Universidad de La Habana
Facultad de Matemática y Computación

# Sistema de Autenticación Central

Autor:
**Nadia González Fernández**

Tutores:
**Lic. Roberto Martí Cedeño**

Trabajo de Diploma
presentado en opción al título de
Licenciado en Ciencia de la Computación

Noviembre de 2022

github.com/nala7/central_authentication_system

Dedicación

# Agradecimientos

Agradecimientos

# Opinión del tutor

Opiniones de los tutores

# Resumen

Resumen en español

# Abstract

Resumen en inglés

# Índice general

# Índice de figuras

# Ejemplos de código

# Introducción

Hoy en día las personas usan muchos sistemas de software independientes con distintas identidades. La identidad del usuario es considerado un conjunto permanente o de larga vida de atributos asociados a la identidad de un usuario. Para obtener acceso a servicios muchos requieren que el usuario esté registrado y haya iniciado sesión. Muchas de las veces los usuarios están registrados en los distintos sitios con el mismo nombre de usuario y con la misma o similar contraseña, lo cual no es la mejor práctica para preservar la seguridad de las cuentas. Por otra parte, es fácil de olvidar las credenciales, lo cual obliga a los gestores del sistema a enviar un correo no encriptado con información confidencial.

Por ello el manejo de múltiples nombres de usuarios y contraseña es una tarea molesta del Internet actual.

Teniendo en cuenta todos esos problemas de seguridad, cada vez más administradores de sitios web deciden delegar esos servicios de administración y autenticación a terceros, entidades externas especializadas en ese tipo de actividad. Pueden alojar, almacenar, administrar y proteger los datos de los usuarios de un proveedor de servicios en particular. Ofrecen una interfaz de programación de aplicaciones (conocida también por la sigla API, en inglés, *Application Programming Interface*), que pueden proporcionar acceso a datos de usuario para aplicaciones externas. Estas soluciones permiten compartir los datos con más de un sistema web, permitiendo así el Inicio de Sesión Único (SSO) para aplicaciones de terceros.[1]

El Nodo Central de la Universidad de La Habana tiene entre sus responsabilidades dar las credenciales digitales a todos los usuarios de la Universidad. Los trabajadores y estudiantes de la institución registran sus datos personales en las bases de datos de recursos humanos y secretaria docente respectivamente. Esa información es utilizada más adelante para que los usuarios puedan autenticarse en los distintos servicios de la institución, respaldado por sus sistemas de origen.

En este trabajo se presenta una solución a los problemas de autenticación que se ajusta a las necesidades y posibilidades de la Universidad de La Habana.

# Motivación

Actualmente vivimos en un mundo donde la tecnología tiene un papel protagónico. La Universidad de La Habana en los últimos años ha estado inmersa en el proceso de transformación digital que lleva a cabo nuestro país, como continuidad de la estrategia de informatización de la sociedad cubana, que pretende integrar las tecnologías digitales a todos los ámbitos de la sociedad.

En este sentido, nuestra sede universitaria ha avanzado en la digitalización de la información y los procesos y trabaja para facilitar el acceso a la red de todos sus estudiantes y trabajadores. A la par de estos avances, debe crecer también la seguridad de los sistemas para evitar brechas de información y asegurar la privacidad de los usuarios.

Garantizar un acceso seguro y sencillo a los recursos de la red es esencial para alcanzar este objetivo. Por consiguiente, se deben establecer mecanismos que cuenten con un alto nivel de seguridad y permitan identificar quién realmente está autorizado para acceder a los recursos del sistema.

La creación de una plataforma central de autenticación resolvería muchos de estos problemas y mejoraría la experiencia de los usuarios. Estos solo tendrían una cuenta para acceder a todos los servicios que ofrece la Universidad, por lo que no tendrían que lidiar con formularios de inicio de sesión cada vez que vayan a acceder a un servicio.

# Antecedentes

Actualmente todos los usuarios de la Universidad de La Habana se almacenan en dos Protocolos Ligeros de Acceso a Directorios (en inglés: Lightweight Directory Access Protocol, también conocido por sus siglas como LDAP). En los dos directorios se guardan las cuentas de correo y los datos de sus usuarios. En uno ellos se registra la información de los estudiantes y en el otro la de los trabajadores. A partir de estos, todos los sitios web y aplicaciones de la universidad, individualmente, verifican la pertenencia del usuario a la institución.

Cada aplicación y servicio tiene un servicio de autenticación individual que dependen de los dos mencionados directorios que contienen a los usuarios.

# Problemática

La Universidad brinda servicios como Wi-Fi, correo, proxy y EVEA (Entorno Virtual de Enseñanza Aprendizaje), imprescindibles para el funcionamiento de la institución. También brinda servicios más especializados como los sistemas de contabilidad, recursos humanos, inventarios, el registro de notas, entre otros. Todos estos

servicios utilizan la autenticación de forma individual, lo cual significa que un usuario tiene diversas cuentas a pesar de pertenecer todas a la misma institución.

El sistema implementado actualmente es poco eficiente y requiere de intervención humana constante para corregir y/o restablecer el apropiado funcionamiento de los servicios de autenticación. Esto genera tiempos elevados de respuesta y dificulta el mencionado proceso de transformación digital que está siendo llevado a cabo por nuestra Universidad.

# Objetivo

Con el propósito de presentar una propuesta para solucionar la problemática expuesta anteriormente, se plantean los siguientes objetivos:

### Objetivo General

- Diseñar e implementar un sistema de autenticación centralizada para todos los usuarios de La Universidad de La Habana.

### Objetivos Específicos

- Generar los servicios de autenticación con compatibilidad con todas las tecnologías existentes y previstas en la institución.
- Implementar la gestión de control de acceso a todos los servicios ofrecidos por el Nodo Central de forma extensible a futuros servicios y fuentes de datos.

# Estructura de la Tesis

# Capítulo 1

# Estado del Arte

En este capítulo se brindan las definiciones de herramientas utilizadas para la autenticación. También se realiza un estudio sobre el estado del arte de las mismas. Además, se brindan razones para incluir su utilización como parte de la solución propuesta.

## 1.1. Inicio de Sesión Único

El Inicio de Sesión Único ( en inglés *Single Sign-On* o también conocido por sus siglas SSO) ha sido ampliamente adoptado para la autenticación en línea debido a su utilidad y seguridad que ofrece. Este es un método de autenticación que permite a los usuarios iniciar sesión con un único conjunto de credenciales en varios sistemas de software independientes. El inicio de sesión único facilita que un usuario no tenga que iniciar sesión en cada aplicación que use. Con este servicio los usuarios pueden acceder a todas las aplicaciones necesarias sin tener que autenticarse con otras credenciales.[2]

En el mundo digital actual, los usuarios acceden a múltiples sistemas para llevar a cabo sus quehaceres. A medida que aumenta la cantidad de sistemas, también aumenta la cantidad de credenciales de cada usuario y, por lo tanto, también incrementa la posibilidad de perderlas u olvidarlas. El inicio de sesión único se puede utilizar para resolver muchos problemas relacionados con múltiples credenciales para diferentes aplicaciones. El acceso de inicio de sesión único al centro de autenticación principal permite a los usuarios obtener acceso a todos los demás recursos disponibles. SSO ayuda a mejorar la productividad del usuario y del desarrollador al evitar que el usuario recuerde varias contraseñas y también reduce la cantidad de tiempo que el usuario dedica a escribir varias contraseñas para iniciar sesión. SSO también simplifica la administración mediante la gestión de credenciales únicas en lugar de múltiples credenciales. Facilita la gestión de los derechos de un usuario que llega, cambia de función dentro o sale de la empresa, para integrar rápidamente aplicaciones adicionales,

delegar derechos de acceso durante las vacaciones sin aumentar la carga de trabajo de la mesa de ayuda. [3]

Grandes empresas como Google, Facebook y Microsoft utilizan estos servicios. En particular Google permite a sus usuarios iniciar sesión una única vez para tener acceso a todos los servicios de la empresa que se encuentran en la nube. Cuando se configura SSO, los usuarios pueden iniciar sesión en terceros proveedores de identidad y luego acceder a las aplicaciones de Google directamente sin un segundo inicio de sesión [4]. Por ejemplo, si se accede a un servicio de Google como Gmail, se autentica automáticamente en YouTube, AdSense, Google Analytics, y otras applicaciones de Google. Del mismo modo, si cierra la sesión de su Gmail u otras aplicaciones de Google, se cerrará automáticamente la sesión de todas las aplicaciones; esto se conoce como Cierre de Sesión Único (en inglés: *Single Logout*) [5]

## 1.2. Autenticación basada en tokens

Los tokens de acceso se utilizan en la autenticación basada en tokens para permitir que una aplicación acceda a una API. Cuando un usuario se autentica y es autorizado correctamente, la aplicación recibe un token de acceso que luego utiliza como credencial cuando llama a la API. El token pasado informa a la API que el portador del token ha sido autorizado para acceder y realizar un grupo de acciones. [6]

tipos de tokens: https://auth0.com/docs/secure/tokens/access-tokens

un token consiste en datos relacionados con la identidad de un usuario particular. Los usuarios pueden obtener dichos tokens a través de una combinación de nombre de usuario/contra- seña, lo que les permite acceder a los recursos solicitados por un periodo específico de tiempo. Durante este periodo no se requieren métodos adicionales de autenticación. Una caracte- rística importante de los tokens es que permiten ser heredados a otros usuarios, por ello, la autenticación basada en tokens es adecuada cuando un token puede proveer acceso a múltiples servicios. [7]

Existen diversos protocolos de autenticación basados en token populares los más populares entre ellos son: SMAL 2.0 [8]; OpenID [9] [10] y OAuth [11] [12].

Tokens are system generated arbitrary construct that asserts the identity of what it claims to be [13]Token-based authentication embodies the exchange of client authentication credentials for a server generated authentication token; and for subsequent client requests to access SaaS resources, the tokens are sent as part of the request in the HTTP header to the server. This reuse of the same user access token for accessing protected resources governed by certain policies can be a challenge, especially when a resource access policy is updated and the user access token is still valid. In fact, this can introduce a Time- Based vulnerability (timing attack) on the protected resources; and with multiple users accessing the resource, the vulnerability index can increase exponentially. Hence, an authentication and authorization model that limits such vul-

nerabilities and enhances secure resource access have been proposed and evaluated. [14]

## 1.3. OpenID Connect

*OpenID Connect* (también OIDC)es una capa de identidad simple implementada a partir del protocolo *OAuth 2.0*. Permite a los clientes verificar la identidad del usuario final en función de la autenticación realizada por un servidor de autorización, así como obtener información básica del perfil del Usuario final de manera interoperable y similar al protocolo REST. [15].

*OpenID Connect* es uno de los protocolos de tipo *Single Sign-On* más utilizados para delegar la autenticación. También tiene un formato simple, por lo que ha ganado popularidad y es soportado por grandes empresas como Google, IBM, Microsoft, Amazon y PayPal [10]. La nueva versión es compatible con *API* y puede ser usado por aplicaciones nativas y móviles. También define mecanismos opcionales más robustos para firmas y cifrados. [15]

Este protocolo está basado en *OAuth 2.0* por lo que tiene todas las ventajas de este protocolo. Sin embargo, lo extiende ya que ofrece facilidades para obtener más información de la identidad de los usuarios eficientemente. Permite un flujo de información adicional que genera un *id-token* que contiene datos del usuario. De esta forma las aplicaciones no solo tienen acceso a los permisos de los usuarios, sino también obtienen información sobre la identidad de los usuarios. [15][1]

## 1.4. Otros protocolos de autenticación

### 1.4.1. SAML

Lenguaje de Marcado para Confirmaciones de Seguridad, conocido como SAML (en inglés: *The Security Assertion Markup Language*) es un marco de trabajo que permite expresar asertos[1] acerca de la identidad, los atributos y las autorizaciones de un sujeto con el objetivo de facilitar las relaciones entre distintas empresas, así como las relaciones de estas con sus usuarios. Este marco de trabajo permite a las compañías crear identidades federadas, lo cual les facilita las tareas de gestión de perfiles, autenticación y autorización de usuarios. El caso típico de uso es el de *Single-Sign-On* (SSO), que permite a los usuarios acceder a diversos sitios en la federación con una única autenticación. [16]

[1]Los asertos definen las afirmaciones de seguridad de una entidad dentro de un sistema. Estas afirmaciones pueden ser de tres tipos: asertos de autenticación, de atributos y de decisiones de autorización.

SAML y OpenID Connect son protocolos de identificación, diseñados para autenticar a los usuarios. También proporcionan datos de identidad para el control de acceso y como método de comunicación para la identidad de un usuario.

SAML durante muchos años ha proporcionado un medio seguro de intercambio de datos de identidad, por lo que muchas organizaciones confían en él. También es muy rico en funciones y cubre una amplia gama de requisitos de identidad.

OIDC, al ser más nuevo y en se encuentra desarrollo, todavía está rezagado con respecto a SAML en términos de características. Sin embargo, para muchas aplicaciones donde solo se necesita un requisito simple para los datos de identidad básicos, particularmente en el espacio del consumidor, OIDC es muy útil, ya que es mucho más fácil de usar que SAML y no requiere el manejo pesado de XML que requiere SAML.

Actualmente, SAML se usa prinipalmente para la identificacion de ciudadanos del gobierno y la autenticación empresarial. Sin embargo, esto está comenzando a cambiar, con sistemas más modernos que utilizan OIDC en lugar de SAML. Esto se debe a que OIDC permite un procesamiento de datos más liviano que SAML, utilizando tokens JSON (token de ID) en lugar de XML. OIDC es ideal para usar con aplicaciones móviles y aplicaciones web de una sola página, donde el uso de SAML sería complicado. [17]

OIDC se adapta a las condiciones del Nodo Central de la Universidad de La Habana ya que se cuentan con pocos recursos, por lo que es necesario un procesamiento ligero de información.

### 1.4.2. Kerberos

Kerberos es una conexión de software que se emplea en una red grande para establecer la identidad declarada de un usuario. Utiliza una combinación de encriptación y Bases de Datos distribuidas de tal forma que un usuario pueda registrarse y comenzar una sesión desde cualquier computadora localizada en la red mediante la obtención de tickets para servicios de un servidor especial conocido como TGS (servidor despachador de tickets); cada ticket contiene información para identificar al usuario o servicio encriptada con la clave privada para el servicio. Como sólo Kerberos y el servicio conocen dicha clave, se considera que el mensaje está genuinamente originado en la fuente y que no fue adulterado en el transporte del mismo. El ticket otorgado por el TGS contiene una nueva clave de sesión que solo conoce el cliente y el servicio afectado. Esta clave será utilizada para encriptar las transacciones que ocurren durante la sesión. Una de las ventajas es que el ticket tiene un tiempo de vida específico, y una vez que éste expira, debe solicitarse un nuevo ticket al TGS para poder seguir utilizando el servicio. Para cada servicio se requiere un ticket distinto. Otra ventaja es que el usuario no debe reingresar la password cada vez que requiere un servicio,

porque si el ticket TGS no expiró puede reusarlo para pedir otro ticket de servicio deseado. Por este motivo, el tiempo de vida del ticket TGS deberá ser mayor que el tiempo de vida del ticket de servicio.

## 1.5. LDAP

El Protocolo Ligero de Acceso a Directorios (en inglés: *Lightweight Directory Access Protocol*, también conocido por sus siglas de LDAP) es un conjunto de protocolos de licencia abierta que son utilizados para acceder a la información que está almacenada de forma centralizada en una red. Este protocolo se utiliza a nivel de aplicación para acceder a los servicios de directorio remoto. [18]

LDAP está basado en estándares implementados sobre TCP/IP. Permite a los clientes interactuar directamente con los servidores de los directorios: almacenar y consultar información, buscar datos filtrados, autenticar usuarios, entre otros.

Este protocolo es utilizado actualmente por muchos sistemas que apuestan por el software libre al utilizar distribuciones de Linux para ejercer las funciones propias de un directorio activo en el que se gestionarán las credenciales y permisos de los usuarios y estaciones de trabajo en redes LAN corporativas en conexiones cliente/servidor.

Un directorio remoto es un conjunto de objetos que están organizados de forma jerárquica, tales como: nombre, claves, direcciones, etc. Estos objetos estarán disponibles para una serie de clientes conectados mediante una red, normalmente interna o LAN, y proporcionarán las identidades y permisos para esos usuarios que la utilicen.

LDAP está basado en el protocolo X.500 para compartir directorios, y contiene esta información de forma jerarquizada y mediante categorías para proporcionarnos una estructura intuitiva desde el punto de vista de la gestión por parte de los administradores.

Estos directorios se utilizan generalmente para contener información virtual de usuarios, para que otros usuarios accedan y dispongan de información acerca de los contactos que están aquí almacenados. Además es capaz de comunicarse de forma remota con otros directorios LDAP situados en servidores que pueden estar en el otro lado del mundo para acceder a la información disponible. De esta forma se crea una base de datos de información descentralizada y completamente accesible.

El sistema de autenticación vigente en el Nodo Central verifica sus usuarios con dos sistemas implementados con LDAP. Este protocolo se adapta a las necesidades y condiciones actuales de la Universidad ya que es *Open Source* (OSS o código abierto).

## 1.6. Active Directory

Directorio Activo (en inglés: Active Directory, conocido también por sus siglas AD) es un servicio de directorios desarrolado por Microsoft que permite almacenar información como usuarios y dispositivos en una base de datos centralizada y jerárquica. AD brinda servicios como autenticación, políticas de acceso y administración de grupos.

Active Directory almacena información sobre objetos en la red y hace que esta información sea fácil de encontrar y usar para administradores y usuarios. Active Directory utiliza un almacén de datos estructurados como base para una organización lógica y jerárquica de la información del directorio.

La seguridad está integrada con Active Directory a través de la autenticación de inicio de sesión y el control de acceso a los objetos del directorio. Con un solo inicio de sesión en la red, los administradores pueden administrar los datos del directorio y la organización en toda su red, y los usuarios autorizados de la red pueden acceder a los recursos en cualquier lugar de la red. La administración basada en políticas facilita la gestión incluso de la red más compleja. [19]

| **Comparación entre LDAP y AD** | | |
|---|---|---|
| | **LDAP** | **AD** |
| **Nombre Completo** | Protocolo Ligero de Acceso a Directorios | Directorio Activo |
| **Función** | Protocolo | Proveedor de servicios de directorios |
| **Standard** | Código Abierto | Propietario |
| **Sistemas Soportados** | Multiplatforma: Windows, Linux, macos | Para aplicaciones y usuarios de Windows |
| **Uso principal** | Consultar y modificar elementos en proveedores de servicios de directorio | Proveer autenticación, políticas, administración de grupos y usuarios, y muchos otros servicios en forma de una base de datos de directorio |

## 1.7. Okta

## 1.8. Gluu

Gluu es una plataforma de código abierto gratuita que proporciona a las organizaciones un servicio de autenticación y autorización para las aplicaciones web y móvil.

Permite configurar SSO (Single Sign-On) en aplicaciones que tengan soporte para OpenID Connect, SAML o CAS para identidades federadas.

## 1.9. Auth0

Auth0 es una plataforma en la nube que ofrece la autenticación y la autorización como un servicio. Auth0 dispone de herramientas para simplificar la autenticación de las aplicaciones y APIs ya que hace uso de estándares como OAuth2.0, OpenID Connect, SAML 2.0, JSON Web Token o WS-Federation, ofreciendo SSO (Single Sign-On) a entornos empresariales

## 1.10. Keycloak

Keycloak es un software de código abierto que permite el *Single Sign-On* o Inicio de Sesión Único con *Identity Management* y *Access Management* para aplicaciones y servicios modernos. Esta herramienta facilita la protección de aplicaciones y servicios con poca o ninguna codificación. Un Proveedor de identidad (en inglés: *Identity Provider*, también conocido por sus siglas IdP), permite que una aplicación (a menudo llamada *Service Provider* o SP) delegue su autenticación. [20]

Este software está escrito en Java y es compatible de forma predeterminada con los protocolos de federación de identidad SAML v2 y OpenID Connect (OIDC) / OAuth2. Está bajo licencia de Apache y es mantenido por Red Hat. [20]

### 1.10.1. Características

Los usuarios se autentican en Keycloak en lugar de hacerlo en las aplicaciones. Esto significa que no es necesario que cada aplicación tenga un formulario de inicio de sesión, autentique a los usuarios o almacene sus datos. Una vez entren en Keycloak, los usuarios no tendrán que iniciar sesión en las demás aplicaciones conectadas al software.

Lo mismo sucede cuando un usuario cierra sesión. Keycloak ofrece cierre de sesión único, lo cual significa que los usuarios solo tienen que desconectarse en una de las aplicaciones para salir de su cuenta en el resto.

Otra prestación de Keycloak son las federaciones de usuarios, que facilitan la compatibilidad con LDAP y otros servidores de directorios activos. También admite la implementación de servicios propios para usuarios guardados en otros tipos de almacenamientos como en bases de datos relacionales.

Keycloak ofrece como herramienta una consola de administración de cuentas, a través de la cual los usuarios pueden administrar sus propias cuentas. Pueden actua-

lizar su perfil, cambiar sus contraseñas y configurar la autenticación en dos pasos. También pueden administrar sus sesiones y visualizar el historial de su cuenta.

Otra característica es que es una herramienta extensible porque permite la eliminación, adición y modificación de las bases de datos de usuarios, los métodos de autenticación y los protocolos. Está basada en protocolos estándares y soportan OpenID Connect, OAuth 2.0 y SAML. [20]

Keycloak facilita añadir la autenticación y un servicio seguro a aplicaciones. Permite que los desarrolladores se centren en la funcionalidad empresarial al no tener que preocuparse por los aspectos de seguridad de la autenticación. También posibilita la unificación de los métodos de autenticación de distintas aplicaciones sin modificarlas.

Gluu es otra de las tecnologías que tienen prestaciones y ventajas similares a Keycloak. Es un servicio *Open Source* que soporta *SAML*, *OpenID Connect, SSO* y *OAuth 2.0*. Sin embargo Gluu es un sistema que requiere de 8 GB de RAM y 40 GB de espacio en disco, mientras que Keycloak solo necesita de 512 Mb de RAM y 1 GB de disco. Por ello Keycloak se ajusta más a los recursos que se tienen en la Universidad de La Habana.

### 1.10.2. JSON Web Token

*JSON Web Token* (abreviado JWT) es un estándar abierto basado en JSON propuesto por IETF (RFC 7519) para la creación de tokens de acceso que permiten la propagación de identidad y privilegios [21].

Esta tecnología define una forma compacta y autónoma para transmitir de forma segura información entre las partes como un objeto JSON. Esta información es verificada y confiable ya que se encuentra firmada digitalmente. Los JWT se pueden firmar usando un secreto (con el algoritmo HMAC) o un par de claves públicas / privadas usando RSA.

JSON Web Token es un método compacto y autónomo para transmitir información, se basa en una cadena de texto que tiene 3 partes (Header, payload, signature) codificadas en Base64, separadas por un punto que es estregado a los clientes de una API como llave de acceso. Keycloak utiliza JWT para transmitir una llave secreta de acceso a usuarios con privilegios. [22]

### 1.10.3. REST API

Se utiliza API REST como medio de prueba para ejecutar peticiones HTTP que posteriormente serán aseguradas con un Json Web Token emitido por Keycloak. La arquitectura REST enfoca a todo lo que lo conforma como un recurso. Los servicios web REST son livianos, altamente escalables y mantenibles. Se usan muy comúnmente

para el intercambio de información, es el estándar más lógico, eficiente y generalizado en la creación de API para servicios de internet.

El siguiente esquema muestra como se realiza el flujo de información para autenticarse a través de Keycloak:

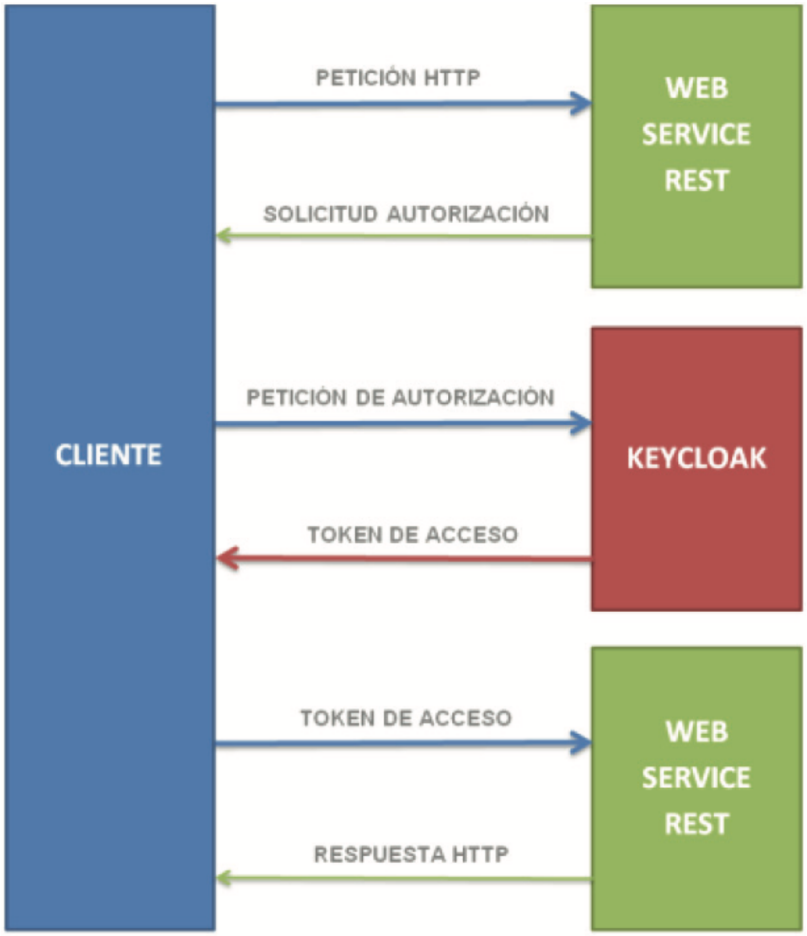


Figura 1.1

Este flujo asegurar que solo los usuarios a los que se les otorgó el acceso al servicio web puedan consumir dicha API. De esta forma se evita el acceso no autorizado a información que se transmite bajo un protocolo no seguro como lo es el protocolo HTTP. [22]

# Capítulo 2

# Propuesta de Sistema Central de Autenticación

En el siguiente capítulo (...)

## Hipótesis

La Universidad de la Habana cuenta con varios sistemas en la red donde se utiliza el método tradicional de usuario/contraseña como mecanismo de seguridad para acceder a diversas aplicaciones, este proceso de autenticación se hace muy complejo al tener que acceder a cada uno de ellos de forma independiente. Con cada servicio nuevo se debe crear un sistema de autenticación que garantice la seguridad de sus datos y se deben hospedar la información de los usuarios repetidas veces, lo cual utiliza una mayor cantidad de recursos y es más propenso fallas.

Para eliminar estas dificultades se realiza un sistema central de autenticación que se basa en el método Inicio de Sesión Único. Esta propuesta tiene como objetivo un aumento en la productividad, tener mayor facilidad de acceso a los recursos, funciones de autenticación a través de una única plataforma, una administración sencilla de credenciales y sobre todo garantizar un aumento de la seguridad. Mediante este servicio el usuario podrá registrarse en el sistema una sola vez, con lo cual podrá acceder a todos los recursos sin tener que volver a autenticarse.

### Requisitos del Software:

- Unificar las distintas fuentes de datos.
- Que el usuario inicie sesión y, hasta que cierre sesión, sea capaz de realizar operaciones sin tener que volver a introducir credenciales.

- El servicio ofrecido al cliente debe permitir que el usuario extienda la sesión una vez pasado el tiempo de expiración de la misma sin que volver a introducir credenciales.
- Reconocer la identidad de los usuarios durante el proceso de autenticación para garantizar un adecuado control de acceso a recursos del sistema.
- Proteger los recursos del sistema, permitiendo que estos sean solamente usados por aquellos usuarios a los que se les ha concedido autorización.
- Que el usuario inicie sesión con dos únicos campos: nombre de usuario y contraseña. Este requisito será suficiente para garantizar la interoperabilidad del sistema, que debe ser capaz de generar un objeto encriptado con toda la información relativa a dicho usuario y viajar por la red de comunicaciones entre las distintas entidades.
- La respuesta obtenida al iniciar sesión de forma exitosa debe ser un objeto que le dé portabilidad y reusabilidad al software.
- En caso de obtener un inicio de sesión erróneo, retornar un error
- Garantizar el control de errores y excepciones.
- La evaluación de permisos de acceso

El software consta de varias etapas:

