UNIVERSIDAD DON BOSCO

VICERRECTORÍA ACADÉMICA FACULTAD DE INGENIERÍA

TRABAJO DE GRADUACIÓN

DEFINICIÓN DE LOS COMPONENTES PARA EL DESARROLLO DE UNA SOLUCIÓN SSO EN ENTORNOS EMPRESARIALES

PARA OPTAR AL GRADO DE

MAESTRO EN ARQUITECTURA DE SOFTWARE

ASESOR

MAESTRO JOSÉ MARIO HERNÁNDEZ

PRESENTADO POR

ANTONIO VLADIMIR MARTEL AVELAR

SAÚL ÁLVAREZ PACHECO

ANTIGUO CUSCATLÁN, LA LIBERTAD, EL SALVADOR, CENTRO AMÉRICA

AGOSTO 2019

Índice

Introducción

La diversidad de aplicaciones a las que un usuario debe tener acceso, requiere de forma individual un proceso de autenticación que valide las credenciales de ingreso a dichos aplicativos. Esta es una tarea habitual en entornos empresariales donde cada usuario tiene sus credenciales de acceso, pero al considerar este escenario por cada uno de los aplicativos se torna en una tarea compleja por la cantidad de datos de acceso que los usuarios deben administrar y recordar, esto no es práctico para los usuarios dado que suelen ser diferentes por cada aplicación y existen políticas que se aplican independiente por aplicación, por ejemplo: longitud, formato, tiempo de caducidad, y de nuevo se exige a los usuarios actualizar sus registros para considerar estos nuevos datos de acceso.

Es una realidad que, a los numerosos sistemas que se debe acceder por un proceso de autenticación, las aplicaciones que se ejecutan requieren en su forma básica y simple las credenciales de acceso de los usuarios. Dichas credenciales se deben recordar (por parte del usuario) y ser validadas correctamente por cada aplicación. Lo anterior es una forma tradicional de administración de usuarios, pero lleva a una situación desventajosa porque demanda al usuario a administrar sus credenciales por cada aplicación a la que accede. Además, existe una administración aislada de usuarios entre las aplicaciones, lo que lleva a duplicar mantenimientos del mismo.

Hay otros escenarios, por ejemplo, cuando un usuario deja de laborar para la empresa o bien se debe de remover o brindar el acceso a diferentes aplicaciones. En este escenario, se debe inactivar/eliminar el usuario por cada aplicación.

Se sugiere un modelo para una *gestión de acceso unificado* que facilite a los administradores de sistemas y a los usuarios finales a utilizar un único conjunto de credenciales para todos los sistemas a los que tiene acceso.

El proceso de autenticación suele ser un componente independiente y no reutilizable en los entornos empresariales y sus portafolios de aplicaciones basadas en la web, por lo tanto se da paso a una exploración de soluciones de terceros para, de cierta forma, centralizar la autenticación y explorar los componentes con el propósito de diseñar la arquitectura de una solución de esta naturaleza. Lo anterior permite analizar las soluciones actuales, evaluar los costos y beneficios y de igual forma tener los conceptos y guía para realizar el desarrollo interno de un servicio de autenticación y no depender de terceros.

El documento está organizado en 5 partes: la primera parte presenta el planteamiento del problema, antecedentes y alcance de la investigación; la segunda presenta la revisión de la literatura relevante; la tercera parte describe la investigación realizada, siendo esta de enfoque cuantitativo, alcance descriptivo y de tipo proyectiva; la cuarta parte presenta los resultados de la investigación; y la última parte describe un diseño de propuesta que define los componentes para el desarrollo de una solución SSO en entornos empresariales.

Capítulo I - Planteamiento del problema

1.1 Descripción

Dado que no se cuenta con la visibilidad de los compontes de una solución SSO, resulta que en los entornos empresariales no existe realmente una arquitectura clara de cómo desarrollar una solución de este tipo.

Si bien ya hay soluciones de terceros para esta situación, pero demandan, en algunas de gran envergadura, altos costos para su implementación. Estas soluciones permiten al equipo de TI gestionar el acceso a cualquier aplicación del dominio, y en algunas soluciones no importa si son empleados, socios, clientes, que las aplicaciones estén en la nube o en un data center (detrás de un firewall), sean aplicaciones web o móviles, la gestión de usuarios se vuelve una tarea ágil gracias a una plataforma estandarizada, de igual forma TI se beneficia ya de que las interrupciones de usuarios (relacionadas al proceso de autenticación) suelen ser menos.

SSO ayuda a que las tecnologías de información se vuelvan más seguras, hacer a los usuarios finales más productivos y a mantener el cumplimiento del negocio.

De igual forma en este contexto de soluciones SSO existe una gran variedad de conceptos y elementos técnicos que la investigación aborda con el fin de definirlos y poder luego identificarlos de forma clara para elaborar un marco de trabajo que ayude a la implementación de una solución, como resultado de la presente investigación.

Lo anterior permite que equipos de TI, en entornos empresariales, puedan tomar de base esta investigación y analizando los resultados, permitirles decidir si es mejor tercerizar el acceso unificado (soluciones ya en el mercado) o proceder con un desarrollo interno y dar valor agregado al incluir elementos que otras soluciones no tengan o necesiten modificarse según lo requiera el negocio.

1.2 Antecedentes

Hay situaciones en que las empresas cuentan con una gran variedad de aplicaciones en diferentes tecnologías y varios servidores, por lo que es necesario tener un inventario de aplicaciones y llevar el control de quienes tienen acceso y el tipo (nivel) que cada uno de los usuarios tiene.

En entornos empresariales, se tienen varias aplicaciones con administración independiente de usuarios y roles, lo que conlleva a estructuras de datos tales como bases de datos o servidores de autenticación que implementan (por ejemplo) un Active Directory por cada una de las aplicaciones. En algunos casos, se cuenta con una gestión centralizada si son conjuntos de aplicaciones adquiridos de forma global como un ERP, CRM, entre otros.

La arquitectura de un sistema SSO se puede clasificar en dos tipos [1]: simple y compleja, y estos a la vez, dependiendo del ámbito de operación, proveen cierta configuración. En esta investigación el ámbito de acción del SSO es para aplicaciones web, por lo tanto, se analizará la configuración denominada Web Single Sign-On (WSSO).

WSSO [2] opera sobre aplicaciones y recursos que se acceden mediante la web, su objetivo principal es de autenticar a los usuarios en las diferentes aplicaciones a las que accede, evitando que el usuario ingrese sus credenciales más de una vez. Los usuarios acceden a las aplicaciones web por medio de un navegador, el WSSO implementa el uso de cookies [3] que le permiten recordar la información en vista que el protocolo HTTP no maneja estados.

En la construcción de un framework de SSO se deben considerar elementos de comunicación entre todos los componentes que componen la arquitectura, como pueden ser los proveedores de identidad, los proveedores de servicios, las aplicaciones de negocio, y la navegación del usuario. La comunicación, como todas, se debe realizar mediante la implementación de protocolos que garanticen la seguridad de la misma y contribuya a estandarizar el manejo de peticiones y respuestas.

1.3 Objetivos

Objetivo General

Proponer un modelo de arquitectura para la implementación de un SSO en aplicaciones web para entornos empresariales, tomando como referencia las tecnologías y soluciones actuales.

Objetivos Específicos

1. Formular las características tecnológicas que implementan las soluciones de SSO.
2. Detallar la construcción de un marco de trabajo genérico (framework) para SSO.
3. Analizar la implementación de una solución SSO considerando el costo de adquirir un servicio o herramienta de un proveedor y el costo de realizar un desarrollo interno, tomando de base la arquitectura sugerida en la presente investigación.

1.4 Justificación

En los ambientas empresariales se tienen un gran número de aplicaciones desarrolladas internamente o por contratación externa, estas pueden correr en un servidor de aplicaciones o varios y pueden ser de la misma tecnología o diferentes. Implementar una solución de SSO puede ser una decisión estrategia que genera varias ventajas, entre ellas la seguridad y es que las políticas de seguridad establecidas se emplean a todas las aplicaciones, se puede monitorear todas las peticiones, identificar accesos no válidos. Facilita a los usuarios el manejar un único conjunto de credenciales para el acceso a las aplicaciones y para el equipo de TI este componente

de autenticación, siendo vital en entornos empresariales, trabaja su comunicación de forma segura implementando protocolos y cifrados en la red. Todo el proceso de inicio de sesión se delega a una entidad externa.

La implementación de una solución SSO, en su esencia básica, autentica al usuario una única vez y le permite acceder a otras aplicaciones (que forman parte del acceso centralizado) sin necesidad de volver a autenticar. Permite al usuario saltar entre las diferentes aplicaciones sin necesidad de iniciar sesión nuevamente. De igual forma este componente se vuelve vital en la configuración de los sistemas y al verse afectado, el daño se expande a todos los que lo consumen. Si un hacker accede a un sistema, bajo esta configuración, entonces este puede acceder a todas las aplicaciones que integran la solución, o bien puede comprometer el servidor del SSO y simplemente no será posible acceder a ningún sistema.

Existen soluciones de paga que exponen todos los beneficios antes mencionados y otras funcionalidades que les permite diferenciarse de las otras opciones de software SSO. De igual forma es posible acceder a la documentación de los marcos de trabajo que estas soluciones implementan, a los medios de comunicación como lo son los protocolos y certificados, a la configuración de un repositorio centralizado de usuarios (similar a como actualmente lo pueden hacer con bases de datos locales), a los servidores de aplicación y cómo manejar el inicio de sesión y uso de cookies u otros componentes que permitan persistencia en la navegación.

Esta investigación explora los componentes de las soluciones actuales, ventajas y desventajas para permitir al usuario valorar, de entre las opciones, cual es mejor para su necesidad. En un segundo punto se propone una solución de arquitectura que integre todos los componentes y requisitos de forma ordenada tal que, puede ser utilizado para realizar un desarrollo interno y tener la visión completa de que es y cómo se construye un middleware, en este caso un SSO. Se debe considerar los costos de adquirir una solución comercial o bien realizar un desarrollo interno, ventajas que se acoplen a la necesidad del negocio donde se desea implementar, y lo más significativo es definir el diseño de arquitectura que todo desarrollo de software necesita para guiar el proceso de implementación.

1.5 Delimitación

La presente investigación lleva por objetivo el diseño de una solución SSO para entornos empresariales donde utilizan diferentes tecnologías en sus aplicaciones web. El primer punto es revisar las soluciones actuales de SSO para observar y analizar las tecnologías y componentes involucrados que habilitan un modelo de gestión de acceso unificado, una solución SSO.

En un segundo punto es tomar las características necesarias para el funcionamiento de una solución SSO, y así describir el dominio del problema, los requerimientos funcionales y no funcionales, especificación de la interfaz de usuario para el portal del SSO, interfaces para la comunicación entre los componentes: aplicaciones web y los servidores de autenticación.

Se propone un diseño de arquitectura con los componentes y la orquestación entre ellos, detallados a nivel macro, se establecen los modelos de datos necesarios para definir el SSO y adaptar las aplicaciones para la integración a la nueva plataforma, diseño de interfaces (inicio de sesión único, portal web), procesos de integración para acoplar las aplicaciones al nuevo flujo de autenticación manejado por el SSO.

Además, se proporciona al lector todo el contexto para que, de forma objetiva, analice el camino más conveniente a tomar para la implementación de un SSO, ya sea como una solución tercerizada o realizar un desarrollo interno, tomando de base la arquitectura propuesta. Se consideran los costos y beneficios de soluciones actuales, así como los aspectos que se deben mejorar y a la vez valorar si son suficientes para cubrir la necesidad del negocio.

Capítulo II – Estado del arte

2.1 Antecedentes

2.1.1 Tecnologías SSO actuales

Un sistema SSO es aquel al que las aplicaciones delegan el proceso de identificación de usuarios, y este a la vez responde a las aplicaciones para notificar si se permite o no el acceso una vez concluido el proceso.

Una solución de tipo SSO permite a los usuarios autenticarse de manera segura con múltiples aplicaciones y sitios web al iniciar sesión solo una vez, con un solo conjunto de credenciales (nombre de usuario y contraseña). Con SSO, la aplicación o el sitio web al que el usuario intenta acceder depende de un tercero de confianza [4] para verificar que los usuarios son quienes dicen ser. De igual forma implementa el nivel de autorización a los usuarios usualmente mediante la aplicación y creación de políticas de acceso a recursos.

Estas soluciones presentan beneficios considerados principalmente en tres categorías [5]:

1. Facilidad de uso: Los usuarios finales no tienen que recordar una gran cantidad de nombres de usuarios y contraseñas. Simplifica el acceso.

2. Implementación: Habilita una gestión centralizada de usuarios y autorizaciones, evitando que sean las propias aplicaciones las que implemente estos mecanismos y delegan al SSO su manejo. El proceso de aprovisionamiento de usuarios se agiliza en todas las aplicaciones involucradas. De igual forma, la activación o baja de usuarios se replica en las aplicaciones lo cual lo vuelve de cierta forma un proceso automatizado. Se integra como un middleware en el contexto de las aplicaciones, permitiendo la integración de diferentes fabricantes, sistemas y tecnologías.

3. Seguridad: Las políticas de contraseña aplican a todas las aplicaciones, fortaleciendo el uso de las mismas y evitando malas prácticas de contraseña o bien contraseñas débiles a estar presentes y comprometer los datos e información en las aplicaciones. Se debe considerar la sensibilidad de tener la validación de credenciales centralizada, y es que de igual forma compromete los procesos permitiendo que un intruso puede acceder a todas las aplicaciones que implementan la solución SSO.

Es importante especificar realmente lo que es un sistema SSO y entender las diferencias con un almacén de contraseñas [4] o también llamado gestor de contraseñas, que en inglés se identifica como "password vault".

- Password vault es un software que almacena usuarios y contraseñas para múltiples programas o sitios, en una ubicación segura y un formato encriptado. Los usuarios

pueden acceder al password vault a través de un único nombre de usuario y contraseña. El password vault luego les proporciona la contraseña para el sitio web al que intentan acceder. Con el almacenamiento de contraseñas se puede tener el mismo conjunto de credenciales, pero debe ingresarlo cada vez que se navega a un sitio web o aplicación diferente.

- Con SSO, después de iniciar sesión a través de la solución, se puede acceder a todas las aplicaciones y sitios web aprobados por la empresa sin tener que iniciar sesión nuevamente. En general, SSO se considera más seguro y elimina la necesidad del usuario a administrar varias contraseñas, se reduce la frecuencia con que se debe iniciar sesión y el volumen de credenciales almacenadas.

La arquitectura de un sistema SSO se puede clasificar en dos tipos [1]:

- Simple: En el que el sistema SSO es único y otorga acceso a los usuarios de un único dominio de seguridad.
- Complejo o federado: Es una arquitectura propia de sistemas federados, en los que existe más de un sistema de autenticación (SSO), y entre los que existe algún mecanismo de interrelación o confianza. Generalmente este tipo de sistemas están compuestos por varios dominios de seguridad, habiendo un mecanismo SSO en cada uno de ellos.

Existe una delimitación del concepto SSO para entornos web, denominado Web-SSO (WSSO), que de igual forma permite a los usuarios un inicio de sesión único para poder acceder a un grupo de aplicaciones web que requieren autenticación.

Dependiendo del ámbito de operación, las soluciones proveen cierta configuración [2]:

E-SSO: Enterprise Single Sign-On, sirve como una autenticación primaria que interactúa con aplicaciones secundarias, con el fin de completar los datos de inicio de sesión almacenados en los servidores de las aplicaciones. Es un sistema heterogéneo que gestiona la autenticación de usuario en entornos integrados al SSO. Es necesario que las aplicaciones secundarias tengan la capacidad de deshabilitar la pantalla de login.

W-SSO: Web Single Sign-On, opera sobre aplicaciones y recursos que se acceden mediante la web y su objetivo principal es de autenticar a los usuarios en las diferentes aplicaciones a las que accede, evitando que el usuario ingrese sus credenciales más de una vez. Los usuarios que aún no han sido autenticados son redirigidos a un servidor de autenticación o servicio web que mediara el acceso.

Kerberos [6]: Es un protocolo de autenticación de red, creado por el MIT. Los usuarios se registran en un servidor y estos obtienen un ticket (TGT, del término ticket-granting ticket), que es usado por las aplicaciones cliente para obtener acceso. La seguridad se basa en el uso de criptografía de clave secreta, que permite al usuario demostrar su identidad a un servidor (y viceversa) a través de una conexión de red insegura.

Identidad federada: Corresponde a una solución de *gestión de identidad*, que permite usar las credenciales disponibles en un sistema de autenticación en otros, y esto a la vez de una misma organización u otras externas. Se basa en el "círculo de confianza" entre las diferentes partes y hace uso de estándares para el intercambio de información entre dominios. Una ventaja es que no se requiere dar acceso a sistemas o compartir componentes tecnológicos, seguridad y autenticación.

OpenID: Sistemas de autenticación distribuidos y descentralizados, en la que cada aplicación debe autenticarse en un servidor OpenID mediante parámetros en la URL. Los sitios web que implementan OpenID no requieren una cuenta de usuario, sino el identificador creado en el servidor.

Los recursos centralizados son gestionados por proveedores de identidad y de servicios [7], conocidos en inglés como *Identiy Provider* (IdP) y *Service Provider* (SP) respectivamente.

- IdP: Se refiere a una estructura de datos que contiene las identidades de los usuarios y maneja toda la información necesaria para la autenticación. Es posible conectar los usuarios a los recursos tecnológicos que lo requieren, desde una ubicación centralizada.
- SP: Es una entidad que proporciona servicios web. Ejemplos de SP incluyen servicios de aplicaciones, de almacenamiento y servicios de internet. Un SP confía en un proveedor de identidad de confianza (IdP) o servicio de token de seguridad (STS) para la autenticación y autorización. En el modelo WS-Federation, un proveedor de servicios se denomina "parte que confía" (*Relying Party* en inglés).

Proveedor es una forma genérica de referirse tanto a los IdP como a los SP, para términos sencillos y en relación con la gestión de identidades, un IdP se puede describir como un proveedor de servicios para almacenar perfiles de identidad y ofrecer incentivos a otros SP con el objetivo de federar las identidades de los usuarios. Sin embargo, se debe tener en cuenta que los IdP también pueden proporcionar servicios más allá de los relacionados con el almacenamiento de los perfiles de identidad.

2.1.2 Soluciones por suscripción

Existe una gran variedad de soluciones SSO, y analizarlas todas no es posible. La presente investigación toma como referencia dos puntos clave para definir las soluciones SSO a analizar.

Como primera referencia se aprecia la opinión de un sitio de revisión de productos para compradores de tecnología empresarial, llamado *IT Central Station* [8].

A continuación, una pequeña reseña del sitio [9]: "Es un sitio de alta calidad y confianza en el que los comentarios son escritos por usuarios reales. Nuestro proceso de autenticación triple con perfiles de LinkedIn, vigilancia policial comunitaria y la supervisión humana aseguran revisiones, sobre los productos, sean 100% auténticas... Los compradores de tecnología ahora

investigan productos a través de la web y crean una lista corta de proveedores incluso antes de que comiencen a hablar con ellos".

IT Central Station dispone un trabajo de investigación donde se comparan las mejores soluciones y proveedores de SSO. El reporte *"SSO Buyer's Guide and Reviews May 2019"* [10], de descarga libre, es basado en más de 65 experiencias de usuario reales con los productos más populares.

Una segunda referencia es la popularidad de las soluciones, y la discusión [11][12] que se genera en torno a estas respecto de cuál es mejor, las similitudes y diferencias que existen.

Se concluye entonces analizar las siguientes soluciones por suscripción:

Solución	Número de vistas	Número de veces comparado con otro producto
Okta	1°	1°
Auth0	2°	2°
OneLogin	5°	3°

Tabla 1: IT Central Station, posición según factor de calificación [10].

Okta [13]

Es un servicio de gestión de identidades de nivel empresarial para aplicaciones basadas en la web, tanto en la nube como detrás de un firewall. Provee un sistema integrado que conecta de forma segura a cualquier persona, a través de cualquier dispositivo, a las tecnologías que necesiten.

Con más de 6,000 integraciones predefinidas para proveedores de aplicaciones e infraestructura, los clientes de Okta pueden usar de manera fácil y segura. Más de 6,100 organizaciones, entre ellas 20th Century Fox, JetBlue, Nordstrom, National Geographic, Western Union, DocuSign, confían en Okta para ayudar a proteger las identidades de sus fuerzas de trabajo y empleados.

Top de comparaciones [10]

- Auth0 vs. Okta - Comparado el 30% del tiempo.
- OneLogin vs. Okta - Comparado el 17% del tiempo.
- Microsoft Azure Active Directory Premium vs. Okta - Comparado el 16% del tiempo.

Características de Okta SSO:

Inicio de sesión único.

- Verifica la contraseña de un solo uso (*OTP* por sus siglas en inglés de One-Time Passsword).
- Integración confiable para SSO en todas sus aplicaciones web y móviles, con un motor de federación completo y políticas de acceso flexible.
- Integración a cualquier aplicación o API moderna: Se debe proporcionar una URL y configurar la integración con el asistente de Okta.
- Soporte de integración con:
  SWA (Secure Web Authentication): Uso de credenciales para inicio de sesión.
  SAML 2.0: Uso del protocolo para iniciar sesión en las aplicaciones.
  OpenID Connect: Uso del protocolo para iniciar sesión en las aplicaciones.
- Soporte para servir como una federación IdP o SP.

Directorio seguro con integración

- Un almacén de usuarios flexible y seguro, integración a AD / LDAP en múltiples dominios y restablecimiento de contraseñas de AD / LDAP de autoservicio.
- Política de contraseñas con opciones de complejidad.
- Política de contraseña basada en grupo.
- Almacenamiento y transformación de atributos enriquecidos para soportar escenarios de autorización y SAML enriquecidos basados en atributos.
- Integración de IDP de terceros con SAML de entrada y OpenID Connect de proveedores de identidad externos.

Informes de seguridad en tiempo real

- Visor de eventos e informes incorporados para descubrir y solucionar anomalías de seguridad y acceso.

Autenticación adaptativa

- Acceso seguro para todos los usuarios con autenticación de dos factores a través de Okta Verify OTP, incluido para todos los clientes de SSO.
- Establece políticas inteligentes de acceso y autenticación basadas en el contexto (ubicación, dispositivos, red) de inicio de sesión.
- Informes y auditorías simples: registros de autenticación detallados, como intentos de inicio de sesión, con informes predefinidos para auditorías y fácil integración con herramientas de seguridad.

Machine Learning [14]

- Solución de autenticación basada en riesgo con capacidades de aprendizaje automático (machine learning): Anunciado en San Francisco el 2 de abril de 2019, las organizaciones ahora pueden automatizar las prácticas de seguridad e implementar técnicas de autenticación simplificadas.

Cosas que se pueden mejorar según opiniones de profesionales, publicadas IT Central Station:

"Todavía tuvimos que escribir varios programas / scripts internos para completar el proceso de aprovisionamiento del usuario. Okta no tiene la capacidad de aprovisionar cuentas de buzón para Exchange local o en un entorno híbrido O365. La función Group Push de Okta a AD no funcionó de manera confiable en nuestro entorno". James Lambert, ingeniero de sistemas Sr. en una compañía de salud con 5,001-10,000 empleados

"Las llamadas al servicio web RESTful y su respuesta parecen un poco lentas". Jaskeerat Singh, consultor en una empresa de servicios tecnológicos con 201-500 empleados.

Elementos de la arquitectura [15]:

Escalabilidad: Capacidad para manejar automáticamente una cantidad creciente de trabajo y el potencial de ser ampliado para adaptarse a ese crecimiento.

Confiabilidad: Capacidad para realizar sus funciones y operaciones previstas sin experimentar fallas.

Seguridad: Procesos, herramientas y políticas para prevenir, detectar y responder a amenazas. "Al maximizar el aislamiento en una arquitectura multiusuario, Okta garantiza un 99,9% de tiempo de actividad y cero tiempos de inactividad planificados. De hecho, mantuvimos un tiempo de actividad del 100% en 2019 hasta la fecha, un tiempo de actividad del 99,9955% en 2018 y un tiempo de actividad del 99,9995% en 2017, incluso cuando aumentamos el 290% en autenticaciones por mes". Héctor Aguilar - CTO @ Okta, Jon Todd - arquitecto jefe @ Okta.

Como referencia, Okta describe las características de la arquitectura detrás de sus soluciones y comparte datos acerca de la carga de trabajo que recibe:

- 2.1Billones Identidades de aplicación gestionadas y aseguradas por Okta
- 550Millones solicitudes web por día
- 70Millones autenticaciones por día
- Mitigación de riesgos
- Implementación escalonada y Rollback
- Independencia del proveedor de infraestructura
- Ajuste de la carga de trabajo
- Escalabilidad horizontal y vertical
- Aislamiento geográfico

Precios [16]

Todos los productos tienen un precio por usuario por mes y se facturan anualmente. El precio indicado es para casos de uso típicos. $ 1,500 por contrato mínimo al año.

- Single Sign-On: $2 por mes, por usuario.
- SSO con autenticación adaptativa: $5 por mes, por usuario.
- Descuentos por volumen están disponibles.

OneLogin [17]

En las empresas, la transformación digital se ve frenada por la naturaleza fragmentada de un entorno de TI híbrido. OneLogin rompe esa barrera, centralizando el acceso a las aplicaciones locales y en la nube. La plataforma de administración de acceso unificada (*UAM* por sus siglas en inglés de Unified Access Management) de OneLogin permite a los usuarios acceder de forma sencilla y segura a las aplicaciones y los datos que necesitan, en cualquier momento y en cualquier lugar.

Con el portal de inicio de sesión único de OneLogin, los usuarios solo tienen que ingresar un conjunto de credenciales para acceder a sus aplicaciones web en la nube y detrás del firewall, a través de computadoras de escritorio, teléfonos inteligentes y tabletas. Aumenta enormemente la productividad al tiempo que mantiene los datos seguros.

La seguridad de contraseña basada en políticas y la autenticación multifactor de OneLogin, aseguran que solo los usuarios autorizados tengan acceso a datos confidenciales. Puede implementar políticas de contraseña más exigentes, como la longitud requerida, la complejidad y las restricciones en la reutilización de la contraseña, así como el tiempo de espera de la sesión y la política de autoservicio para restablecer la contraseña para aumentar la protección sin impedir a sus usuarios.

Más de 2,500 clientes empresariales [18] confían a nivel mundial en OneLogin para asegurar sus aplicaciones, organizaciones como Pacific Life, Yammer, Airbus son algunos de sus clientes que delegan la gestión de identidad y acceso a OneLogin.

Top de comparaciones [10]

- Okta vs. OneLogin - Comparado el 51% del tiempo
- Auth0 vs. OneLogin - Comparado el 12% del tiempo
- Microsoft Azure Active Directory Premium vs. OneLogin - Comparado el 6% del tiempo

Características de OneLogin SSO:

Portal SSO

- Con el portal de inicio de sesión único de OneLogin, los usuarios solo tienen que ingresar un conjunto de credenciales para acceder a sus aplicaciones web en la nube y detrás del firewall, a través de computadoras de escritorio, teléfonos inteligentes y tabletas. Esto aumenta enormemente la productividad al tiempo que mantiene los datos seguros. La seguridad de contraseña basada en políticas y la autenticación multifactor de OneLogin aseguran que solo los usuarios autorizados tengan acceso a datos confidenciales.
- Políticas de contraseña más exigentes, como la longitud requerida, la complejidad y las restricciones en la reutilización de la contraseña, así como el tiempo de espera de la sesión y la política de autoservicio para restablecer la contraseña para aumentar la protección sin impedir a sus usuarios.
- Fácil adición de aplicaciones personales, que no requieren la participación de TI.
- Soporte de 21 lenguajes.
- Habilita la técnica de almacén de contraseñas para aplicaciones no federadas.

Inicios de sesión múltiples

- El sistema de autenticación de inicio de sesión único permite crear cualquier número de inicios de sesión para el mismo tipo de aplicación. Si se tienen diferentes entornos de producción y de pruebas, la funcionalidad de inicio de sesión múltiple ahorra tiempo.

Inicio de sesión social

- La autenticación social permite a los usuarios finales iniciar sesión en OneLogin utilizando sus credenciales de proveedor de identidad social de servicios como Facebook, Google+, LinkedIn y Twitter. Esto proporciona una experiencia más ágil, ya que no se necesita crear una contraseña de OneLogin para acceder a las aplicaciones dentro del portal.
- Inicio de sesión compartido para aplicaciones que no admiten varios usuarios al mismo tiempo.

Enlaces de inicio de aplicaciones y enlaces profundos

- Los usuarios no siempre tienen que acceder a las aplicaciones a través del portal SSO de OneLogin. Muchas veces, las aplicaciones se inician a través de enlaces en correos electrónicos, como notificaciones de intercambio de documentos o invitaciones a reuniones. Simplemente haga clic en el enlace y OneLogin le permite iniciar sesión automáticamente.
- Pre-integrado con miles de aplicaciones web y nuevas aplicaciones se agregan cada día.

Soporte de integración con

- SAML 2.0: Uso del protocolo para iniciar sesión en las aplicaciones. Kits de herramientas SAML de código abierto para cinco plataformas de desarrollo web: PHP, Python, Ruby, Java, .Net.
- OpenID Connect: Uso del protocolo para iniciar sesión en las aplicaciones.

