attribute requests”) y emite aserciones de atributos.

7.2.2 Proveedor de Servicio o Service Provider

El Proveedor de Servicio es aquel recurso que se desea tener acceso mediante la autenticación. Existen distintas formas de implementar un Proveedor de Servicio, entre los más conocidos es a través de la instalación de un módulo de Apache pero también se conocen plugins de los principales CMS como Drupal, Joomla, Wordpress, entre otros.

El Proveedor de Servicio gestiona los recursos protegidos cuyo acceso se basa en la información que recibe del proveedor de identidad.

Figura 8 Componentes del SP



Fuente: El autor

- **Assertion Consumer Service**
  Representa el interfaz utilizado para comunicarse con el Servicio Single Sign-On del Proveedor de Identidad. Procesa la aserción de autentificación devuelta por éste (o bien, por el Artifact Resolution Service, dependiendo del caso de uso), inicia una petición de atributos al IdP (opcional), establece un contexto de seguridad en el Proveedor de Servicios para el usuario actual y redirige al cliente al recurso deseado.

- **Attribute Requester**
  Una vez que ha sido establecido un contexto de seguridad en el Service Provider, el Attribute Requester puede llevar a cabo un intercambio de atributos comunicándose con la Attribute Authority del Proveedor de Identidad.

- **Access Control**
  El Proveedor de Servicios debe proveer algún medio para evitar el libre acceso a los recursos protegidos, permitiendo la intervención del Proveedor de Identidad para supervisar el control de acceso.

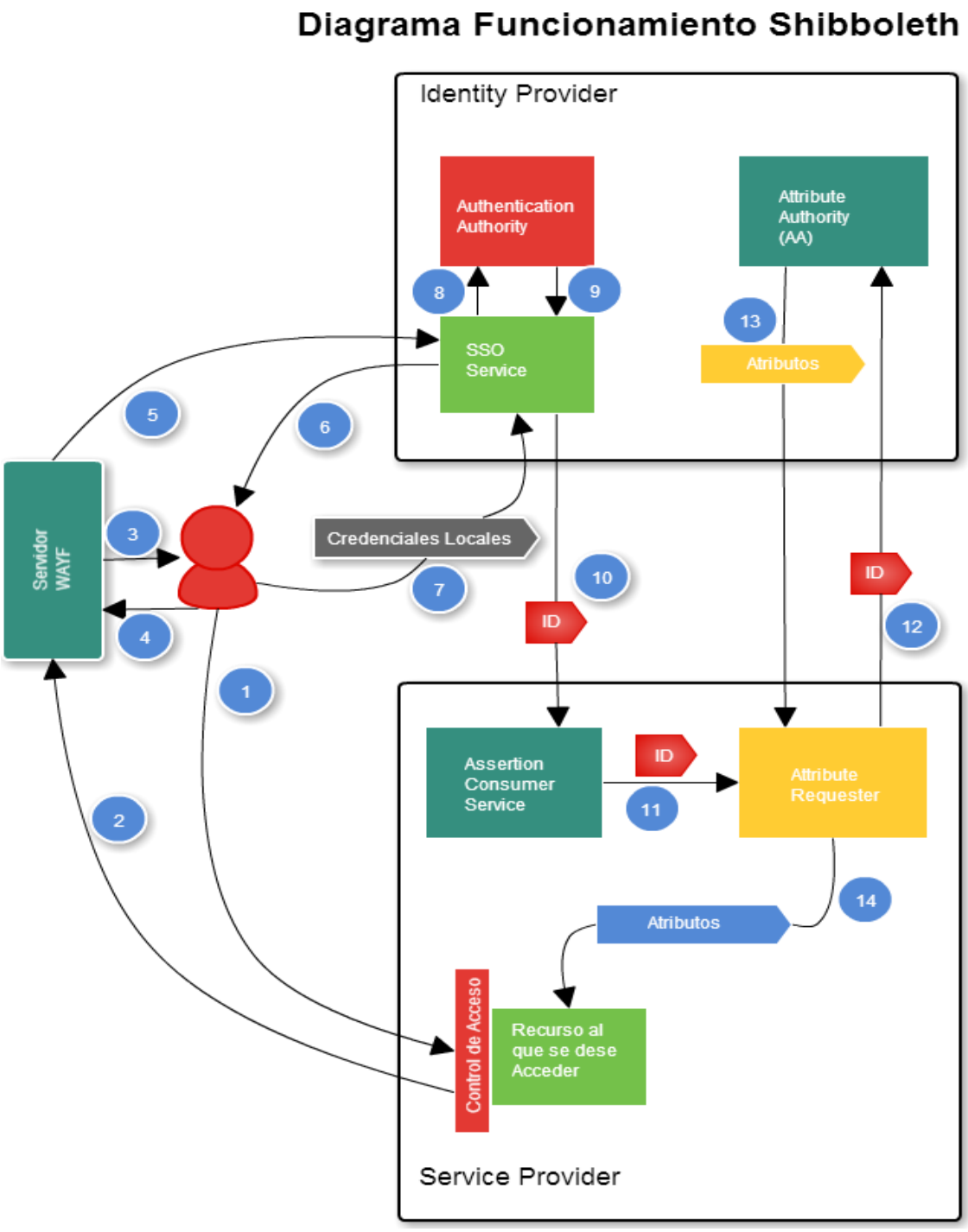
7.2.3 Servidor de Descubrimiento (WAYF)

El Servicio de Descubrimiento provee también una interfaz web para el usuario final que consta de una lista de Proveedores de Identidad los cuales pueden ser seleccionados para realizar la autenticación, esta información la provee un archivo que contiene la información de los distintos Proveedores de Identidad, a esto se le conoce como el problema de WAYF (Where are you from?).