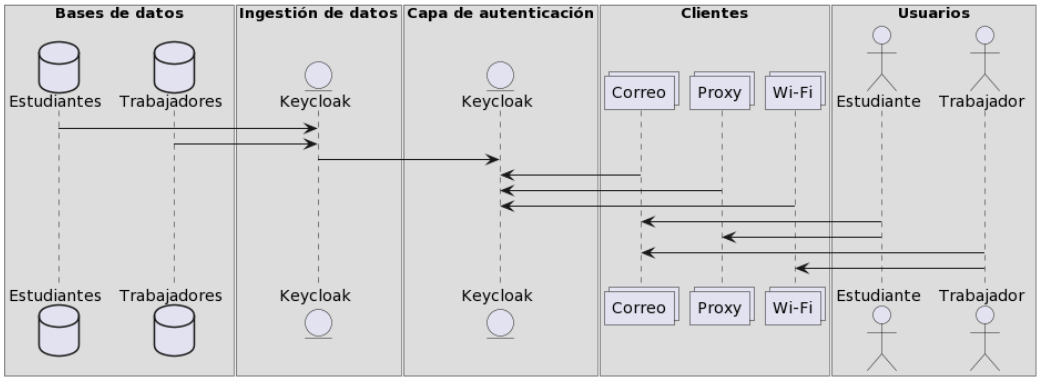


Figura 2.1

## Bases de Datos

Actualmente los estudiantes de toda la Universidad de La Habana al matricularse se inscriben en secretaría en el Sistema de Gestión de la Nueva Universidad (SIGENU). En este sistema se almacenan todos los datos de los alumnos: datos personales, proveniencia, notas, grados.

Por otra parte, los datos de los profesores y resto de trabajadores de la Universidad son guardados en bases de datos de ASSETS, software contratado. Cada unidad presupuestada de la Universidad tiene su propio ASSET donde almacena los datos de recursos humanos y del inventario.

El Nodo Central es el responsable de todas las comunicaciones de la Universidad de La Habana. Presta servicios a todas las facultades, desde las que se encuentran en la Colina, hasta facultades externas como Economía, el Jardín Botánico y la Quinta de los Molinos. Cada facultad tiene su sistema independiente donde gestiona a todo su personal que responde a sus necesidades.

La información es almacenada en varias bases de datos, lo cual hace difícil la gestión de todos los usuarios de la Universidad. Crear un sistema único al cual todas las facultades pudieran acceder de forma remota simplificaría el trabajo, no solo a los programadores , sino también a los usuarios. Por ejemplo, cada estudiante podría matricularse en su propia facultad y se evitarían las aglomeraciones y la confusión creada todos los años.

Por otra parte sería complicado cambiar el software utilizado por toda. Como se la institución. Como mencionó anteriormente, algunas facultades se encuentran alejadas de La Colina, lo cual dificulta la comunicación y la instalación de nuevos sistemas. También se debe tener en cuenta que la experiencia del usuario sería diferente con un nuevo software. El personal trabajador de la Universidad es de diversas edades, por lo que podría ser complejo la adaptación a una nueva interfaz.

Por lo tanto en el presente trabajo se ha decidido no cambiar los sistemas. La unificación de las bases de datos de usuario de toda la Universidad de La Habana es un proceso complejo que no se encuentra en los objetivos del presente trabajo.

Por otra parte, se necesita un nombre de usuario único para cada trabajador y estudiante de la institución. Al utilizarse distintas fuentes de datos, no se puede garantizar que todas las bases de datos tengan las mismas estructuras. Sin embargo, todos los usuarios tienen una cuenta de correo que los identifica unívocamente, por lo que utilizar este campo como nombre de usuario garantiza que cada persona es identificado.

## Ingestión de Datos

La ingesta de datos es el proceso mediante el cual se introducen datos, de diferentes fuentes, estructura o características dentro de otros sistemas de almacenamiento o procesamiento de datos. [23]

Teniendo varias fuentes de datos, para garantizar una ingestión de datos exitosa, la solución debe ser capaz que leer información de distintas fuentes de datos.

Por otra parte el estado de un usuario puede cambiar y sus permisos de acceso pueden variar. Por ejemplo, a los estudiantes se les puede dar de baja o un trabajador puede terminar su contrato y por lo tanto se les debe quitar sus credenciales de forma inmediata. Un estudiante después de graduarse puede pasar a se trabajador de la institución, por lo que sus permisos de acceso deben ser cambiados. Otro caso sería cuando un estudiante o trabajador sale una licencia, deben suspenderse sus credenciales temporalmente hasta que la persona regrese.

Se necesita que el sistema de autenticación se actualice constantemente ya que las bases de datos cambian sus datos.

La Universidad se encuentra en constante cambio. Por ejemplo, en 2017 culminó el proceso de adscripción del Instituto Superior de Diseño (ISDI) a la Universidad de La Habana [24]. Para ello fue necesario incorporar todos los datos de los usuarios del ISDI al sistema de autenticación vigente para otorgarles las credenciales correspondientes. Por ello, el nuevo sistema debe ser capaz de admitir nuevas fuentes de datos de forma sencilla en un futuro.

En el nodo central, la ingesta de datos se hace a través de LDAP. (explicar brevemente para qué se usa LDAP en el nodo y cómo ingesta los datos)

## Capa de Autenticación

La autenticación es un área clave en la seguridad de la información. En la modernidad los usuarios necesitan acceder a muchos servicios digitales imprescindibles para su vida cotidiana. Las contraseñas basadas en caracteres alfanuméricos han sido las más comunes en todo tipo de sistemas por su fácil implementación. [**rodriguez2018seguridad**]

El proceso de autenticación consiste en la verificación de la identidad de un usuario o una entidad [25]. Los usuarios de la institución utilizan diariamente distintos servicios, que van desde acceder a Internet por la red WI-FI o por proxy, hasta servicios web como EVEA y correo. Cada servicio tiene su propio sistema de autenticación.

Resulta engorroso crearse una cuenta en cada uno de los sistemas y autenticarse cada vez que uno acceda a un sitio nuevo. La creación de un sistema en el cual todos los usuarios de la Universidad puedan autenticarse con su correo de la institución y

su contraseña facilitaría la interacción diaria con los servicios ofrecidos por el Nodo Central.

Para ello se debe garantizar la seguridad de las cuentas. Por ejemplo, los estudiantes utilizan la máquinas de los laboratorios y se autentican con su cuenta personal. Es común que las personas olviden cerrar sus sesiones, por lo que la cuenta debe expirar al pasar de un tiempo prudencial, de lo contrario, su información puede ser utilizada por otra persona.

Se utilizará la autenticación basada en tokens ya que permite

En este caso la capa de autenticación es la encargada de recibir las peticiones de los distintos clientes y verificar la identidad de un usuario a partir de la información recopilada por la Ingestión de datos.

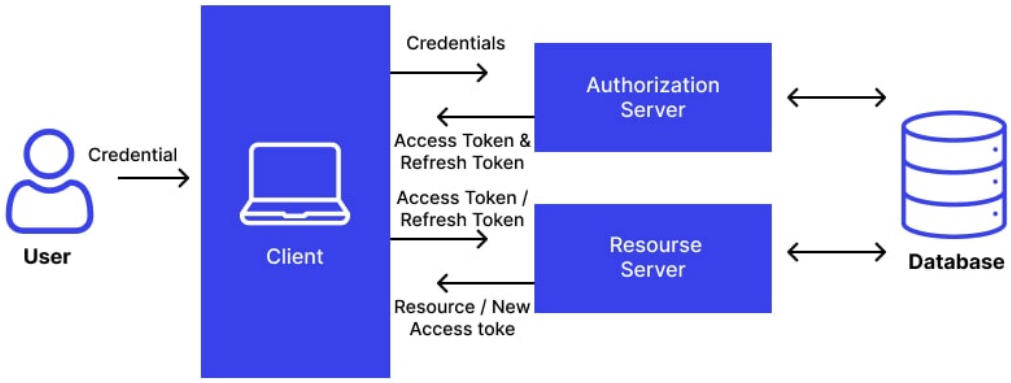


Figura 2.2

## Clientes

El Nodo Central brinda infraestructura a los servicios de la Universidad vitales para el correcto funcionamiento de la institución. Entre responsabilidades se encuentran la gestión de correo, el acceso a Internet, las cuentas de usuarios y en el caso de los profesores gestión de cuentas de VPN. También tiene como clientes todos los servicios web de la casa de altos estudios, entres ellos el sitio oficial de la Universidad https://www.uh.cu/, los sistemas utilizados en recursos humanos, en los departamentos de contabilidad

Cada cliente tiene su propio sistema de autenticación, lo cual no es eficiente ya que utiliza una mayor cantidad de recursos porque cada uno tiene su registro de usuario en bases de datos individuales. También crea brechas de seguridad ya que no todos

cumplen los estándares que deberían y existe más información en la red expuesta a ataques externos.

El software debe ser capaz de admitir nuevos clientes fácilmente. Uno de los clientes más importantes es el Entorno Virtual de Enseñanza y Aprendizaje, más conocido como EVEA, creado en 2018. Este es una plataforma informática encargada de orientar la comunicación pedagógica entre los universitarios que intervienen en el proceso educativo. También, tiene la misión de crear espacios o comunidades organizadas en torno al estudio. [26]

Este entorno virtual representó un apoyo para el sistema universitario cuando los estudiantes no podían asistir a sus facultades debido a la pandemia provocada por la covid-19 en diciembre del 2019 [27] . En la primera mitad del año 2020 la cantidad de usuario aumentó considerablemente en poco tiempo. El sistema era lento y tenía muchas fallas. Uno de los mayores retos fue lograr registrar a todos sus usuarios y que estos se pudiera autenticar. De haber existido un sistema que ya autenticara a todos los estudiantes y profesores de la Universidad, los programadores habrían tenido un problema menos a la hora de modificar la plataforma para adaptarla al nuevo uso que se le daría.

# Usuarios

**Estructura del Cap 2**

- Descartar otras opciones
- Describir brevemente por qué Keycloak (despues en mas detalle cada componente)
- Propuesta: Diagrama
- - qué es la etapa/componente - cuál es problema concreto - propuesta de solución
- **Bases de datos:**
  qué es: dónde y en qué formato se almacena la información
  problema: Hay muchas facultades, es dificil cambiar como se introducen los datos.
  propuesta: no cambiar nada, tiene que tener la base de datos una columna con valores únicos que pueda ser usado como nombre de usuario

- **Ingestión de datos**
  Qué es: definición problema: - Aceptar varias fuentes de datos
  - Actualizarse live. por si un estudiante se gradúa, un trabajador culmina su contrato, un trabajador empieza un contrato, un estudiante comienza primer año, estudiante pasa a ser trabajador, estudiante regresa de licencia.
  - Contraseña ?????
  - Añadir fuentes de datos dinámicamente
  propuesta: keycloak, why?

- **Capa de autenticación**
  Qué es: definición
  problema:
  - unificar
  - seguridad, computadoras compartidas en el laboratorio
  - refresh
  - logout,
  propuesta: keycloak (ya se está usando para ingestión de datos mantenimiento, simplicidad, facilidad de aprendizaje, menos recursos)

- **Clientes**
  Qué es: definición, en la UH: correo, evea
  Problema: todos distintos, nada en común, seguridad del sistema de autenticación
  Propuesta: que los servicios utilicen el API de keycloak para autenticar
  tu tesis no modifica los clientes existentes, pero ilustra cómo hacerlo con un cliente de ejemplo desarrollado en Python.

- **Usuarios**
  Qué es: personas jóvenes, personas mayores
  Problema: confuso

  Los usuarios se ven obligados a tener más de unas credenciales lo cual crea confusión. Frecuentemente se olvidan los nombres de usuarios o contraseñas o se utilizan las mismas credenciales en varios sitios, lo cual es una mala práctica desde el punto de vista de la seguridad.
  Propuesta: sus servicios no van a cambiar, pero única contraseña, seguridad añadida sin cambiar la experiencia de usuario

# Capítulo 3

# Detalles de Implementación y Experimentos

## Elección de herramientas

En este capítulo se toman decisiones claves respecto a qué herramientas utilizar para lidiar con problemas claves que surgen al autenticarse

### 3.0.1. Keycloak

# Conclusiones

Conclusiones

- El rendimiento del nuevo sistema es más eficiente ya que en cada iteración añade solo los nuevos usuarios en lugar de borrar y recrear cada noche todos sus usuarios. - El tiempo de activación de un usuario (el tiempo que transcurre entre que un usuario es dado de alta por recursos humanas y que el usuario tenga acceso a todos los servicios provistos por el Nodo Central como correo electrónico y acceso a internet) es más corto. - Sistema de autenticación más fiable

# Recomendaciones

Recomendaciones

# Bibliografía


[1] R. Kutera y W. Gryncewicz, «Single sign on as an effective way of managing user identity in distributed web systems. The ActGo-Gate project case study,» Informatyka Ekonomiczna, n.º 2 (40), 2016 (vid. págs. 1, 6).

[2] Microsoft. (2022). «What is single sign-on in Azure Active Directory?» Dirección: `https://learn.microsoft.com/es-es/azure/active-directory/manage-apps/what-is-single-sign-on` (vid. pág. 4).

[3] V. Radha y D. H. Reddy, «A survey on single sign-on techniques,» Procedia Technology, vol. 4, págs. 134-139, 2012 (vid. pág. 5).

[4] Google. (2022). «About SSO,» dirección: `https://support.google.com/` (vid. pág. 5).

[5] Auth0. (2022). «Single Sign-On,» dirección: `https://auth0.com/docs/authenticate/single-sign-on` (vid. pág. 5).

[6] ——, Access Tokens — auth0.com, `https://auth0.com/docs/secure/tokens/access-tokens`, 2022 (vid. pág. 5).

[7] A. Banerjee y M. Hasan, «Token-Based Authentication Techniques on Open Source Cloud Platforms,» Sistemas y Telemática, vol. 16, n.º 47, 2018 (vid. pág. 5).

[8] S. Cantor, I. J. Kemp, N. R. Philpott y E. Maler, «Assertions and protocols for the oasis security assertion markup language,» OASIS Standard (March 2005), págs. 1-86, 2005 (vid. pág. 5).

[9] D. Recordon y D. Reed, «OpenID 2.0: a platform for user-centric identity management,» en Proceedings of the second ACM workshop on Digital identity management, 2006, págs. 11-16 (vid. pág. 5).

[10] C. Mainka, V. Mladenov, J. Schwenk y T. Wich, «Sok: Single sign-on security—an evaluation of openid connect,» en 2017 IEEE European Symposium on Security and P IEEE, 2017, págs. 251-266 (vid. págs. 5, 6).

[11] D. Hardt, «The OAuth 2.0 authorization framework,» inf. téc., 2012 (vid. pág. 5).

[12] J. Richer y A. Sanso, OAuth 2 in action. Simon y Schuster, 2017 (vid. pág. 5).

[13] K. Zheng y W. Jiang, «A token authentication solution for hadoop based on kerberos pre-authentication,» en 2014 International Conference on Data Science and Advanced A IEEE, 2014, págs. 354-360 (vid. pág. 5).

[14] O. Ethelbert, F. F. Moghaddam, P. Wieder y R. Yahyapour, «A JSON token-based authentication and access management schema for cloud SaaS applications,» en 2017 IEEE 5th International Conference on Future Internet of Things and Cloud (F IEEE, 2017, págs. 47-53 (vid. pág. 6).

[15] OpenID. (2022). «Welcome to OpenID Connect,» dirección: `https://openid.net/connect/` (vid. pág. 6).

[16] R. Sánchez Guerrero, «Estudio y puesta en marcha de una infraestructura de gestión de identidad federada basada en SAML 2.0,» Tesis de mtría., 2009 (vid. pág. 6).

[17] N. Naik y P. Jenkins, «Securing digital identities in the cloud by selecting an apposite Federated Identity Management from SAML, OAuth and OpenID Connect,» en 2017 11th International Conference on Research Challenges in Information Science ( IEEE, 2017, págs. 163-174 (vid. pág. 7).

[18] LDAP. (2022). «LDAP,» dirección: `https://ldap.com/` (vid. pág. 8).

[19] Microsoft. (2022). «Active Directory Domain Services Overview,» dirección: `https://learn.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/virtual-dc/active-directory-domain-services-overview` (vid. pág. 9).

[20] Keycloak Documentation, `https://www.keycloak.org/documentation` (vid. págs. 10, 11).

[21] J. Bradley, N. Sakimura y M. B. Jones, JSON Web Token (JWT), mayo de 2015. dirección: `https://www.rfc-editor.org/rfc/rfc7519.txt` (vid. pág. 11).

[22] C. Muyón y F. Montaluisa, «Métodos de seguridad de la información para proteger la comunicación y los datos de servicios web REST en peticiones HTTP utilizando JSON Web Token y Keycloak Red Hat Single Sign On,» Revista Ibérica de Sistemas e Tecnologias de Informação, n.º E29, págs. 198-213, 2020 (vid. págs. 11, 12).

[23] C. Fernández Pérez, «Aplicación web para la gestión de procesos de ingesta de datos en entorno de BI/Big Data,» 2020 (vid. pág. 16).

[24] I. S. de Diseño. (2022). «Historia del Instituto,» dirección: `https://www.isdi.co.cu/index.php/site/historia/Historia#:~:text=Al%20culminar%20el%20proceso%20de,mes%20de%20enero%20de%202017.` (vid. pág. 16).

[25] J. L. Teherán Sierra y col., «Mecanismo de autenticación y control de acceso para Software-Defined Networking-SDN,» 2014 (vid. pág. 16).

[26] G. Mok Rodríguez, E. Carmona Fernández, C. A. García Santoya y M. Embarek El-Bah Valdés, Evea: Una puerta hacia otra forma de Estudio, abr. de 2022. dirección: `http://www.cubadebate.cu/especiales/2022/04/02/evea-una-puerta-hacia-otra-forma-de-estudio/` (vid. pág. 17).

[27] R. Ferrer, «Pandemia por COVID-19: el mayor reto de la historia del intensivismo,» Medicina intensiva, vol. 44, n.º 6, pág. 323, 2020 (vid. pág. 18).