Documentación técnica para desarrolladores en el uso de estándares de comunicación, autenticación, llamadas a APIs, reportes.

Autenticación adaptativa [19]

- Aprovecha el aprendizaje automático (Machine Learning) para realizar evaluaciones de riesgo dinámicas que pueden detectar inicios de sesión de alto riesgo y desencadenar la autenticación multifactor.
- Las puntuaciones de riesgo se calculan en función de la reputación de la red, la ubicación geográfica, la identificación del dispositivo y las anomalías de tiempo.

Cosas que se pueden mejorar según opiniones de profesionales, publicadas IT Central Station

- "Necesita aumentar la cantidad de conectores disponibles para conectarse a los diferentes puntos finales". ManojKumar
- "El tiempo de inactividad con OneLogin es algo que seguirá mejorando, pero está bien. El precio es bastante competitivo en comparación con otros productos". ManojKumar
- "Facilitar la personalización de la interfaz de usuario, debería ser más fácil configurar los ajustes de OneLogin en función de las necesidades del cliente". ManojKumar - Gerente Regional de Operaciones en una empresa de servicios tecnológicos con 10,001+ empleados.

Precios [20]

Todos los productos tienen un precio por usuario por mes y requieren un número mínimo de usuarios.

Plan de arranque

- Single Sign-On con soporte estándar
- $2 por mes, por usuario - 25 usuarios mínimos.

Plan de empresa

- SSO con seguridad basada en políticas, multifactor y gestión avanzada de usuarios
- $4 por mes, por usuario - 10 usuarios mínimos.

Plan ilimitado

- Gestión total de la identidad para la empresa compleja
- $8 por mes, por usuario - 5 usuarios mínimos.

Complementos (100 usuarios mínimos)

- Virtual LDAP: Integración con VPN, NAS
- OneLogin Desktop: Extiende el SSO a macOS o máquinas Windows.

- Autenticación adaptativa: Utiliza el aprendizaje automático para la autenticación inteligente. Crea perfiles de usuario basados en ubicaciones de inicio de sesión típicas, horarios, direcciones IP, etc., y luego desafía los intentos de inicio de sesión anormales.
- Extiende la gestión de identidad a las aplicaciones heredadas (*legacy*)

Auth0 [21]

Proporciona autenticación y autorización como un servicio. Cualquier aplicación (escrita en cualquier idioma o en cualquier pila) se puede conectar a Auth0 y definir los proveedores de identidad que se desea usar.

En función de la tecnología de las aplicaciones, se elige uno de los SDK (o llamada a las API) y se conecta a la aplicación. Ahora, cada vez que un usuario intenta autenticarse, Auth0 verificará la identidad y enviará la información requerida a la aplicación.

En la cartera de clientes de Auth0, se encuentran: Safari, Sprinkrl, JetPrivilege

Características de Auth0 SSO:

- Gratis para proyectos open-source sin ánimo de lucro.
- Autorización máquina a máquina: Facilita la comunicación segura entre su API y los clientes externos no interactivos, así como las API internas con solo tocar un interruptor y protocolos basados en estándares.
- Autenticación multifactor con guardián: Guardián iOS o Android para recibir notificaciones.
- Detección de anomalías: Mantiene a los usuarios y servicios a salvo de filtraciones de contraseñas e intrusos. Protege y notifica a usuarios cuando se filtran las credenciales o cuando alguien está tratando de forzar la entrada en su cuenta.
- Apoyo principal (premier): Amplio acuerdos de soporte 24x7x365.
- Conectores para Active Directory y LDAP
- Conexiones personalizadas a bases de datos a través de JavaScript que corre en un servidor Auth0.

Soporte de integración con

- OAuth 1, OAuth 2: Estándar de autorización que le permite a un usuario otorgar acceso limitado a sus recursos en un sitio, a otro sitio, sin tener que exponer sus credenciales
- JSON Web Tokens: Estándar abierto que define una forma compacta y autónoma para transmitir de manera segura información entre las partes como un objeto JSON.
- SAML 2.0: Formato de datos basado en XML de estándar abierto.
- OpenID Connect: Capa de identidad que se encuentra sobre OAuth 2 y permite la verificación fácil de la identidad del usuario.

- WS-Federation: Estándar desarrollado por Microsoft, y usado extensivamente en sus aplicaciones.

Auth0 proporciona integraciones de SSO para los siguientes servicios, se mencionan los principales:

- Dropbox
- Office 365
- SalesForce
- SharePoint
- Slack
- Microsoft Dynamics
- Windows Active Directory RMS
- Adobe Echosign
- CloudBees

Top de comparaciones [10]

- Okta vs. Auth0 - Comparado el 71% del tiempo
- OneLogin vs. Auth0 - Comparado el 10% del tiempo
- SAP Customer Data Cloud vs. Auth0 - Comparado el 7% del tiempo

Cosas que se pueden mejorar según opiniones de profesionales, publicadas IT Central Station:

"Pueden hacer un mejor trabajo explicando lo que se supone que debe hacer a continuación para seguir correctamente un enfoque idiomático para usar la solución más allá de simplemente pasar un token de JWT a un servidor y hacer que el servidor verifique y luego firme para validar el token". Michael M - Gerente de Servicios Tecnológicos

Precios [22]

Los precios son manejados en dos modos de suscripción, y estas a la vez bajo tres categorías dependiendo de las necesidades de la empresa. De igual forma los precios dependen del tipo y cantidad de usuarios que se desean adquirir.

Los tipos de usuario pueden ser regulares o de empresa, y se diferencian según el tipo de conexión que se requiere para ellos. De igual forma hay otra clasificación de usuarios, y estos son basados en la autenticación que requieren.

Se debe elegir usuarios con clasificación externa, aquellos externos a la compañía que deben ser autenticados en las aplicaciones o APIs desarrolladas. Usuarios con clasificación interna son empleados de la compañía que de igual forma deben ser autenticados en las aplicaciones o APIs desarrolladas.

Categoría	Suscripción mensual	Suscripción anual (con esta suscripción se obtiene un mes gratis)
Desarrollador	• $13 – 1,000 usuarios activos regulares • $850 – 50,000 usuarios activos regulares	• $143 – 1,000 usuarios activos regulares • $9,350 – 50,000 usuarios activos regulares
PRO, usuarios externos	• $59 - 100 usuarios activos regulares y 100 usuarios activos de empresa. • $1,625 - 10,000 usuarios regulares activos y 500 usuarios activos de empresa	• $649 - 100 usuarios activos regulares y 100 usuarios activos de empresa. • $17,857 - 10,000 usuarios regulares activos y 500 usuarios activos de empresa
PRO, usuarios externos con autenticación máquina a máquina (respectivos a configuración anterior)	• 5,000 tokens de acceso y configuración de usuarios seleccionados: $74 • 5,000 tokens de acceso y configuración de usuarios seleccionados: $1,640	• 5,000 tokens de acceso y configuración de usuarios seleccionados: $814 • 5,000 tokens de acceso y configuración de usuarios seleccionados: $18,040
PRO, usuarios internos	• $200 – 100 empleados • $1,100 – 1,000 empleados	• $2,200 – 100 empleados • $12,100 – 1,000 empleados
EMPRESA	N/A - Contactar al proveedor	N/A - Contactar al proveedor

Tabla 2: Precios por categoría y tipo de suscripción.

Si la necesidad de usuarios sobrepasa lo detallado en la tabla anterior, se debe contactar con el proveedor.

2.1.3 Guía para selección de candidatos

Las organizaciones analizan las características para seleccionar candidatos aceptables para proveedores de servicios SSO en entornos empresariales, y buscan que las soluciones sean de bajo costo, siempre y cuando cumplan con las especificaciones necesarias para cubrir los requisitos de un inicio de sesión unificado [23]. Se sugiere analizar las siguientes características:

Fijación de precios

- Los precios normalmente en las organizaciones es la primera clave al momento de la toma de decisión para adquirir una solución o implementarla en las Empresas.

Facilidad de uso

- La complejidad debe ser lo más sencilla posible para poder ser administrado E implementado.

Implementación

- Se debe identificar que la solución pueda ser implementada dentro de la empresa que pueda ser acoplada en la estructura organización con la que se cuenta.

Políticas de seguridad

- Uno de los controles más verificados es la parte de la seguridad, refiriéndose a este como uno de los puntos más críticos en las aplicaciones.

Autenticación SAML

- Para poder manejar las autenticaciones y autorizaciones de mejor manera se tiene como referente el protocolo SAML.

Protección con contraseña

- Los diferentes controles de seguridad uno de ellos y el más crucial en muchos casos es la parte de las contraseñas, en líneas generales existen diferentes niveles de seguridad desde: Bajas, Medianas y altas.
- Cada una de ellas comprende ciertas diferencias en cuanto la longitud y complejidad desde la inclusión de caracteres especiales, números, letras mayúsculas y minúsculas, además se deben manejar diferentes controles técnicos de protección de usuario.
- Las configuraciones para estos controles de seguridad y niveles según se la política de seguridad de cada organización debe ser permitida.

Autenticación Multifactor

- Existen diferentes factores en las aplicaciones que requieren validaciones del usuario adicionales por ello uno de los requisitos necesarios es el manejo de una doble autenticación para ello las soluciones deben de proveer una validación de token u/o certificados de validación.

Soporte

- La mayoría de las aplicaciones SSO deberá cumplir con un requisito importante para las organizaciones es poder adaptar las soluciones a sus políticas para ello la parte del mantenimiento debe estar siempre.

2.2 Componentes de un SSO

2.2.1 Autenticación

Este componente en los SSO se encarga de verificar y validar las credenciales del usuario para brindar el acceso a las aplicaciones.

Para comprender este componente veremos el proceso tradicional de los inicios de sesión y como se establece dentro del SSO.

1. Autenticación tradicional.

   Las autenticaciones tradiciones se refiere a que cada usuario debe de autentificarse en cada aplicación en la que desee acceder este proceso de manera individual para todas las aplicaciones se cuenta con un usuario y contraseña [24].

   El proceso de autenticación tradicional:

- Envía la información del usuario (usuario y contraseña).
- Realiza la verificación con los datos almacenados (Active Directory, bases de datos, entre otros).
- Se valida la petición del usuario y en caso sea exitoso se concede el acceso a las aplicaciones.

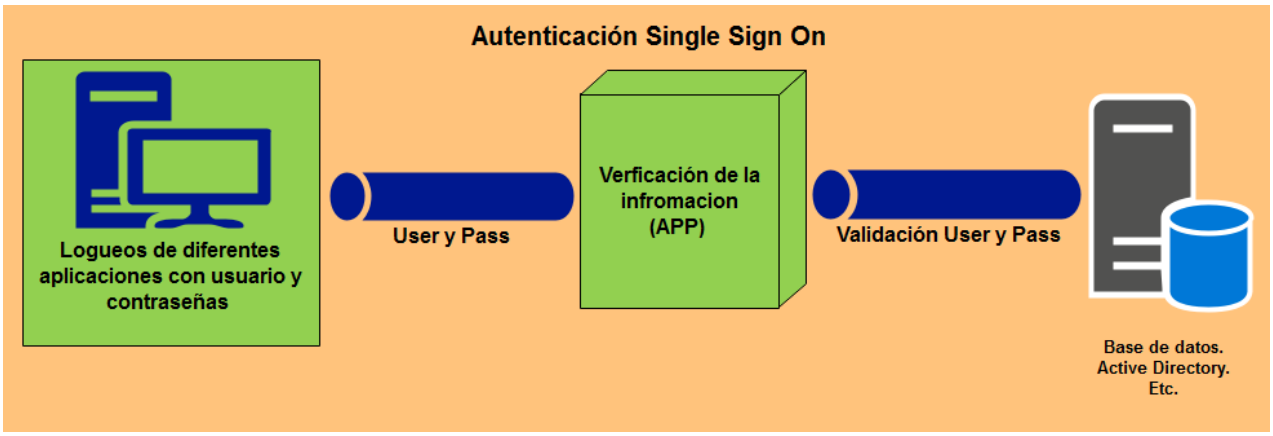

Figura 1: Proceso tradicionales en la autenticación. Elaboración propia

La autenticación tradicional establece que cada una de las aplicaciones necesita realizar el proceso de login y así obtener el acceso a las aplicaciones.

- Login de aplicaciones: Cada vez que se desea ingresar a una aplicación, se debe realizar el proceso.

- La verificación: Esta acción se realiza internamente en la aplicación la cual recibe como parámetros las credenciales (usuario y contraseña)
- Validación: Se realiza la verificación de los datos en un repositorio de datos (Active Directory, base de datos, otros) y cuando las credenciales del usuario son válidas, obtiene el acceso a la aplicación.

2. Autenticación SSO.

Las soluciones SSO toma la responsabilidad de realizar la autenticación de las credenciales de usuarios, de este modo el SSO se encarga de velar la veracidad de las credenciales que sea pertenecientes y correctas de parte del usuario, cuando se realiza la verificación y validación y es autenticado se regresa un comprobante(Token) el cual se utiliza para enviarlo a la aplicación a la que se desea acceder cuando la aplicación lo recibe el y realiza la validación contra la entidad de autenticación (SSO). Además de validar el comprobante y su vigencia. Si es correcto, la aplicación da acceso al usuario, de este modo cuando desea ingresar a otra aplicación solamente se debe presentar el comprobante para realizar la autenticación nuevamente para la otra aplicación [25].

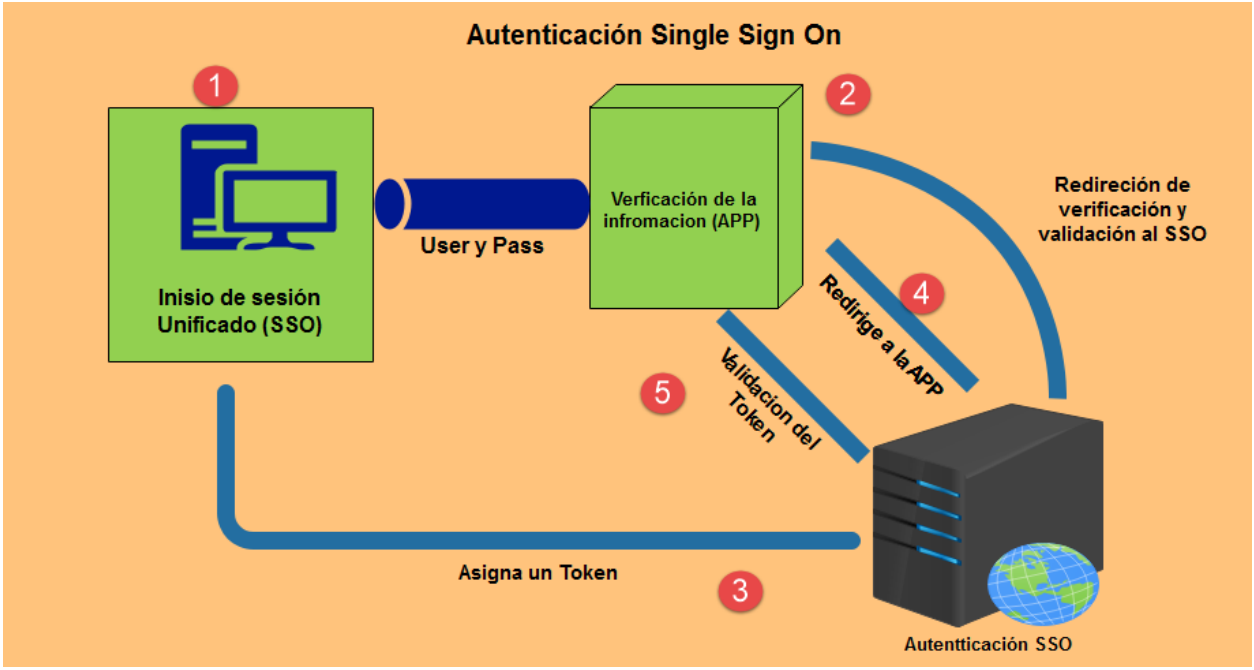

Figura 2: Proceso en la autenticación SSO. Elaboración propia

La autenticación SSO, a diferencia del proceso tradicional, solamente se realiza una vez y da acceso a todas las aplicaciones. Detalle del proceso:

1. Ingreso de las credenciales (cuando no se ha iniciado sesión) se dirige a la solución SSO.
2. Se realiza la verificación y validación de la autenticación dentro del SSO.
3. Asigna el token el cual se envía al usuario para que pueda tener acceso a la aplicación.
4. Se redirige nuevamente a la aplicación, cuando esto se realiza ya se cuenta con el token por el lado del usuario.

5. La aplicación realiza la validación del token contra el del SSO si es el asignado este ingresa a la aplicación de manera exitosa.

Al realizar el proceso de autenticación y desea ingresar a otra aplicación. El SSO asigno un token al usuario y lo envía, El SSO verifica y validad el token para que la aplicación permita al usuario ingresar.

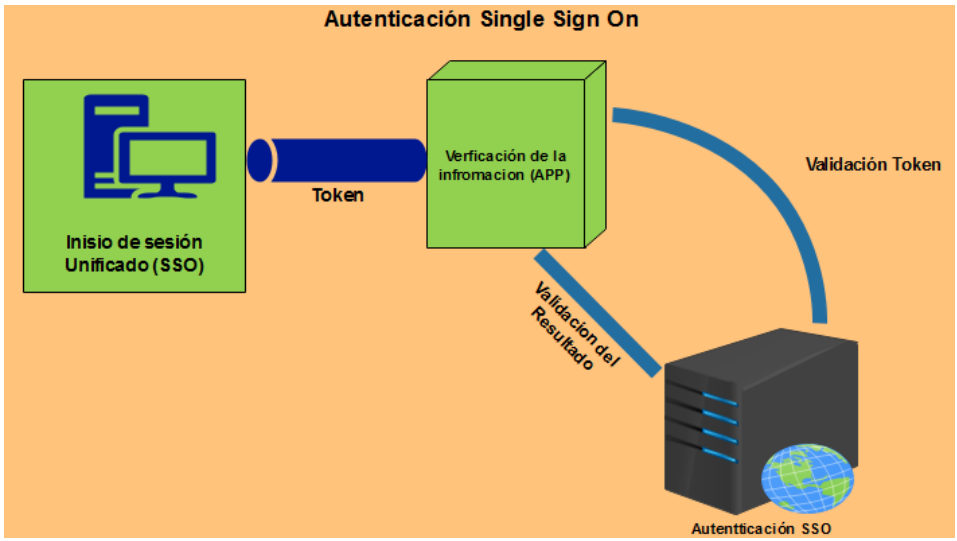

Figura 3: Procesos de autenticación mediante Token. Elaboración propia

Cuando el token es invalido o caduco su tiempo de validez se redirecciona al SSO para realizar la autenticación [26].

2.2.2 Comunicación

La comunicación en un sistema SSO debe ser segura, pues es un punto crítico y debe garantizar la integridad de las peticiones y respuestas que viajan en ella. Una arquitectura SSO maneja la comunicación dependiendo de los elementos que integre [27], como lo son el protocolo de comunicación, los proveedores de servicios y de identidades.

Dependiendo de los elementos seleccionados, la comunicación utiliza conceptos como: token, certificados de seguridad, llaves secretas de cliente y servidor, cookies, y demás. Todo organizado entre los proveedores y las aplicaciones que consumen los servicios [28].

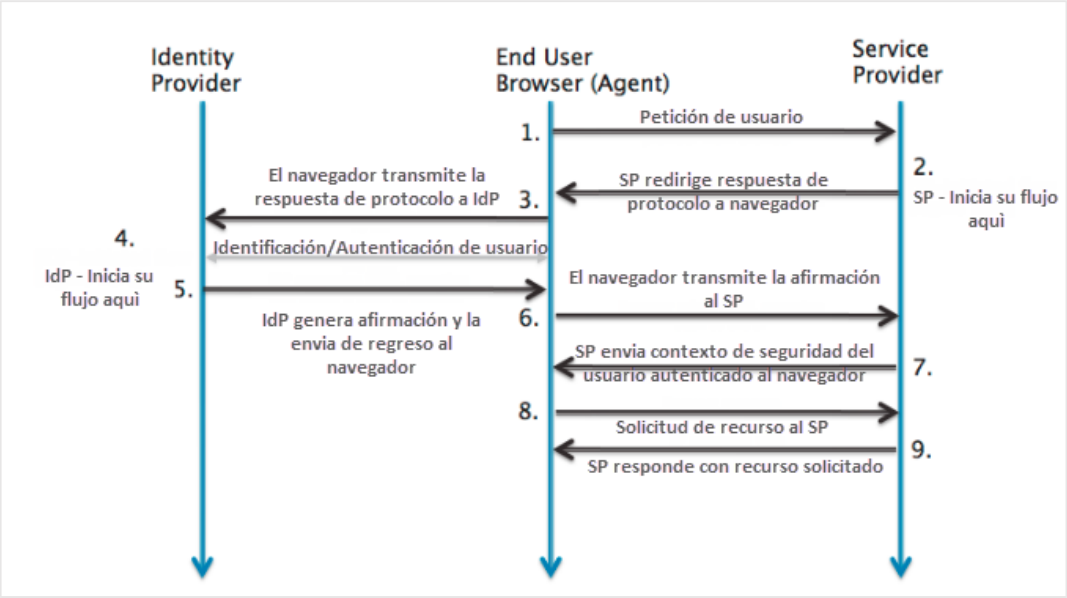

Figura 4: Comunicación entre los componentes de SSO [28].

2.2.3 Seguridad

Los componentes de seguridad en los SSO Son muy importantes en este tipo de soluciones ya que de cierto modo solo bastaría con un único usuario y contraseña para tener acceso de manera completa sobre las aplicaciones alojadas en la solución.

Algunos de los atributos que preparan este componente de las soluciones SSO tenemos indispensables: Proveedor de Identidad (IDP), Proveedores de Servicios (SPs). Servicio de Token de Seguridad (STS) [29].

En el siguiente diagrama podemos comprender como se complementa cada uno de los componentes de seguridad antes mencionados para efectos de entendimiento a la comunicación entre sí de los componentes aquí mencionado tomaremos el diagrama siguiente.

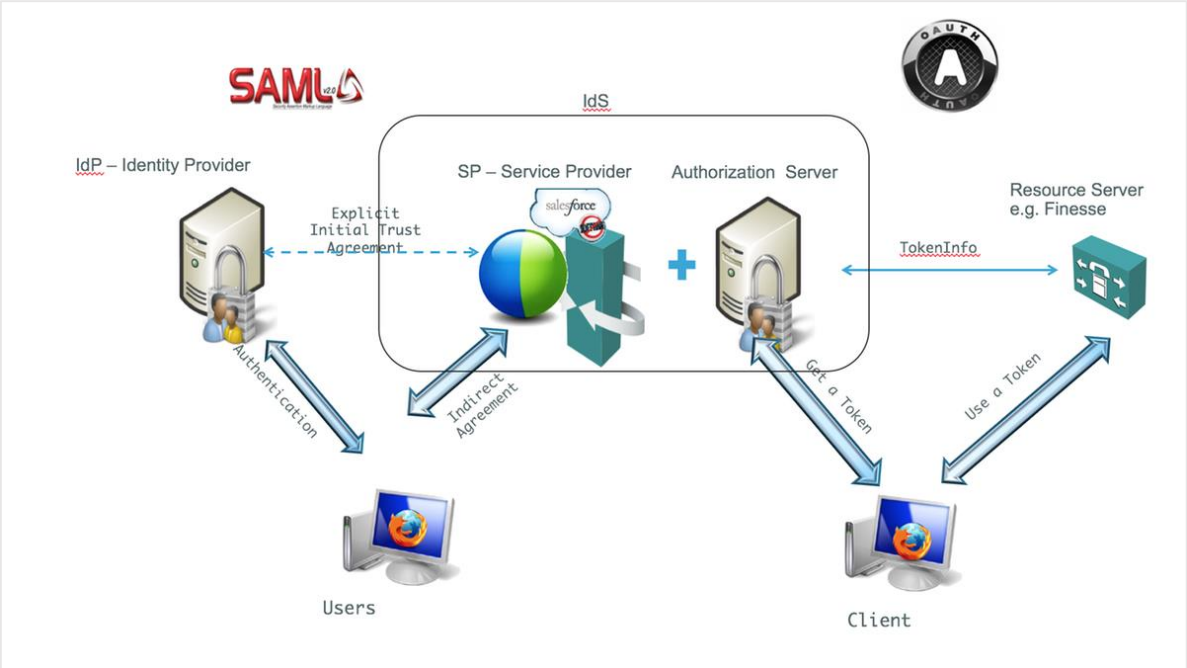

Figura 5: Seguridad en las aplicaciones SSO [29].

Se logra apreciar en la imagen anterior como se integran los componentes de seguridad antes mencionados los cuales se detallan cada uno por separado en el presente documentos en los siguientes apartados.

2.3 Servidores de identidad

El espacio de administración de identidades es complejo, con varios componentes diferentes. La gestión de la identidad sustenta a la mayoría de las organizaciones, es el sistema nervioso central de la infraestructura de TI de una organización [30]. Le dice a los usuarios y recursos de TI quién puede hacer qué y sobre qué recursos. A medida que las organizaciones se hacen más grandes, el trabajo se vuelve más complejo y crítico.

Los sistemas de control de identidad y acceso dentro de una organización abarcan varios recursos diferentes. Comienza con el servicio de directorio, que a menudo se conoce como el proveedor de identidad hasta los servicios de inicio de sesión único (SSO) y de autenticación multifactor (MFA) de las aplicaciones web. El IdP, sin embargo, es el cerebro de cualquier infraestructura de gestión de identidad.

2.3.1 Active Directory Federation Services (ADFS)

Active Directory [31] (AD) es un producto de Microsoft que consta de varios servicios que se ejecutan en Windows Server para administrar los permisos y el acceso a los recursos en red.

AD almacena los datos como objetos. Un objeto es un elemento único, como un usuario, grupo, aplicación o dispositivo, como una impresora. AD clasifica los objetos por nombre y

atributos, por ejemplo, el nombre de un usuario puede incluir la cadena de nombre, junto con la información asociada con el usuario, como contraseñas y claves de Secure Shell (SSH).

El servicio de federación de AD [32] (ADFS) es una solución de inicio de sesión único (SSO) creada por Microsoft. Como un componente de los sistemas operativos de Windows Server, proporciona a los usuarios acceso autenticado a aplicaciones que no son capaces de usar la autenticación de Windows integrada (*IWA* por sus siglas en inglés de Integrated Windows Authentication) a través de Active Directory (AD).

Desarrollado para brindar flexibilidad, ADFS brinda a las organizaciones la capacidad de controlar las cuentas de sus empleados mientras simplifican la experiencia del usuario: los empleados solo necesitan recordar un único conjunto de credenciales para acceder a múltiples aplicaciones a través del SSO.

ADFS administra la autenticación a través de un servicio de proxy alojado entre AD y la aplicación de destino. Utiliza una confianza federada, que vincula ADFS y la aplicación de destino para otorgar acceso a los usuarios. Esto permite a los usuarios iniciar sesión en la aplicación federada a través de SSO sin necesidad de autenticar su identidad directamente en la aplicación.

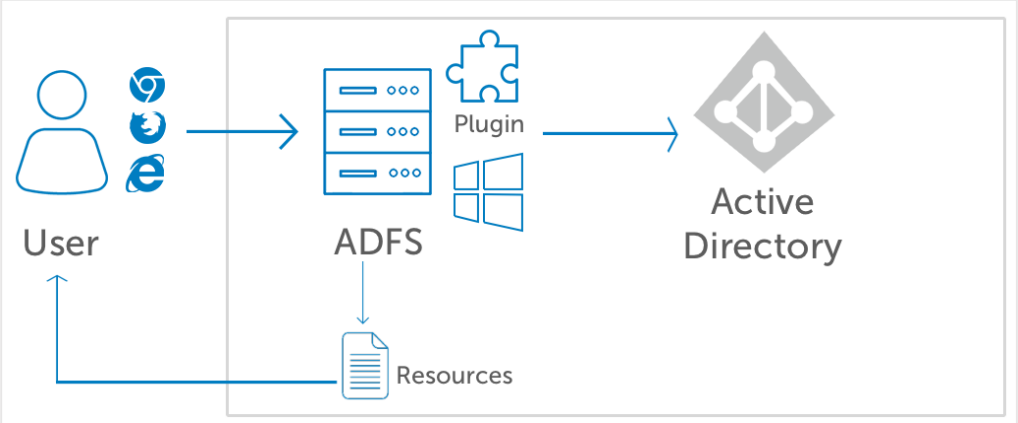

Figura 6: Integración con AD [33].

El proceso de autenticación generalmente sigue estos cuatro pasos [32]:

1. El usuario navega a una URL proporcionada por el servicio ADFS.
2. El servicio ADFS luego autentica al usuario a través del servicio AD de la organización.
3. Al autenticarse, el servicio ADFS proporciona al usuario un reclamo de autenticación.
4. El navegador del usuario luego reenvía este reclamo a la aplicación de destino, que otorga o niega el acceso en función del servicio de confianza federada creado.