Esquema de Funcionamiento

En la figura se describe todo el proceso que se realiza al interior de Shibboleth cuando un usuario desea acceder a un recurso protegido. En este proceso de funcionamiento intervienen los tres elementos mencionadas anteriormente. Proveedor de Identidad, Proveedor de Servicios y Servicio de Descubrimiento.

Figura 9 Funcionamiento Shibboleth



Fuente: El autor

La interacción de funcionamiento descrita en el diagrama se define de la siguiente manera:

1 El usuario intenta acceder a un recurso protegido del Proveedor de Servicio, el cual se encuentra situado “detrás” de un control de acceso.

2 El control de acceso no conoce al usuario ni de qué sistema proviene, por lo que lo redirige al servidor WAYF.

3 El Servidor WAYF inicia un proceso para averiguar el Proveedor de Identidad que él usuario desea. El proceso puede ser automático o, como en este caso, interactivo, presentando ante el usuario una lista de posibles Proveedores de Identidad, de entre los cuales el usuario elegirá una.

4 El usuario le indica al Servidor WAYF en qué Proveedor de Identidad desea autentificarse.

5 El Servidor WAYF redirige al usuario al Proveedor de Identidad adecuado.

6 El Servicio Single Sign-On, que forma parte del IdP elegido, pregunta al usuario por sus credenciales en dicho sistema, con el fin de autentificarlo.

7 El usuario responde enviando al Proveedor de Identidad sus credenciales locales (por ejemplo, el nombre de usuario que posee en dicho sistema y su contraseña).

8 El Servicio Single Sign-On comprueba que las credenciales del usuario son correctas y envía una petición de autentificación SAML a la Authentication Authority dentro del mismo Proveedor de Identidad.

9 La Authentication Authority devuelve una aserción SAML de autentificación como respuesta a la petición del Servicio Single Sign-On.

10 El Proveedor de Identidad genera un identificador único (ID) y redirige al usuario al Proveedor de Servicios, para que entregue la aserción de autentificación al Assertion Consumer Service.

11 El Assertion Consumer Service valida la aserción que acaba de recibir, crea una sesión de seguridad para el usuario y transfiere el control de ejecución al Attribute Requester.

12 El Attribute Requester utiliza el identificador que generó el Proveedor de Identidad en el paso 10 para solicitar los atributos del usuario. La solicitud va dirigida a la Attribute Authority, situada en el Proveedor de Identidad.

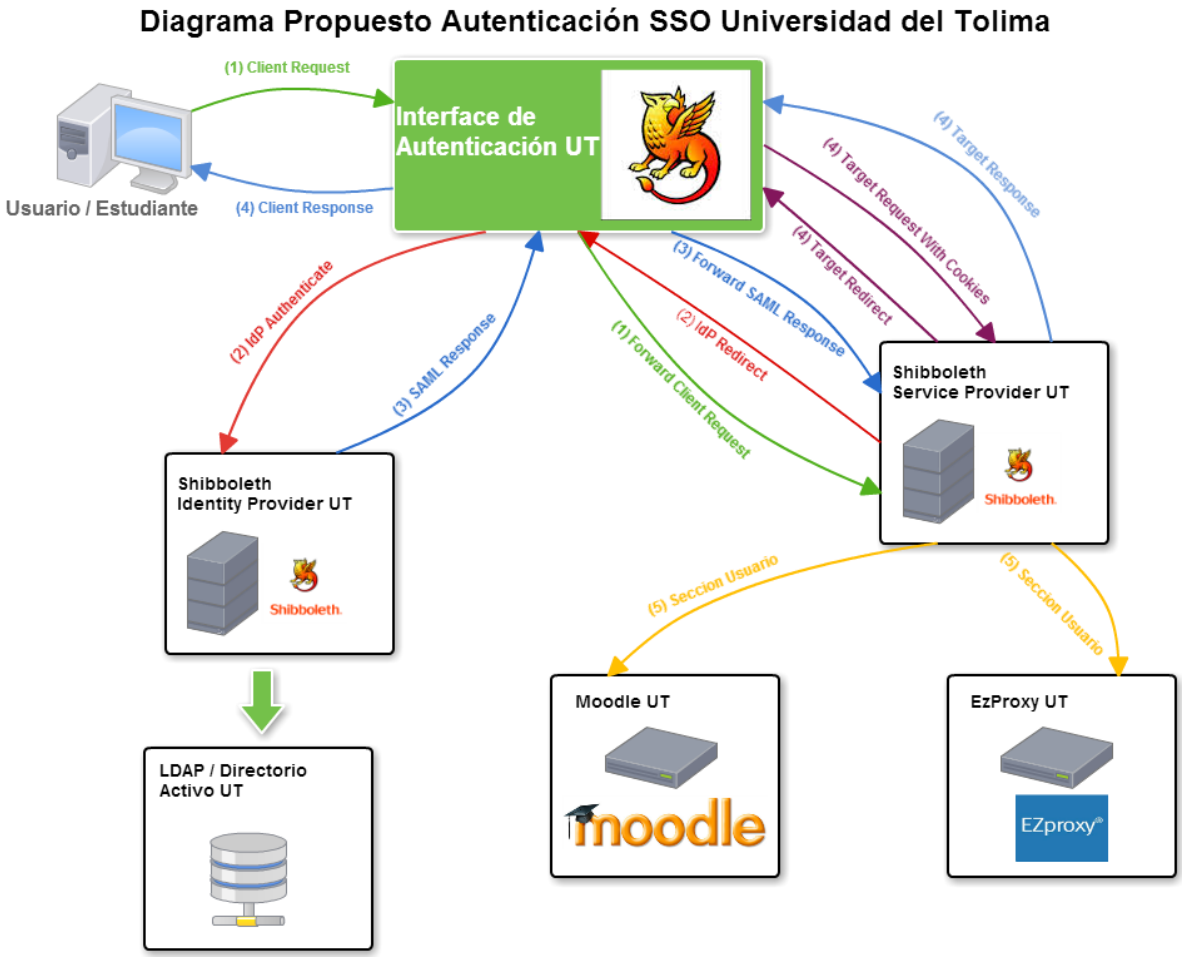
13 La Attribute Authority del IdP responde con una aserción SAML de atributos. Qué y cuántos atributos componen la respuesta, depende de la política de entrega de atributos que establezca el Proveedor de Identidad.

