Universidad Tecnológica de La Habana “José Antonio Echeverría”

Facultad de Ingeniería Informática

Aplicación web de código abierto para la gestión y autenticación de usuarios basada en Directorio Activo.

Autor: Carlos Daniel Vilaseca Illnait
Tutores: Dra. C. Raisa Socorro Llanes,
Dra. C. Lisandra Bravo Ilisastigui

La Habana, Cuba,
Septiembre 2024

Resumen

El documento presenta una solución propuesta para la gestión y autenticación de usuarios basada en Directorio Activo a través de LDAP. Se abordan temas como la historia detrás de esta propuesta, la selección del cliente LDAP que permita la comunicación con el directorio desde la aplicación, cómo se logra que la solución sea configurable y personalizable, las pruebas realizadas y los desafíos encontrados.

Palabras clave: gestión de usuarios, código abierto, Directorio Activo, LDAP, personalización.

Abstract

The document presents a proposed solution for user management and authentication based on Active Directory through LDAP. It addresses topics such as the history behind this proposal, the selection of the LDAP client that allows communication with the directory from the application, how to make the solution configurable and customizable, the tests performed and the challenges encountered.

Keywords: user management, open source, Active Directory , LDAP, personalization.

Índice

Índice de tablas

Índice de figuras

Introducción

La gestión de usuarios y la autenticación en entornos de red son procesos esenciales en cualquier organización [1], [2], [3]. Los directorios activos desempeñan un papel clave al centralizar y asegurar el control de accesos a los diferentes servicios y recursos empresariales [3]. Actualmente, estas herramientas son ampliamente utilizadas en empresas de diversos tamaños para gestionar usuarios, grupos y unidades organizativas de manera unificada (Figura 32).

Un Directorio Activo (AD por sus siglas en inglés) es una base de datos jerárquica utilizada para almacenar y gestionar información sobre los recursos de la red, como usuarios, dispositivos y servicios. Proporciona una estructura centralizada que permite a los administradores controlar permisos y acceso a los recursos de manera segura y eficiente [4], [5], [18]. En este contexto, el protocolo LDAP (Lightweight Directory Access Protocol) se utiliza para interactuar con los directorios activos, facilitando la búsqueda, consulta y modificación de la información almacenada en ellos [4], [6], [7], [8], [9], [10].

Cada empresa establece sus propias políticas y controles de seguridad, los cuales influyen en cómo se debe gestionar la información y la estructura organizativa mediante un AD. La implementación de un AD debe, por tanto, adaptarse a los requerimientos específicos de cada entidad, presentando un

desafío cuando se buscan soluciones que armonicen con sus necesidades y regulaciones de seguridad [3], [4], [5].

Existen dos grandes grupos de directorios activos en el mercado: los de pago y los de código abierto. Las soluciones comerciales, como las ofrecidas por Microsoft, destacan por su facilidad de uso y alto nivel de integración, pero también generan una dependencia tecnológica significativa y pueden no ser viables para organizaciones con presupuestos limitados . En contraste, los directorios activos de código abierto eliminan la necesidad de costosas licencias y ofrecen independencia tecnológica, aunque suelen ser más complejos de implementar y mantener [1], [7], [17].

A pesar de las ventajas que ofrecen las soluciones de software libre y código abierto (SLCA), estas suelen presentar desafíos significativos en términos de personalización y simplicidad. Su enfoque en satisfacer necesidades específicas puede generar complejidades técnicas que demandan recursos especializados para su personalización y mantenimiento. La falta de flexibilidad para adaptarse a los requerimientos particulares de cada empresa puede comprometer tanto la seguridad como la funcionalidad organizacional. Además, las restricciones arquitectónicas inherentes y los diseños originales que no consideraron la personalización pueden hacer que estas herramientas sean menos intuitivas y más difíciles de ajustar [31], [32], [33].

A partir de esta situación problemática se identifica como problema a resolver: Las soluciones de AD de código abierto y libres (SLCA)

presentan un nivel insuficiente de adaptabilidad y facilidad de uso, lo que dificulta su implementación y personalización en entornos específicos. Para solucionar el problema se tiene como objeto de estudio los servicios de AD y las herramientas asociadas para su administración, con un enfoque particular en su integración y gestión mediante el protocolo LDAP. El campo de acción se delimita a los sistemas de AD y las herramientas de gestión de tipo SLCA basadas en LDAP.

Como hipótesis se plantea que desarrollar una herramienta de gestión de AD de código abierto que ofrezca un mayor nivel de personalización a través de archivos de configuración permitirá mejorar la adaptabilidad y facilidad de uso, sin comprometer la seguridad y estabilidad, en comparación con las soluciones de SLCA existentes.

Para demostrar esta hipótesis se plantea como objetivo general crear una consola de administración de código abierto para AD que demuestre una mejora en la personalización y facilidad de uso en comparación con las herramientas de gestión existentes.

A partir de este objetivo general, se derivan los siguientes objetivos específicos y tareas:

1. Analizar los requisitos de la aplicación y la personalización del sistema:

    1.1. Documentar los requisitos funcionales y no funcionales que debe cumplir la aplicación.

1.2. Analizar diferentes casos de uso para identificar las opciones de personalización.

1.3. Documentar los requisitos de personalización, incluyendo la interfaz de usuario y ajustes de seguridad.

2. Seleccionar tecnologías adecuadas:

2.1. Evaluar diferentes clientes LDAP disponibles en el mercado, considerando factores como compatibilidad, rendimiento y facilidad de integración.

2.2. Elegir el cliente LDAP que mejor se alinee con los requisitos funcionales y no funcionales previamente definidos.

3. Seleccionar tecnologías adecuadas:

3.1. Evaluar diferentes clientes LDAP disponibles en el mercado, considerando factores como compatibilidad, rendimiento y facilidad de integración.

3.2. Elegir el cliente LDAP que mejor se alinee con los requisitos funcionales y no funcionales previamente definidos.

4. Implementar la arquitectura y funciones básicas de la aplicación:

4.1. Establecer la arquitectura base del proyecto y configurar el ambiente de desarrollo necesario.

4.2. Configurar el cliente LDAP seleccionado y las herramientas asociadas para iniciar el desarrollo.

4.3. Desarrollar mecanismos de autenticación que interactúen con el AD utilizando el cliente LDAP seleccionado.

4.4. Implementar funcionalidades críticas para la gestión de usuarios y grupos utilizando el cliente LDAP, incluyendo operaciones de lectura, eliminación y actualización.

5. Realizar pruebas para asegurar el correcto funcionamiento del sistema:

   5.1. Diseñar y ejecutar pruebas de integración que verifiquen el correcto funcionamiento del sistema en su conjunto, desde la autenticación hasta la gestión de recursos.

   5.2. Documentar los resultados de las pruebas y realizar los ajustes necesarios basados en los hallazgos.

   5.3. Extender el conjunto de pruebas de integración para abarcar nuevas funcionalidades y garantizar la estabilidad y compatibilidad del sistema ante cambios y actualizaciones futuras.incluyendo operaciones de lectura, eliminación y actualización.

6. Simplificar y documentar el proceso de despliegue:

   6.1. Identificar y documentar estrategias y herramientas que simplifiquen el proceso de instalación y configuración inicial de la aplicación.

6.2. Utilizar contenedores Docker para simplificar el proceso de puesta en marcha de la aplicación.

6.3. Elaborar documentación detallada del proceso de despliegue.

7. Desplegar documentación:

7.1. Crear y estructurar la documentación técnica que incluya la descripción del sistema, la arquitectura, y las guías de desarrollo.

7.2. Documentar referencias de API y/o archivos de configuracion que sean claras y accesibles.

7.3. Publicar la documentación en un sitio web accesible.

8. Desplegar demo:

8.1. Configurar el servidor donde se va a desplegar.

8.2. Configurar Docker para simplificar el despliegue.

8.3. Configurar CI/CD para despliegues automáticos.

8.4. Ejecutar el primer despliegue exitoso.

Como valor práctico con la realización de este trabajo se espera un diseño de software de una herramienta para la gestión de AD que sea personalizable y fácil de desplegar.

La estructura del informe se organiza de la siguiente manera:

Capítulo 1 Fundamentación teórica. En este capítulo se presenta una explicación teórica sobre los conceptos y tecnologías

fundamentales para la gestión de usuarios y AD. Se analizarán en profundidad temas como la importancia de la gestión de usuarios en entornos digitales, los principios de seguridad y autenticación, y el funcionamiento de los directorios activos y LDAP. También se abordarán las diferentes herramientas existentes para la gestión de AD, sus ventajas y limitaciones, y se establecerá el marco teórico que sustenta la propuesta de solución planteada en este trabajo.

Capítulo 2 Descripción de la propuesta de solución. Este capítulo aporta una explicación detallada sobre la propuesta de solución a los problemas identificados en la gestión de AD. Se describirá la arquitectura de la aplicación web de código abierto propuesta, sus funcionalidades principales, y cómo se abordarán los requisitos de personalización y simplicidad de despliegue. Además, se explicarán las decisiones de diseño y las tecnologías seleccionadas para el desarrollo de la herramienta, así como los beneficios esperados en términos de seguridad, estabilidad y facilidad de uso.

Capítulo 3 Validación de la propuesta de solución. Este capítulo está dedicado a la validación de la propuesta de solución mediante la realización de pruebas exhaustivas. Se diseñarán y ejecutarán pruebas de integración para verificar el correcto funcionamiento del sistema en su conjunto, desde la autenticación hasta la gestión de recursos. Los resultados de estas pruebas se documentarán y se realizarán los ajustes necesarios basados en los hallazgos.