ADFS nace de la necesidad de superar los desafíos de autenticación creados por AD [32] en un mundo en línea cada vez más conectado. AD e IWA han establecido limitaciones cuando se trata de la autenticación moderna, y no pueden autenticar a los usuarios que acceden a aplicaciones integradas de AD externamente. Este es un desafío en el lugar de trabajo moderno,

donde los usuarios a menudo necesitan acceder a aplicaciones que no son de su propiedad o administradas por su organización de AD.

ADFS resuelve el problema de los usuarios que necesitan acceder a las aplicaciones integradas de AD mientras trabajan de forma remota, ofreciendo una solución flexible mediante la cual pueden autenticarse utilizando sus credenciales de AD de organización estándar a través de una interfaz web. Permite a los usuarios de una organización acceder a las aplicaciones de otra organización más allá del ámbito de su dominio en AD.

La configuración de los roles en ADFS se describen en el Anexo 1.

2.3.2 Open LDAP

OpenLDAP es una implementación de código abierto del protocolo Lightweight Directory Access Protocol (LDAP) desarrollada por el proyecto OpenLDAP [34].

OpenLDAP comienza con el advenimiento de LDAP, creado en la década de 1990 por Tim Howes y sus colegas en la Universidad de Michigan [35]. LDAP se utilizó para crear rutas que podrían usarse para autenticar sistemas, aplicaciones basadas en servidores y bases de datos, entre muchos otros recursos de TI.

LDAP pronto sería utilizado por los desarrolladores para crear OpenLDAP, una implementación de servidor de código abierto de LDAP. El protocolo, junto con Kerberos, también sirvió como uno de los núcleos de Active Directory, que se convertiría en el servicio de directorio comercial más popular.

En 1998, se formó el proyecto OpenLDAP [36]. Comenzó a partir del software de la Universidad de Michigan y limpiaron los problemas conocidos en el código, expandieron la portabilidad de su plataforma y comenzaron una importante reingeniería.

La reestructuración significativa del código ha dado como resultado estructuras internas notablemente flexibles que permiten la construcción de "back-ends" de bases de datos para acceder a los datos almacenados en diferentes tecnologías de almacenamiento de bajo nivel (Berkeley DB, SQL, LDAP, shell, meta, etc.).

Otra reestructuración introdujo superposiciones (*overlays* el término en inglés) que brindan acceso a la lógica de las operaciones de directorio y permiten la introducción de nuevas capacidades sin modificar ninguno de los códigos principales de OpenLDAP. Todas estas extensiones se pueden cargar dinámicamente según sea necesario.

La suite de OpenLDAP incluye [37]:

- Slapd - demonio LDAP autónomo (servidor) (por sus siglas en inglés de stand-alone LDAP daemon).
- bibliotecas implementando el protocolo LDAP.
- Utilidades, herramientas y clientes de muestra.

La arquitectura del servidor OpenLDAP tiene dos niveles, uno es el “frontend” que maneja el procesamiento del protocolo y las conexiones de redes. El segundo nivel es el backend que hace el trabajo real de almacenar o recuperar datos en respuesta a las solicitudes LDAP. Los backends se pueden compilar estáticamente en slapd, o cuando el soporte del módulo está habilitado, se pueden cargar varios backends y a la vez varias instancias de cada backend activas por vez.

Generalmente una petición LDAP es recibida por el frontend, es decodifica y transferida a un backend para su procesamiento [38]. Cuando la petición es completada por el backend, se devuelve un resultado al frontend, quien luego transfiere el resultado al cliente LDAP. Una sobreposición (overlay) es un componente de software que puede ser insertado entre el frontend y el backend. Puede interceptar peticiones y lanzar otras acciones en ellas antes que el backend las reciba, de igual forma puede actuar sobre los resultados que llegan del backend antes que estos alcancen el frontend.

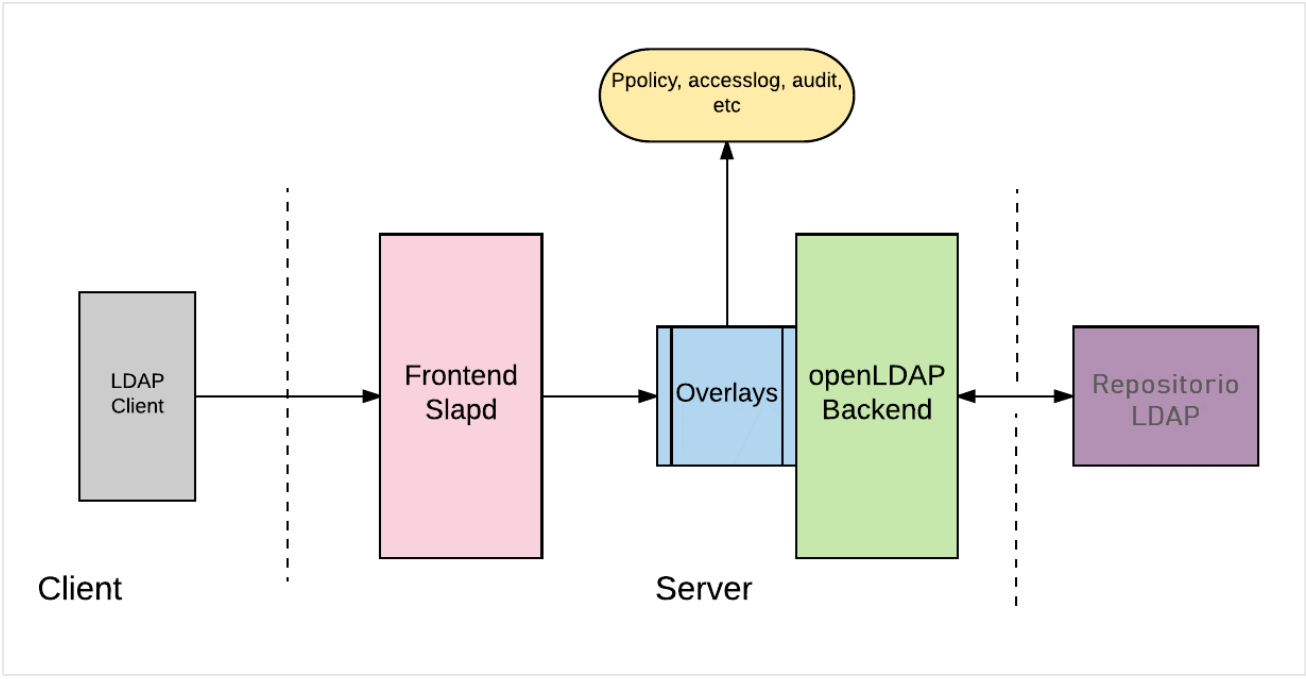

Figura 7: Servidor OpenLDAP [39].

2.3.3 Autenticación en servidor de correos POP3

La autenticación en los servidores de correos POP3 es manejada por los *SMTP* (Simple Mail Transfer Protocol) este protocoló provee la comunicación por medio de textos.

La Autenticación del SMTP se lleva a cabo por medio del protocolo SMTP AUTH luego que se realiza la autenticación lo recibe el POP3 desde el servidor remoto al cliente local. Para almacenarlos y organizarlos de esta manera podemos acceder a los correos, aunque ya se encuentre de manera offline. Cuando ya se encuentran de manera local son removidos del servidor. El POP3 maneja los puertos 110 y 995 en donde el primero de estos no es cifrado y es

el predeterminado en cambio el puerto 995 realiza la conexión de manera Cifrada. De esta forma al utilizar el SMTP como protocolo de transferencia utiliza el puerto 465 para el cifrado [40].

El método de acceso al servidor de correos se realiza por medio de TCP/IP de manera encriptada utilizando el puerto 995, Los comandos de POP3 están compuestos por un número limitado de caracteres de tres a cuatro incluyendo un parámetro o más. Las palabras claves diferencian mayúsculas y minúsculas las cuales consisten en ASCII y estos se separan por un único espacio a diferencia de las palabras claves los argumentos cuentan con una longitud de hasta 40 caracteres.

Las sesiones de POP3 se manejan con tres estados diferentes:

1. Authorization: al establecer la conexión TCP y se ha recibido respuesta del servidor POP3. Un ejemplo de la respuesta del servidor POP3:

S: +OK POP3 Server Ready

Existen dos posibles mecanismos para la AUTHORIZATION por medio de un usuario y contraseña o el comando APOP, es necesario que al menos uno de estos mecanismos sea efectuado en el POP3, al establecer el acceso al agente de correos se adquiere un acceso exclusivo, esto para que los mensajes no sean eliminados o modificados antes que la sesión ingrese al estado de UPDATE, cuando este proceso se efectúa correctamente se ejecuta el indicador de estado positivo es entonces ingresa al estado de TRANSACTION [40]. En este estado no se cuenta con marcas de mensajes como eliminados, no se brinda acceso al agente de correo, cuando no se adquiere un bloqueo, por lo que es denegado el acceso y no es posible analizar, se genera un indicador negativo a la petición cuando el bloque es del servidor, este responde con la liberación del bloqueo previamente al rechazar el comando. Si logra generar el indicador con el estado negativo el servidor cierra la conexión, Si no se realice el cierre de la conexión, los clientes pueden generar un comando nuevo de AUTENTICACIÓN e iniciar nuevamente, de igual forma si no se desea el cliente puede generar el comando QUIT [41].

Al realizar la apertura del agente de correo en el servidor se asigna un número de mensaje a cada uno de los mensajes además de tomar el tamaño de los mensajes en los octetos. De este modo tendríamos el primer mensaje dentro de gestor de correos con la asignación “1” el segundo “2”, sucesivamente. De esta manera se agregan los números a los mensajes dentro del agente de correos, los comandos POP3 y las respuestas se representan los números de mensajes el tamaño es expresado en base 10 conocido como decimales. El estado de AUTHORIZATION maneja el QUIT de la siguiente manera:

QUIT

Arguments: none

Restrictions: none

Possible Responses:

+OK

Examples:

C: QUIT

S: +OK dewey POP3 server signing off

2. Transaction: cuando se realiza la identificación en el servidor POP3, bloqueado y abierto el buzón de correos de forma correcta. Se ha establecido el estado de TRANSACTION, en esta instancia el cliente puede emitir cualquiera de los comandos POP3. Luego de cada comando el servidor genera una repuesta, en definitiva, el cliente ejecuta el comando QUIT y la sesión POP3 cambia a UPDATE [40].

Los comandos validos en el estado de TRANSACTION:

STAT

Arguments: none

Restrictions:

may only be given in the TRANSACTION state

El servidor POP3 genera la respuesta positiva con la información para el agente de correos, la información se genera en una lista llamada "lista de despliegue" en el agente de correos. Para los agentes de correos es requerido sea un formato determinado para las listas despegables. Las respuestas positivas se conforman por "+OK" continuo de un solo espacio, el número de mensajes. Seguido un Espacio y su tamaño en octetos, esta nota no es requerida en el tamaño del agente de correos, en la implementación mínima deben finalizar la línea en respuestas con un par de CRLF, Permiten al cliente realice el analices de los mensajes en el agente de correos. Los mensajes marcados como eliminados no aparecen en el total de los mensajes del agente de correos [40].

Possible Responses:

+OK nn mm

Examples:

C: STAT

S: +OK 2 320

LIST [msg]

Arguments:

a message-number (optional), which, if present, may NOT

refer to a message marked as deleted

Restrictions:

may only be given in the TRANSACTION state

Se establece un argumento y el servidor envía una respuesta positiva con una línea que contiene información para el mensaje, esta línea es llamada como lista escáner para cada mensaje.

Cuando no se establece ningún argumento y el servidor envía una respuesta positiva la respuesta no contiene una línea de información si no multilínea. Después de la inicial +OK para cada uno de los mensajes por medio del agente de correos. El servidor POP3 responde con la lista de escáner para el mensaje. si no hay mensajes en el agente de correos, responde sin la lista de escaneo y genera una respuesta positiva con el octeto de determinación y un par de CRLF [40].

Se requiere que todos los servidores POP3 mantenga un formato establecido para los listados de escaneo. La lista de exploración consta del número de mensajes, con un único espacio y el tamaño del mensaje en los octetos. Así se establece el cálculo para el tamaño del mensaje porque todos los mensajes que se transmiten tienen mismo formato, el conteo del número de octetos para el mensaje dentro del host en el servidor puede ser diferente del número de octeto determinado en el mensaje por las convenciones locales para elegir el fin de la línea. Un ejemplo, si el host del servidor representa de manera interna el final de la línea como un solo carácter, el servidor POP3 solamente contara cada vez que aparezca el carácter en un mensaje como dos octetos. No se establece ningún requisito sobre lo que sigue en el tamaño del mensaje dentro de la lista de escaneo, se debe terminar la línea de respuesta con un par de CRLF en las implementaciones mínimas, las más avanzadas si puede contener más información, como se analiza desde el mensaje, (se describen en el Anexo 2) de la estructura del mensaje y los diferentes casos.

3. Update: El servidor POP3 libera todos los recursos adquiridos durante el estado anterior y cierra la conexión TCP. Cuando se ejecuta el comando de salida QUIT en el estado de TRANSACTION inicia la sesión UPDATE, además si se genera el comando QUIT desde la AUTHORIZATION se finaliza, pero no se ingresa al estado UPDATE [40].

Si se finaliza la sesión y no es por el lado del cliente no se genera el comando QUIT el servidor no ingresa al estado UPDATE y los mensajes se mantienen y no son eliminados del agente de correos.

QUIT

Arguments: none

Restrictions: none

El servidor POP3 identifica y elimina los mensajes que han sido marcados previamente como eliminados en la entrega de correos, genera la respuesta sobre el estado de la operación. Al generarse un error como la pérdida o escases de un recurso que se localizan a eliminar el mensaje, el agente de correos responde con la eliminación de algún número de mensajes o ninguno de los que han sido marcados como eliminados.

Cuando finalmente se ejecuta los eventos en el estado UPDATE este ha eliminado correctamente los mensajes o no y se libera el bloqueo al acceso del agente de correos y se cierra la conexión TCP.

2.3.4 Autenticación en Bases de Datos

La autenticación como tal se basa en el proceso para validar si la persona que está tratando de ingresar es la que debe ser.

Las bases de datos es la serie de datos organizados y relacionados entre sí, los cuales son almacenados por medio de sistemas de información y almacenados de manera correcta para luego poder realizar la manipulación de estas [42].

Unas características que mencionar son:

- Independencia lógica y física de los datos.
- Redundancia mínima.
- Acceso concurrente por parte de múltiples usuarios.
- Integridad de los datos.
- Consultas complejas optimizadas.
- Seguridad de acceso y auditoría.
- Respaldo y recuperación.

La autenticación en base de datos es muy sencilla de comprender e implementar, inicialmente se debe comprender cuál es la estructura en la que se establece la autentificación en las bases de datos. Existen tres niveles de seguridad para poder realizar la autenticación en las bases de datos.

El primer nivel de Acceso se establece en el servidor de este modo gestionar quienes pueden acceder al servidor, de igual forma se gestionan los roles que se está mandando desde la aplicación donde se encuentra Login.

Segundo nivel es estando en el servidor de base de datos se debe asignar un usuario para tener acceso a las bases de datos que desea acceder, esto para cada una de las bases de datos que se desee ingresar el login de manera análoga [42].

El tercer nivel de acceso se establece cuando el acceso a la base de datos está abierto y este tiene acceso a sus objetos donde obtendrá la información sobre las credenciales de autenticación enviadas de esta forma se envía una respuesta de autorización o denegación. De este modo tiene acceso a las aplicaciones.

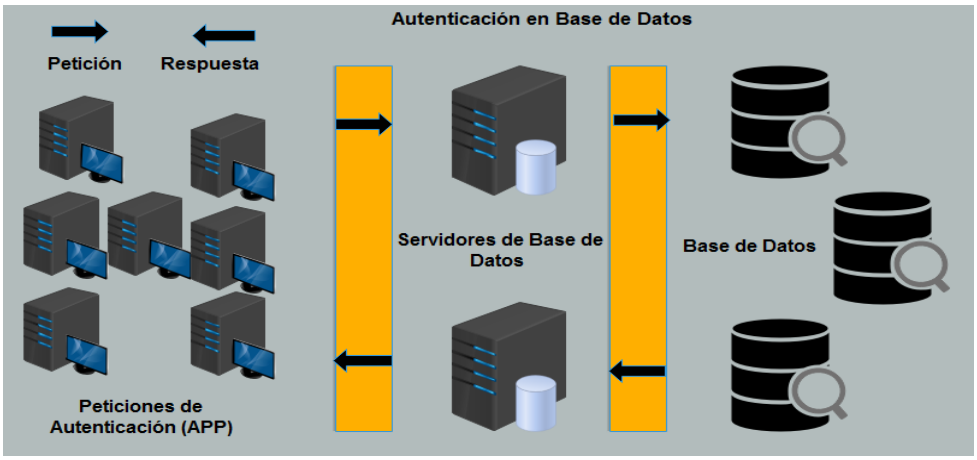

Figura 8: Autenticación en base de datos. Elaboración propia

La información no se envía como texto plano a través de la red de manera segura ya que se implementas diferentes medidas de seguridad en la data enviada uno de los controles como tal es el manejo de las contraseñas con un hash estas funciones criptográficas ayudan a que las contraseñas sean seguras además de nivel de complejidad de la estructura misma de la contraseña al ser de una longitud y combinaciones de números caracteres y letras los manejos de bloqueos por intentos fallidos y expiración de contraseñas de igual forma existen diferentes controles de seguridad.

2.4 Servidores WEB

Los servidores web se conocen como servidor HTTP en donde se procesa una aplicación del lado del servidor ejecutando la conexión bidireccional, asíncronas con el cliente de este modo suministra una respuesta del lado del cliente a cualquier aplicación, las respuestas recibidas del lado del cliente son compiladas y ejecutadas en el navegador.

Se estudiarán dos servidores web, uno de código abierto para plataformas Unix, Microsoft Windows, Macintosh conocido como Apache HTTP Server. Otro es un conjunto de servicios para la plataforma Microsoft, conocido como Internet Information Services (IIS), que es especialmente utilizado en servidores web.

Existen otros servidores web en las mismas clasificaciones, pero todos contemplan un objetivo y es el manejo de servicios web por medio del protocolo HTTP.

A nivel de popularidad según Web Technology Surveys establece que los servidores Apache HTTP Server e Internet Information Services son de los más utilizados [43].

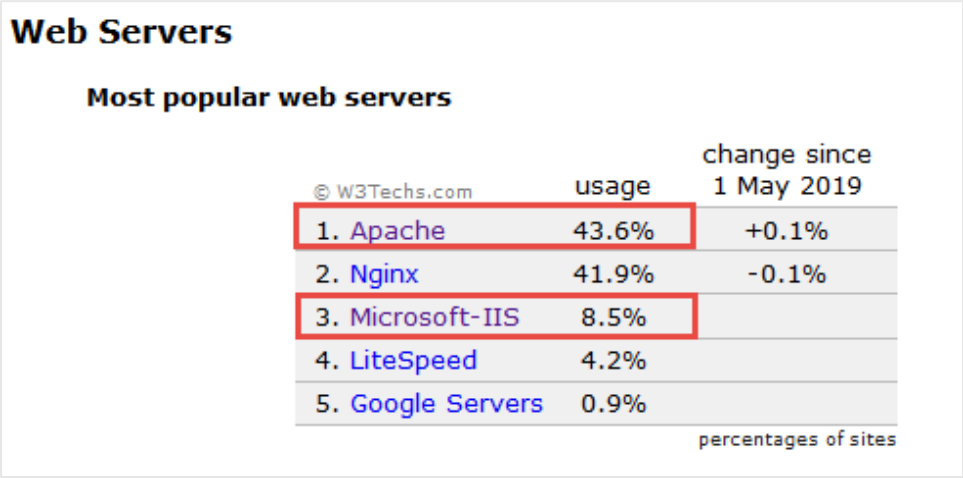

Figura 9: Popularidad de los Servidores Web [43].

2.4.1 Apache HTTP Server

El servidor web Apache es uno de los más mencionados en la actualidad y se establece con mayor presencia en el mercado, es de código abierto y su nombre oficial es "Apache HTTP Server" el cual ha sido desarrollado por "Apache Software Foundation".

El servidor web acepta solicitudes del cliente (ej. navegadores) y envía las respuestas a la petición que se solicita (Ej. Página web). Maneja diferentes módulos que aseguran más funciones a su software como lo son MPM (el manejo de procesamientos múltiples) o MOD_SSL para poder habilitar la complejidad con SSL v3 y TLS [44].

Las características comunes que se incluye Apache son:

- .htaccess
- IPv6
- FTP
- HTTP/2
- Perl, Lua, y PHP
- Anulación del ancho de banda
- WebDAV
- Balanceo de carga
- Re-escritura de URL
- Rastreo de sesión
- Geo ubicación basada en dirección IP

Las configuraciones en apache se realizan por medio de directivas en archivos planos, se tiene como archivo principal de configuración el "httpd.conf" y contiene una directiva por cada línea.

La estructura de Apache se establece por módulos, se configuran a través de las directivas en cada uno de módulos. Los módulos que pertenecen a Apache se pueden clasificar en tres: Módulos Base, Módulos Multiproceso, Módulos Adicionales [44].

- Módulos Base y Multiprocesos: La unión de estos módulos es posible porque los módulos Base son funciones básicas del servidor y el Multiproceso tiene como principal funcionamiento unir los puertos de la máquina, aceptando las peticiones y enviando atender las peticiones. Los módulos base y multiproceso (se describen en el Anexo 3).
- Módulos Adicionales: Cualquier otro módulo que le añada una funcionalidad al servidor. Cuando se desea agregar una utilidad al servidor se añade un nuevo módulo de este modo limitamos la instalación de software. Además, las funcionalidades básicas y las configuraciones para los módulos se manejan en el fichero de httpd.conf. Los módulos adicionales (se describen en el Anexo 4).

El funcionamiento interno de Apache HTTP Server se muestra en la siguiente imagen:

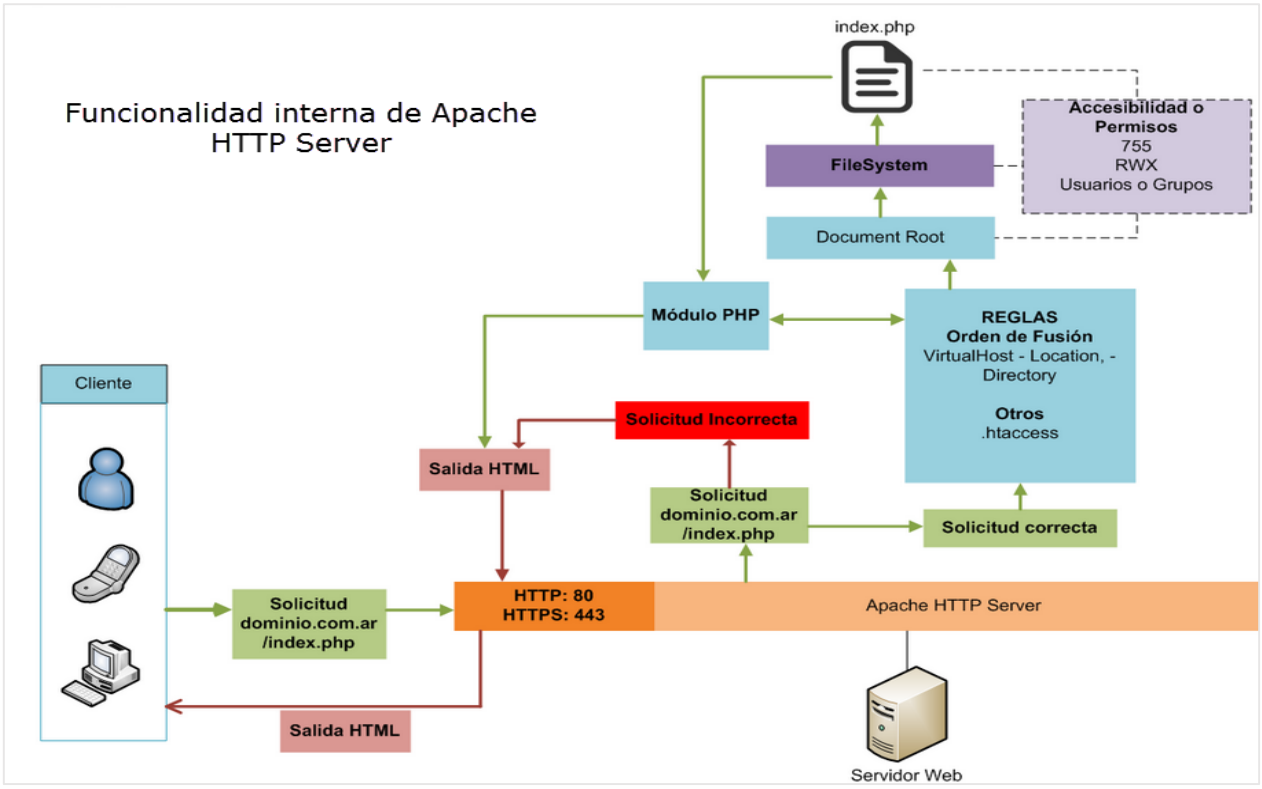

Figura 10: Mecanismo de funcionalidad de Apache [76].

La arquitectura es simple conociendo el funcionamiento interno del servidor web (Apache).

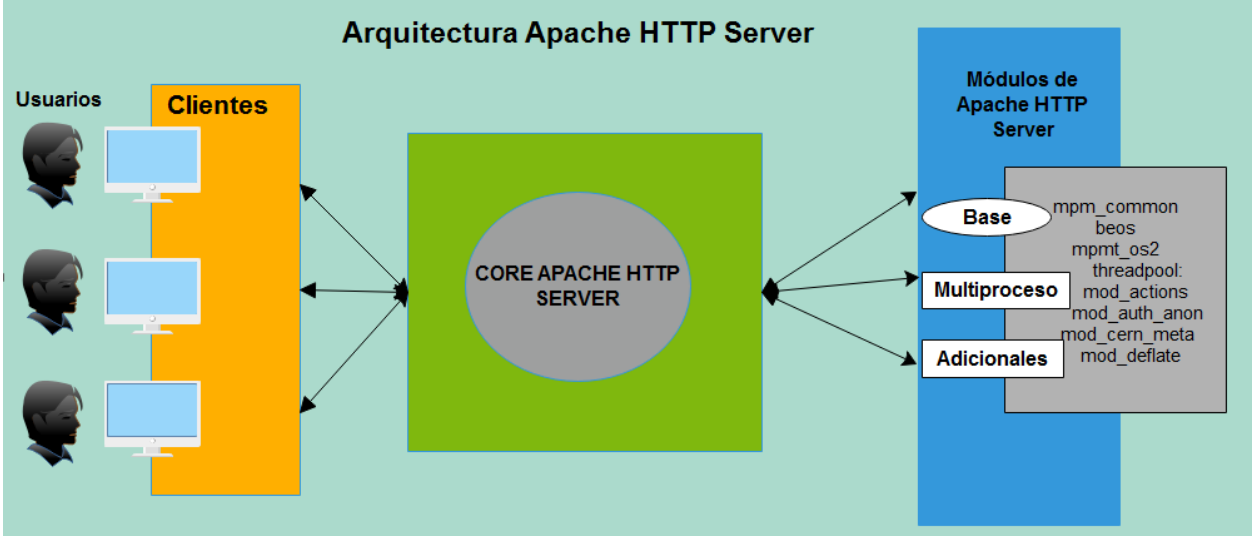

Figura 11: Arquitectura de servidor Web Apache HTTP Server. Elaboración propia

Servicios: Apache HTTP Server administra las solicitudes / reglas y devuelve los documentos web. Las respuestas del lado del servidor Apache siempre son en formato HTML, de este modo se exponen las aplicaciones web y se configuran los módulos necesarios. Apache comprende las acciones desarrolladas en cada una de las aplicaciones en diferentes lenguajes y utiliza diferentes módulos para generar una respuesta HTML y ser mostrada al cliente (navegador).

Cuando se realiza la solicitud del cliente se inicia con el nombre del dominio y su index por defecto, para generar toda la comunicación vía cliente servidor, Apache recibe la solicitud, verifica las configuraciones y la procesa. El flujo para acceder a un recurso es:

- Si el recurso existe y es accesible, ya sea en el htdocs o directorio, asigna los permisos y se ejecuta el módulo del lenguaje especifico.
- Envía una respuesta HTML plano, lo regresa al cliente (navegador) y finaliza con el código HTTP 200.
- Si la solicitud es correcta pero el recurso no existe o hay un error interno de configuración de Apache, se genera un error del tipo HTTP 404 como respuesta HTML.
- También puede ser que en el servidor este mal configurado y obtenemos errores del tipo HTTP 500.

Seguridad: La seguridad de los servidores web debe ser siempre un punto importante en su implementación, los controles incluyen la instalación de los últimos parches de seguridad, restringir los accesos por direcciones IP, así como ocultar la versión e información relaciona a las configuraciones del servidor [45]. Además, Apache trabaja en una cuenta propia y grupos de usuarios que al utilizar el mod_security permite:

- Filtración simplificada.

- Filtraciones basadas en las expresiones regulares (por las cadenas de textos).
- Validaciones en la codificación de las URL.
- Validar la codificación apegada al estándar de codificación de caracteres (Unicode).
- Auditables.
- Prevención sobre los ataques con cadenas de caracteres vacíos (NULL Byte).
- Limitar la memoria en subida.
- Enmascarar la identidad del servidor.