14 El Service Provider utiliza los atributos recibidos para decidir si permite al usuario acceder al recurso deseado, o bien rechaza dicho intento de acceso.

7.3 Modelo de Implementación Universidad del Tolima

A continuación se describe un modelo de implementación de autenticación federada mediante el uso de la herramienta Shibboleth, con dos de sus componentes esenciales para su funcionamiento, como lo son el Identity Provider (IdP) y Service Provider (SP). Centralizando los datos de usuario mediante un servidor LDAP, sirviendo los usuarios al componente Identity Provider que junto con el componente Service Privider servirán como un sistema único de autenticación SSO para las plataformas Moodle y Ezproxy con que cuenta la Universidad del Tolima.

Figura 10 Diagrama SSO Universidad del Tolima



Fuente: El autor

Siguiendo el modelo propuesto se requiere de la instalación de:

- Servidor LDAP
- Servidor con el componente de Shibboleth IdP
- Servidor con el componente de Shibboleth SP
- Servidor de prueba con una instalación de Moodle
- Servidor de Prueba con una Instalación de Ezproxy

Para las instalaciones se trabaja con sistema operativo Linux Centos, debido a que son de uso libre, compatibilidad con los componentes a instalar y al amplio soporte de repositorios de software.

LDAP

El siguiente procedimiento está basado en un sistema operativo Linux Centos 6.5 x86_64, el mismo puede variar según la distribución utilizada, para la instalación en otros sistemas operativos puede consultar la documentación del software en: http://www.openldap.org/

Instalar OpenLDAP Server y Cliente mediante los paquetes.

```
# yum install openldap openldap-servers openldap-clients –y
```

Los archivos de configuración quedan en la ubicación /etc/openldap

Copiar el archivo slapd.conf en /etc/openldap

```
# cp /usr/share/openldap-servers/slapd.conf.obsolete /etc/openldap/slapd.conf
```

Crear una contraseña para el administrador (rootdn):

```
# slappasswd

New password:
Re-enter new password:
{SSHA}GtG8bcLGeN/rf1iStKFK2pu0C2EZf/RX
```

Copiar la contraseña generada en el archivo slapd.conf y realizar las siguientes modificaciones:

```
vi /etc/openldap/slapd.conf
```

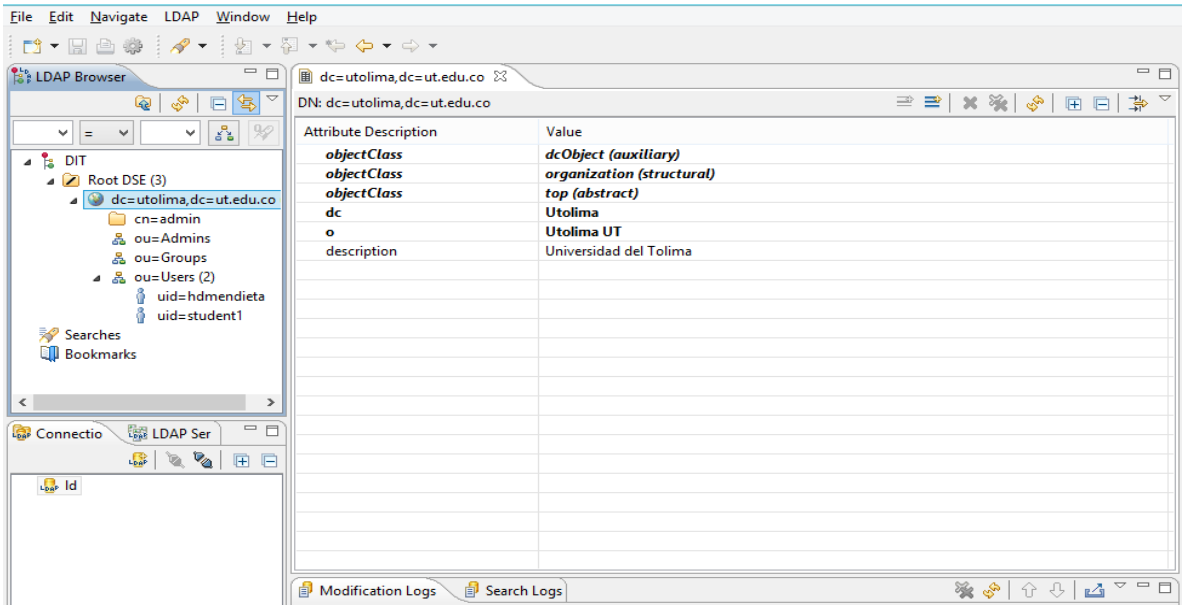
```
access to *
by dn.exact="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth" read
by dn.exact="cn=Manager,dc=example,dc=com" read
 by * none


database       bdb
suffix         "dc=example,dc=com"
checkpoint     1024 15
rootdn         "cn=Manager,dc=example,dc=com"
rootpw         {SSHA}GtG8bcLGeN/rf1iStKFK2pu0C2EZf/RX
loglevel       256
sizelimit      unlimited
```

Iniciar el servicio slapd

```
# service slapd start
```

Realizar conexión con un cliente para verificar la instalación y realizar administración del servidor LDAP

Figura 11 Comprobando la instalación del servidor LDAP



Fuente: El autor

7.3.1 **Proveedor de identidad (Identity provider)**

El siguiente procedimiento está basado en un sistema operativo Linux Centos 6.5 x86_64, el mismo puede variar según la distribución utilizada, para la instalación en otros sistemas operativos puede consultar la documentación del software en: https://wiki.shibboleth.net/confluence/display/SHIB2/IdPInstall.

Descargar el software del Shibboleth Identity Provider

Descargar el software para la instalación de Shibboleth Identity Provider

#: cd /usr/local/src/
#: wget http://shibboleth.net/downloads/identity-provider/latest/shibboleth-identityprovider-2.4.0-bin.tar.gz

Descomprimir el software en /usr/local/src
#: tar xvf shibboleth-identityprovider-2.4.0-bin.tar.gz

Instalar Tomcat 6 mediante el gestor de paquetes de Centos

#: yum install –y java-1.6.0-openjdk tomcat6

Comprobar la versión de Java que se utiliza en el sistema

#: alternatives –config java

Agregar Tomcat6 al inicio del sistema

#: chkconfig tomcat6 on

Configuración de Tomcat 6 para desplegar la webapp del Software Shibboleth Identity Provider

Se Preparar Tomcat 6 para la ejecución de Shibboleth IdP, se Configurar SSL para Tomcat y se Configurar el usuario Tomcat y asignar la ruta a las librerías

Copiar las librerías avaladas al directorio de Tomcat 6
#: cp -r /usr/local/src/shibboleth-identityprovider-2.4.0/endorsed/ /usr/share/tomcat6/

Descargar la librería dta-ssl y ubicarla en el directorio de Tomcat 6
#: cd /usr/share/tomcat6/lib/
#: wget https://build.shibboleth.net/nexus/content/repositories/releases/edu/internet2/middleware/security/tomcat6/tomcat6-dta-ssl/1.0.0/tomcat6-dta-ssl-1.0.0.jar

Se modificar el usuario Tomcat y agregar la ruta a las librerías avaladas, para esto se modifica el archivo /etc/sysconfig/tomcat6 y modificar/agregar la siguiente información:
#: vi /etc/sysconfig/tomcat6

TOMCAT_USER="root"
JAVA_ENDORSED_DIRS="/usr/share/tomcat6/endorsed"

Modificar el archivo /usr/share/tomcat6/conf/server.xml y agregar el siguiente contenido.
#: vi /usr/share/tomcat6/conf/server.xml

```
<Connector
port="443"
protocol="HTTP/1.1"
SSLEnabled="true"
maxThreads="150"
scheme="https"
secure="true"
clientAuth="false"
sslProtocol="TLS"
keystoreFile="/opt/shibboleth-idp/credentials/idp.jks"
keystorePass="contraseña" />

<Connector port="8443"
  protocol="org.apache.coyote.http11.Http11Protocol"
SSLImplementation="edu.internet2.middleware.security.tomcat6.DelegateToApplicationJSSEImplementation"
  scheme="https"
  SSLEnabled="true"
  clientAuth="true"
  keystoreFile="/opt/shibboleth-idp/credentials/idp.jks"
  keystorePass="contraseña" />
```

El parámetro keystoreFile debe ser modificado con la ruta al archivo del Certificado Digital que asegura el dominio idp.<dominio asignado por la institución> en este caso utsso.ut.edu.co y el parámetro keystorePass debe ser modificado con la contraseña del Certificado Digital.

Se recomienda utilizar el Certificado Digital en formato Public Key Cryptography Standards (PKCS) #12 (.p12 o .pfx).

NOTA: El script de instalación generará unos Certificados Digitales autofirmados para el funcionamiento del software pero se recomienda utilizar Certificados Digitales emitidos por autoridades de certificación bien conocidas.

Crear el archivo /usr/share/tomcat6/conf/Catalina/localhost/idp.xml y agregar el siguiente contenido:

#: vi /usr/share/tomcat6/conf/Catalina/localhost/idp.xml

```
<Context docBase="/opt/shibboleth-idp/war/idp.war"
privileged="true"
antiResourceLocking="false"
antiJARLocking="false"
unpackWAR="false"
swallowOutput="true" />
```

Esto crea un contexto de despliegue de la aplicación web del IdP, por defecto la ruta al archivo es /opt/shiboleth-idp/war/idp.war, recuerde modificar la ruta si el software ha sido instalado en un directorio diferente.

Instalación del Software Shibboleth Identity Provider

Definir la variable de entorno JAVA_HOME

#: export JAVA_HOME=/usr/lib/jvm/jre-1.6.0/

Ejecutar el script de instalación de Shibboleth IdP

#: cd /usr/local/src/ shibboleth-identityprovider-2.4.0/
#: ./install.sh

El script solicitará la siguiente información:

- Ingresar la ruta del directorio donde desea que sea instalado el software de Shibboleth IdP, por defecto la ruta es /opt/shibboleth-idp/

- Ingresar el FQDN (Fully Qualyfied Domain Name) de su IdP, se recomienda nombrarlo como idp.<utsso.ut.edu.co.

- Ingresar la contraseña del certificado que será generado por el script, si utiliza un Certificado Digital no autofirmado esto no afectará.

- Se logrará ver el mensaje “BUILD SUCCESSFUL” luego de la ejecución correcta del script.

Verificación de la instalación

Agregar el servidor Tomcat 6 al inicio del sistema y posteriormente iniciar el servicio.

#: chkconfig tomcat6 on
#: service tomcat6 start

Verificar el despliegue de la aplicación en los logs del servidor Tomcat 6.

#: tail –f –n 50 /usr/share/tomcat6/logs/catalina.out

Verificar en la tabla de procesos del sistema la ejecución del servidor Tomcat 6

#: ps ax | grep tomcat

Mediante el comando netstat los puertos SSL 443 y 8443 verificar que estén abiertos para el establecimiento de conexiones.

#: netstat -tupan | grep 443

Agregar las reglas del firewall iptables y reiniciar el servicio iptables

#: vi /etc/sysconfig/iptables.