El detalle de las configuraciones de seguridad mínima en Apache se describe en el Anexo 5.

2.4.2 Internet Information Services

Internet Information Services (IIS) para Windows® Server es un servidor web flexible, seguro y manejable para hospedar cualquier "servicio" en la web. Desde la transmisión de medios a las aplicaciones web, la arquitectura abierta y escalable de IIS maneja las tareas más exigentes. La versión más reciente es IIS 10.0, para Windows 10 y Windows Server 2016.

Desde la versión 7 de se proporciona una arquitectura de procesamiento de solicitudes que incluye [46]:

- El Servicio de Activación de Procesos de Windows (*WAS* por sus siglas en inglés de Windows Process Activation Service), que permite que los sitios utilicen protocolos distintos de HTTP y HTTPS.
- Un motor de servidor web que se puede personalizar agregando o eliminando módulos. Los módulos son características individuales que el servidor utiliza para procesar solicitudes. Existen módulos nativos que pueden ser agregados dependiendo de la necesidad: HTTP, seguridad, contenido, compresión (en las tuberías), caché, registro y diagnóstico, módulo de soporte gestionado. Además de los módulos nativos, IIS permite utilizar módulos de código administrado para ampliar la funcionalidad de IIS.
- Tuberías integradas (Integrated Pipeline) de procesamiento de solicitudes de IIS y ASP.NET.

IIS contiene varios componentes que realizan funciones importantes para las aplicaciones y servidor web. Cada componente tiene responsabilidades, como escuchar las solicitudes realizadas al servidor, administrar procesos y leer archivos de configuración. Estos componentes incluyen:

Escuchas de protocolo, como HTTP.sys (Hypertext Transfer Protocol Stack):

Escuchas de protocolo (protocol listeners) reciben solicitudes específicas de protocolo, las envían a IIS para su procesamiento y luego envían las respuestas a los solicitantes. Por ejemplo, cuando un navegador cliente solicita una página web de Internet, el oyente HTTP,

HTTP.sys, recoge la solicitud y la envía a IIS para su procesamiento. Una vez que IIS procesa la solicitud, HTTP.sys devuelve una respuesta al navegador del cliente.

De forma predeterminada, IIS proporciona HTTP.sys como el protocolo que escucha las solicitudes HTTP y HTTPS. HTTP.sys se introduce en IIS 6.0 como un escucha de protocolo específico de HTTP para solicitudes HTTP. HTTP.sys sigue siendo la escucha HTTP en IIS 7 y versiones posteriores, pero incluye soporte para SSL (por sus siglas en inglés de Secure Sockets Layer).

HTTP.sys proporciona los siguientes beneficios:

- ✓ Caché en modo kernel. Las solicitudes de respuestas almacenadas en caché se sirven sin cambiar al modo de usuario.
- ✓ Cola de solicitud en modo kernel. Las solicitudes causan menos sobrecarga en el cambio de contexto porque el kernel envía las solicitudes directamente al proceso de trabajo correcto. Si no hay un proceso de trabajo disponible para aceptar una solicitud, la cola de solicitudes en modo kernel retiene la solicitud hasta que un proceso de trabajo la retire.
- ✓ Solicitud de pre-procesamiento y filtrado de seguridad.

Servicio de publicación World Wide Web (servicio WWW)

El Servicio WWW lee la información de configuración de la metabase de IIS y usa esa información para configurar y actualizar el HTTP listener, HTTP.sys. Además, el servicio WWW inicia, detiene, recicla, monitorea y administra los procesos de trabajo que procesan las solicitudes HTTP.

Supervisa el rendimiento y proporciona contadores de rendimiento para los sitios web y para el caché de IIS.

Servicio de activación de procesos de Windows (WAS)

Administra la configuración del grupo de aplicaciones y los procesos de trabajo en lugar del Servicio WWW. Esto le permite utilizar la misma configuración y modelo de proceso para los sitios HTTP y no HTTP.

Los grupos de aplicaciones (application pool) separan las aplicaciones por límites de proceso para evitar que una aplicación afecte a otra aplicación en el servidor.

WAS lee cierta información de archivos de configuración y pasa esa información a los adaptadores de escucha en el servidor. Los adaptadores de escucha son componentes que establecen la comunicación entre WAS y escuchas de protocolo, como HTTP.sys. Una vez que los adaptadores de escucha reciben información de configuración, configuran sus escuchas de protocolo relacionadas y preparan a las escuchas para escuchar las solicitudes.

IIS 7 y versiones posteriores utilizan un sistema de configuración basado en XML para almacenar las configuraciones de IIS [47]. El sistema de configuración se introdujo con ASP.NET y se basa en un sistema jerárquico de sistema de administración que utiliza archivos *.config. Los archivos de configuración principales son:

- ✓ ApplicationHost.config
- ✓ Administration.config
- ✓ Redirection.config

Procesamiento de solicitud HTTP en IIS [46]

1. Cuando un navegador de cliente inicia una solicitud HTTP para un recurso en el servidor web, HTTP.sys intercepta la solicitud.
2. HTTP.sys se pone en contacto con WAS para obtener información del almacén de configuración.
3. WAS solicita información de configuración del almacén de configuración, applicationHost.config.
4. El servicio WWW recibe información de configuración, tales como el grupo de aplicaciones y la configuración del sitio.
5. El servicio WWW utiliza la información de configuración para configurar HTTP.sys.
6. WAS inicia un proceso de trabajo para el grupo de aplicaciones al que se realizó la solicitud.
7. El proceso de trabajo procesa la solicitud y devuelve una respuesta a HTTP.sys.
8. El cliente recibe una respuesta.

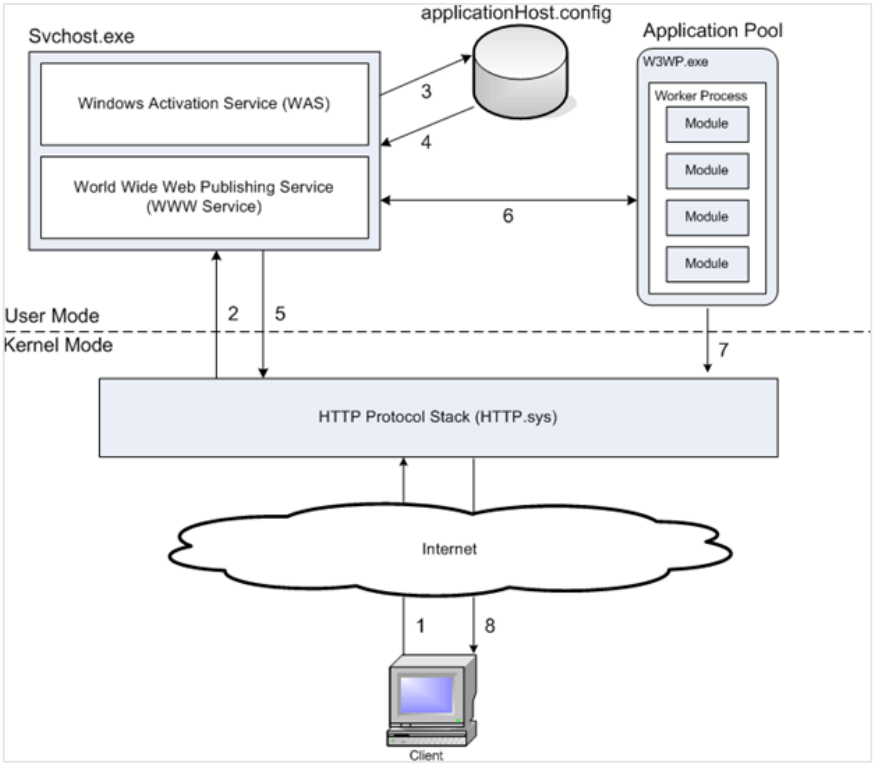

Figura 12: Procesamiento de solicitudes [46].

El “proceso de trabajo” al que se hace referencia es una instancia del ejecutable W3WP.exe que se relaciona con un grupo de aplicaciones (application pool). En un proceso de trabajo, una solicitud HTTP pasa a través de varios pasos ordenados, llamados eventos. En cada evento, un módulo nativo procesa parte de la solicitud, como autenticar al usuario o agregar información al registro de eventos. Cuando la solicitud pasa a través de todos los eventos, la respuesta se devuelve a HTTP.sys.

Algunos elementos de seguridad que se integra en el servidor IIS:

- Aislamiento automático de sitios web [58]: IIS 7.0 ofrece un mayor aislamiento de la aplicación al proporcionar a los procesos de los trabajadores una identidad completamente única y una configuración de espacio aislado de forma predeterminada, lo que reduce aún más los riesgos de seguridad. IIS 7.0 incluye el aislamiento automático del grupo de aplicaciones y puede proteger miles de sitios web en un solo servidor. Esto permite que cada sitio web se ejecute en su propio espacio de memoria con una identidad única generada automáticamente, lo que ayuda a garantizar que las aplicaciones no se vean afectadas por otros fallos o violaciones de seguridad de las aplicaciones que se ejecutan en el mismo servidor. Esta capacidad permite a las organizaciones consolidar más sitios web en menos servidores, y aumenta la seguridad y confiabilidad para todos los sitios web que se ejecutan en un host compartido.
- Autorización de URL [49]: IIS 7.0 almacena las reglas de autorización de URL en el archivo web.config de una aplicación, de modo que las reglas de autorización que protegen contra el acceso no autorizado siguen el contenido, incluso cuando el contenido se mueve a un servidor diferente o incluso a un nuevo dominio. IIS 7.0 también admite la autorización de URL de ASP.NET para todos los tipos de solicitudes de contenido web en la canalización integrada.
- Solicitud de filtrado incorporada [49]: El filtrado de solicitudes de IIS 7.0 permite a los administradores implementar políticas de aceptación de URL tanto global como por URL. El filtrado de solicitudes ayuda a proteger el servidor al garantizar que solo se procesen las solicitudes válidas. Los administradores pueden aumentar la seguridad del servidor web al proporcionar múltiples opciones de filtrado que pueden evitar que se procesen las URL maliciosas o incorrectas.

2.5 Comunicación entres servidores

2.5.1 Protocolos de comunicación

La presente investigación estudia los estándares dominantes de web abierta para identidad en línea [50]. Los más destacados para la integración de soluciones SSO, son: *SAML, OAuth 2.0* y *OpenID Connect.*

Previo a la descripción formal de estos protocolos, se detallan puntos clave que se deben considerar para la correcta interpretación de estos [50]:

- OAuth 2.0 es un marco de autorización, no un protocolo de autenticación. OAuth 2.0 se puede usar para muchas tareas interesantes, una de las cuales es la autenticación de personas.
- OAuth 2.0 es un estándar ligeramente reciente que fue desarrollado por Google y Twitter para habilitar los inicios de sesión de Internet optimizados. OAuth 2.0 utiliza una metodología similar a SAML para compartir información de inicio de sesión. SAML proporciona más control a las empresas para mantener sus inicios de sesión SSO más seguros, mientras que OAuth 2.0 es mejor en dispositivos móviles y usa JSON.
- OpenID Connect es un "perfil" de OAuth 2.0 diseñado específicamente para la liberación y autenticación de atributos.
- Facebook y Google son dos proveedores de OAuth 2.0 que se puede utilizar para iniciar sesión en otros sitios de internet, como por ejemplo acceder a Spotify.

SAML

Security Assertion Markup Language (*SAML* por sus siglas en inglés) es un estándar abierto [51] que permite la comunicación segura de identidades entre organizaciones mediante las funciones de autenticación y autorización.

Las transacciones SAML utilizan XML para estandarizar las comunicaciones entre el proveedor de identidad (IdP) y los proveedores de servicios (SP). SAML es el enlace entre la autenticación de la identidad de un usuario y la autorización para usar un servicio.

OASIS aprobó SAML 2.0 en 2005, y el estándar cambió significativamente de la versión 1.1, tanto que las versiones son incompatibles. SAML habilita el SSO, un término que significa que los usuarios inician sesión una vez, y esas mismas credenciales son reutilizadas para el inicio de sesión en otros proveedores de servicios.

El trabajo de SAML es transferir la identidad del usuario de un lugar (el proveedor de identidad) a otro (el proveedor de servicios). Esto lo hace a través de un intercambio de documentos XML que so firmados digitalmente para comprobar que el mensaje proviene de una fuente de confianza.

Se describe la secuencia de eventos para el inicio de autenticación único que detalla cómo trabaja el protocolo SAML [52] [75]:

1. El usuario solicita acceso a un recurso protegido.
2. El SP verifica si ya está autenticado dentro del sistema. Si es así, salta al paso 7; si no lo está, el SP inicia el proceso de autenticación.
3. La aplicación identifica el origen del usuario (por subdominio de la aplicación, dirección IP del usuario o similar) y redirige al usuario al IdP apropiado, solicitando la autenticación. Esta es la solicitud de autenticación (SAML 2.0 AuthnRequest).

4. Después de la posible identificación del usuario, el flujo de SSO reanuda.
5. El IdP construye la respuesta de autenticación en forma de un documento XML (una afirmación, que se conoce como SAML Assertion, por el término en inglés) que contiene el nombre de usuario o la dirección de correo electrónico del usuario, lo firma con un certificado X.509 y publica esta información en el proveedor de servicios.
6. El proveedor de servicios, que ya conoce al IdP y tiene una huella digital de certificado, recupera la respuesta de autenticación y la valida utilizando la huella digital del certificado.
7. Se establece la identidad del usuario y se le proporciona acceso a la aplicación.

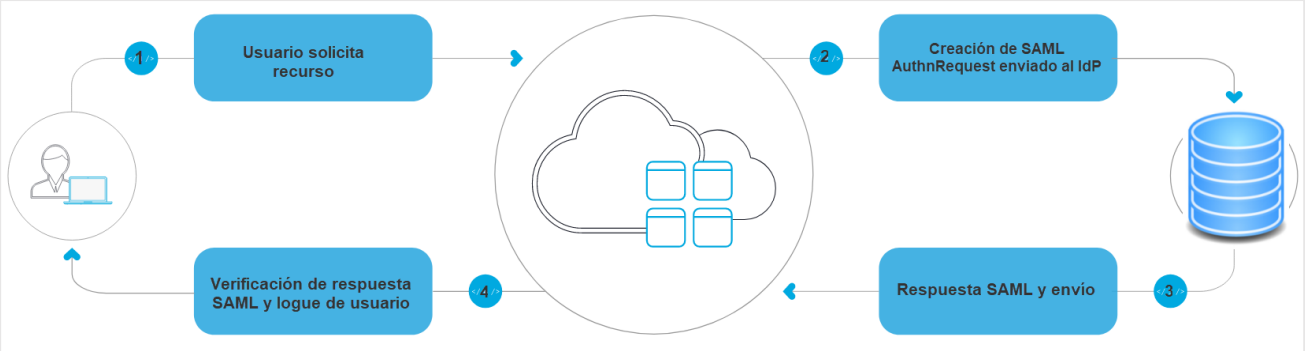

Figura 13: Flujo SSO empezado por SP [53].

Hay tres tipos diferentes de afirmaciones SAML (Assertion) [51]

1. Autenticación: Demuestran la identificación del usuario y proporcionan el tiempo en que el usuario inició sesión y qué método de autenticación utilizaron.
2. Atributo: La aserción de atribución pasa los atributos SAML al proveedor de servicios; los atributos SAML son datos específicos que proporcionan información sobre el usuario.
3. Decisión de autorización: dice si el usuario está autorizado para usar el servicio o si el proveedor de identidad denegó su solicitud debido a una falla de contraseña o falta de derechos sobre el servicio.

Beneficios de la autenticación SAML [54]

Estandarización: SAML es un formato estándar que permite una interoperabilidad perfecta entre sistemas, independientemente de la implementación. Elimina los problemas comunes asociados con la implementación y la arquitectura del proveedor y la plataforma específica.

Mayor seguridad: La seguridad es un aspecto clave del desarrollo de software, y para entornos empresariales, es extremadamente importante. SAML proporciona un único punto de autenticación, que ocurre en un IdP seguro. Luego, SAML transfiere la identidad a los

proveedores de servicios. Esta forma de autenticación garantiza que las credenciales no abandonen el límite del cortafuego.

Documentación: El protocolo SAML está ampliamente documentado en los productos SSO comerciales para facilitar la integración. Algunas soluciones como OneLogin proveen incluso cajas de herramientas (toolkit como término en inglés) para cinco plataformas de desarrollo web [53].

A continuación, se detalla un ejemplo de SAML AuthNRequest. Se detalla la aplicación que sirve como IdP y el certificado de seguridad para ser validado, entre otro.

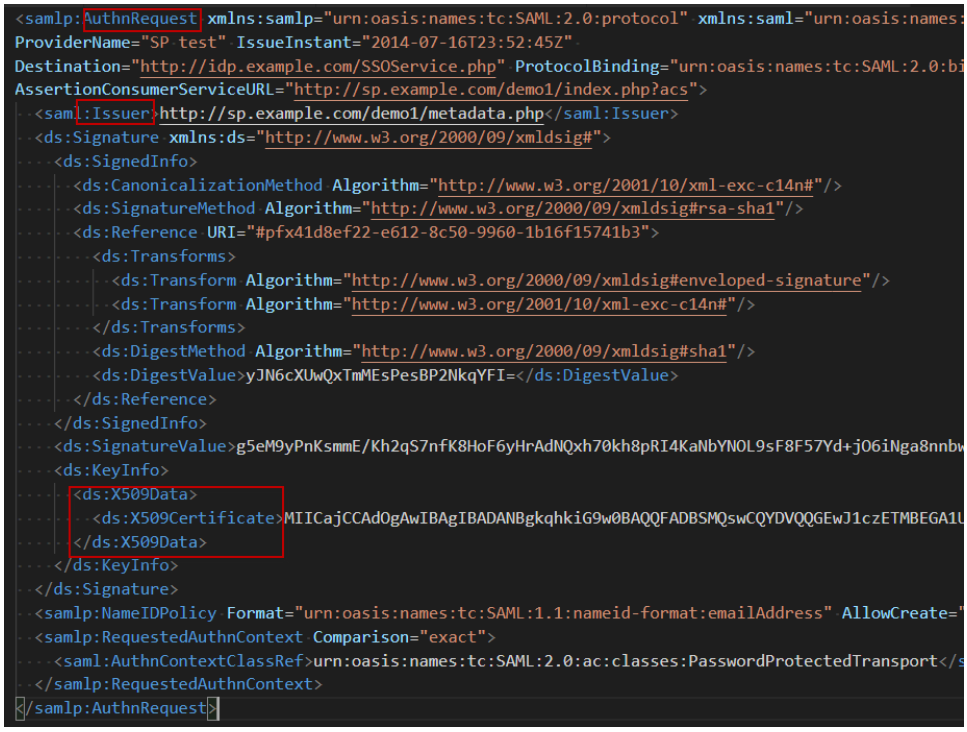

Figura 14: Representación de SAML. Elaboración propia

OAuth 2.0

Es un protocolo estándar de la industria para la autorización [55]. OAuth 2.0 reemplaza el trabajo realizado en el protocolo original de OAuth creado en 2006 que fue desarrollado, en parte, para compensar las deficiencias de SAML en las plataformas móviles, y está basado en JSON en lugar de XML. OAuth 2.0 se enfoca en la simplicidad de desarrollo del cliente y permite a las aplicaciones obtener acceso limitado a cuentas de usuario en un servicio HTTP, como Facebook, Twitter, GitHub, otros.

La autenticación del usuario se delega al servicio que aloja la cuenta del mismo y autoriza a las aplicaciones de terceros el acceso a dicha cuenta de usuario. Proporciona flujos de autorización para aplicaciones web, escritorio y móviles.

OAuth define cuatro roles [56] [57]

Rol propietario del recurso: La entidad que puede otorgar acceso a un recurso protegido. Normalmente este es el usuario final. El acceso de la aplicación a la cuenta del usuario se limita al "alcance" de la autorización otorgada (acceso de lectura o escritura).

Rol cliente: El cliente es la *aplicación* que desea acceder a la cuenta del *usuario*. Antes de que pueda hacerlo, debe ser autorizado por el usuario, y dicha autorización debe ser validada por la API.

Rol servidor de recursos / API: El servidor que aloja los recursos protegidos. Esta es la API a la que desea acceder.

Rol servidor de autorización / API: El servidor que autentica al propietario del recurso y que emite los tokens de acceso después de obtener la autorización correspondiente. La API de igual forma permite refrescar y revocar un conjunto de tokens. Este servidor funciona como un proveedor de identidades IdP.

A continuación, se ilustran como los roles interactúan entre sí [56], en lo que llamamos a este punto un flujo abstracto o genérico. Es cómo generalmente interactúan entre sí.

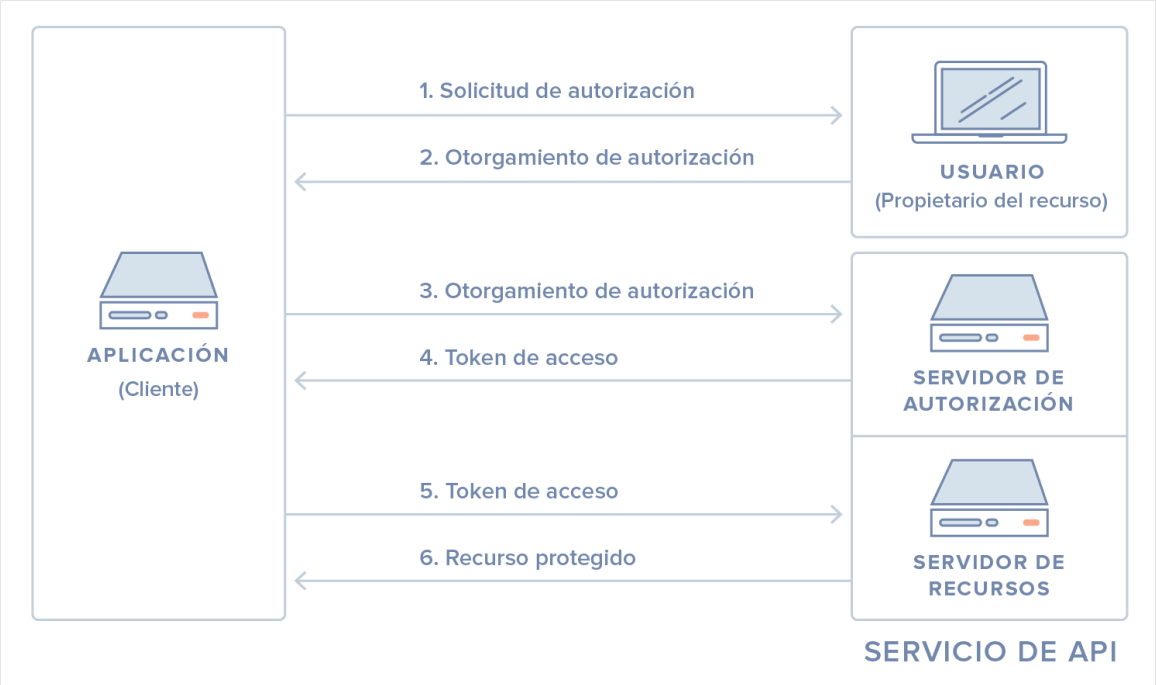

Figura 15: Flujo abstracto [56].

Una "concesión" (o “grant” por el término en inglés) de OAuth 2.0 es la autorización dada (u "otorgada") al cliente por el usuario [58]. Ejemplos de concesiones (“grants”) son "código de

autorización" y "credenciales de cliente". Cada concesión de OAuth tiene un flujo correspondiente.

La especificación del marco de autorización OAuth 2.0 define cuatro flujos para obtener un token de acceso. Estos flujos se denominan tipos de concesión. El flujo a utilizar depende principalmente del tipo de aplicación que se está desarrollando [59].

- Código de autorización: Utilizado por las aplicaciones web que se ejecutan en un servidor. Esto también es utilizado por las aplicaciones móviles, utilizando la técnica de clave de prueba (Proof Key por su término en inglés) para el intercambio de código (PKCE por sus siglas en inglés de Proof Key for Code Exchange).
- Implícito: Utilizado por aplicaciones centradas en JavaScript (aplicaciones de una sola página) que se ejecutan en el navegador del usuario.
- Credenciales de contraseña del propietario del recurso: Utilizadas por aplicaciones de confianza.
- Credenciales del cliente: Se utiliza para la comunicación máquina a máquina.

Puntos finales (endpoints por el término en inglés) utilizados en el protocolo [59]:

Punto final de autorización (Authorization endpoint, por el término en inglés): Se utiliza para interactuar con el propietario del recurso y obtener la autorización para acceder al recurso protegido. Utiliza los siguientes parámetros de solicitud:

- response_type: Le dice al servidor que tipo de concesión utilizar, y la respuesta es basada en este parámetro. Los valores pueden ser “code” o “token”.
- client_id: El ID de la aplicación que solicita autorización.
- redirect_uri: URL a la que se redirige en una respuesta exitosa.
- scope: Una lista de permisos, delimitada por espacios, que la aplicación requiere.
- state: Utilizado con fines de seguridad.

Este punto final (endpoint) es utilizado por la concesión “código de autorización”, que emite un AuthorizationCode. De igual forma es utilizado por la concesión “Implícita”, que emite un token de acceso (AccessToken, por el término en inglés).

La diferencia en los tipos de respuesta es que un AuthorizationCode es una cadena opaca (opaque strings: cuando no se usa una API personalizada), que debe intercambiarse con un token de acceso en el punto final del token. Por su parte, un AccessToken es una cadena opaca (o un JWT en la implementación Auth0, es decir utiliza una API personalizada) que denota quién ha autorizado qué permisos (ámbitos) a qué aplicación.

Punto final de token (Token endpoint, por el término en inglés): Obtiene un token de acceso o un token de actualización. Es utilizado por todos los flujos, excepto por el flujo implícito.

En el flujo de código de autorización, la aplicación intercambia el código de autorización que obtuvo del punto final de autorización para un token de acceso.

En el flujo de credenciales del cliente y credenciales de contraseña del propietario del recurso, la aplicación se autentica utilizando un conjunto de credenciales y luego obtiene un token de acceso.

Antes de utilizar OAuth, se debe registrar la o las aplicaciones con el servicio [56]. Esto se hace a través de un formulario de registro en la parte del "desarrollador" o "API" del sitio web del servicio, en el cual proporcionarás la siguiente información y otra según sea el caso:

- Nombre de la aplicación
- Sitio web de la aplicación
- Redirect URI o Callback URL

Redirect URI es donde el servicio reorienta al usuario después de que se autorice o deniegue su solicitud. Es la parte de la aplicación que manejará códigos de autorización o tokens de acceso obtenidos en las respuestas de OAuth 2.0.

La seguridad en este protocolo se logra después de que se registra la aplicación, y es que el servicio emite las "credenciales del cliente" (client_id) en forma de un identificador de cliente y un secreto de cliente. El identificador de cliente es una cadena pública que utiliza la API de servicio para identificar la aplicación y para generar las URL de autorización que se presentan a los usuarios. Una vez la aplicación solicita el acceso a la cuenta de un usuario, el secreto de cliente (client_secret) se utiliza para autenticar la identidad de la aplicación al API de servicio. Es importante recordar que se debe mantener la confidencialidad entre las aplicaciones y la API.

El client_id es un identificador público para todas las aplicaciones. Por su parte el client_secret es un secreto conocido solo por la aplicación y el servidor de autorizaciones. Debe ser lo suficientemente aleatorio como para que no se pueda adivinar, lo que implica evitar el uso de bibliotecas UUID (*universally unique identifier*, término en inglés) comunes que a menudo tienen en cuenta la marca de tiempo o la dirección MAC del servidor que lo genera.

OPEN ID

Es el sistema de autenticación digital descentralizado, donde se puede identificar en una página web por medio de la URL, la cual además es verificada por medio de un servidor llamado un proveedor de identidad.

Este mecanismo de autenticaciones se basa en identificares conocidos como URIs o XRIs se asigna un identificador el cual se garantiza que el usuario lo controla, el usuario decide cuál será su proveedor de identidad.

Alguno de los componentes en OPEN ID témenos:

Core: Open ID implementa el protocolo Oauth para realizar una comunicación directa al usuario y el proveedor Open ID la cual se realiza por medio de los Token ID que opera en un JSON web Token (JWT) el cual consiste en manejar las notificaciones sobre la autenticación de los usuarios.

Para comprender como se establece los JSON Web Token en el Core de Open ID, estableceremos que se utiliza la propiedad de autorización ya una vez que el usuario ha iniciado sesión, las solicitudes subsiguientes incluirán el JWT, estableciendo al usuario acceso a las rutas, servicios y los recursos que se permiten con el token. En inicio de sesiones unificadas es una característica se utiliza ampliamente con JWT en la actualidad esto sucede por sus mínimas cargas y su capacidad para ser utilizadas como facilitador en los dominios [60].

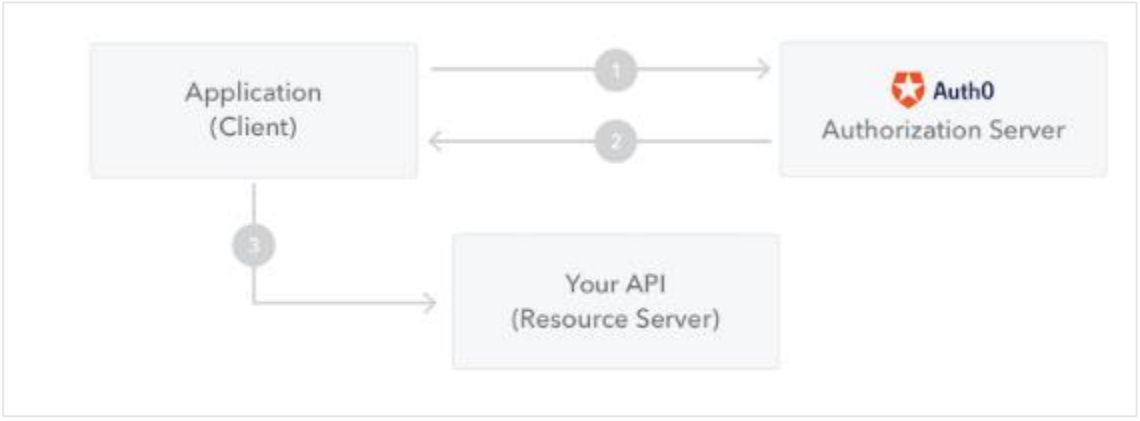

Figura 16: Estructura de JSON WEB TOKEN [60].

1. Se solicita del lado del cliente o aplicación la autorización al servidor de autenticación. Para que se logre generar se realiza por medio de uno de los flujos de autorización. En este caso con una aplicación compatible en OpenID.
2. Al realizar la asignación de la autorización el servidor devuelve un token para tener acceso a las aplicaciones.
3. Se utiliza la aplicación el token que se asignó para el acceso al recurso que está protegido por una API.

Discovery: OpenID Connect Discovery es capaz de realizar la identificación del proveedor OpenID para el usuario, el relying party en este caso no necesita tener el conocimiento previo del proveedor OPENID.

Registro de clientes Dinámicos: El relying party deberá ser registrado con un proveedor OpenID, cuando el cliente es externo o usuario final pueda hacer uso de los servicios o recursos de OpenID se administra el proceso de los relying parties externos, registrando los clientes de manera dinámica con el proveedor OpenID [61].

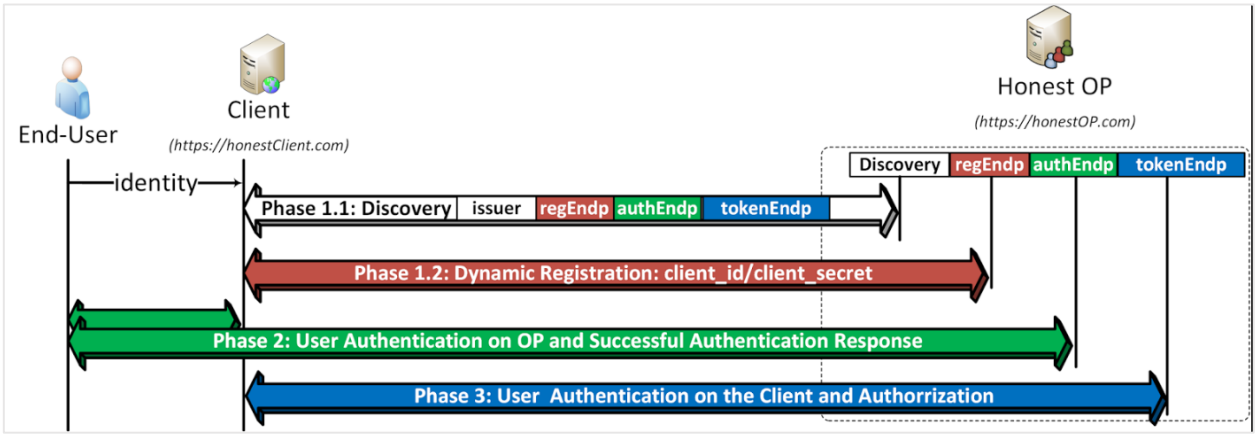

Figura 17: Registros dinámicos en OpenID [77].

Inicios de sesiones únicas: cuando el usuario final visita al cliente e inicia la autenticación de su sesión única para la autenticación en el cliente para poder obtener acceso a sus recursos. Para ello se ingresa la identidad del usuario.

Administración de Sesión: Se utiliza en la administración de las sesiones de los usuarios finales en OpenID. Además, monitorea y actualiza los estados en los inicios de sesiones del usuario, se establece que el relying party puede cerrar la sesión de un usuario si es registrado desde el endpoint del proveedor OpenID.

OpenID Endponits OP

- Endpoit de Registro: Registra los clientes con el OP, para que pueda utilizar los servicios en la autenticación.
- Endpoint de autorización: Los usuarios deben de autenticarse al OP. realizando un proceso de autenticación que corresponda y autorizar al cliente para que pueda acceder a los recursos solicitados.
- Endpoint de token: La comunicación para el cliente se realiza por medio de tokenEndp, Se asigna el token ID y autentica los usuarios, la comunicación que se establece directamente entre el cliente y OP [61].
- Endpoint de información de usuario (userinfoEndp): Envía la información sobre el usuario autenticado como correo electrónico, direcciones, números telefónicos, géneros entre otros, cuando se realiza el acceso a los recursos que el cliente utilizara como su token para el acceso que se ha obtenido en la autenticación de OpenID.

Seguridad OpenID

La seguridad en el protocolo de comunicación es esencial ya que es el canal de transacción de la información que a nivel de soluciones tecnológicas es el insumo vital. Para ello se muestran algunos de los ataques a los cuales se está expuesto y como poder mitigar el riesgo que sucedan dichos ataques:

Robo de cookies y tokens: Las sesiones locales almacenan los tokens o en la cookie del cliente, estas cookies son creadas y son interceptadas en la red ya sea tomadas por scrips XSS maliciosos los cuales son ingresados en alguna página bajo el dominio de los servicios [62]. El modo que se utiliza es conocido como "tokens del portador" para comunicarse con sus actores estos tokens son de procedencia no confiable (se describen en el Anexo 6).

Suplantar los tokens: Cuando se realiza un enlace contextual se está realizando la suplantación de los tokens, el servicio es incapaz de proporcionar los parámetros para los estados durante la ejecución de las solicitudes de autorización para poder asegurar las peticiones con las respuestas [62]. Este parámetro normalmente es el valor que enlaza a la sesión del navegador (se describen en el Anexo 7).

Redirección abierta: Estos ataques son los que toman los parámetros de las URL y se redirigen al usuario un valor que se asocia del parámetro sin las validaciones [62], esto lo hace vulnerable y se utilizan normalmente en los ataques de phishing para que los usuarios realicen visitas a sitios maliciosos sin darse cuenta de ello (se describen en el Anexo 8).

Suplantación de OP: Cuando se realiza la suplantación OP malicioso que crea un token para la autenticación que contiene los identificadores que el OP no puede controlar [62]. De este modo se accede a la cuenta del usuario desde el relying party (Confiar en la sesión) (se describen en el Anexo 9).

Confusión de llave: Las vulnerabilidades en la implementación de gestión de llaves del relying party, dando por resultado el uso de una clave de confianza. Actúa como un OP malicioso y utiliza la fase de asociación para establecer un secreto compartido con el relying party de destino [62]. Posteriormente, confunde al relying party de manera que pueda utilizar la clave compartida de otro OP valido cuando en realidad, es el perteneciente del OP malicioso (se describen en el Anexo 10).

Ataque de endpoint malicioso: los ataques de por medio de inyección de endpoints malicioso se establecen en tres fases esto para poder romper la seguridad del protocolo de comunicación [62] (se describen en el Anexo 11).

2.5.2 Servicios Web SOAP/REST

La W3C define los Servicios Web como sistemas de software diseñados para soportar una interacción interoperable máquina a máquina sobre una red [63]. Los Servicios Web suelen ser APIs Web que pueden ser accedidas dentro de una red (principalmente internet), utilizan un

protocolo (generalmente HTTP), un formato en los mensajes de comunicación (XML o JSON) y son ejecutados en el sistema que los aloja.

SOAP

Simple Object Access Protocol (SOAP) es un protocolo estándar que es definido por dos objetos en diferentes procesos para la comunicación por el intercambio de datos XML. Este protocolo es utilizado para los servicios web, se ejecuta en los sistemas operativos para poder realizar la comunicación con las diferentes aplicaciones por medio del uso de la trasferencia de hipertexto y los lenguajes de enmarcado extensible como XML como base en el intercambio de información [64] [65].

El protocolo simple de acceso a datos SOAP es el núcleo de servicios Web, el cual proporciona un mecanismo estándar de empaquetado de mensajes, además facilita la comunicación al estilo RPC entre el cliente y servidor de manera remota [66].

Ventajas [66]:

- No se asocia a ningún lenguaje: se puede desarrollar la aplicación en cualquier lenguaje de programación que se desee utilizar ya que SOAP no especifica una API, de este modo la implementación se trabaja en el lenguaje de programación seleccionado.
- No se asocia a ningún protocolo de transporte: SOAP no específica cómo se deberán asociar los mensajes con HTTP. Los mensajes no son más que documentos XML esto hace que pueda ser transportado por cualquier protocolo que pueda transmitir texto, no específica ninguna infraestructura de objetos distribuidos, el mayor número de sistemas de objetos distribuidos se logran extender algunos de ellos ya admiten SOAP.
- Aprovecha estándares existentes: SOAP aprovecha XML en las codificaciones de los mensajes de esta manera se especifican los esquemas de XML, además que no se debe definir el medio de transporte de los mensajes este puede realizarlo por medio de los existentes como HTTP y SMTP
- Interoperabilidad entre múltiples entornos: Todas las aplicaciones desarrolladas en el estándar pueden comunicarse por medio de los mensajes SOAP con las aplicaciones que estén bajo otras plataformas, Por ejemplo, una aplicación de escritorio que se ejecute en una PC puede comunicarse con una aplicación del back–end ejecutándose en un mainframe capaz de enviar y recibir XML sobre HTTP.

Desventajas [66]:

- Por el manejo del formato XML, se puede considerar SOAP como una tecnología que se implementa en los middleware.

- Al utilizar HTTP como el protocolo de transporte los roles de las partes que interactúan son estáticos, solo el cliente podrá los servicios de la otra.

Mensajería en SOAP

Proporciona un mecanismo estándar para el empaquetado de mensajes este se compone de un sobre que contiene el mensaje e información de cabecera que se utiliza para describir el mensaje [64]

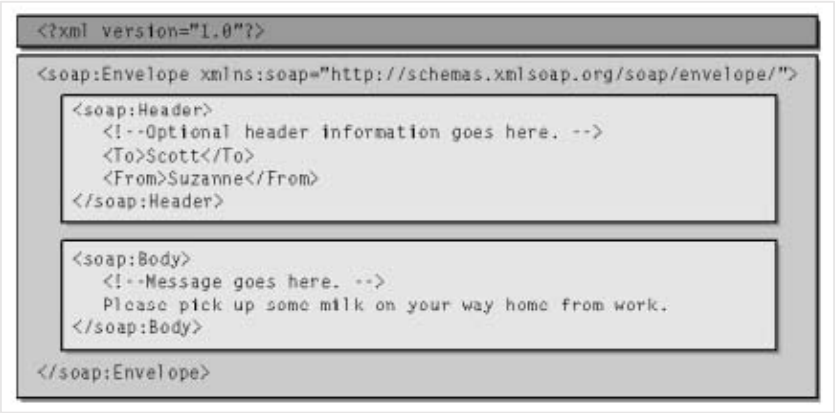

Figura 18: Estructura de mensaje SOAP [64].

El mensaje debe estar dentro del sobre de SOAP que ha sido construido, este se compone de un elemento Envelope, Header y Body la cabecera es el elemento hijo del sobre, luego el cuerpo seguido por la cabecera en el cuerpo se especifican la carga de datos del mensaje y en la cabecera se especifican los datos adicionales que no sean pertenecientes al cuerpo del mensaje [65] (se describen en el Anexo 12).

REST

Representational State Transfer (REST), es un estilo de arquitectura de software para los sistemas distribuidos web. El término fue introducido en la tesis doctoral de Roy Fielding en el año 2000, quien es uno de los principales autores de la especificación HTTP.

La arquitectura se refiere a una colección de principios, que definen como los recursos son definidos y diseccionados. REST describe la interfaz de transmisión de datos específicos de un domino sobre HTTP sin una capa adicional, como hace SOAP.

REST no se considera un estándar y como se ha mencionado es un estilo de arquitectura. Sin embargo, está basado en los siguientes estándares:

- HTTP

- URL
- Representación de recursos: XML/HTML/JSON
- MIME types: Tipos de contenido que la llamada REST retorna (*Content-Type*, por el término en inglés). Estos pueden ser text/xml, text/json, text/html, entre otros.

La motivación de esta arquitectura reside en intentar usar todas las características de la web para el intercambio de mensajes entre computadoras distribuidas. REST utiliza identificadores de recursos uniformes (URI) unificados para identificar los recursos.

Principios de la arquitectura REST, de acuerdo a Rafael Navarro Marset [63]

- Escalabilidad de la interacción con los componentes: La arquitectura es basada en la web, millones de usuarios y equipos se conectan a ella sin comprometer el rendimiento de esta.
- Puesta en funcionamiento independiente: Servidores antiguos deben ser capaces de entenderse con clientes actuales (y viceversa). HTTP permite la extensibilidad mediante el uso de las cabeceras, a través de las URIs, a través de la habilidad para crear nuevos métodos y tipos de contenido.
- Generalidad de interfaces: Gracias al protocolo HTTP, cualquier cliente puede interactuar con cualquier servidor HTTP sin ninguna configuración especial.
- Compatibilidad con componentes intermedios: Los intermediaros pueden ser tipos de proxys para Web. Algunos de ellos, las caches, se utilizan para mejorar el rendimiento. Otros permiten reforzar las políticas de seguridad: firewalls. Por último, otro tipo importante de intermediarios, gateway, permiten encapsular sistemas no propiamente Web. Por tanto, la compatibilidad con intermediarios nos permite reducir la latencia de interacción, reforzar la seguridad y encapsular otros sistemas.

Los elementos que participan en REST son

- URI: Una cadena de caracteres que identifica los recursos en una red de forma unívoca. Se dice entonces que los "recursos" son identificados por medio de las URI que los desarrolladores del servicio (API) habilitan y documentan.
- Verbos HTTP: Se debe especificar el verbo que el "recurso" utiliza. Esto es importante ya que el desarrollo de la API supone ejecutarse bajo un verbo en específico definido por los desarrolladores de la misma. De igual forma el verbo que cada recurso utiliza debe estar documentado. Generalmente se hace una analogía contra las operaciones asociadas a las bases de datos, ejemplo:

Acción	HTTP	SQL
Crear	PUT	Insert
Consultar	GET	Select
Actualizar	POST	Update
Eliminar	DELETE	Delete

Tabla 3: Analogía verbos HTTP

Cada mensaje contiene toda la información necesaria para comprender y completar la petición, lo que significa que la comunicación entre las partes no guarda estado alguno. Definición de elementos en llamada REST:

- Códigos de estado: Al completar la llamada al recurso solicitado, el servidor regresa un código de estado de la respuesta. Estos códigos sirven para el desarrollador de la aplicación que consume el recurso, conozca si la solicitud realizada fue exitosa, ha fallado o bien el recurso no se encontró (entre otros estados).
- Cabeceras HTTP: Son los parámetros que se envían en una petición o respuesta HTTP, al cliente o al servidor para proporcionar información esencial sobre la transacción en curso. El formato de las cabeceras es 'Cabecera: Valor' y son incluidas en la llamada al recurso.
- Contenido guiado por MIME: Se refiere a los encabezados "MIME media type" o Content-Type. Este valor corresponde al formato del contenido o del archivo que se transmite por la web. Se compone de un tipo y un subtipo: tipo/subtipo.

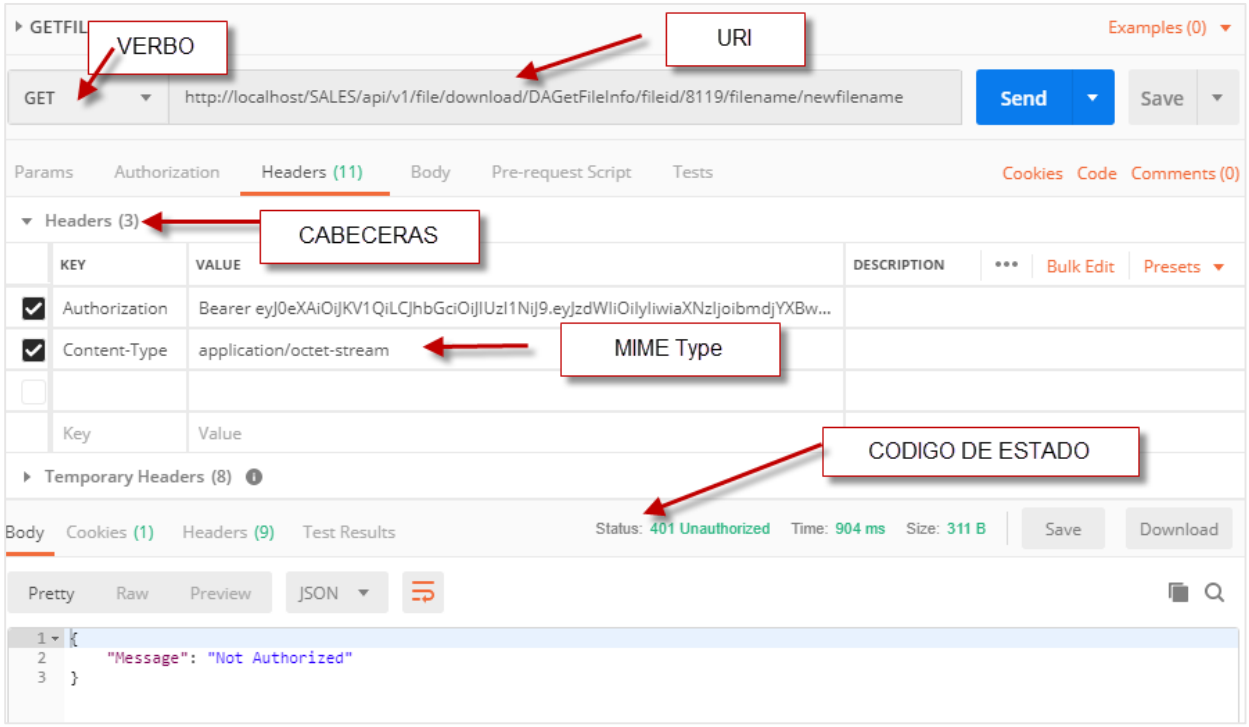

Figura 19: Ilustración de elementos sobe llamada REST utilizando el software Postman. Elaboración propia

De igual forma es posible implementar REST con el estándar XML. Existe bastante discusión de cuál servicio es mejor entre SOAP y REST, y la decisión de implementar uno u otro depende de varios factores como lo pueden ser: el tipo de entorno (aislado, conocidos), el

número de usuarios que consumen el recurso, y otras consideraciones que están fuera del alcance de la presente investigación.

2.6 Seguridad en la comunicación de aplicaciones

2.6.1 Certificados de Seguridad

Un *certificado de clave pública* es un punto de unión entre la clave pública de una entidad y uno o más atributos referidos a su identidad [67]. El certificado garantiza que la clave pública pertenece a la entidad identificada y que la entidad posee la correspondiente clave privada. La entidad identificada se denomina *sujeto del certificado* o *subscriptor.*

Para que un certificado digital sea válido, este debe ser emitido por alguna autoridad certificadora, ya que si uno se certifica a sí mismo no hay ninguna garantía de que su identidad sea la que anuncia, y, por lo tanto, no debe ser aceptada por un tercero que no lo conozca. Un certificado es una estructura de datos firmada que enlaza una clave pública a una entidad.

Se debe verificar que una autoridad certificadora ha emitido un certificado y así detectar si un certificado no es válido. Para evitar la falsificación de certificados, la entidad certificadora después de autentificar la identidad de un sujeto firma el certificado digitalmente.

El formato de certificados X.509 es un estándar del ITU-T (*International Telecommunication Union-Telecommunication Standarization Sector*) y el ISO/IEC (*International Standards Organization / International Electrotechnical Commission*) que se publicó por primera vez en 1988. El formato de la versión 1 fue extendido en 1993 e incluye dos nuevos campos que soportan el control de acceso a directorios. Después de emplear el X.509 v2 para intentar desarrollar un estándar de correo electrónico seguro, el formato fue revisado para permitir la extensión con campos adicionales, dando lugar al X.509 v3, publicado en 1996 [67].

La criptografía de la clave pública se basa en un par de claves pública y privada para cifrar y descifrar el contenido [68]. Las claves están relacionadas matemáticamente, y el contenido cifrado mediante el uso de una de las claves solo se puede descifrar utilizando el otro. La clave privada se mantiene en secreto. La clave pública suele estar incorporada en un certificado binario, y el certificado se publica en una base de datos a la que pueden acceder todos los usuarios autorizados.

El certificado X.509 es generalmente utilizado como medida de seguridad en la comunicación de un SSO. Todas las soluciones comerciales analizadas en esta investigación, al igual que los protocolos de comunicación, soportan este certificado y a la vez existe una amplia documentación para la integración con estos productos.

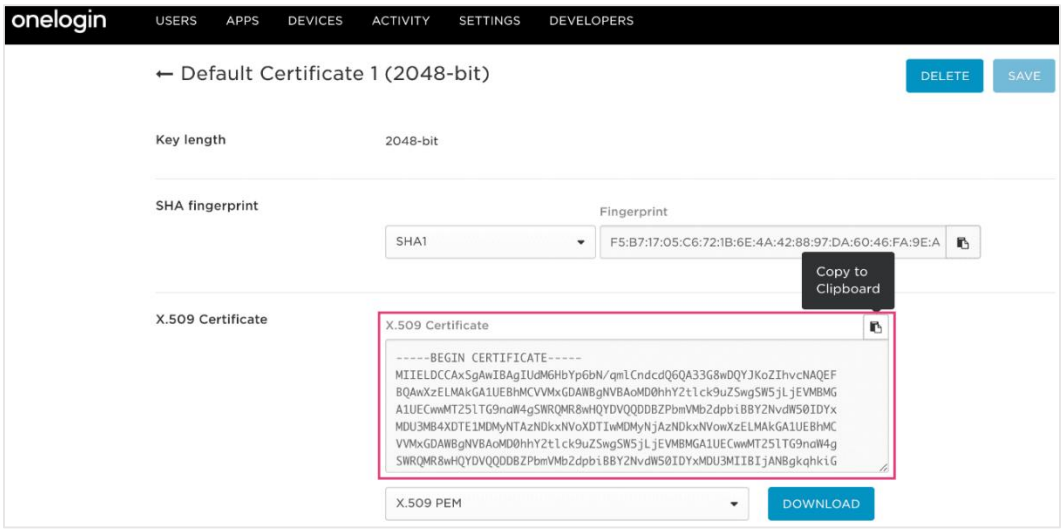

Figura 20: Ejemplo de configuración de certificado en OneLogin [69].

2.6.2 Tokens

Los Token son una cadena de texto que contiene un significado, una validación o un identificador según el caso en el que sea utilizado.

Normalmente los tokens se generan del lado de un servidor de autenticaciones para la obtención de un Identificador (ID Token). El token toma un valor para ser utilizado como llaves de acceso [70].

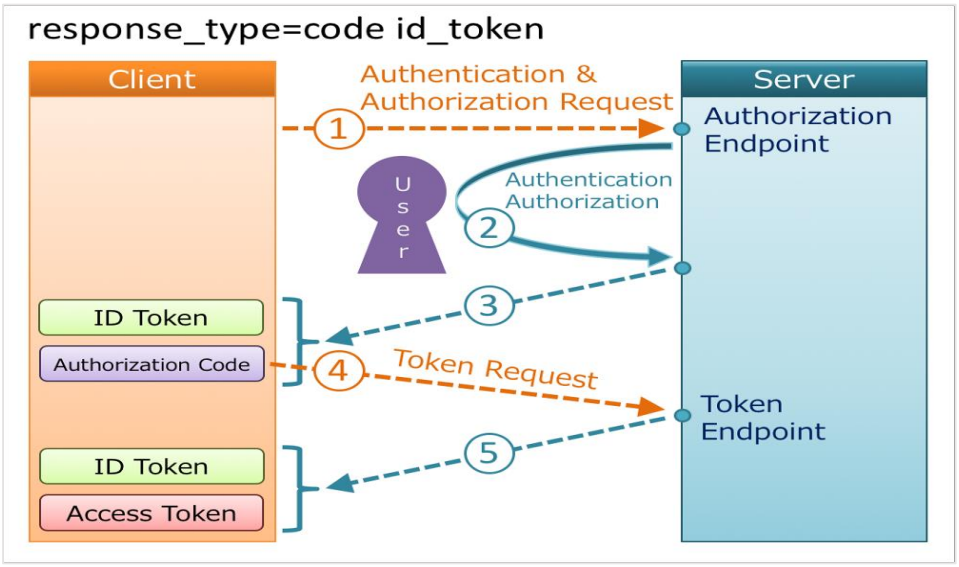

Figura 21: Tipos de respuesta en la autenticación con Token [70].

El JSON Web Token es el más utilizado en los protocolos de autenticación ya que se puede definir como el token más seguro para validar una autenticación como la autorización de un inicio de sesión o ejecución de un proceso [60].

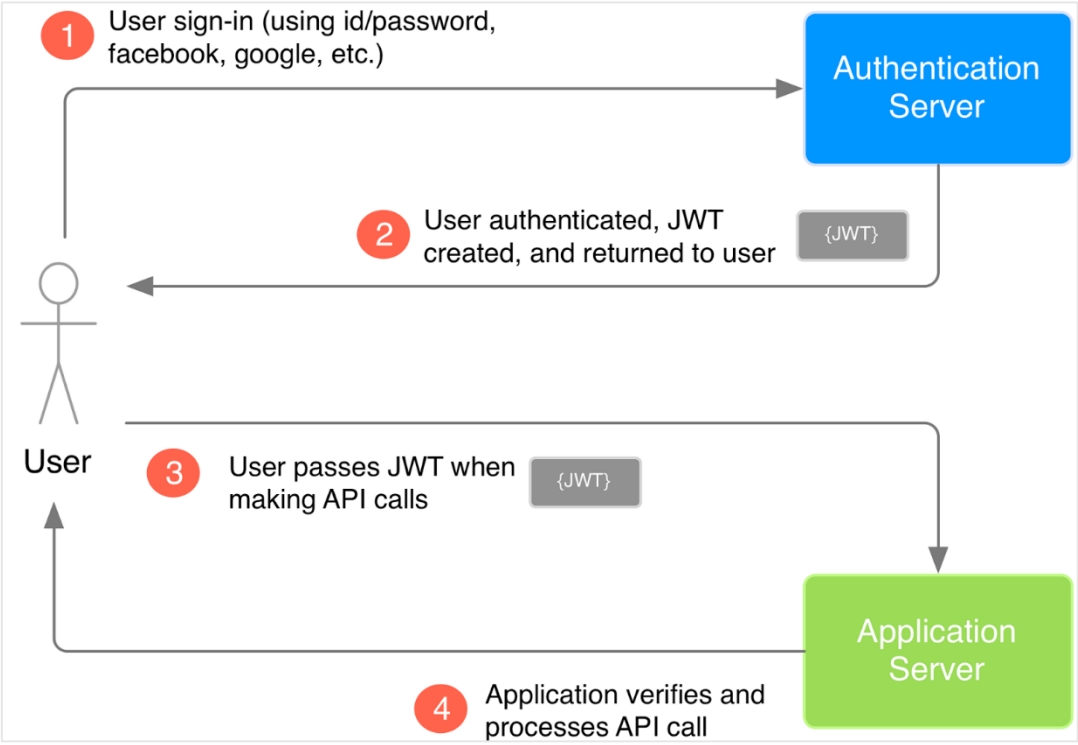

Figura 22: Estructura de la autenticación por medio de JSON WEB TOKEN [60].

2.7 SSO como un Middleware

El middleware es un software de conectividad diseñado para ayudar a administrar la complejidad y heterogeneidad propia de los sistemas distribuidos. El middleware construye una especie de puente entre los diferentes sistemas al habilitar comunicaciones y transferencia de datos.

Es capaz de administrar aplicaciones dispares tanto dentro de una sola organización o entre organizaciones independientes [71]. Una solución de tipo SSO cumple con esta definición dado que, dependiendo de la configuración, cuenta con entornos y componentes federados que permiten autenticar usuarios de una compañía en otra sin problemas, o incluso entre aplicaciones que son de dominios diferentes pero que se integran con SSO federados.

Para entornos empresariales un SSO funciona como una capa de software de servicios, que al igual a un middleware permite a las aplicaciones interoperar entre los componentes de la organización, a pesar de las posibles diferencias tecnológicas que existen entre los componentes.

El middleware provee un conjunto de APIs [71] que se exponen a las aplicaciones para que estas las consuman. En un SSO, la API principal es el servicio para la autenticación de usuarios. Inherente a este proceso, provee funciones para: ubicar aplicaciones, seguridad en la comunicación (por ejemplo, la autenticación multifactor), gestión de usuarios y políticas, entre otras que se han mencionado en esta investigación.

Al igual que un middleware, la comunicación en el SSO es mediante el envío de paquetes de datos que se denominan "mensajes" [71]. En un SSO, la comunicación se habilita con el uso de estándares que se denominan "protocolos de comunicación". La presente investigación describe algunos de estos estándares que, como mínimo, cumplen las siguientes características [71]:

- Integridad en el envío del mensaje.
- Uso de encabezados en el mensaje.
- Uso de propiedades, atributos o parámetros.
- Manejo de estructuras de datos, como bien lo son JSON, XML.

El middleware habilita la comunicación maquina a máquina, al igual que un SSO se habilita comunicación entre las API y los clientes externos no interactivo, pero depende del alcance de la solución SSO habilitar esta funcionalidad.

El middleware permite dos modos básicos de comunicación: síncronos y asíncronos [71]. Un SSO de igual forma soporta esos dos modos, pero depende de cada componente y tecnología, los recursos que están disponibles de un modo u otro.

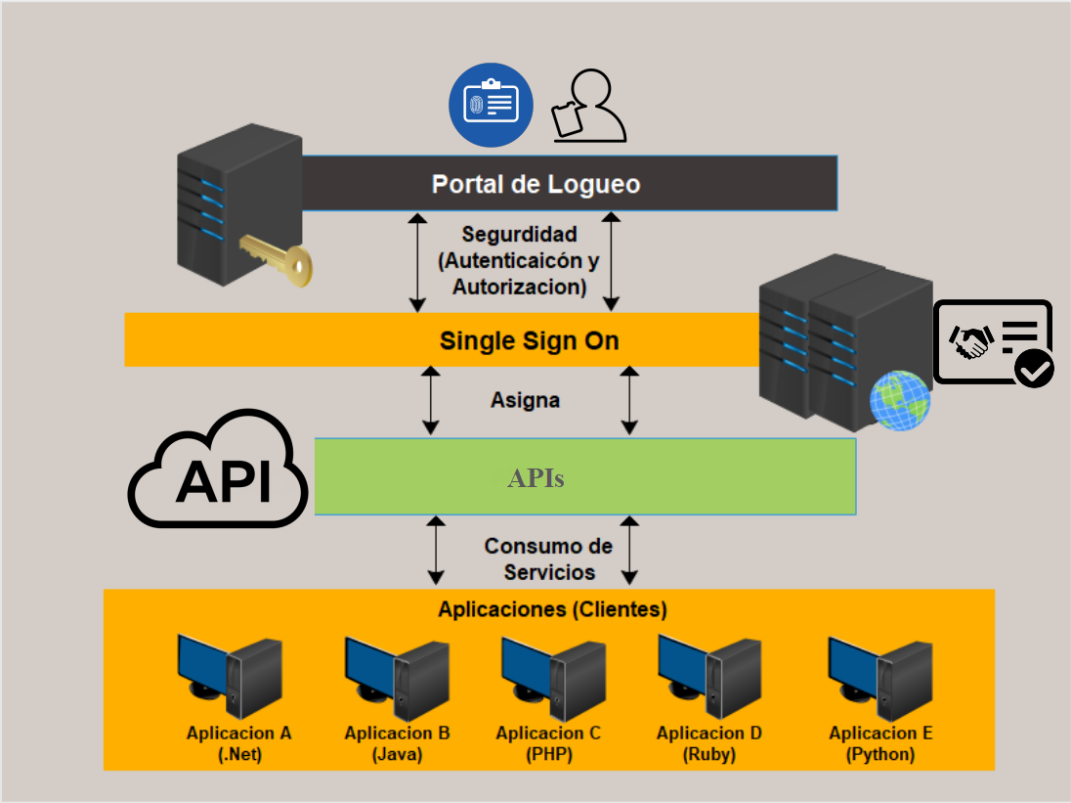

Figura 23: SSO como middleware. Elaboración propia

Capítulo III – Metodología de investigación

3.1 Tipo de investigación

Según el enfoque es *cuantitativa*, como la define Sampieri dado que: plantea el problema, hay una revisión de la literatura, construcción de marco teórico, recolección de datos y de igual forma es objetiva [72].

Para el enfoque cuantitativo, el mismo Sampieri clasifica la investigación según su alcance. En este sentido, la presente investigación es de alcance *descriptiva*. "miden, evalúan o recolectan datos sobre diversos conceptos (variables), aspectos, dimensiones o componentes del fenómeno a investigar. En un estudio descriptivo se selecciona una serie de cuestiones y se mide o recolecta información sobre cada una de ellas, para así (valga la redundancia) describir lo que se investiga" [73].

Una tercera clasificación de la investigación es por el tipo, siendo la presente una investigación *proyectiva* y es que se proporcionan las características de un diseño arquitectónico. Los autores Martha Nelly Córdoba y Carolina Monsalve, la definen: "Consiste en encontrar la solución a los problemas prácticos, se ocupa de cómo deberían ser las cosas para alcanzar los fines y funcionar adecuadamente. Consiste en la elaboración de una propuesta o de un modelo, para solucionar problemas o necesidades de tipo práctico, ya sea de un grupo social, institución, un área en particular del conocimiento, partiendo de un diagnóstico preciso de las necesidades del momento, los procesos explicativos o generadores involucrados y las tendencias futuras" [74].

3.2 Unidades de análisis

Objetivo específico	Unidad de análisis
Formular las características tecnológicas que implementan las soluciones de SSO.	Soluciones SSO actuales Soluciones por suscripción: Okta, Auth0, OneLogin
Detallar la construcción de un marco de trabajo genérico (framework) para SSO.	Tecnologías para implementación Servidores de autenticación Interfaces Middleware Protocolos de comunicación

	Servicios web
Analizar la implementación de una solución SSO considerando el costo de adquirir un servicio o herramienta de un proveedor y el costo de realizar un desarrollo interno, tomando de base la arquitectura sugerida en la presente investigación.	Portafolio de soluciones web. Costos de adquisición. Costos de desarrollo. Costos en infraestructura.

Tabla 4: Unidades de análisis por objetivo específico

3.3 Variables

Unidad de análisis	Variables
Soluciones SSO actuales	Robustez Escalabilidad Compatibilidad
Soluciones por suscripción: Okta, Auth0, OneLogin	Adaptable Integra con más de un servidor web Servidores de autenticación Interfaces
Tecnologías para implementación. Servidores de autenticación Interfaces Middleware Protocolos de comunicación Servicios web	Manejo de mensajes Rapidez de procesamiento Manejo de cargas Arquitectura Autenticación
Portafolio de soluciones web Costos de adquisición. Costos de desarrollo.	Gestiona los inicios de sesión Políticas de seguridad Autenticación multifactor Mantenimiento. Tipos de suscripciones. Recursos humanos: cambios, tiempo.

Costos en infraestructura.	Equipo para desarrollo. Mitigación de riesgos. Infraestructura.

Tabla 5: Variables por unidades de análisis

3.4 Procedimiento de investigación y desarrollo de instrumentos

1. Para las soluciones SSO actuales, y las diferentes soluciones por suscripción como (Okta, Auth0, OneLogin). Se establece un número finito de variables con las cuales se pretende realizar la recopilación de información concreta sobre las Soluciones SSO y sus componentes. De esta manera poder conocer más sobre los componentes y atributos que lo conforman, así poder definir cada uno de ellos en el entorno de las soluciones, cuál es su auténtica finalidad en las soluciones de un único inicio de sesión, para poder comprender la estructura que se debe buscar al momento de desarrollar un framework de SSO.

Para recopilar información se utilizará un instrumento: *lista de chequeo para la investigación de soluciones actuales de single sign on y soluciones por suscripción* (se describen en el Anexo 13).

2. Para la unidad de análisis de Tecnologías para implementación, Servidores de autenticación, Interfaces y Middleware, se desarrolla la evaluación por medio de un número de variables que englobaran la información necesaria para conocer el desempeño de cada componente en la implementación de las soluciones SSO como manejo de los diferentes servidores de autenticación sus interfaces y los middlewares.

Estas variables se enumeran junto con sus características en el cuadro resumen *tecnologías para la implementación de la solución single sign on, servidores de autenticación, interfaces, middleware, protocolos de comunicación y servicios web* (se describen en el Anexo 14).

3. Para la unidad de análisis de Portafolio de soluciones, las aplicaciones web y como se integra la comunicación por medio del protocolo SAML, como los diferentes servicios que soporta como SOAP, REST.

Estas variables se miden con el formulario *lista de chequeo para la investigación de portafolio de soluciones web, costos de adquisición, desarrollo e infraestructura* (se describen en el Anexo 15).

3.4.1 Procedimiento

Variables	Procedimiento	Análisis
Robustez. Escalabilidad. Compatibilidad. Adaptable. Integra con más de un servidor web. Servidores de autenticación. Interfaces. Manejo de mensajes. Rapidez de procesamiento. Manejo de cargas. Arquitectura. Autenticación SAML. Maneja autenticación por servidores POP3. Gestiona los inicios de sesión. Políticas de seguridad. Autenticación multifactor. Mantenimiento. Tipos de suscripciones. Recursos humanos: cambios, tiempo. Equipo para desarrollo. Mitigación de riesgos. Infraestructura.	Observación de las características. Análisis del contenido de las tecnologías para detallar si cumple o no cumple. Identificar en qué consisten los diferentes componentes de las soluciones SSO. Conocer sobre las aplicaciones web y su integración en las soluciones SSO. Identificar el manejo de mensajes en los protocolos de comunicación. Identificar el Procesamiento de datos en los SSO. Comprender la arquitectura de los componentes de un SSO. Manejar las diferentes formas de autenticación en las soluciones SSO. Idealizar la mejor opción en las Soluciones SSO como Servidor Web. Realizar juicio de valor sobre los servicios web SOAP y REST. Desarrollar un estudio excautivo para la administración de accesos de manera centralizada.	Análisis de componentes necesarios para que un SSO sea robusto y escalable. Plano de los componentes de las soluciones SSO Para lograr construir un marco de trabajo para un SSO empresarial. Análisis de protocolos de comunicación, seguridad y datos para la integración del SSO y las aplicaciones web. Análisis para identificar los diferentes servicios necesarios para las soluciones SSO.

Tabla 6: Variables por unidades de análisis

Capítulo IV – Análisis y discusión de resultados

4.1 Dominio del problema

La presente investigación analiza algunas soluciones comerciales SSO y a la vez se justifica por qué el estudio de esas soluciones, considerando que existen una gran variedad de ellas. Se detalla, en lo posible, las características funcionales y aspectos técnicos para su integración con las aplicaciones web.

Lo siguiente que el marco teórico aborda son los diferentes componentes que utilizan las soluciones SSO, siendo estos básicos y requeridos (de bajo nivel) con funciones específicas que representan a otros componentes (de más alto nivel) en la configuración de las soluciones SSO:

- Estructuras de datos que contienen las identidades de los usuarios (id, claves, atributos, etc.) que sirven como IdP.
- Los servidores web que proveen a los SP para el procesamiento de peticiones.
- Configuraciones que habilitan al SP como un componente escalable, seguro y que garantice la integridad de los datos.

Como último aspecto en el análisis, se detallan las tecnologías necesarias que facilitan la integración de todos los componentes mencionados. Desde los protocolos de comunicación, manejo de sesiones, peticiones por medio de tokens, consumo de servicios web, intercambio de mensajes entre maquinas con estándares como XML y JSON.

Se provee una visión del SSO como un producto que da valor agregado a los entornos donde se implementa, y como este sirve de intermediario a lo que llamamos un middleware. Delegando la funcionalidad básica como los es la autenticación de usuarios en entornos empresariales.

Ante la necesidad de implementar una solución SSO, se plantea la problemática de ¿cómo hacerlo?

Desde una visión en la que se realiza un desarrollo interno de este tipo de producto (SSO), el equipo implementador debe analizar todos los aspectos técnicos y características esperadas que se deben incluir. Dada la gran variedad de opciones. No es una tarea fácil hacer la selección de componentes y menos aún la correcta orquestación de estos. Sin duda, la arquitectura sugerida debe obedecer a la necesidad de cada empresa donde se desea implementar, pero el análisis previo es el mismo, donde se descubren los componentes a utilizar (muy probablemente ya se tengan alguno de esos).

Hay aproximaciones de soluciones SSO, como se ha detallado en el marco teórico, pero basándose en la definición formal de este tipo de productos, se descubre que carecen de algunas características, con lo que se pretende establecer un marco de trabajo genérico que integre los componentes necesarios y den valor agregado en una eventual implementación.

4.1.1 Resolución de la problemática

Las soluciones SSO manejan la autenticación de los usuarios dentro de las aplicaciones de una organización por medio de un único conjunto de credenciales, si bien la mayoría cumple con los requerimientos mínimos de esta funcionalidad, no siempre se logra utilizar al máximo dado que algunos de los componentes que este utiliza no son compatibles o adaptables con las aplicaciones de la empresa, lo que obliga a realizar algunos cambios a estas. Por lo general las soluciones SSO son adquiridas por suscripción, ya sea anual o mensual, y estas a la vez dependen de la cantidad de usuarios adquiridos, el soporte y los módulos adicionales. Esto implica un costo relativamente alto que algunas empresas no son capaces de costear y por consiguiente no lo adquieren, a pesar de tener la necesidad de este tipo de producto.

La presente investigación con un enfoque cuantitativo elaborada en los capítulos anteriores y con el desarrollo de los diferentes instrumentos, genera la información sobre las variables asignadas con las que se identifican los componentes en las soluciones SSO y así definir un marco de trabajo para que las organizaciones lo tomen como referencia y posteriormente realicen el desarrollo interno de una solución SSO.

Se han establecido tres componentes principales:

Autenticación: se encarga de verificar y validar las credenciales del usuario que habilitan el acceso a las aplicaciones. Se detallan los atributos que se debe contemplar y la especificación de las tecnologías a utilizar.

Comunicación: este elemento y sus atributos son los que se encargan de establecer la integración de la solución con las aplicaciones web. La comunicación se puede establecer en diferentes tecnologías para este tipo de soluciones, se definen los protocolos de comunicación y se identifican los servidores y servicios web que la establecen.

Seguridad: el elemento de seguridad en todas las soluciones siempre debe ser un control de mucha eficiencia y calidad por la sensibilidad que este componente implica y mayor aún en las aplicaciones web empresariales. Se identifica el manejo de tokens en las soluciones SSO y el uso de certificados que agregan seguridad y confianza a la comunicación. Todos los componentes involucrados manejan su propia seguridad, y está sujeta a las diferentes tecnologías que implementan.

4.1.2 Propuesta de solución

Se establecen los componentes para desarrollar un modelo de solución SSO en entornos empresariales, el cual despliega un portal web para funciones administrativas y a la vez se exponen servicios web para que las diferentes aplicaciones puedan consumirlos. La implementación de este servidor web administra el flujo de peticiones de autenticación entre las aplicaciones que delegan este proceso al SSO. Se le conoce como proveedor de servicios (SP).

El SP no contiene a los usuarios, para ellos se apoya de un proveedor de identidades (IdP) que gestiona la estructura de datos donde se encuentran realmente las credenciales de los

usuarios y otros atributos que se consideren necesarios o requeridos. Identifica las aplicaciones a las que los usuarios tienen derecho acceder, e importante mencionar que es un repositorio centralizado con lo que la administración de usuarios se vuelve una tarea más ágil, dado que es el único espacio que se debe modificar para guiar el proceso de autenticación.

La comunicación entre los componentes identificados y las aplicaciones web empresariales deben seguir un estándar, para ello se utilizan lo que se conocen como protocolos de comunicación. Estos protocolos implementan configuraciones específicas, donde se establece la información que manejan en la comunicación, elementos de seguridad, y es común para peticiones y respuestas. Sirve como un bus de control de datos, que garantiza la integridad de la información que se procesa, como puede ser validar que las peticiones tengan un origen confiable (mediante un certificado de clave pública y privada) y que solo se requiera una única entrada de credenciales para acceder de forma transparente (mediante uso de tokens) a todas las aplicaciones que se integran a la solución SSO.

4.2 Discusión de resultados

El objetivo es la recopilación de la información exhaustiva sobre las diferentes unidades evaluadas por medio de las variables asignadas y los instrumentos adecuados para la recopilación de la información la cual se ha clasificado para cada unidad un número diferentes de variables.

Resultado del Instrumento *(Lista de chequeo).*

Las primeras unidades evaluadas para la recopilación de la información la comprenden las soluciones actuales SSO y las soluciones por suscripción, desarrollando la evaluación por medio de las siguientes variables asignadas (Robustez, Escalabilidad, Compatibilidad, Adaptable, Integra con más de un servidor web, Servidores de autenticación, Interfaces). Generando un resultado que nos indica que las aplicaciones actuales por suscripción se establecen de buena manera en las empresas, sin embargo, no todas cumplen un cien por ciento de eficiencia en las necesidades de la empresa, en algunos casos sería necesario adquirir más de una aplicación, podemos evaluar las valoraciones según se realizaron los instrumentos en el (se describen en el Anexo 13).

Valoración de cumplimiento en cuatro categorías sobre las variables de las SSO.

Ponderación	Valoración
0% a 25%	Regular
26% a 50%	Bueno
51% a 75%	Muy Bueno
76% a 100%	Excelente

Tabla 7: Valoración de cumplimiento.

Resultados de las variables evaluadas para las soluciones por suscripción, basados en la información recolectada en el marco teórico y documentación externa a la investigación. De

igual forma se consideran opiniones de terceros, como ejemplo reseñas de revistas tecnológicas como *IT Central Station.*

OKTA	
Variable	**Valoración**
Robustez	50%
Escalabilidad	75%
Compatibilidad	75%
Adaptable	90%
Servidores de autenticación	95%
Interfaces	85%

AUTH0	
Variable	**Valoración**
Robustez	85%
Compatibilidad	80%
Adaptable	75%
Integración con más de un servidor web	90%
Servidores de autenticación	95%
Interfaces	85%

ONELOGIN	
Variable	**Valoración**
Compatibilidad	75%
Adaptable	90%
Servidores de autenticación	75%
Interfaces	87%

Resultado del Instrumento *(Cuadro resumen).*

Las unidades evaluadas en este instrumento incluyen aspectos específicos de las soluciones de terceros y los componentes tecnológicos que se integran a ellas.

En las *Tecnología para la implementación*, se describen los SP y aspectos de seguridad en cuatro de las cinco unidades evaluadas, indicando manejo de mensajes, procesamiento, manejo de cargas y arquitectura de procesamiento de peticiones.

Para *Servidores de autenticación,* se describe el manejo de cargas, arquitectura y autenticación de los IdP siendo estos cuatro componentes los que se analizan y vierten a este cuadro resumen.

Las *interfaces* son resumidas para aspectos de integración, uso de APIs y herramientas que faciliten o asisten el trabajo de implementación/adaptación de soluciones. Describe conceptos importantes que definen la interacción entre componentes, las diferentes arquitecturas que implementan y la autenticación.

Se provee una visión del SSO como un *middleware* y las funcionalidades que este debe poseer para servir como pieza tecnológica intermedia en los entornos empresariales.

Protocolos de comunicación para su correcta interpretación y resaltar similitudes y diferencias entre los tres que la investigación analiza. Indica aspectos de mensajes, arquitecturas y autenticación.

Por último, se tiene los servicios WEB que incluyen SOAP y REST como tecnologías que facilitan el consumo de los diferentes proveedores. Se indican manejo de mensajes, manejo de cargas y arquitectura. El cuadro resumen se identifica en el (se describen en el Anexo 14).

Resultado del Instrumento *(Lista de chequeo).*

Las unidades evaluadas en este instrumento son referentes a las soluciones Web de las empresas, como los costos de adquisición de una solución SSO, los costos de desarrollo de una solución SSO y los costos que son requeridos en la infraestructura al momento de adquirir o desarrollar una solución de inicio de sesión único. Generando un resultado que las soluciones web de las organizaciones siempre deberán sufrir algún ajuste para poder adaptar una solución SSO según sea requerido, por lo cual se genera un costo de mantenimiento para adecuarlas, al SSO que se contrate, de este modo analizamos cuales serían los costos de adquisición de una solución y los posibles ajustes que debemos realizar en las aplicaciones, además de realizar el análisis de cuál sería el costo de desarrollo, ya que la infraestructura presentara cambios por la adquisición y desarrollo de la solución, podemos evaluar las valoraciones según se realizaron los instrumentos en el (se describen en el Anexo 15).

Valoración de cumplimiento en cuatro categorías:

- Para aplicaciones web (*muy bajo, bajo, medio, alto)*
- Para costos de adquisición, desarrollo e infraestructura *(regular, bueno, muy bueno, excelente)*

Ponderación	**Valoración**
0% a 25%	Regular / Muy bajo
26% a 50%	Bueno / Bajo
56% a 75%	Muy Bueno / Medio
76% a 100%	Excelente / Alto

Tabla 8: Valoraciones de cumplimiento

Resultados de las variables evaluadas para las aplicaciones Web.

Aplicaciones Web	
Variable	**Valoración**
Gestiona los inicios de sesión	80%
Políticas de seguridad	75%
Autenticación multifactor	50%
Mantenimiento.	25%

Resultados de las variables evaluadas para los costos de adquisición, desarrollo e infraestructura.

Costos de Adquisición	
Variable	**Valoración**
Autenticación multifactor	Bajo
Mantenimiento.	Medio
Tipos de suscripciones.	Medio
Infraestructura	Bajo

Costos de Desarrollo	
Variable	**Valoración**
Políticas de seguridad	Medio
Mantenimiento.	Bajo
Recursos humanos: cambios, tiempo.	Medio
Equipo para desarrollo	Medio
Infraestructura	Bajo

Costos de Infraestructura	
Variable	**Valoración**
Mantenimiento.	Medio
Recursos humanos: cambios, tiempo.	Bajo
Mitigación de riesgos	Medio
Infraestructura	Medio

4.3 Requerimientos

Un verdadero sistema SSO requiere que el usuario provea sus credenciales de inicio de acceso solamente una vez, durante una sesión, con lo que se obtiene acceso a múltiples aplicaciones sin que estas soliciten la autenticación nuevamente.

Por otra parte, debido a lo sofisticado que suponen ser estas soluciones SSO, algunas implementaciones se les denomina SSO simplificadas, en las que el usuario en efecto utiliza el mismo conjunto de credenciales de acceso, pero es requerida por cada aplicación o servicio que se desea utilizar (se debe proveer por cada recurso).

Como se ha estudiado en la presente investigación, las soluciones SSO vienen con una diversidad de variantes, pero se analiza que todos comparten un modelo común. Desde un punto de vista arquitectónico, un sistema SSO siempre involucra cinco entidades lógicas:

1. IdP (proveedor de identidades): Entidad que administra y autentica la información de identidad de los usuarios y proporciona las identidades confirmadas a otros proveedores de servicios.
2. SP (proveedor de servicios): También conocido como *parte que confía - RP* (por sus siglas en ingles de Relying Party). Es una entidad que proporciona servicios web, que a la vez confía en un proveedor de identidad de confianza (IdP) para la toma de decisiones sobre autenticación y autorización.
3. Usuario: Persona que asume una identidad digital particular de un IdP para acceder a los recursos y servicios protegidos gestionados por los SP. Un usuario es un miembro de la fuerza laboral: empleado, ejecutivo, administrador, socio, cliente, etc.
4. Agente Usuario: Es una aplicación de software que se ejecuta en una computadora personal, dispositivo móvil o cualquier dispositivo que interactúa con el IdP y SP en nombre del usuario, suele ser un navegador web o una aplicación móvil.
5. Protocolo: Es un acuerdo de formatos de mensajes y mecanismos de transporte entre los IdP, los RP y los agentes de usuario diseñados para intercambiar identidades afirmadas entre los IdP y los SP. Un protocolo de comunicación en las soluciones SSO puede ser un estándar abierto o una especificación propietaria.

En la implementación de un SSO, este componente se suma al sistema nervioso central de la infraestructura de TI de una organización. Es importante considerar la sensibilidad que implica tener la validación de credenciales centralizada, desde todas las ópticas posibles:

Aspectos negativos, como lo pueden ser:

- Permitir a intrusos acceder a todas las aplicaciones que implementan la solución SSO (no solo a unas).
- La ausencia o poco desarrollo de los requerimientos no funcionales: escalabilidad, seguridad, eficiencia entre otros, que traen consecuencias a corto y largo plazo.
- Bloquear el trabajo de los usuarios, en el escenario donde el SSO simplemente no funciona o está bajo mantenimiento y no se tienen políticas de actualización, ya sea a nivel de infraestructura (con un servidor auxiliar, si fuera necesario) o ventanas de tiempos para ejecutar el despliegue.

Aspectos positivos, como lo pueden ser:

- Permite a los usuarios utilizar un único conjunto de credenciales, facilitándoles la tarea de gestionar su usuario y contraseña. Fuera del contexto de SSO donde en efecto deben recordar un usuario y contraseña por cada aplicación que se desea utilizar.

- Políticas de seguridad implementadas en todas las aplicaciones integradas a la solución. Incluye las políticas de credenciales como lo son: longitud, tiempo de validez, formato, forzar un cambio, etc.
- Liberación de los recursos de TI en el contexto donde los usuarios no recurren a este departamento (o en menor medida) por problemas de autenticación. De igual forma les permite (a TI) dar de alta y baja de manera ágil a los usuarios según sea necesario.
- Mayor seguridad mediante el uso de controles para la autenticación (si aplica): autenticación multifactor, machine learning, autenticación basada en el contexto (por lugar de ubicación, dispositivo, red, etc.)
- Informes con registros de autenticación detallados, intentos fallidos de inicio de sesión, actividad sospechosa.
- Se ve al SSO como un componente de bajo acoplamiento que permita una integración fácil.
- Permite la reutilización de código (o piezas) ya de que las aplicaciones no gestionan la autenticación de usuarios, solo deben delegar este proceso al SSO.

Desde la perspectiva del usuario, el flujo de interacción con las aplicaciones no se debe ver afectado. Es decir, basta con saber las credenciales que debe utilizar y es el SSO quien detrás de escena lo autentica, no de forma local sino según la empresa ha definido (IdP en uso).

Si hay un pequeño cambio, y es que el *agente de usuario* al detectar una sesión valida omite el paso de autenticación sobre el recurso solicitado por el *usuario*. Esto último es la evidencia que el usuario tiene para determinar que en efecto se implementa un SSO.

Como una buena práctica, se implementan *puertas traseras* (*backdoors* por el término en inglés) que se traducen en una configuración tal que permite a súper usuarios (administrador) omitir la autenticación externa (SSO IdP) e iniciar sesión de forma local en la aplicación.

Esto último obedece a escenarios donde puede haber un error en la configuración de la comunicación, o algo cambia en los puntos finales. Tener una puerta trasera disponible para que los administradores la utilicen para acceder a un sistema bloqueado se vuelve extremadamente importante.

Independiente de cómo se inicie el flujo en el SSO, ya sea por el SP o el IdP, las entidades, los aspectos positivos y negativos y todo lo mencionado se mantiene. La presente investigación incluye evidencia del flujo iniciado por el SP, de igual forma las figuras ilustradas siguen ese esquema. Es una delimitación en la presente investigación.

5.1 Diseño de arquitectura

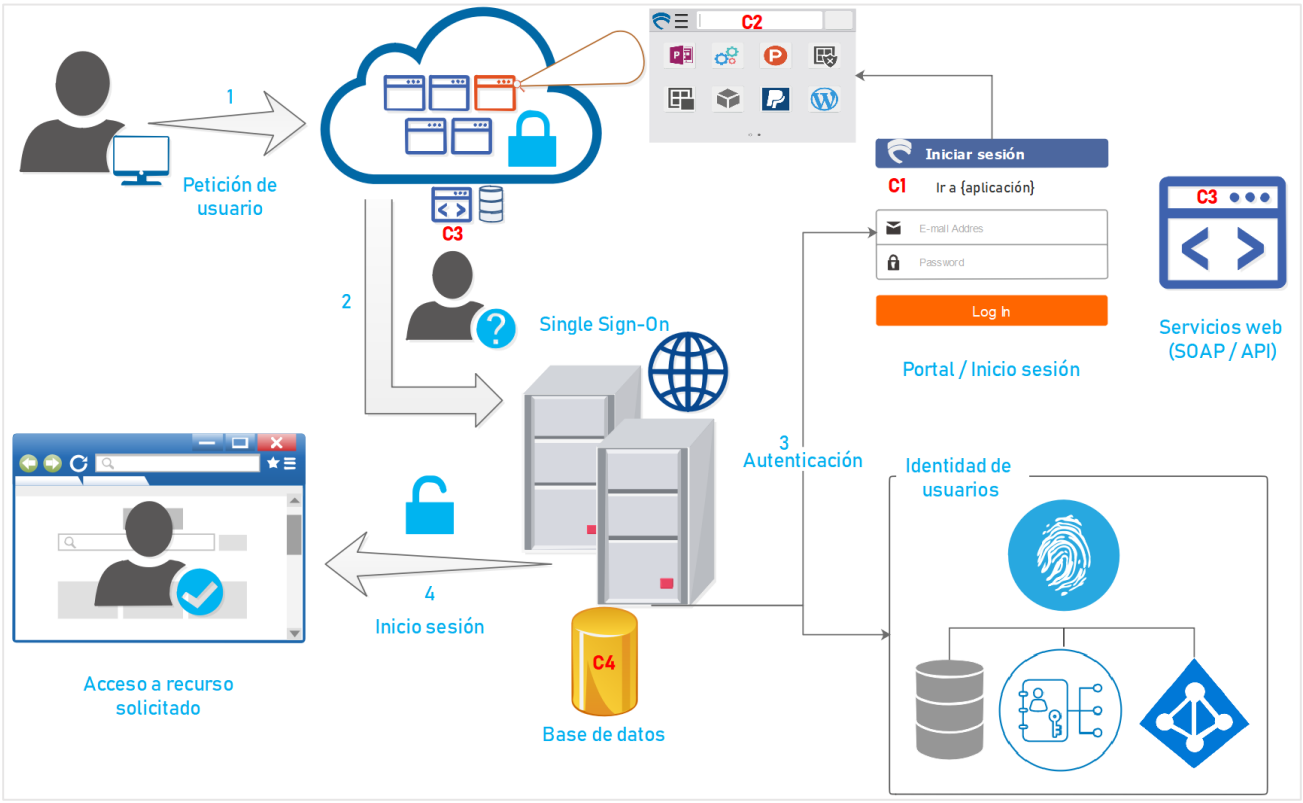

Figura 24: Diseño propuesto de arquitectura SSO. Elaboración propia

La figura 24 ilustra la propuesta de arquitectura, a un nivel macro, de una solución SSO con sus componentes y el flujo de interacción entre estos. La propuesta integra las entidades lógicas que se han mencionado en el capítulo IV y de igual forma analizadas en el capítulo II, sin embargo, el cambio más evidente y sustancial es que no se reflejan por separado los proveedores de identidad y de servicios, sino que se propone un diseño que los integre como un solo componente al que identificamos como servidor de Single Sign-On.

Dado que esta propuesta tiene una delimitación para entornos empresariales, y sirve como modelo para un desarrollo interno, los componentes exponen una relación de confianza inherente entre ellos. Sin embargo, siempre existe la posibilidad de saltar esa confianza y es por ello que se implementan mecanismos de seguridad para validar que las peticiones provienen de un origen confiable y no suponerlo.

El servidor SSO expone principalmente cuatro componentes:

- Pantalla de inicio de sesión (C1): Este componente es reutilizable por todas las aplicaciones, y es aquí que se solicitan las credenciales de usuario. Se identifica el

recurso al que se solicita acceso. Ej. La pantalla muestra una leyenda del tipo “Ir a FINANZAS” o “Ir a SCM”. Esto se logra con parámetros enviados en la URL.

- Portal web (C2): Para usuarios finales (usuarios regulares que utilizan las aplicaciones en su día a día de trabajo) es un nuevo componente que les permite visualizar todas las aplicaciones a las que tiene acceso. Para usuarios administradores este portal web habilita menús de configuración necesarios para la integración de las aplicaciones.
- Servicios web (C3): El servidor SSO y las aplicaciones exponen servicios web necesarios para validar si un usuario ya ha sido autenticado, y también que sea de una fuente confiable.
- Acceso a estructura de datos que contienen las identidades de usuarios (C4): Clases de conexión a los diferentes repositorios que contienen las identidades de los usuarios. El servidor SSO hace uso de estos conectores para consumir la información del usuario y decidir si es válido o no. Dado que el IdP y el SP se integran en un solo componente, la comunicación interna es tradicional y no es necesario implementar algún protocolo.

Las aplicaciones web actuales que desean integrarse al mecanismo de SSO, deben proveer un cambio mínimo que en esencia permite tres cosas:

1. Seguridad: Por medio de un clave que es compartida con el servidor SSO.
2. Puerta trasera: Un mecanismo que habilita la autenticación de las aplicaciones para ejecutarse de forma local, para ello se define un campo lógico que identifica si se debe redirigir al inicio de sesión único o hacer uso de una URL alterna para autenticarse de forma local.
3. Exponer un servicio web para ser consumido por el servidor SSO.

Esta arquitectura es para aplicaciones web que buscan integrarse a un entorno que centraliza el proceso de autenticación y tomar ventaja de esta implementación.

A continuación, se hace un detalle específico de los componentes del servidor SSO y los elementos técnicos que se deben considerar como parte del marco de trabajo genérico propuesto por la investigación.

5.2 Modelado de datos

Se establecen los diferentes modelos de datos para integrar las aplicaciones y el portal de configuraciones de la solución SSO.

5.2.1 Modelado de datos para las aplicaciones web

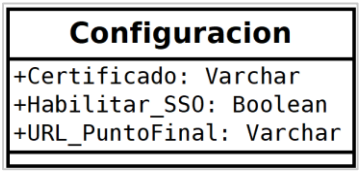

Figura 25: Modelo de datos aplicaciones web. Elaboración propia

Se define una entidad en las aplicaciones web para habilitar la integración a la solución SSO. Se establecen los siguientes atributos:

Certificado: Almacena la información de la llave compartida con el servidor SSO y es requerido para comprobar la confianza entre ambas partes.

Habilitar_SSO: Habilita una puerta trasera para acceder a la aplicación en caso existan problemas con la autenticación en el servidor SSO. El administrador puede actualizar este campo directamente en la base de datos en caso sea necesario. Como resultado, cuando se navega a la aplicación y se detecta que *Habilitar_SSO* es *falso*, entonces internamente se redirige a una URL, en el dominio de la aplicación, que muestra una pantalla de inicio de sesión alterna para que usuarios puedan acceder con sus credenciales asignadas en la aplicación. Es importante recalcar que la autenticación no se realiza contra el SSO ni utiliza el componente de *Identidad de Usuarios*.

URL_PuntoFinal: Identificador de la aplicación web que contiene el inicio de sesión centralizado. Es necesario para saber el recurso al que se debe redirigir el usuario donde procede a autenticarse. Ejemplo: *https://portalsso/servicio/autenticacion/*

5.2.2 Modelado de datos para el SSO

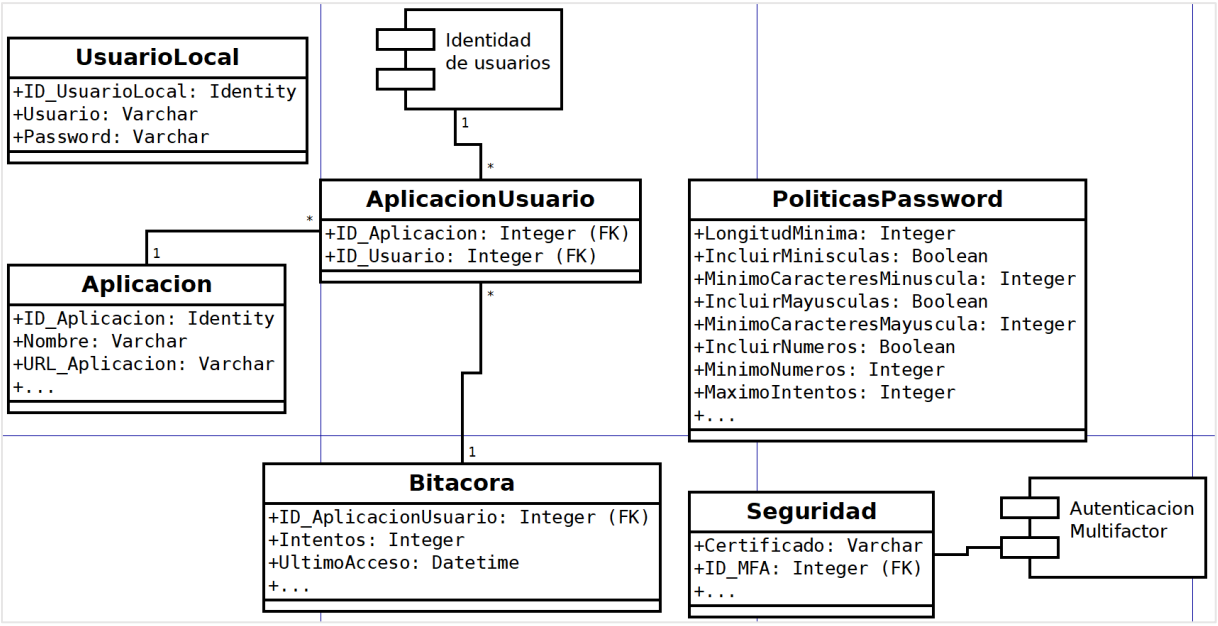

Figura 26: Modelo de datos servidor SSO, campos y tipos de datos sugeridos. Elaboración propia

Se definen varias entidades relacionadas a la administración del portal web y configuraciones para gestionar políticas de contraseñas y establecer controles de seguridad. A continuación, un detalle de las entidades sugeridas en este modelo:

Seguridad: Almacena la información de la llave compartida (certificado) que se utiliza para comprobar la confianza entre el servidor SSO y las aplicaciones que solicitan el recurso. Se sugiere agregar otro control de seguridad para validar a los usuarios (autenticación multifactor) que tendrá configuraciones muy particulares según sea requerido. Los mecanismos de doble autenticación (MFA) dependen de cada producto, pueden ser por: notificaciones, mensajes de texto, llamadas telefónicas, generación de tokens, entre otros. Ejemplo de autenticación multifactor utilizando un producto de Microsoft:

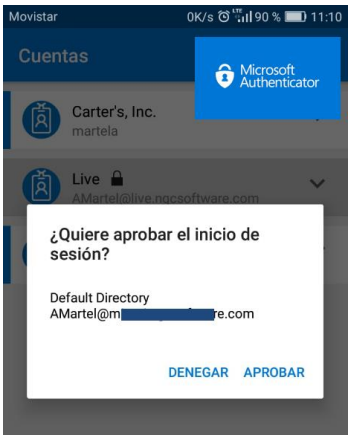

Figura 27: Microsoft Authenticator notificación de acceso para inicio de sesión por usuario.

Aplicacion: Catálogo de las aplicaciones con los datos relevantes para la correcta identificación del recurso solicitado. Sirve para identificar el nombre de la aplicación que se muestra en la pantalla de inicio de sesión único, en base a un parámetro enviado en la URL. Como ejemplo, al acceder el siguiente recurso (*https://portalsso/servicio/autenticacion/?url=https%3A%2F%2Ffinanzas%2FQA),* nótese que la URL resultante incluye la URL de la aplicación solicitante y el servidor SSO busca esa URL en sus aplicaciones para resolver el nombre que se debe mostrar. Como medida de seguridad en las consultas del lado del servidor SSO, el valor enviado en la URL debe incluirse en un objeto de tipo *parámetro* para evitar *SQL injection*. Se debe considerar que cada tecnología expone mecanismos para prevenir este tipo de ataques.

De igual forma el servidor SSO debe conocer la URL de la aplicación solicitante para consumir un servicio y verificar la confianza entre las partes. Se explica a mayor detalle en 5.5.

AplicacionUsuario: Relaciona cuáles son las aplicaciones que el usuario tendrá acceso para que pueda ser mostrada en el portal. Los usuarios (del entorno empresarial) son obtenidos de un repositorio externo dado que no residen en el servidor SSO.

PoliticasPasswod: Asigna diferentes políticas de seguridad en las contraseñas de usuarios. Toma efecto cuando un usuario solicita un cambio de contraseña o bien un administrador del portal crea un usuario y le asigna una contraseña.

Bitacora: Mantiene un historial sobre la actividad de los usuarios en las diferentes aplicaciones a las que solicita acceso. Si el desarrollo lo requiere, se sugiere implementar una bitácora con soporte para reportes y generar indicadores para establecer componentes del tipo: *machine-learning, MFA para ciertos usuarios (en demanda y por contexto), etc.*

Usuario Local: Al igual que en las aplicaciones web, se sugiere implementar un mecanismo de puerta trasera. La pantalla alterna de inicio de sesión en el portal puede ser habilitada solo para los usuarios que estén definidos en un repositorio local del servidor SSO.

5.3 Diseño inicio de sesión único

Figura 28: Pantalla para inicio de sesión único. Elaboración propia

Se habilita una pantalla para el inicio de sesión único, se establece la misma para realizar el acceso a las aplicaciones de manera directa, como al portal web de la solución SSO.

Solicitudes de acceso:

- Ingresar al portal web los recursos solicitado es: https://portalsso/servicio/autenticacion/
- Ingresar a una aplicación de manera directa el recurso solicitado es: https://portalsso/servicio/autenticacion/?url=aplicacion_a_ingresar

Formas de acceso:

Acceso al portal: Al solicitar el recurso para ingresar al portal web, presenta la pantalla para inicio de sesión única, para que se realice la autenticación y concibe el acceso al menú de aplicaciones.

Acceso directo: Cuando se solicita directamente el recurso de la aplicación a la que se desea acceder, se presenta la pantalla para el inicio de sesión único, cuando es realizada la autenticación concibe el acceso a la pantalla principal de la aplicación web.

5.4 Diseño Portal Web.

Se establece el diseño de un portal web al que los usuarios finales y administradores tienen acceso y que de igual forma se integra al ambiente SSO, es decir que el proceso de autenticación utiliza la estructura de datos *Identidad de usuarios* para identificar a los usuarios. El modelado de datos para el SSO sugiere, al igual que en las aplicaciones regulares, la implementación de un mecanismo de puerta trasera.

El portal consta de dos principales pantallas: *Tablero de aplicaciones* y *Menú de opciones*.

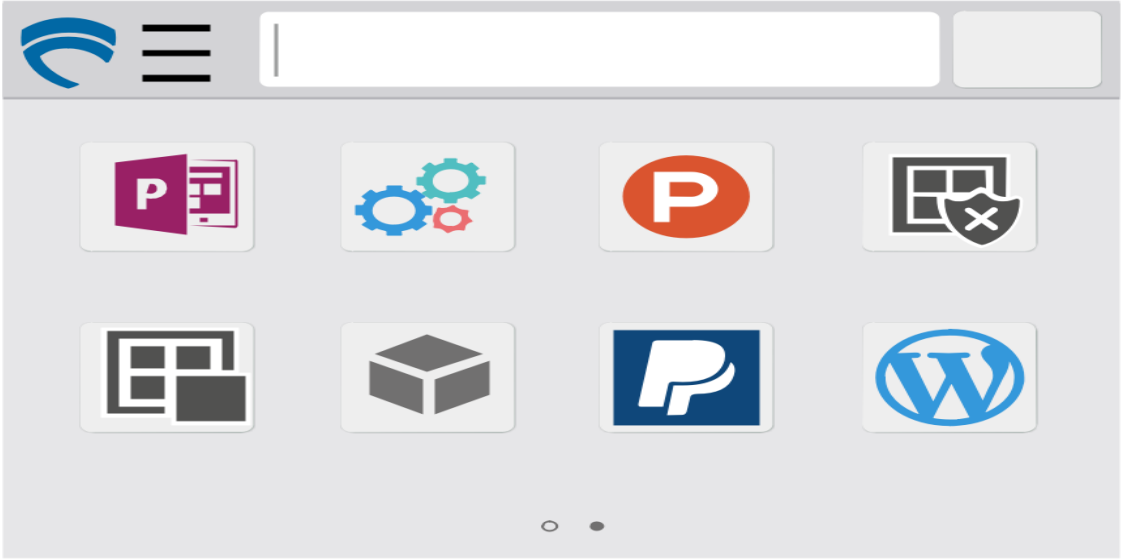

Figura 29: Tablero (dashboard) con los enlaces a las diferentes aplicaciones que el usuario tiene acceso.

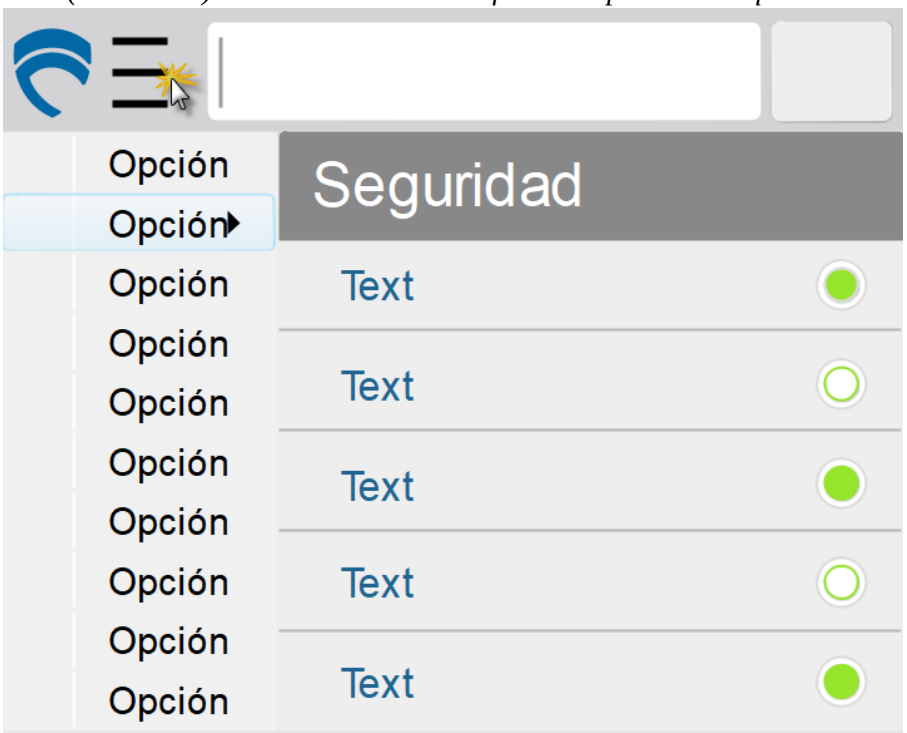

Figura 30: Menú de opciones para los usuarios. Elaboración propia

En la figura 29 se establece la pantalla principal del portal donde los usuarios visualizan las aplicaciones asociadas a las que poseen acceso. La pantalla muestra los iconos que hacen referencia a las aplicaciones, así mismo se incluye una pequeña leyenda para identificar la aplicación.

En el diseño propuesto, en la parte superior izquierda de la figura 29 se encuentra un menú desplegable que muestra las opciones según el tipo de usuario (la figura 30 despliega este menú):

- Usuario administrador*:* Muestra las opciones necesarias para la configuración de las aplicaciones, gestión de usuarios, políticas de seguridad, etc.
- Usuario final: Opciones que se requieran, ejemplo: cambio de contraseña, configuración de tablero,

5.5 Procesos de integración

Se identifican tres actores en los procesos que permiten la integración de las aplicaciones web al entorno SSO.

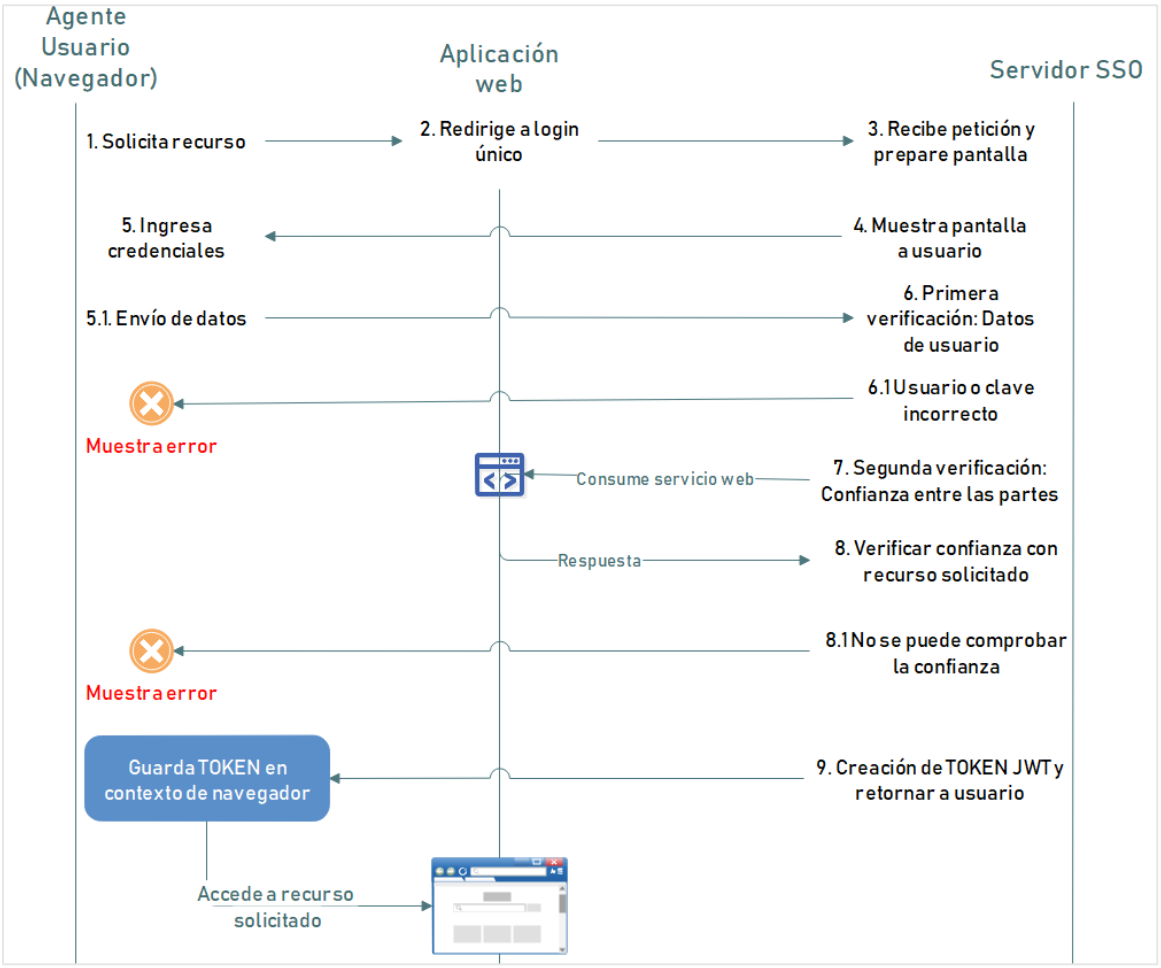

Figura 31: Proceso caso 1 "primer acceso a la aplicación". Elaboración propia

Se muestra el flujo de acciones sugerido para autenticar a los usuarios haciendo uso del servidor SSO. Entre las *acciones* mostradas, queda a discreción de los implementadores las técnicas que debe aplicar para lograr el flujo sugerido. Algunas observaciones:

- La verificación de usuarios se realiza utilizando la estructura de datos que contiene la *Identidad de usuarios*. Dicha estructura es únicamente accesible por el servidor SSO.
- La verificación de confianza es realizada mediante el cifrado de datos utilizando el *certificado* que cada aplicación contiene, y es comparados (el dato cifrado) contra uno generado por el servidor SSO. Si las codificaciones son iguales, entonces se confirma la confianza entre las partes. Este paso es importante, y el hecho de que una aplicación contenga un certificado distinto no permite que ese recurso sea accedido por los usuarios utilizando el servidor SSO.
- No se ilustra el acceso por medio de la puerta trasera a las aplicaciones, sin embargo, el proceso y justificación se ha detallado en los subtemas anteriores.
- Se sugiere uso de JSON Web Token (JWT) dado que es un estándar para el manejo de autenticación. El token es generado por el servidor SSO, pero no se almacena en él, siempre reside en el contexto del navegador.
- En la acción de *acceder a recurso solicitado*, los implementadores deben definir el proceso interno para realizarlo. No es algo que el modelo propuesto realice, se limita a dar un indicador de quién y a que recurso tiene acceso. Cada aplicación debe proveer el mecanismo para generar el contexto del usuario autenticado dentro de la misma.

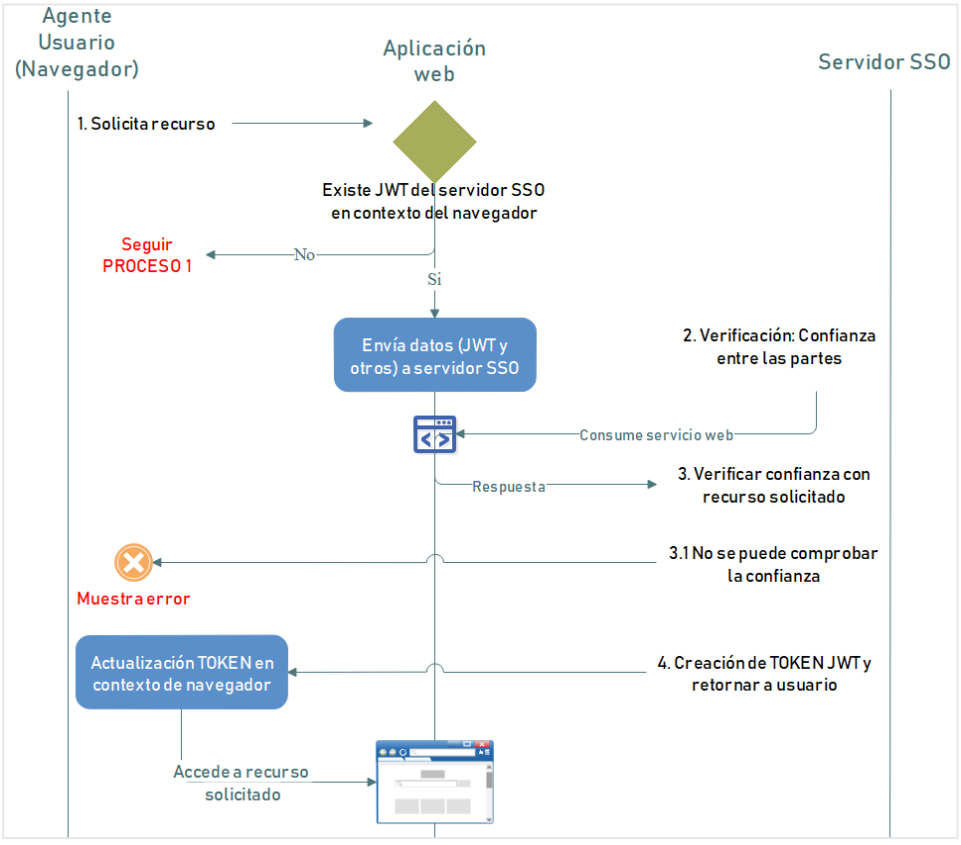

Figura 32: Proceso caso 2 "acceso a otro recurso". Elaboración propia

Capítulo VI – Conclusiones

Se realiza el estudio de las tecnologías SSO actuales, como punto de partida, para identificar los diferentes componentes que integran y sí comprender como coexisten dentro de la solución, con el objetivo de realizar el análisis y obtener un juicio de valor al momento de definir la propuesta de diseño para la implementación de una solución de inicio de sesión único.

Considerando los componentes para una solución SSO, según el modelo propuesto en el capítulo cinco de la investigación, la selección de tecnologías a utilizar en el desarrollo del software debe ser compatibles y que puedan adaptase a los *procesos de integración*.

6.1 Tecnologías de SSO actuales.

Fuera del contexto de inicio de sesión único (SSO), los usuarios finales hacen malabares con las múltiples credenciales que deben administrar, ya sea que las memoricen o las guarden en algún lugar (un baúl de contraseñas, archivos de texto, etc.). De igual forma el personal de TI debe estar disponible para asistir a los usuarios en los posibles problemas que se presenten en el proceso de autenticación, tareas como el restablecimiento de la contraseña, dar de alta a un nuevo empleado, remover un usuario de ciertas aplicaciones, etc.

Las soluciones comerciales analizadas en la investigación exponen las características funcionales, la arquitectura, el uso de estándares, las tecnologías soportadas, entre otros. Entre la gran variedad de opciones, es importante recordar que el SSO, en su forma básica, debe simplificar la experiencia de los usuarios en internet respecto a la tarea de inicio de sesión: ingresar las credenciales solo una vez y ganar el acceso a todas las aplicaciones según la configuración. Se establece un único punto de acceso que a la vez es gestionado por TI y se vuelve un componente reutilizable para las aplicaciones que se integran al entorno SSO.

En líneas generales, y en base a las soluciones analizadas, un SSO debe integrar las siguientes características:

- Fácil acoplamiento: Permite a las aplicaciones integrarse a la plataforma SSO de forma ágil, expone documentación para el uso de APIs, entre otros.
- Gestión sencilla: En relación a la integridad de los datos de acceso, al estar en un *gestor centralizado* la sincronización es inherente al proceso. Un cambio de contraseña toma efecto de inmediato y todas las aplicaciones reconocen el cambio de contraseña. Para la administración de usuarios, el dar de alta o baja a un usuario es una tarea sencilla, y no se requiere ir aplicación por aplicación, basta con hacerlo una vez.
- No debe alterar el día a día de los usuarios: El flujo de trabajo de los usuarios no se ve afectado, por el contrario, el acceso a las aplicaciones es transparente y pretende mejorar la experiencia evitando las interrupciones que generan las solicitudes de contraseñas más de una vez, para acceder a los recursos que está autorizado.

- La seguridad: Una característica critica dado que el inicio de sesión único se vuelve parte del sistema nervioso central de la empresa, y cualquier vulnerabilidad compromete la información de la empresa. Día a día se escriben titulares con referencia a violaciones de seguridad o *hackeo* de sistemas y aplicaciones, la seguridad es algo que debe interesar a todos y en este contexto de SSO el equipo de TI juega un rol primario. Se deben prever los mecanismos necesarios para fortalecer la seguridad, ejemplos: autenticación multifactor, indicadores de actividad sospechosa, confianza entre las aplicaciones, manejo de comunicación, robustez de la solución SSO (infraestructura), ventanas de mantenimiento, puertas traseras, etc.

Los productos de terceros cuentan con innumerables configuraciones y beneficios, todos con buenas críticas y casos de estudio exitosos que generan confianza al punto de delegar el proceso de autenticación de usuarios a un agente externo a la empresa u organización. El éxito de la implementación de un producto SSO, en buena medida, depende de la visión estratégica que se tenga, el involucramiento de las partes interesadas (desde la alta gerencia hasta los usuarios finales), componentes adicionales que se deseen incluir (seguridad por contexto, mecanismos de doble verificación de identidad, machine-learning, etc.).

Concluir que un producto y cierta *configuración (estándares, tecnologías, estructuras de datos, seguridad)* es mejor que las demás, basándose solamente en las *fichas técnicas*, no es posible, o por lo menos no se concibe que lo sea, sino que debe analizarse objetivamente y apuntar a que satisfaga la necesidad de la empresa.

Basados en la evaluación de estos productos, y el *diseño de propuesta* (capítulo V de la investigación) se genera una nueva ficha técnica *(adhoc)* para una solución SSO que se integra en un entorno empresarial y contenga la definición de los diferentes componentes que deben ser incluidos. La especificación de este nuevo modelo se basa en el análisis de los productos de terceros, donde se retoman las principales características, ventajas, requerimientos funcionales y no funcionales, arquitecturas, y todo lo necesario para dar un valor agregado a la solución propuesta.

La importancia de este nuevo modelo es lograr minimizar el costo de adquisición e implementación para este tipo de soluciones en entornos empresariales, además de poder administrar los recursos en su totalidad y adaptarlo a las necesidades específicas de la empresa. Se pretende implementar un mecanismo de reutilización en varios componentes de la empresa: infraestructura, centros de datos, aplicaciones web, identidad de usuarios, recursos de TI, etc.

6.2 Recomendaciones

Para poder realizar la implementación de una solución, sea está contratada por un tercero o desarrollada de manera interna dentro de las organizaciones, se deben realizar cambios de los cuales podemos mencionar: Infraestructura, capacidad de hardware, adquisición de equipos nuevos (computadoras, servidores), cambios en las aplicaciones empresariales para integrase a la

solución de inicio de sesión único y cualquier ajuste que sea necesario para su correcta implementación.

Todas las recomendaciones se establecen de manera general para que cada equipo implementador decida las tecnologías a utilizar, ya sea por políticas de la empresa, compatibilidad, conocimiento por parte de los desarrolladores, entre otros. No se pretende apuntar a una(s) tecnología(s) en específico dado que el modelo arquitectónico es factible ejecutarlo utilizando cualquiera de ellas o una combinación de tecnologías.

- Tecnologías de desarrollo a utilizar: Se recomienda se utilicen las que no estén obsoleta o está en pleno desarrollo, que cuenten con soporte, estándares de calidad para el desarrollo.
- Almacenamiento de la información: Se recomienda a las organizaciones implementar mecanismos de respaldo para sus estructuras de datos, replicación de la información, uso de recursos del servidor.
- Procesos**:** Se recomienda el uso de tecnologías que administren de manera óptima los recursos del servidor (RAM, CPU, HDD), manejo de procesos en hilos, manejo de rack para maximizar el espacio en un centro de datos centralizados, flexibilidad en el despliegue de las aplicaciones, garantizar una alta disponibilidad mediante el uso de balanceadores de carga.
- Configuraciones de los servidores:

  La velocidad del reloj: Establecer una velocidad más alta para que las respuestas a las solicitudes web se establezcan de manera óptima.

  Tamaño de caché: Al aumentar la caché se reducen las frecuencias que necesita el procesador para obtener los datos de memoria, lo cual ayuda sobre la capacidad de respuesta del sistema y brinda una mejor experiencia al usuario.

6.3 Costos

6.3.1 Costos de desarrollo

Para llevar a cabo la implementación de una solución se debe conocer que, además de realizar mejoras y cambios, también se requiere de una inversión la cual genera costos. La apuesta es a que los ingresos mejoren y estos sean mayores que los costos actuales y futuros.

Recursos técnicos: Se debe realizar una inversión de equipos informáticos adecuados para el desarrollo de la solución o adquirir diferentes componentes de hardware para potenciar los activos con los que cuenta la empresa.

Recurso humano: Al implementar una solución SSO se debe conocer la magnitud del proyecto, por lo cual se debe establecer el personal que se asignara a su eventual ejecución. Independiente de la metodología de desarrollo a utilizar se debe prever el costo que este requiere,

una forma acertada es hacer la planeación del desarrollo a través de un *cronograma de actividades* y establecer los *tiempos* y *cantidad* de recurso.

Tomando como referencia la experticia del equipo investigador al momento de desarrollar e implementar soluciones a la medida, se logra establecer un consenso de cuáles son las *variables* a tomar en cuenta al momento de calcular los costos del proyecto. De igual forma en los ambientes donde el equipo realiza sus labores diarias, se evidencia este tipo de escenarios donde se establece el mismo patrón para calcular los costos de desarrollo e implementación.

Fórmula para poder realizar el cálculo del costo:

Costo(C) = Recurso (R) X Tiempo (T)

Nota: La fórmula es general y para calcular un mejor costo se deben identificar, en cada una de las fases, cual es la variable a modificar al momento de planificar el desarrollo de ellas. Así conocer en mejor detalle el monto de inversión.

Análisis y diseño: El único costo que tendremos en este apartado es realizar el estudio de la propuesta, analizar los componentes definidos y comprender como se debe establecer la eventual implementación bajo el diseño de la arquitectura que se provee en el presente documento.

Desarrollo y ajustes: Se debe establecer la cantidad de recursos para el desarrollo de la solución y la cantidad de horas, así obtener un costo en el desarrollo y los ajustes a las aplicaciones web que se integraran a la solución,

Costo(C) = Recurso_Asignado (RA) X Tiempo_Desarrollo (TD)

Pruebas: Para poder realizar la verificación y posterior validación que se cuenta con la funcionalidad y calidad necesaria, debemos realizar diferentes tipos de pruebas, con la finalidad llegar a una pre-producción que se puede catalogar como pruebas.

Costo(C) = Recurso_Asignado (RA) X Tiempo_Pruebas (TP)

Producción: Al realizar el lanzamiento de una solución se debe tener asignado el personal para cubrir con diferentes factores técnicos desde la instalación a la ejecución de pruebas para todos los posibles casos del proceso.

En este punto, es posible estimar el costo total de la implementación realizado la sumatoria de los costos establecidos por cada una de las fases:

Costo total = C_Analisis_Disño + C_Desarrollo + C_Pruebas + C_Producción

6.3.2 Costos de adquisición

Las soluciones SSO analizadas en la investigación, manejan un modelo de costos en base al número de usuarios adquiridos y esta fijación de precios (pricing) a la vez varía según el modo de suscripción: mensual y anual.

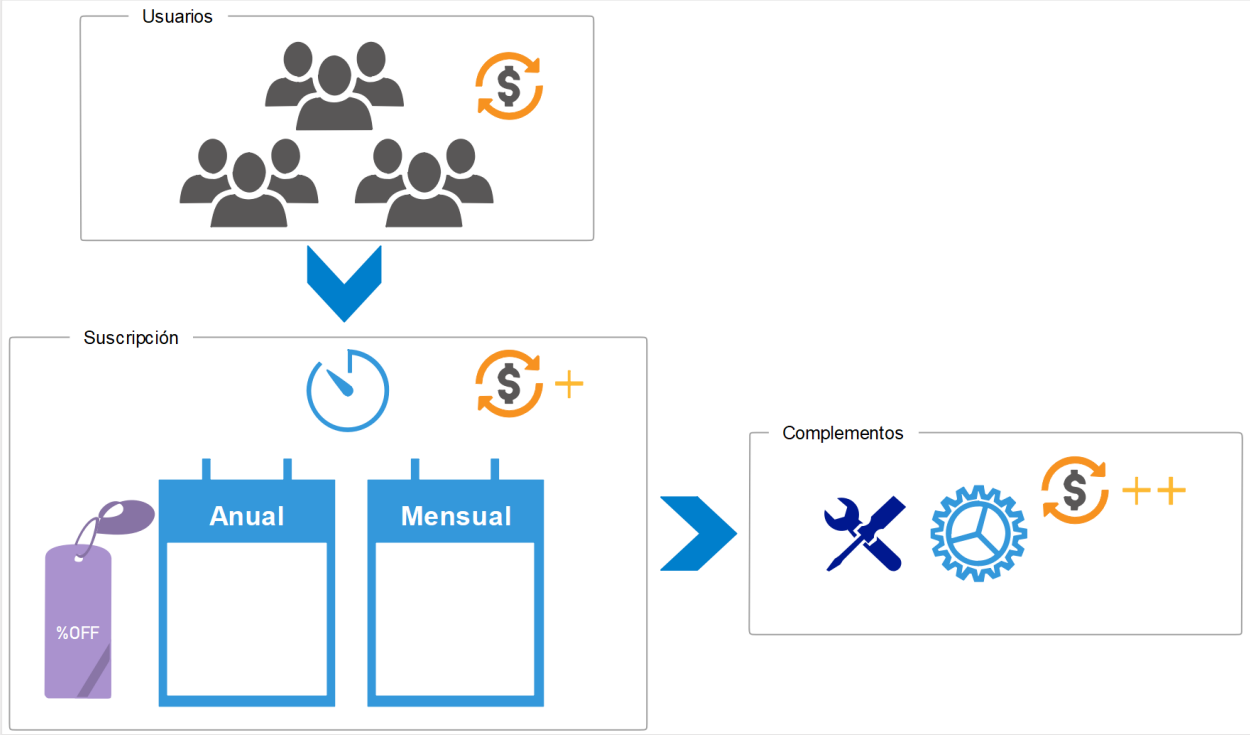

Figura 33: Costos de adquisición. Elaboración propia

Al momento de adquirir un producto de SSO es importante conocer la fijación de precios del proveedor. La figura 33 ilustra como el precio varía según las configuraciones que se adquieren, algunas consideraciones a tomar en cuenta son:

- A mayor número de usuarios, algunos proveedores mejoran el precio base. Los proveedores definen un número mínimo de usuarios que se deben adquirir.
- La suscripción es mensual o anual, y con contratos anuales el proveedor puede mejorar el precio del producto.
- Adicional a la solución SSO, se pueden agregar complementos que den valor agregado a la empresa.
  Algunos complementos son requeridos por las políticas internas de la empresa, como *autenticación multifactor*. Otros complementos son útiles y necesarios para el equipo de TI en la empresa, un ejemplo es la *gestión de APIs, reportes, etc.*
  Otros posibles complementos: *Machine learning, autenticación adaptativa, gestión de acceso contextual, etc.*

- Se debe considerar una mejora al soporte del producto. Ejemplos: Disponibilidad 7/24, tiempos de respuesta, entre otros. Además, se recomienda establecer un acuerdo de nivel de servicio (*SLA* por sus siglas en ingles de Service Level Agreement).

Se establece como un ejemplo la tabla 9 con los productos y servicios para la implementación de un SSO en una empresa, de igual forma se identifican algunos componentes adicionales. La empresa requiere 2,500 usuarios y algunas licencias para usuarios desarrolladores (REST API), por una contratación anual (12 meses).

Productos / Servicios	**Costo**
Administración de usuarios (soporte estándar)	$2.00 mensual por usuario
Autenticación Multifactor	$2.00 mensual por usuario
Soporte 24x7x365	$1,200.00 anual
Integración con VPN	$500.00 anual
REST APIs	$156.00 anual (1,000 usuarios)
Autenticación adaptativa	$960.00 anual

Tabla 9: Ejemplo del costo de adquisición de una solución SSO.

Cálculo del costo de adquisición:

(SSO) 2,500 usuarios X $2 X 12 meses:	$60,000
(MFA) 2,500 usuarios X $2 X 12 meses:	$60,000
(COMPONENTES) $1,200 + $500 + $156 + $960:	$2,816
TOTAL:	$122,816

Para la empresa, es importante identificar el número de usuarios que desea integrar a la solución (debe revisar su portafolio de aplicaciones), decidir el modo de suscripción y los complementos necesarios a incluir en el producto que se desea adquirir.

Referencias bibliográficas

[1] J. I. Martín, "Implantación de un SSO (Single Sign On)", trabajo de fin de máster, Universidad Abierta de Cataluña, 2008. [Último acceso: 12 May 2019] Disponible en: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/28021/6/nacho_martinTFM0114memoria.pdf

[2] Tecnoinver (2015, Sep 16). Qué es Single Sign-On o Autenticación Única. [En línea]. [Último acceso: 15 May 2019] Disponible en: https://www.tecnoinver.cl/que-es-single-sign-on-o-autenticacion-unica/

[3] MDN Web Docs (2019, May 18). HTTP cookies. [En línea]. [Último acceso: 01 Jul 2019] Disponible en: https://developer.mozilla.org/es/docs/Web/HTTP/Cookies

[4] Onelogin. How does single sign-on work. [En línea]. [Último acceso: 13 Jun 2019] Disponible en: https://www.onelogin.com/learn/how-single-sign-on-works

[5] J. Bonneau, C. Herley, P. C. van Oorschot, F. Stajano, "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes", in 2012 IEEE Symposium on Security and Privacy, San Francisco, CA, USA, 2012. [Último acceso: 20 May 2019] Disponible en: https://ieeexplore.ieee.org/abstract/document/6234436

[6] MIT (2019, Ene 08). What is Kerberos. [En línea]. [Último acceso: 22 Jun 2019] Disponible en: https://web.mit.edu/kerberos/#what_is

[7] Empowerid. Service Identity Providers. [En línea]. [Último acceso: 25 May 2019] Disponible en: https://www2.empowerid.com/learningcenter/technologies/service-identity-providers

[8] IT Central Station. [En línea]. [Último acceso: 10 Jul 2019] Disponible en: https://www.itcentralstation.com/

[9] IT Central Station. the leading product review site for enterprise technology buyers. [En línea]. [Último acceso: 10 Jul 2019] Disponible en: https://marketing.itcentralstation.com/about/

[10] IT Central Station (2019, Ene). Single Sign-On (SSO). [En línea]. [Último acceso: 12 Jul 2019] Disponible en: https://www.itcentralstation.com/products/comparisons/auth0_vs_okta_vs_onelogin

[11] FinancesOnline (2019, Feb 28). Compare Auth0 vs Okta Identity Cloud. [En línea]. [Último acceso: 12 Jun 2019] Disponible en: https://comparisons.financesonline.com/auth0-vs-okta-identity-cloud

[12] FinancesOnline (2019, Feb 28). Compare OneLogin vs Auth0. [En línea]. [Último acceso: 13 Jun 2019] Disponible en: https://comparisons.financesonline.com/onelogin-vs-auth0

[13] Okta. Single Sign-On. [En línea]. [Último acceso: 05 May 2019] Disponible en: https://www.okta.com/products/single-sign-on/

[14] Okta (2019, Abr 02). Okta Launches Risk-Based Authentication Solution with Machine Learning Capabilities, Enhancing Adaptive Multi-factor Authentication and Adaptive Single Sign-On Products. [En línea]. [Último acceso: 06 May 2019] Disponible en: https://www.okta.com/press-room/press-releases/okta-launches-risk-based-authentication-solution-with-machine-learning-capabilities/

[15] H. Aguilar, J. Todd, "A Paradigm Shift in Scale for Identity and Access Management", Okta, 2016. [En línea]. [Último acceso: 12 May 2019] Disponible en: https://www.okta.com/resources/whitepaper/scaling-okta-to-10-billion-users

[16] Okta. Identity Cloud Pricing. [En línea]. [Último acceso: 30 May 2019] Disponible en: https://www.okta.com/pricing/#it-single-sign-on

[17] Onelogin. Secure Signgle Sign-On (SSO) Solution. [En línea]. [Último acceso: 18 Jun 2019] Disponible en: https://www.onelogin.com/product/sso

[18] Onelogin. [En línea]. [Último acceso: 05 Jul 2019] Disponible en: https://www.onelogin.com/

[19] Onelogin. Cloud-Based IAM for the Modern Enterprise. [En línea]. [Último acceso: 15 Jul 2019] Disponible en: http://resources.onelogin.com/SB-OneLogin-Product-Datasheet.pdf

[20] Onelogin. Pricing. [En línea]. [Último acceso: 30 May 2019] Disponible en: https://www.onelogin.com/product/pricing

[21] Auth0. Auth0 Overview. [En línea]. [Último acceso: 15 Jun 2019] Disponible en: https://auth0.com/docs/getting-started/overview

[22] Auth0. Pricing. [En línea]. [Último acceso: 30 May 2019] Disponible en: https://auth0.com/pricing

[23] Soluciones para la pequeña empresa MB (2019 Jul). Las mejores soluciones de inicio de sesión único para empresas en 2018. [En línea]. [Último acceso: 30 May 2019] Disponible en: https://es.mobbybusiness.com/4623best-single-sign-on-solutions-for-business-in-2018

[24] O. Blancarte (2017, Jul 10) Single SingOn (SSO). [En línea]. [Último acceso: 15 Jul 2019] Disponible en: https://www.oscarblancarteblog.com/2017/07/10/single-singon-sso/

[25] GlobalSign. Acceso Centralizado y Control de Autorización. [En línea]. [Último acceso: 22 Jul 2019] Disponible en: https://www.globalsign.com/es/gestion-de-identidades-y-accesos/single-sign-on/

[26] M. A. Alvarez (2018, Abr 11). Autenticación por token. [En línea]. [Último acceso: 15 Jul 2019] Disponible en: https://desarrolloweb.com/articulos/autenticacion-token.html

[27] S. Cantor (2017, May 08), Proveedores de Identidad. [En línea]. [Último acceso: 15 Jun 2019] Disponible en: https://wiki.shibboleth.net/confluence/display/CONCEPT/Home

[28] Okta. SAML. [En línea]. [Último acceso: 15 Jun 2019] Disponible en: https://www.okta.com/integrate/documentation/saml/

[29] Cisco System (2019, Mar 25). Configure the Identity Provider for CiscoIdentity Service to enable SSO [En línea]. [Último acceso: 30 Jun 2019] Disponible en: https://www.cisco.com/c/en/us/support/docs/customer-collaboration/unified-contact-center-express/200612-Configure-the-Identity-Provider-for-UCCX.pdf

[30] R. Bhargave. "What is an Identity Provider (IdP)?", JumpCloud Directory-as-a-service. Mar 2019 [En línea]. [Último acceso: 30 Jun 2019] Disponible en: https://jumpcloud.com/blog/identity-provider-idp/

[31] M. Rouse. "Active Directory", TechTarget. Jun 2018 [En línea]. [Último acceso: 22 Jul 2019] Disponible en: https://searchwindowsserver.techtarget.com/definition/Active-Directory

[32] T. Shyamsundar. "What is ADFS?". Okta. Jun 2018 [En línea]. [Último acceso: 15 Jun 2019] Disponible en: https://www.okta.com/blog/2018/06/what-is-adfs/

[33] Okta. Okta MFA for Microsoft ADFS. [En línea]. [Último acceso: 31 May 2019] Disponible en: https://www.okta.com/integrations/okta-mfa-for-microsoft-adfs/

[34] E. B. Méntrida. "Implantación de un SSO (Single Sign On)" proyecto de fin de master Universidad Abierta de Cataluña, 2018. [Último acceso: 12 May 2019] Disponible en: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/73085/6/ebarquillaTFM0118memoria.pdf

[35] G. Lattimore. "Can OpenLDAP Replace Active Directory?", JumpCloud Directory-as-a-service. May 2019 [En línea]. [Último acceso: 15 Jun 2019] Disponible en: https://jumpcloud.com/blog/openldap-replace-ad/

[36] OpenLDAP. The OpenLDAP Foundation Overview. [En línea]. [Último acceso: 11 Jul 2019] Disponible en: https://www.openldap.org/foundation/

[37] OpenLDAP. OpenLDAP Software. [En línea]. [Último acceso: 11 Jul 2019] Disponible en: https://www.openldap.org/

[38] EcuRed. OpenLDAP. [En línea]. [Último acceso: 20 Jul 2019] Disponible en: https://www.ecured.cu/OpenLDAP

[39] Devopsldeas (2017, Sep 25). Planning of LDAP DIT Structure and Config of Overlays (access, ppolicy). [En línea]. [Último acceso: 19 Jun 2019] Disponible en: https://devopsideas.com/planning-of-ldap-dit-structure-and-config-of-overlays-access-ppolicy/

[40] J. Myers, M. Rose (1996, May). Post Office Protocol - Version 3, Universidad Carnegie Mellon [En línea]. [Último acceso: 1 Jun 2019] Disponible en: https://tools.ietf.org/html/rfc1939

[41] J. Myers (1994, Dic). POP3 AUTHentication command, Universidad Carnegie Mellon [En línea]. [Último acceso: 5 Jun 2019] Disponible en: https://tools.ietf.org/html/rfc1734

[42] M. Cañas, "Guía Diseño de seguridad de una Base de Datos", Universidad Don Bosco, 2018. Disponible en: http://www.udb.edu.sv/udb_files/recursos_guias/informatica-ingenieria/base-de-datos-i/2019/i/guia-12.pdf

[43] Web Technology Surveys. [En línea]. [Último acceso: 28 Jun 2019] Disponible en: https://w3techs.com/

[44] SLaYeR, Lechon, "Administración y Gestión de un Servidor Web Apache", 2006. [Último acceso: 12 May 2019] Disponible en: http://index-of.co.uk/SERVIDORES/apache.pdf

[45] CAPACITY Information Technology Academy (2016). Cómo Mejorar La Seguridad De Un Servidor Web Apache. [En línea]. [Último acceso: 2 Jun 2019] Disponible en: http://blog.capacityacademy.com/2013/11/26/como-mejorar-la-seguridad-de-servidor-web-apache/

[46] R. Templin, Equipo IIS (2007, Nov 15). Introduction to IIS Architectures. [En línea]. [Último acceso: 10 Jun 2019] Disponible en: https://docs.microsoft.com/es-es/iis/get-started/introduction-to-iis/introduction-to-iis-architecture

[47] R. Anderson (2016, Sep 25). IIS Configuration Reference. [En línea]. [Último acceso: 11 Jun 2019] Disponible en: https://docs.microsoft.com/en-us/iis/configuration/

[48] Microsoft ©. Enhanced Server Protection. [En línea]. [Último acceso: 12 Jun 2019] Disponible en: https://www.iis.net/overview/security/enhancedserverprotection

[49] Microsoft ©. Access Protection. [En línea]. [Último acceso: 12 Jun 2019] Disponible en: https://www.iis.net/overview/security/accessprotection

[50] Gluu Inc. OAuth vs SAML vs OpenID Connect. [En línea]. [Último acceso: 15 Jun 2019] Disponible en: https://www.gluu.org/resources/documents/articles/oauth-vs-saml-vs-openid-connect/

[51] J. Petters (2018, Ago 21). What is SAML and How Does it Work?. [En línea]. [Último acceso: 10 May 2019] Disponible en: https://www.varonis.com/blog/what-is-saml/

[52] D. Carru (2016, Feb 29). SP vs IdP Initiated SSO. [En línea]. [Último acceso: 12 May 2019] Disponible en: https://blogs.oracle.com/dcarru/sp-vs-idp-initiated-sso

[53] Onelogin. Overview of SAML. [En línea]. [Último acceso: 15 May 2019] Disponible en: https://developers.onelogin.com/saml

[54] P. Otemuyiwa (2016, Dic 06). How SAML Authentication Works. [En línea]. [Último acceso: 15 May 2019] Disponible en: https://auth0.com/blog/how-saml-authentication-works/

[55] OAuth 2.0. [En línea]. [Último acceso: 1 Jul 2019] Disponible en: https://oauth.net/2/

[56] M. Anicas (2014, Jul 21). An Introduction to OAuth 2. [En línea]. [Último acceso: 1 Jul 2019] Disponible en: https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2

[57] OAuth 2.0. Authorization Framework. [En línea]. [Último acceso: 10 Jul 2019] Disponible en: https://auth0.com/docs/protocols/oauth2

[58] Okta. OAuth 2.0. [En línea]. [Último acceso: 1 Jul 2019] Disponible en: https://developer.okta.com/authentication-guide/auth-overview/#oauth-2-0

[59] OAuth 2.0. Protocol flow. [En línea]. [Último acceso: 27 Ju 2019] Disponible en: https://auth0.com/docs/protocols/oauth2#protocol-flow

[60] JWT. Introduction to JSON Web Tokens. [En línea]. [Último acceso: 27 Jun 2019] Disponible en: https://jwt.io/introduction/

[61] M. Macías, "OpenID en el SIR", presentación, Universidad Politécnica de Valencia, 2011. [Último acceso: 15 Jul 2019] Disponible en: http://www.rediris.es/difusion/eventos/foros_movilidad-identidad/2011/ponencias/doc/MiguelMacias-openID.pdf

[62] R. Peña, M. Lombardo, "OpenID connect y la seguridad de la identidad digital", Universidad Tecnológica de Panamá, 2017. [Último acceso: 26 Jun 2019] Disponible en: https://revistas.utp.ac.pa/index.php/ric/article/download/1758/2498

[63] R. Navarro, "REST vs Web Services. Modelado, Diseño e Implementación de Servicios Web", 2007. [Último acceso: 26 Jun 2019] Disponible en (vista previa): https://www.scribd.com/document/102965400/Rest-v-S-Servicios-Web-SOAP

[64] B. González (2004, Jul 07). Simple Object Access Protocol. [En línea]. [Último acceso: 2 Jul 2019] Disponible en: https://desarrolloweb.com/articulos/1557.php

[65] Guru99. SOAP Web Services Tutorial. [En línea]. [Último acceso: 2 Jul 2019] Disponible en: https://www.guru99.com/soap-simple-object-access-protocol.html

[66] EcuRed. Simple Object Access Protocol. [En línea]. [Último acceso: 2 Jul 2019] Disponible en: https://www.ecured.cu/Simple_Object_Access_Protocol

[67] S. Talens. Introducción a los certificados digitales. [En línea]. [Último acceso: 25 May 2019] Disponible en: https://www.uv.es/sto/articulos/BEI-2003-11/certificados_digitales.html

[68] Microsoft © (2018, May 30). X509 Public Key Certificates. [En línea]. [Último acceso: 25 May 2019] Disponible en: https://docs.microsoft.com/en-us/windows/desktop/seccertenroll/about-x-509-public-key-certificates

[69] Really Simple Systems. Single Sign-on – OneLogin Integration. [En línea]. [Último acceso: 16 Jun 2019] Disponible en: https://support.reallysimplesystems.com/single-sign-on/

[70] T. Kawasaki (2017, Oct 30). Diagrams of All The OpenID Connect Flows. [En línea]. [Último acceso: 17 Jun 2019] Disponible en: https://medium.com/@darutk/diagrams-of-all-the-openid-connect-flows-6968e3990660

[71] "Unidad V Enfoque Middleware", notas de clase para Integración de Sistemas Informáticos, MAS, Universidad Don Bosco, M. Hernández, 2017.

[72] H. Sampieri, F. Collado, B. Lucio. Similitudes y diferencias entre los enfoques cuantitativo y cualitativo. Metodología de la investigación. 4a. edición. México D.F.: McGraw-Hill, 2006, cap. 1, pp.3-29. ISBN 970-10-5753-8.

[73] H. Sampieri, F. Collado, B. Lucio. Definición del alcance de la investigación a realizar: exploratoria, descriptiva, correlacional o explicativa. Metodología de la investigación. 4a. edición. México D.F.: McGraw-Hill, 2006, cap. 5, pp.99-117. ISBN 970-10-5753-8.

[74] M. Córdoba, C. Monsalve. TIPOS DE INVESTIGACIÓN: Predictiva, proyectiva, interactiva, confirmatoria y evaluativa [En línea]. [Último acceso: 1 Feb 2019]. Disponible en: http://2633518-0.web-hosting.es/blog/didact_mate/9.Tipos%20de%20Investigaci%C3%B3n.%20Predictiva%2C%20Proyectiva%2C%20Interactiva%2C%20Confirmatoria%20y%20Evaluativa.pdf

[75] U. Brazo (2017, Nov 17). SAML: Qué es, para qué se usa, cómo funciona. [En línea]. [Último acceso: 20 Jul 2019] Disponible en: https://cioperu.pe/articulo/24726/saml-que-es-para-que-se-usa-como-funciona/?p=3

[76] M. Ortiz (2015, Nov 25). Cómo funciona Apache HTTP Server. [En línea]. [Último acceso: 20 Jul 2019] Disponible en: http://migueleonardortiz.com.ar/curso-arquitectura-web/como-funciona-apache-http-server/920

[77] V. Mladenov, C. Mainka (2015, Oct 15). Attacking OpenID Connect 1.0 - Malicious Endpoints Attack. [En línea]. [Último acceso: 20 Jul 2019] Disponible en: https://web-in-security.blogspot.com/2015/10/attacking-openid-connect-10-malicious.html

Anexos