-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT

-A INPUT -m state --state NEW -m tcp -p tcp --dport 8443 -j ACCEPT

Reiniciar el servicio iptables

service iptables restart

Verificar que el IdP está en ejecución sin errores

#: curl -k https://localhost/idp/profile/Status

También puede verificar ingresando al siguiente URL desde un equipo remoto https://utsso.ut.edu.co/idp/profile/Status/ donde podrá observar una página en blanco con un pequeño “ok” que indica que está funcionando correctamente

Configurar autenticación con un servidor LDAP

Se configuración de los parámetros de conexión del servidor LDAP para la autenticación de los usuarios, se configurar el método de autenticación del Manejador de Autenticación y se configurar el método de autenticación que utiliza el IdP por defecto.

Editar el archivo /opt/Shibboleth-idp/conf/login.conf ig con la siguiente información

```
ShibUserPassAuth {

// Example LDAP authentication
// See: https://wiki.shibboleth.net/confluence/display/SHIB2/IdPAuthUserPass
   edu.vt.middleware.ldap.jaas.LdapLoginModule required
      ldapUrl="ldaps://ldap.example.com:636/"
      baseDn="ou=people,dc=example,dc=org"
      bindDn="uid=IdPServiceAcct,ou=people,dc=example,dc=org"
      bindCredential="password"
      userFilter="uid={0}";
};
```

Modifique los campos con la información correspondiente a su directorio LDAP, tales como el URL con el puerto 389 para el protocolo LDAP y 636 para el protocolo

LDAPS, el BaseDN (Base Distinguished Name) que es el árbol de los usuarios de su directorio.

Los campos bindDN y bindCredential son las credenciales de un usuario del directorio LDAP con privilegios para realizar consultas en el mismo directorio (Se recomienda crear un usuario específico para dicha gestión).

Configurar el método de autenticación, para esto edite el archivo /opt/shibboleth-idp/conf/handler.xml y realice las siguientes modificaciones:

- Descomentar el método de autenticación UsernamePassword:

```
<!--  Username/password login handler -->
<ph:LoginHandler xsi:type="ph:UsernamePassword"
jaasConfigurationLocation="file:///opt/shibboleth-idp/conf/login.config">
<ph:AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport</ph:AuthenticationMethod>
</ph:LoginHandler>
```

- Comentar/Eliminar el método de autenticación RemoteUser:

```
<!-- Login Handlers -->
<!--
<ph:LoginHandler xsi:type="ph:RemoteUser">
<ph:AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified
</ph:AuthenticationMethod>
</ph:LoginHandler>
-->
```

Editar el archivo /opt/shibboleth-idp/conf/relying-party.xml y modificar en la sección Relying Party Configuration la siguiente línea:

<rp:DefaultRelyingParty provider="https://idp.example.com/idp/shibboleth" defaultSigningCredentialRef="IdPCredential">

Quedando como sigue:

<rp:DefaultRelyingParty provider="https://idp.example.com/idp/shibboleth" defaultSigningCredentialRef="IdPCredential" defaultAuthenticationMethod="urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport">

Reiniciar el servidor Tomcat 6

#: service tomcat6 restart

Figura 12 Prueba del funcionamiento del IdP obteniendo los metadatos

Fuente: El autor

7.3.2 Proveedor de servicio (service provider)

Pasos para la Instalación del Shibboleth Service Provider

El siguiente procedimiento está basado en un sistema operativo Linux Centos 6.5 x86_64, el mismo puede variar según la distribución utilizada, para la instalación en otros sistemas operativos puede consultar la documentación del software en: https://wiki.shibboleth.net/confluence/display/SHIB2/Installation.

Instalar servidor web Apache

Instalar el servidor web Apache mediante el gestor de paquetes yum

#: yum –y install httpd

Edite el archivo /etc/httpd/conf/httpd.conf y edite las siguientes directivas:

Servername spsso.ut.edu.co

UseCanonicalName On

Configurar Apache solo para conexiones seguras mediante SSL/TLS

Instalar el módulo de SSL para apache

#: yum –y install mod_ssl

Editar el archivo /etc/httpd/conf/httpd.conf y active el módulo RewriteEngine agregando las siguientes reglas al inicio del archivo.

#: vi /etc/httpd/conf/httpd.

RewriteEngine On

RewriteCond %{HTTPS} off

RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}

Esto direcciona las solicitudes HTTP hechas a través del puerto 80 a utilizar el protocolo HTTPS mediante el puerto 443, por lo que debe verificar que el servicio esté utilizando el puerto 80 descomentando la siguiente directiva:

Listen 80

Desbloquear los puertos 80 y 443 en los cortafuegos por defecto del sistema, edite el archivo /etc/sysconfig/iptables y agregue las siguientes reglas:

#: vi /etc/sysconfig/iptables

-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT

Luego se reiniciar el servicio iptables

#: service iptables restart

Iniciar el servidor web Apache

#: service httpd start

Se verifican los puertos 80 y 443 mediante el siguiente comando:

#: netstat –tupan | grep httpd

NOTA: Por defecto Apache utilizará los Certificados Digitales autofirmados generados por defecto en la instalación.

Los Certificados Digitales del servidor Apache no interfieren en la instalación del Proveedor de Servicio de Shibboleth, por esto debe realizar el cambio de Certificados Digitales para el servicio web a conveniencia.

Instalar el módulo de Shibboleth

Instalar el módulo de Shibboleth para el servidor web Apache

Agregar los repositorios de Shibboleth

#: curl -o /etc/yum.repos.d/security:shibboleth.repo
http://download.opensuse.org/repositories/security://shibboleth/CentOS_CentOS-6/security:shibboleth.repo

Instalar el software de Shibboleth y reiniciar el servicio de httpd para cargar el módulo Shibboleth para Apache.

#: yum –y install shibboleth
#: service httpd restart

Agregar el servicio al inicio automático del sistema y luego iniciarlo

#: chkconfig shibd on
#: service shibd start

Verificar el estado del servicio Shibboleth

curl –k https://localhost/Shibboleth.sso/Status

Configurar el Proveedor de Servicio

La configuración del servicio de Shibboleth consiste principalmente en realizar cambios en el archivo /etc/shibboleth/shibboleth2.xml

Se establecer el EntityID adecuado, se configurar el SP para utilizar Cookies seguras y se establecer el correo del administrador del SP

Editar el archivo /etc/shibboleth/shibboleth2.xml
#: vi /etc/shibboleth/shibboleth2.xml

Modifique la siguiente línea y establezca el atributo EntityID acorde al FQDN de su SP:

<ApplicationDefaults
entityID="https://spsso.ut.edu.co/shibboleth"
REMOTE_USER="eppn persistent-id targeted-id">

Modificar la siguiente línea para que el SP maneje Cookies de forma segura, el resultado de la modificación debe ser:

<Sessions lifetime="28800" timeout="3600"
checkAddress="false"
relayState="ss:mem" handlerSSL="true"
cookieProps="; path=/; secure; HttpOnly">

Modifique la siguiente sección con el correo electrónico de la persona encargada de la administración del Proveedor de Servicio:

<Errors supportContact="admi@ut.edu.co"
helpLocation="/about.html"
  styleSheet="/shibboleth-sp/main.css"/>

Configurar el Certificado Digital de nuestro SP

Configurar el Service Provider para que utilice el Certificado Digital correcto
Ubicar el Certificado Digital en la siguiente ruta /etc/shibboleth/certs/ (Recomendado)

#: mkdir /etc/shibboleth/certs/
#: mv /path/to/certificate_and_key/* /etc/shibboleth/certs/

Editar el archivo /etc/shibboleth/shibboleth2.xml y modifique las rutas para que utilice propiamente su Certificado Digital junto con la Llave Privada

#: vi /etc/shibboleth/shibboleth2.xml

<!-- Simple file-based resolver for using a single keypair. -->

<CredentialResolver type="File"
key="/etc/shibboleth/certs/<certificado>.key"
certificate="/etc/shibboleth/certs/<certificado>.crt"/>

-key es el atributo de la Llave Privada

-certificate es la ruta propiamente del Certificado Digital

Reiniciar el servicio shibd

#: service shibd restart

Figura 13 Prueba del funcionamiento del servicio SP obteniendo los metadatos

Fuente: El autor

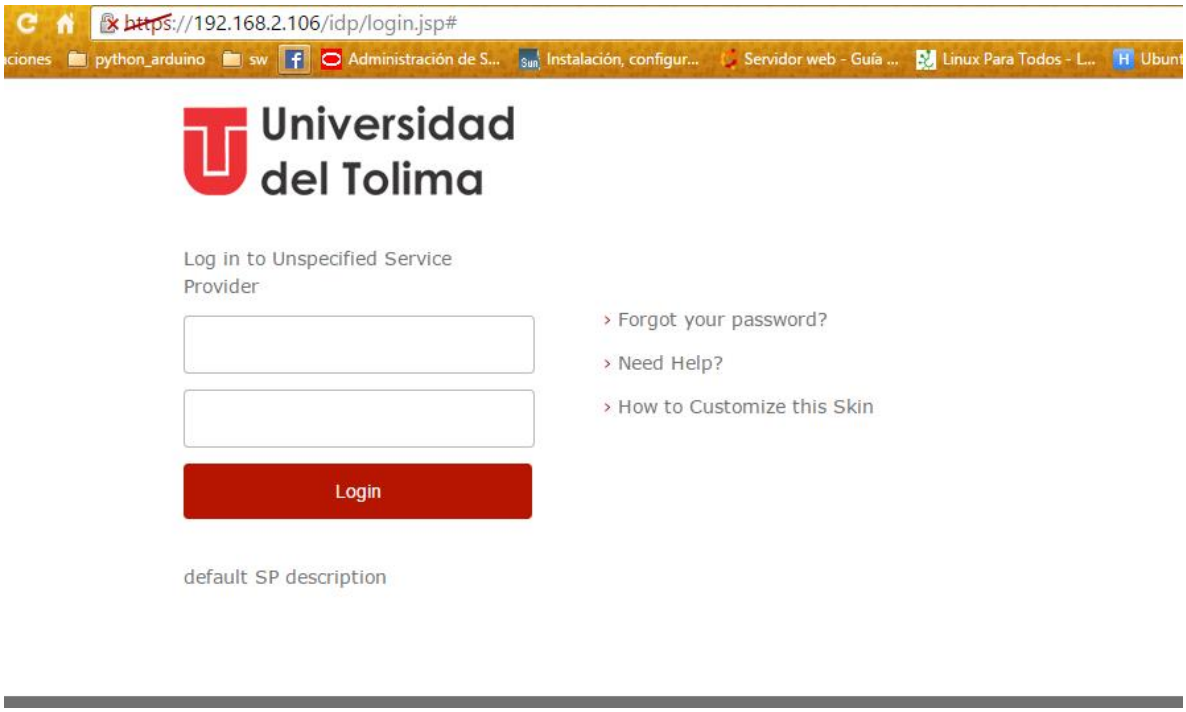
Ingresar a nuestro SP mediante Single Sign On

Realizar el inicio de sesión en nuestro Service Provider con nuestras credenciales institucionales.

Ingresar mediante un navegador web al siguiente URL:

https://spsso.ut.edu.co/secure

Figura 14 Prueba de Ingreso



Fuente: El autor

8 NOMBRES DE LAS PERSONAS QUE PARTICIPAN EN EL PROCESO

El desarrollo del proyecto de investigación será desarrollado por estudiantes de la Especialización de Seguridad informática de la UNAD y coordinados por el director de proyectos, descritos en la siguiente tabla:

Tabla 1 Participantes

NOMBRE	ROLES	PERFIL PROFESIONAL
MARTÍN CAMILO CANCELADO RUIZ	Director proyecto	✓ Ingeniero de Sistemas
HERNAN DARIO MEDIETA	Estudiante	✓ Ingeniero de Sistemas ✓ Tecnólogo en electrónica
FRANCISCO ANDRADE NAVARRO	Estudiante	✓ Ingeniero de Sistemas ✓ Tecnólogo en electrónica

9 RECURSOS DISPONIBLES

a. **Materiales:** Se disponen de los siguientes recursos para llevar acabo el desarrollo del proyecto:

   - ✓ Servidores
   - ✓ Red cableada y Wi-Fi
   - ✓ Computadores de escritorio
   - ✓ Portátiles
   - ✓ Servidor de Bases de datos

b. **Institucionales:** Se disponen acceso y documentación de los diferentes sistemas de información con que cuenta la institución.

   - ✓ Plataforma académica
   - ✓ Plataforma administrativa
   - ✓ Moodle
   - ✓ Ezproxy

c. **Financieros:** Para la financiación de dicho proyecto se cuenta con un presupuesto por parte de la administración de la Universidad del Tolima, que ayudara a realizar la puesta en marcha y sostenimiento de la misma, es de aclarar que se cuanta con el ambiente, equipos, sistemas de información, redes y tecnologías necesarias para llevar acabo el desarrollo del proyecto, que se realizara por medio de soluciones de software libre.

Tabla 2 presupuesto

Item	Elemento	Cantidad	Valor Unitario	Valor Total
1	servidores	1	$8.000.000	$8.000.000
2	Computadores de Escritorio	2	$2.000.000	$4.000.000
3	Swiche	1	$4.200.000	$4.200.000
4	Vmware Exi 5.5 Lic. Académica	1	$2.100.000	$2.100.000
5	Sistema Operativo Linux	5	N/A	N/A
6	Internet	Varios	N/A	$200.000
7	Papelería	Varios	N/A	$20.000
			Total	**18.520.000**

10 CRONOGRAMA

Figura 15 Cronograma de actividades

Título del proyecto: SISTEMA CENTRALIZADO DE GESTIÓN DE USUARIOS PARA LA UNIVERSIDAD DEL TOLIMA.

CRONOGRAMA DEL PROYECTO													
	MESES	M1				M2				M3			
		SEMANAS				SEMANAS				SEMANAS			
Actividad del Proyecto*													
PROCESO DE PLANIFICACIÓN													
Elaborar acta constitución del proyecto													
Validar alcance con directivos													
Revisar documentación													
IDENTIFICACIÓN DE REQUERIMIENTOS													
Identificacion de sistemas													
Identificacion de infraestructura													
Identificacion sistema de seguridad													
Documentar sistema de autenticacion													
Diseñar diagrama													
PROCESO DE PLANIFICACIÓN													
Documetar entregbles del proyecto													
Documentar plan del alcance													
Planificar recurso humano													
Documentar matriz de roles y funciones													
Documentar plantillas de seguimiento													
Documentar recursos disponibles													
PROCESO DE EJECUCIÓN													
Consideraciones sobre la implementación													
Generalidades de la implementación													
caracteristicas de la implementación													
Esquema de implantación													
Implantación													
PROCESO DE SEGUIMIENTO Y CONTROL													
Documentar estado del servicio													
Proceso de cierre del proyecto													
Generar informe final del proyecto													
efectuar modificaciones													
Acta de aceptación													
Revisión y aprobación													
cierre del proyecto													

CONCLUSIONES

- Mediante la construcción del Sistema Single Sing On se constató la posibilidad de interactuar con las diferentes aplicaciones con que cuenta la Universidad del Tolima brindando de esta manera un sistema de autenticación único y seguro, que brinde comodidad a los usuarios.

- El sistema Single Sing On, intercambia información sensible y protege estos datos mediante el uso de canales seguros, por lo que cualquier aplicación que gestione información sensible debe utilizar mecanismos para proteger tal información.

- La consulta de documentación y de foros relacionados a las tecnologías empleadas en el desarrollo del sistema de autenticación Single Sing On, redujo el tiempo de solución de problemas encontrados, por lo que se recomienda hacer uso de estos recursos al momento de implementar un sistema centralizado de usuarios.

BIBLIOGRAFÍA

ALADDIN . (s.f.). *Aladdin*. Obtenido de Gestión de contraseñas. Single Sign On: http://www3.es.safenet-inc.com/news/2006/eToken/gestion_contrasenyas.aspx

BARRANCOS, I. (28 de 12 de 2010). *tecnoquia*. Obtenido de http://tecnoquia.blogspot.com/2010/12/shibboleth-2-gestion-de-identidades.html

BOGOTÁ, A. m. (31 de 12 de 2008). *http://www.alcaldiabogota.gov.co/*. Obtenido de http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=34488

--------.-------- (05 de 01 de 2009). *http://www.alcaldiabogota.gov.co/*. Obtenido de http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=34492

--------.-------- (18 de 10 de 2012). *http://www.alcaldiabogota.gov.co/*. Obtenido de http://www.alcaldiabogota.gov.co/sisjur/normas/Norma1.jsp?i=49981

CAS. (09 de 06 de 2015). *Single Sign-On para la Web*. Obtenido de http://jasig.github.io/cas/4.0.x/planning/Architecture.html

CONFLUENCE. (22 de 10 de 2013). *Confluence*. Obtenido de https://wiki.shibboleth.net/confluence/display/SHIB2/ShibEnabled

ECURED. (09 de 06 de 2015). *EcuRed*. Obtenido de http://www.ecured.cu/index.php/WSO2

INTERGRAPHIC DESING. (15 de 06| de 2012). Obtenido de Single Sign On: Un solo login, múltiples accesos: http://www.intergraphicdesigns.com/blog/2012/06/15/single-sign-on-un-solo-login-multiples-accesos/

IVÁN M. CABALLERO, J. J. (07 de 2003). *Universidad de los Andes.* Obtenido de http://www.criptored.upm.es/guiateoria/gt_m142j.htm

MARTIN, S. (2012). *http://confia.aupa.info/*. Obtenido de http://confia.aupa.info/docs/cursos/2012/noviembre/anexo_software.html

OYUELA, S. G. (31 de 05 de 2014). *Salpicadero*. Obtenido de http://www.josso.org/confluence/display/JOSSO1/JOSSO+-+Java+Open+Single+Sign-On+Project+Home

SIMPLESAMLPHP. (09 de 06 de 2015). *SimpleSAMLphp*. Obtenido de https://simplesamlphp.org/

TRACPOWERED. (2013-2014). *http://www.edgewall.org/*. Obtenido de https://dev.e-taxonomy.eu/trac/wiki/ShibbolethProxy

UNIVERSIDAD DE CORDOBA. (03 de 2010). *Universidad de Cordoba*. Obtenido de Notas sobre requisitos de un sistema de Single Sign On: https://www.uco.es/servicios/informatica/sistemas/doc_ccc/fed_SSO/Requisitos_SSO.html

UNIVERSIDAD DE SEVILLA. (s.f.). *Univerisdad de Sevilla*. Obtenido de Sistema Single Sign-On: https://sso.us.es/integracion/

UNIVERSIDAD DEL TOLIMA. (13 de Agosto de 2013). *Principios y Valores*. Recuperado el 18 de Octubre de 2014, de http://www.ut.edu.co/administrativos/index.php/inti/quienes-somos/principios-y-valores

WEERAWARANA, S., & FREMANTLE, P. (09 de 06 de 2015). *WSO2*. Obtenido de http://wso2.com/products/identity-server/

WIKIPEDIA. (07 de 01 de 2015). *Wikipedia*. Obtenido de http://en.wikipedia.org/wiki/Central_Authentication_Service

ANEXOS

ANEXO A

	Formato para el estudio técnico de los diferentes servicios ofrecidos por la universidad del Tolima.	Versión 1.0

Proyecto:	Estudio SSO	Lugar:	Universidad del Tolima
Elaborado por:	Hernan Dario Mendieta Francisco Andrade N.	Oficina:	Gestión Tecnológica

Los estudiantes de la especialización en seguridad informática de la Universidad Nacional Abierta y a distancia UNAD, nos encontramos realizando una encuesta para conocer el grado de aceptación de los diferentes sistemas de información con que cuenta la universidad esto con el fin de realizar un estudio de aceptación que permita mejorar los servicios prestados, por tal motivo se solicita responder de forma clara, sencilla y veraz.

Código:

Nombre:	
Código	

¿Cuáles de los siguientes servicios utiliza usted en la Universidad del Tolima?
_Moodle _Software académico _Redes libres WI-FI _EzProxy UT (Biblioteca) Otro…

¿Con qué frecuencia utiliza los servicios de la Universidad del Tolima?
_Diario _Semanal _Mensual _Semestral

¿Cuál es el mayor inconveniente que ha tenido en el uso de los servicios ofrecidos por la Universidad del Tolima?
_Servicio deficiente _Complejidad de acceso _Falta de interés _Otro…..

¿En qué lugar del campus hace uso de los diferentes servicios prestados por la Universidad del Tolima?
_Zonas Wi-Fi _Salas de informática

¿Está satisfecho con los servicios prestados por la Universidad del Tolima?
_Totalmente de acuerdo _De acuerdo _En desacuerdo _No los usa

Aprobaciones	
HERNAN DARIO MENDIETA	Firma:
FRANCISCO ANDRADE	Firma:

ANEXO B

	UNIVERSIDAD DEL TOLIMA OFICINA DE GESTION TECNOLOGICA	Formato ep_sso Versión 1.0

Proyecto:	Estudio SSO	Lugar:	Universidad del Tolima
Elaborado por:	Hernan Dario Mendieta Francisco Andrade N.	Oficina:	Gestión Tecnológica

Plataforma a Evaluar	
Plataforma	Moodle
Versión	2.6
Tipo de Autenticación	Local
	Usuario y Contraseña
Origen de Datos de Usuario	Local (Tabla de Interna de Usuarios)
Cifrado de contraseña	SI
Plugins de Autenticación	Base De Datos Externa
	Autenticación por Webservice
	Autenticación Servidor CAS
	Autenticación Servidor LDAP
	Autenticación Shibboleth
	Autenticación Servidor Radius

Observaciones
La plataforma se encuentra autenticado a los usuarios por medio de la taba de usuarios interna, los cuales son creado de forma manual

Aprobaciones	
HERNAN DARIO MENDIETA	Firma:
FRANCISCO ANDRADE	Firma:

ANEXO C

	UNIVERSIDAD DEL TOLIMA OFICINA DE GESTION TECNOLOGICA	Formato ep_sso Versión 1.0

Proyecto:	Estudio SSO	Lugar:	Universidad del Tolima
Elaborado por:	Hernan Dario Mendieta Francisco Andrade N.	Oficina:	Gestión Tecnológica

Plataforma a Evaluar	
Plataforma	EzProxy
Versión	6.0
Tipo de Autenticación	Externa
	Usuario y Contraseña
Origen de Datos de Usuario	Servidor Radius
Cifrado de contraseña	SI
Plugins de Autenticación	Autenticación Servidor LDAP
	Autenticación Shibboleth
	Autenticación Servidor Radius

Observaciones
La plataforma se encuentra autenticado a los usuarios por medio de un servidor Radius, en el cual los usuarios son creado de forma manual

Aprobaciones	
HERNAN DARIO MENDIETA	Firma:
FRANCISCO ANDRADE	Firma: