



**DESARROLLO DE UN PRIVILEGED ACCESS MANAGEMENT
(PAM)**

MÁSTER UNIVERSITARIO EN CIBERSEGURIDAD

2021 - 2022

Nombre y Apellidos del alumno

Carlos Alfredo Camacho Guerrero

Nombre y Apellidos del tutor

David Rodríguez Galiano

Índice de Contenidos

1. Resumen Ejecutivo.....	6
1.1. Palabras Clave:.....	6
2. Abstract	7
2.1. Keywords:	7
3. Agradecimientos.....	8
4. Introducción	9
5. Objetivos.....	12
5.1.1. Objetivos General:	12
5.1.2. Objetivos específicos	12
6. Metodología.....	13
6.1. Tecnologías utilizadas para el desarrollo del PAM.....	15
6.1.1. Java	15
6.1.2. Librerías utilizadas para el acceso SSH.....	16
6.1.3. Framework Web.....	17
6.1.4. Spring Boot	17
6.1.5. Vaadin Flow.....	19
6.1.6. Contenedores de Software.....	20
6.2. Análisis y Diseño de la aplicación	22
6.2.1. Lista de requerimientos.....	22
6.2.2. Diagramas de clase.....	23
6.2.3. Bosquejos de la aplicación	24
6.3. Licenciamiento del proyecto	30
6.4. Desarrollo de la aplicación	30
7. Resultados.....	38
7.1. Publicación del código fuente	51
8. Conclusiones.....	52
9. Limitaciones y Líneas futuras.....	54
9.1. Limitaciones.....	54
9.2. Líneas futuras de Investigación.....	54
10. Glosario de términos.....	55
11. Referencias.....	57
12. Anexos.....	61

Índice de Imágenes

Imagen 1 - Diagrama Propuesto del Desarrollo (Elaboración Propia)	13
Imagen 2 - Componentes del Framework Spring (Spring, 2022).....	18
Imagen 3 - Vaadin Flow con los componentes instanciado en el servidor (Vaadin, 2022).....	20
Imagen 4 - Diferencia entre contenedores Docker y maquinas virtuales (AWS, 2022).....	21
Imagen 5 - Diagrama de clase (Elaboración propia)	24
Imagen 6 - Inicio de la Aplicación (Elaboración propia)	25
Imagen 7 - Listado de Terminales Disponibles (Elaboración propia)	26
Imagen 8 - Acceso terminal (Elaboración propia).....	26
Imagen 9 - Conexión Servidores (Elaboración propia)	27
Imagen 10 - Agregar, edición de conexiones (Elaboración propia)	27
Imagen 11 - Registro de conexiones (Elaboración propia).....	28
Imagen 12 - Registro de rotación de contraseñas (Elaboración propia)	28
Imagen 13 - Usuarios (Elaboración propia).....	29
Imagen 14 - Carpetas y Archivos del Proyecto (Elaboración propia)	31
Imagen 15 - Carpetas del Proyecto (Elaboración propia).....	34
Imagen 16 - Servicios definidos Docker Compose (Elaboración propia)	37
Imagen 17 - Acceso al Sistema (Elaboración propia)	38
Imagen 18 - Entrada del sistema (Elaboración propia).....	39
Imagen 19 - Listado de acceso perfil administrador (Elaboración propia)	40
Imagen 20 - Listado de accesos perfil usuario (Elaboración propia)	40
Imagen 21 - Selección de la terminal para el acceso. (Elaboración propia).....	41
Imagen 22 - Acceso a la terminal por el usuario. (Elaboración propia)	41
Imagen 23 - Registro de conexiones de los usuarios (Elaboración propia).....	42
Imagen 24 - Registro de los comandos y respuesta de la sesión. (Elaboración propia).....	43
Imagen 25 - Administración de terminales (Elaboración propia)	44
Imagen 26 - Ventana de edición de las terminales (Elaboración propia)	44

Imagen 27 - Registro de rotaciones de contraseña (Elaboración propia)	45
Imagen 28 - Gestión de los usuarios (Elaboración propia).....	46
Imagen 29 - Ventana de edición de los usuarios (Elaboración propia)	46
Imagen 30 - Vista Acerca De (Elaboración propia)	47
Imagen 31 - Acceso software utilizando teléfono inteligente (Elaboración propia)	48
Imagen 32 - Vista de inicio utilizando teléfono inteligente (Elaboración propia)	48
Imagen 33 - Menú de las funcionales utilizando teléfono inteligente (Elaboración propia).....	49
Imagen 34 - Vista de gestión de terminales utilizando teléfono inteligente (Elaboración propia).....	49
Imagen 35 - Actualización de terminales utilizando teléfonos inteligentes(Elaboración propia).....	50
Imagen 36 - Documentación del repositorio de control de versiones del proyecto (Elaboración propia).....	51

Índice de tablas

Tabla 1 - Requerimientos del software	22
Tabla 2 - Estructura del Proyecto	31
Tabla 3 - Componentes en la tabla src	35
Tabla 4 - Librerías y componentes destacados	36
Tabla 5 - Comando ejecución Docker Compose	51
Tabla 6 - Licencia AGPL V3	61
Tabla 8 - Archivo Docker para la creación de la imagen	62
Tabla 9 - Archivo Docker Compose	63

1. Resumen Ejecutivo

El trabajo final de máster tiene como objetivo desarrollar un sistema para la gestión de acceso privilegiado (PAM, por sus siglas en inglés, *Privileged Access Management*) vía una plataforma web. La aplicación permite administrar los usuarios y servidores, brindando acceso mediante el protocolo SSH vía Web, ocultando las credenciales de acceso al usuario final y aplicando políticas de rotado de contraseñas.

El PAM permite a los administradores de infraestructura mantener un control de los usuarios que acceden a los servidores o servicios sin exponer las credenciales y manteniendo un registro de toda la actividad realizada por el usuario para fines de auditoría.

1.1. Palabras Clave:

Privileged Access Management, PAM, Web, SSH.

2. Abstract

The objective of the final master's project is to develop a system for Privileged Access Management (PAM) using a Web platform. The application allows users and servers to be managed, providing access through the SSH protocol using the web, hiding access credentials from the end user, and applying password rotation policies.

PAM allows infrastructure administrators to keep track of users accessing servers or services without exposing credentials and by keeping a record of all user activity for auditing purposes.

2.1. Keywords:

Privileged Access Management, PAM, Web, SSH.

3. Agradecimientos.

A Dios, por la fortaleza en afrontar el reto de la maestría en conjunto con todas las demás responsabilidades. A mi familia, por la comprensión en el tiempo prestado, en especial a mi esposa Damarys. Al Prof. David Rodríguez, por la asesoría y los consejos brindados. A la Universidad Camilo José Cela, por brindarme la oportunidad de ser parte en su prestigioso programa de maestría. Al personal docente, por su entrega y dedicación durante todo el proceso. A mis compañeros, por la colaboración y amistad brindada durante toda la maestría.

4. Introducción

La gestión de acceso con privilegios, por sus siglas en inglés, PAM (Privileged Access Management), es un sistema que utilizan las organizaciones para controlar y supervisar los accesos privilegiados por parte de los usuarios y evitar el robo de las credenciales de los sistemas con acceso de administrador (CyberArk, 2022). En una organización, los usuarios con acceso de alto nivel de privilegios, es decir, acceso administrador, dentro de la confianza que es cedida para realizar sus funciones, existe el riesgo de las operaciones que pueden realizar (provocado por un error o mal intencionado) sean afectados los servicios de los sistemas que tienen acceso, evidenciando que es necesario una supervisión de las sesiones de trabajos por parte de los usuarios. De acuerdo con Haber y Rolls, un PAM es una subdisciplina dentro de un *framework* de gobernanza de identidad (Haber y Rolls, 2022). El cual puede ser implementando en conjunto a un Identity Access Management (IAM). Por definición, PAM es una metodología para asegurar, controlar, monitorear y administrar la actividad privilegiada de los recursos. Un PAM incluye múltiples componentes para administrar identidades privilegiadas, cuentas y credenciales y sus correspondientes contraseñas, certificados y claves. El objetivo de un PAM es reducir el riesgo al proporcionar acceso privilegiado solo a los usuarios y recursos que necesitan privilegios administrativos o de raíz para completar una tarea.

Los recursos gestionados bajo un PAM pueden abarcar cualquier recurso informático, desde un sistema operativo, aplicaciones, bases de datos, dispositivos de red, recursos en la nube, entre otros (Haber y Rolls, 2022). Las soluciones PAM brindan a las organizaciones las herramientas necesarias para el acceso privilegiado y proteger todos los activos independientemente de dónde se encuentren y, por lo general, se enfocan en los recursos críticos que contienen la información y la infraestructura más confidenciales (Garbis, Chapman, 2021).

De acuerdo con (Haber y Rolls, 2022), indican algunos de los componentes que son partes de una solución PAM son:

- **Almacenamiento de contraseñas:** la capacidad de una solución para almacenar de forma segura credenciales y cuentas en una caja fuerte de contraseñas para la recuperación manual, automática o mediante programación mediante una identidad.

- **Administración de contraseñas:** la capacidad de realizar funciones de administración en una contraseña asociada con una cuenta. Esto incluye cambios de contraseña utilizadas por usuarios, como cuentas de servicio para aplicaciones.
- **Gestión de sesiones:** la capacidad de la solución PAM para registrar la interacción del usuario o de la aplicación con un comando o una sesión remota, independientemente del protocolo de conexión, almacenando la actividad en la pantalla o mediante el teclado o los movimientos del mouse para futuros análisis y de esa forma detectar un mal uso o vulnerabilidad.
- **Gestión de privilegios:** la capacidad de monitorear, controlar y finalizar todas y cada una de las actividades privilegiadas que ocurren en un recurso, ya sea por parte del usuario o de una aplicación.
- **Gobernanza de gestión de acceso privilegiado (PAM):** la implementación de un PAM debe ir acompañada en las políticas y los procedimientos que requiere una organización para implementarla en las operaciones diarias. (Haber y Rolls, 2022) nos indican que la industria maneja una serie de siglas estándar para ayudar a agrupar y explicar los diferentes componentes que están incluidos en un PAM. Los proveedores rara vez otorgan licencias de forma individual y, por lo general, las organizaciones harán referencia a sus soluciones en estas categorías para cumplir con sus requisitos. Un PAM puede implementar todos o algunos de sus componentes, los proveedores y desarrolladores de los sistemas PAM han categorizado algunas de esas funciones en siglas para identificar de una forma más simple:
 - Account Password Management (APM).
 - Privileged Account and Session Management (PASM).
 - Privileged Session Management (PSM).
 - Session Recording and Monitoring (SRM).
 - Application-to-Application Password Management (AAPM).
 - Privilege Elevation and Delegation Management (PEDM).
 - User Behavior Analytics (UBA).

En la actualidad existen muchas soluciones que implementan algunas o todas las funcionalidades esperadas en un PAM. La empresa consultora (Gartner, 2022), mantiene

una lista de las mejores soluciones PAM valoradas por los usuarios y compara las soluciones implementadas entre productos, del listado de Gartner, los servicios mejor posicionados son: Secret Server de la empresa Delinea, CyberArk e Iraj Privileged Access Manager, según el reporte de (Gartner, 2022). En proyectos de código abiertos, se han encontrado tres repositorios públicos en la plataforma Github (2022), que aborden las funcionalidades de almacenamiento y gestión de contraseñas, gestión de sesiones y gestión de privilegios. Los proyectos identificados son: Vault Project, Teleport y OpenAKC, con una documentación madura y específica, actualizaciones frecuentes y una amplia comunidad de usuarios. Es nuestro interés, por lo importante de mantener un control y supervisión de los usuarios con privilegios, desarrollar un sistema PAM con acceso y administración web, que permita la supervisión de los usuario, seguimiento y registro de las actividades realizada en una sesión, así como la rotación de las contraseñas entre los dispositivos controlados.

5. Objetivos

A continuación, se describe el objetivo específico y los objetivos generales del presente trabajo fin de máster.

5.1.1. Objetivos General:

Desarrollar un sistema de gestión de acceso privilegiado (PAM) para sistemas operativos Linux mediante el acceso de una aplicación Web

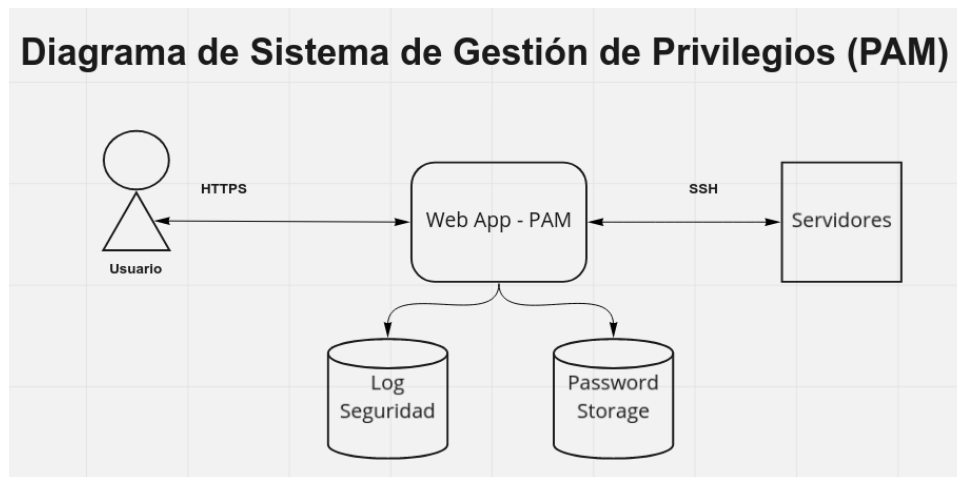
5.1.2. Objetivos específicos

- Implementar una terminal web para acceder de forma segura a los servidores gestionados vía el protocolo SSH.
- Implementar un mecanismo de almacenamiento seguro de las contraseñas de los servidores gestionados.
- Implementar un mecanismo para rotar las contraseñas entre la aplicación web y los servidores gestionados.
- Implementar un sistema de registro (logging) de las actividades de los usuarios en las sesiones de trabajo en los servidores gestionados.
- Brindar una interfaz web para la administración de las tareas de gestión.
- Utilizar tecnologías modernas para el despliegue de aplicaciones autocontenidas.

6. Metodología.

Para el desarrollo de nuestro sistema PAM es necesario abordar una serie de conceptos y herramientas que estaremos utilizando para alcanzar los objetivos propuestos. De la imagen #1, presentamos el esquema a nivel general de nuestra solución.

Imagen 1 - Diagrama Propuesto del Desarrollo (Elaboración Propia)



Los elementos que interactúan en nuestro sistema son:

- a) Usuario: Persona con acceso a la herramienta que utiliza el sistema para la conexión a servidores controlados o la supervisión de las actividades realizadas entre conexiones.
- b) Pila TCP/IP: Responde a Transmission Control Protocol/Internet Protocol, es un modelo que define un conjunto de protocolos que ofrecen un intercambio confiable y de extremo a extremo sobre una red naturalmente no confiable. De acuerdo con (Redeszone, 2022), TCP/IP presenta las siguientes características:
 - Es orientado a conexión.
 - Ofrece confiabilidad y asegura el reordenamiento de la información que es recibida.
 - Permite el control de flujo, de manera que el emisor no sobrepasa la capacidad del equipo receptor.

- Sus funcionalidades se encuentran divididas en capas que tienen tareas específicas.
- c) Hypertext Transfer Protocol (HTTP): Protocolo de comunicación que permite la transferencia de información entre un cliente y un servidor. El protocolo es controlado por el World Wide Web Consortium (W3C) y es utilizado de manera amplia en las aplicaciones que utilizan el protocolo TCP/IP para comunicarse.
- d) Secure Sockets Layer (SSL) y Transport Layer Security (TLS): Son protocolos criptográficos que proporcionan comunicación segura en una red de datos, desarrollados por la Internet Engineering Task Force (IETF) y utilizados ampliamente en las comunicaciones que requieren garantía de seguridad en el Internet.
- e) Hypertext Transfer Protocol Secure (HTTPS): Protocolo de comunicación basado en HTTP que utiliza TLS para la comunicación segura de las comunicaciones entre los sistemas que utilizan mayormente el Internet para comunicarse.
- f) Secure SHell (SSH): Protocolo para la comunicación segura de los accesos remotos a un servidor mediante un canal cifrado. Permite la conexión, transferencia de archivos y redirección del tráfico para ejecutar programas de forma remota. Los servidores SSH permiten implementar las características del protocolo y los clientes interactuar con ellos.
- g) Log de Seguridad: Aplicación encargada de almacenar los registros de los eventos procesados por el PAM, permiten realizar auditorías o ser integrados en aplicaciones de detección de anomalías o incidencias para dar alertas en tiempo real.
- h) Base de datos de Contraseña: Base de datos encargada de almacenar las contraseñas de los usuarios con privilegios en los servidores gestionado por la herramienta. Una característica importante es la encriptación de los datos mediante claves asimétricas.
- i) Claves Asimétricas: Sistema de encriptación basado dos partes, en una llave que realiza la encriptación, conocida como llave privada y otra que permite la descryptar la información, conocida como llave pública.

- j) Aplicación Web PAM: Aplicación que gestiona las funcionalidades de gestión, supervisión y acceso de los sistemas controlados, integrando un conjunto de protocolos, librerías, *frameworks* y tecnologías.

6.1. Tecnologías utilizadas para el desarrollo del PAM

Para el desarrollo del PAM se utilizará como lenguaje de programación y plataforma la tecnología Java, así como varias librerías y *frameworks* asociado al lenguaje.

6.1.1. Java

Java es un lenguaje de programación y plataforma creado por Sun Microsystems (Byous, 1998) liberando su primera versión en el 1996 y adquirido en el 2009 por la empresa Oracle (El País, 2009). Java deriva su sintaxis en gran medida de los lenguajes de programación de C y C++, revolucionando la forma de compilar y distribuir aplicaciones por la compilación de sus archivos en un formato denominado Bytecodes (.class) y utiliza la Máquina Virtual de Java (JVM, siglas en inglés) que sirve de intermediario para ejecutar el programa compilado sin importar la arquitectura primaria del sistema host que ejecuta la JVM (Lindholm, Yellin, Bracha y Buckley, 2014). Java utiliza el paradigma de programación orientada a objeto y permite la programación funcional, incluye librerías en su núcleo central para trabajar con los protocolos que implementan la pila de comunicación TCP/IP y manejo de multiprocesos (*threads*) siendo muy utilizado en aplicaciones que interactúan con el Internet y procesan una cantidad elevada de datos.

En la actualidad, según el índice de (TIOBE, 2022), Java representa el tercer lugar con mayor relevancia en función a los datos recopilados de la industria. Java se encuentra en su versión 18 y permite la creación de aplicaciones en todos los ámbitos tecnológicos: web, escritorios, smart-cards, dispositivos móviles, sistemas empujados, sistema en tiempo real y la nube.

6.1.2. Librerías utilizadas para el acceso SSH

Con el objeto de implementar los requerimientos propuestos, es importante seleccionar la librería de que estará interactuando con el protocolo SSH, en Java existen tres librerías que tienen buen soporte, una comunidad muy activa y utilizadas en proyectos en producción:

- **Apache Mina SSHD:** Utiliza como base el framework Multipurpose Infrastructure for Network Applications (Mina) para crear aplicaciones escalables y de alto rendimiento en aplicaciones que utilizan los protocolos basados en TCP, UDP y comunicación en serie basado en la característica no bloqueante de procesos (NIO, Non-blocking I/O) incorporado en la plataforma Java. La librería es distribuida bajo la licencia Apache License 2.0. Es una implementación nativa en Java, indicando que no requiere ninguna librería adicional para ser integrada en nuestro proyecto. Soporta la versión 2 del protocolo SSH y está diseñada para utilizada con llamadas no bloqueantes, utilizando la interfaz Future, permitiendo las llamadas asíncronas y ejecutando una función de retorno (callback) (Apache Mina, 2022).
- **Java Secure Channel (JSch):** Desarrollada por la empresa JCraft.com, y distribuida bajo la licencia BSD-style. Muy utilizado en proyectos anteriores al 2018. La librería no presenta nuevas más actualizaciones en sus repositorios Maven desde el 2018. Librería nativa en java, implementa la versión 2 del protocolo, y su diseño es basado en llamadas bloqueantes para los comandos y llamadas a flujos de datos vía hilos (Thread). Las fuentes de la librería no están disponibles en repositorios abiertos (JSch, 2022).
- **SSHJ - SSHv2 library for Java:** Desarrollado por Jeroen Van Erp y publicado en la plataforma Github, cuenta con más de 2000 estrellas por parte de los usuarios y actualizaciones constantes. Implementa la versión 2 del protocolo, librería 100% basada en Java, permite llamadas asíncronas y ejecutando funciones de retorno (callback) (SSHJ, 2022).

Basado en la documentación, frecuencia de actualizaciones y diseño del API, fue seleccionado la librería de SSHJ para implementar el proyecto.

6.1.3. Framework Web

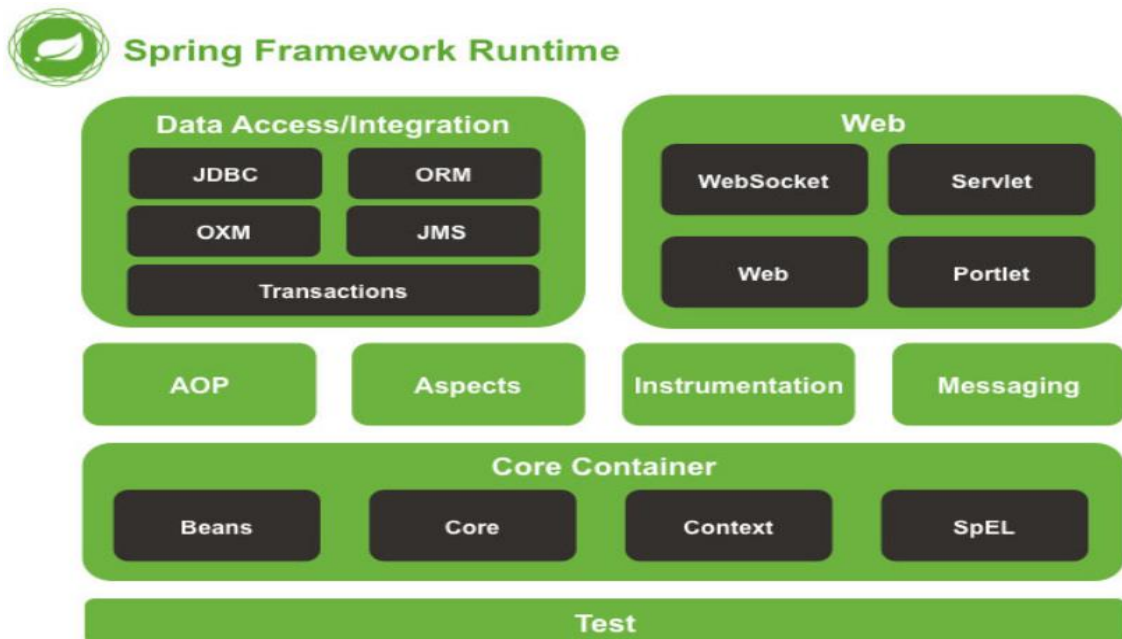
Los *framework* web nos permiten acelerar el desarrollo de aplicaciones por la normalización en la curva de aprendizaje y las funcionalidades que mejoran la productividad (Edix, 2022).

El *framework* seleccionando para desarrollar los componentes orientados a procesar las peticiones de los usuarios en el servidor es Spring Boot y el *framework* para la capa de presentación, donde estarán interactuando los clientes es Vaadin Flow.

6.1.4. Spring Boot

Spring Framework fue creado por Rod Johnson en el 2002 y fue revolucionario en su momento por incluir el concepto de trabajo de convención sobre configuración (CoC, por sus siglas en inglés) y el uso del contenedor de inversión de control (IoC, por sus siglas en inglés) e inyección de dependencias manejado por el contenedor. Para el 2002 las aplicaciones desarrolladas en la plataforma Java utilizaban una aplicación que permitía brindar servicios adicionales al desarrollo, como son la gestión de las conexiones de base de datos, cola de mensajería, llamadas a servicios externos e implementación de especificaciones de Java (JSR) que son parte del estándar, esas aplicaciones se denominan Servidores de Aplicaciones (IONOS, 2019). En la Imagen 2, podemos visualizar los componentes que conforman el *framework* de Spring.

Imagen 2 - Componentes del Framework Spring (Spring, 2022)



Spring Boot es un módulo del proyecto de Spring que fue creado en el 2014 para simplificar el desarrollo de aplicaciones bajo Spring Framework, en el marco de las nuevas tecnologías orientada al uso de contenedores de aplicaciones, los servicios en La Nube y brindar un mejor soporte a las aplicaciones diseñadas bajo la arquitectura de desarrollo basada en microservicios (IBM Cloud Education, 2020).

Las características que implementa Spring Boot:

- Simplifica la creación de aplicaciones listas para ser ejecutadas en un único Jar o War. (Ideal para microservicios y contenedores Docker)
- Implementan herramientas para monitorear y auditar nuestras aplicaciones.
- No necesita código generado o configuración basada en XML.
- Proporciona su propio contenedor Web (Tomcat, Jetty o Undertow), no necesita la generación de archivos War.
- Integrado con herramientas de gestión de proyectos, Maven o Gradle.
- Gran cantidad de librerías relacionadas integradas.

6.1.5. Vaadin Flow

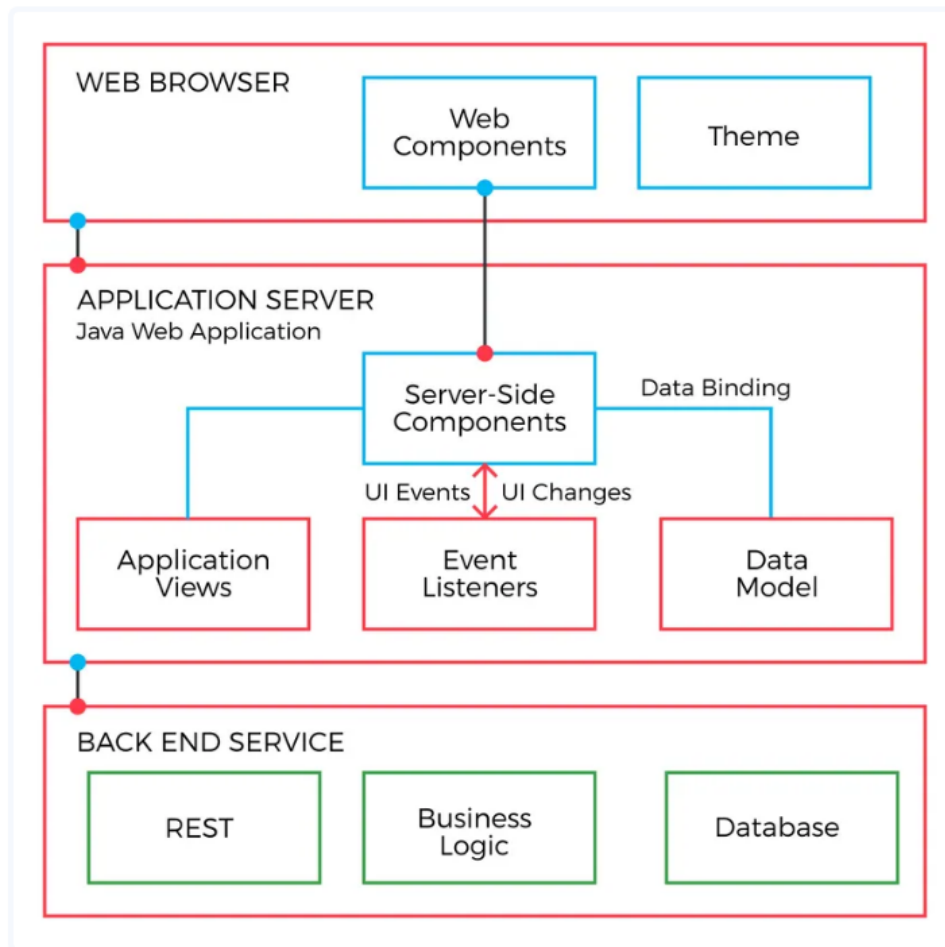
Es un framework de aplicaciones Web RIA (Rich Internet Application) del lado del servidor y del cliente, cuyo objetivo es desarrollar aplicaciones con un comportamiento, funcionalidad y fluidez a las aplicaciones de escritorios. Tiene como característica que está basado en componentes web y soporta los lenguajes disponibles en la máquina virtual de Java (Kotlin, Groovy, Scala, Clojure, entre otros.) (Vaadin Team, 2022).

Como característica importante:

- Implementa el uso de componentes web, estándar en la distribución de componentes en la web, necesarios para una experiencia satisfactoria por el usuario.
- Cuando utilizamos la opción de ejecución en el servidor, la regla de negocio se ejecuta en el mismo, dando una seguridad adicional.
- Integración directa entre el DOM y el servidor sin tocar JS, JSON, entre otras tecnologías.
- Los que tienen una experiencia en desarrollo Desktop adoptan el *framework* de una manera fácil.

En la Imagen 3, se presenta el esquema de comunicación que utiliza Vaadin Flow con los componentes instanciados en el servidor bajo la tecnología Java con los componentes Web y su presentación en el navegador. Vaadin Flow permite la interacción mediante eventos que son reportados desde los componentes web hasta los componentes instanciados en el servidor de aplicación bajo tecnología Java.

Imagen 3 - Vaadin Flow con los componentes instanciado en el servidor (Vaadin, 2022)



6.1.6. Contenedores de Software

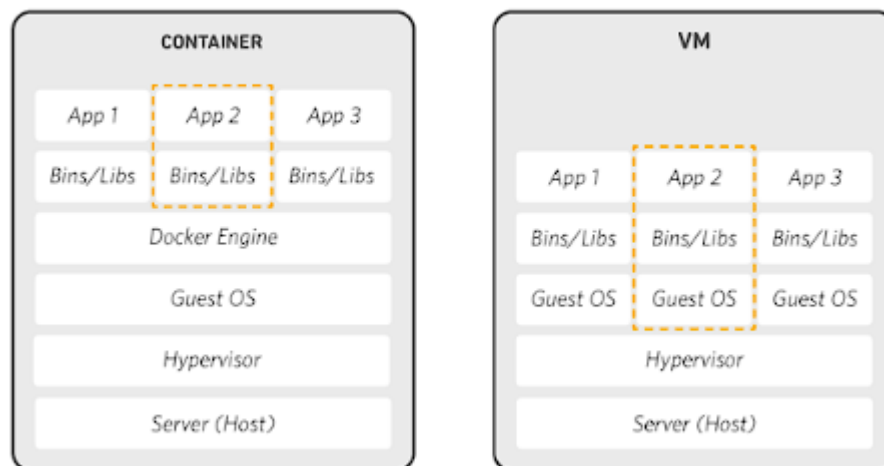
Los contenedores de software emulan el comportamiento actual del sector de transporte logístico, en la manera en como trabajan diferentes cargas y son transportadas por diferentes medios (buques, trenes y camiones de carga) homologando el contenedor de la carga. Aprovechando ese concepto, el software se empaqueta en un software estándar conocido como contenedor, el cual contiene todos los archivos y dependencias necesarias para su ejecución, aislando la aplicación contenida del host que ejecuta el contenedor y homologando su ejecución sin importar el entorno de ejecución (Azure, 2022).

Los contenedores de software nos aportan agilidad, portabilidad y escalabilidad rápida en nuestros desarrollos (Azure, 2022).

6.1.6.1. Docker

Es un motor que permite implementar los contenedores de software, empaquetando las aplicaciones, con sus librerías y permitiendo la ejecución aislada y homogénea de las aplicaciones independiente del entorno. Docker es un sistema operativo para contenedores, eliminando la necesidad de administrar el hardware del servidor (AWS, 2022). En la Imagen 4, podemos observar la diferencia en la forma de ejecución de la máquina virtual (VM) y un contenedor del software, aprovechando los recursos del sistema anfitrión (*host*).

Imagen 4 - Diferencia entre contenedores Docker y máquinas virtuales (AWS, 2022)



Docker utiliza tecnologías presentes en el kernel de Linux, como son los grupos de control y el espaciado de nombres, para dividir y aislar los procesos de la ejecución primaria del equipo anfitrión (*host*), aprovechando los recursos de la infraestructura y conservando la seguridad presente en los sistemas individuales (Red Hat, 2022).

Las ventajas de los contenedores de software basado en Docker se pueden listar en:

- Modularidad.
- Capas y control de versiones en las imágenes.
- Restauración.
- Implementación rápida.

6.1.6.2. Docker Compose

Es una herramienta dedicada a la coordinación de la ejecución simultáneas de varios contenedores de software, para ser inicializadas de forma rápida y fácil mediante un script. Esa coordinación recibe el nombre de orquestación y se utiliza de forma local, es decir, en el computador que sirve de equipo anfitrión (*host*). Docker Compose utiliza un archivo de configuración tipo YAML para la definición de los contenedores, servicios, redes y volúmenes utilizados en la orquestación (Keepcoding, 2022).

6.2. Análisis y Diseño de la aplicación

6.2.1. Lista de requerimientos

De acuerdo con el objetivo general y los objetivos específicos, la relación de requerimientos del software queda planteado en la Tabla 1- Requerimientos del software.

Tabla 1 - Requerimientos del software

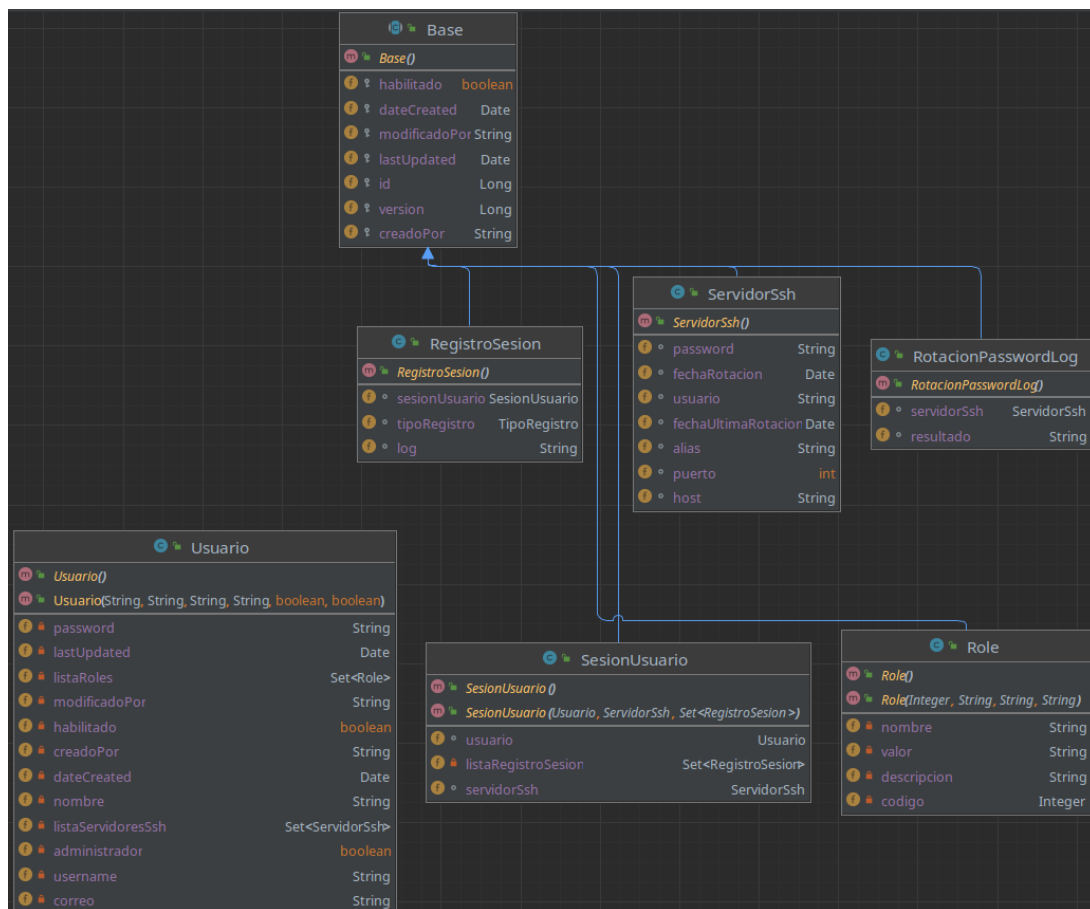
Requerimientos de la aplicación	
#	Requerimiento

1	El sistema debe gestionar la creación, actualización, consulta y eliminación de los usuarios que interactúan con el PAM
2	El sistema debe permitir la gestión de los servidores que serán gestionados por el PAM
3	El sistema debe permitir asignar un servidor a un usuario basado en roles y permisos.
4	El sistema debe gestionar la rotación de contraseñas en los servidores gestionados por el PAM.
5	El sistema debe permitir almacenar los registros de las conexiones realizadas por los usuarios con los servidores gestionados.
6	El sistema debe permitir la conexión vía SSH mediante el PAM.
7	El sistema debe permitir al usuario acceder a una conexión SSH mediante el PAM.
8	El sistema debe utilizar contenedores web para el fácil despliegue de la aplicación.
9	El sistema debe permitir a un superadministrador visualizar la información de las contraseñas de acceso.

6.2.2. Diagramas de clase

De los requerimientos listados en la Tabla 1, el modelado de las clases queda reflejado en la Imagen 4.

Imagen 5 - Diagrama de clase (Elaboración propia)



6.2.3. Bosquejos de la aplicación

Para el diseño visual de la aplicación, fue trabajado con un esquema de web responsivo que permita la adaptación de la vista en función a la disponibilidad de la resolución de la pantalla, sin perder funcionalidades por las limitaciones de visualización que pudiera tener el usuario.

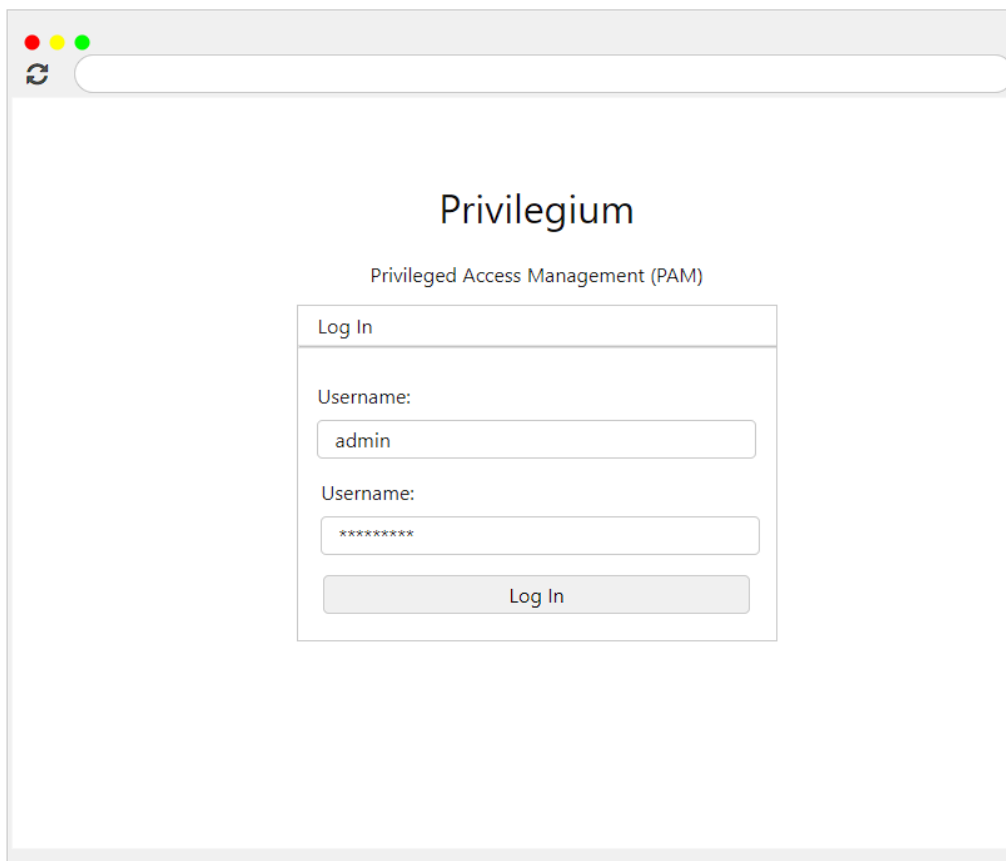
Las funcionalidades plasmadas a nivel diseño permiten alinear lo deseado con lo esperado en el desarrollo de la aplicación. Como guía fueron trabajadas las vistas con las funcionalidades más importantes:

- Acceso a la aplicación (Ver Imagen 5).
- Listado de las terminales asociadas a un usuario (Ver Imagen 6).

- Acceso a terminal (Ver Imagen 7).
- Conexión a los servidores (Ver Imagen 8).
- Nuevo o edición de conexión a servidores (Ver Imagen 9).
- Registros de conexiones (Ver Imagen 10).
- Registros de rotación de contraseñas (Ver Imagen 11).
- Administración de usuarios (Ver Imagen 12).

Los casos que requieren de la gestión de alguna entidad, se utilizará el esquema presentando en la Imagen 9, el caso aplica en los procesos de gestión de usuarios.

Imagen 6 - Inicio de la Aplicación (Elaboración propia)



The image shows a web browser window displaying the login page of the 'Privilegium' Privileged Access Management (PAM) application. The browser's address bar is empty. The page has a light gray background. At the top center, the word 'Privilegium' is displayed in a large, bold, black font. Below it, 'Privileged Access Management (PAM)' is written in a smaller, regular black font. A white rectangular box with a thin gray border contains the login form. The form has a title 'Log In' at the top. It includes two 'Username:' labels, each followed by a text input field. The first input field contains the text 'admin', and the second contains a series of asterisks '*****'. At the bottom of the form is a gray button labeled 'Log In'.

Imagen 7 - Listado de Terminales Disponibles (Elaboración propia)

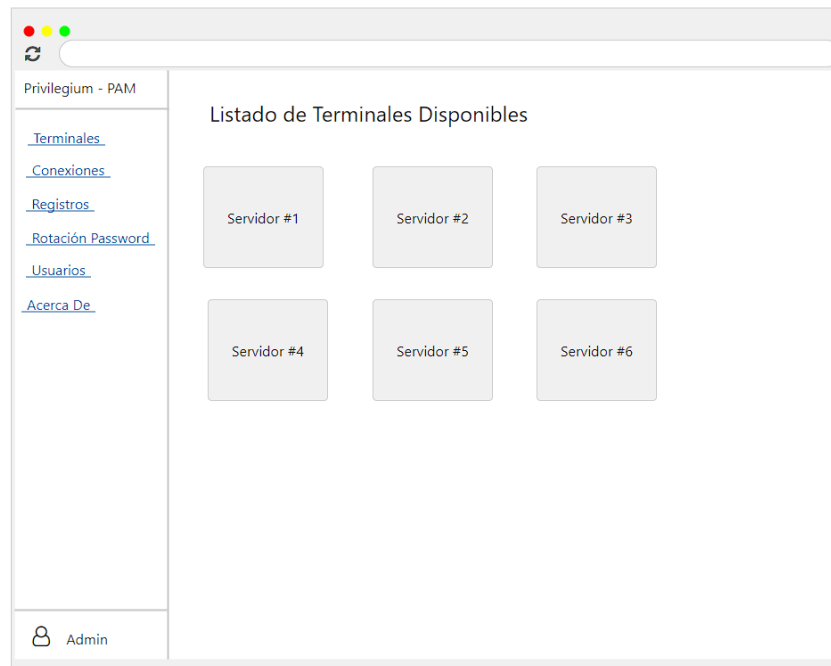


Imagen 8 - Acceso terminal (Elaboración propia)

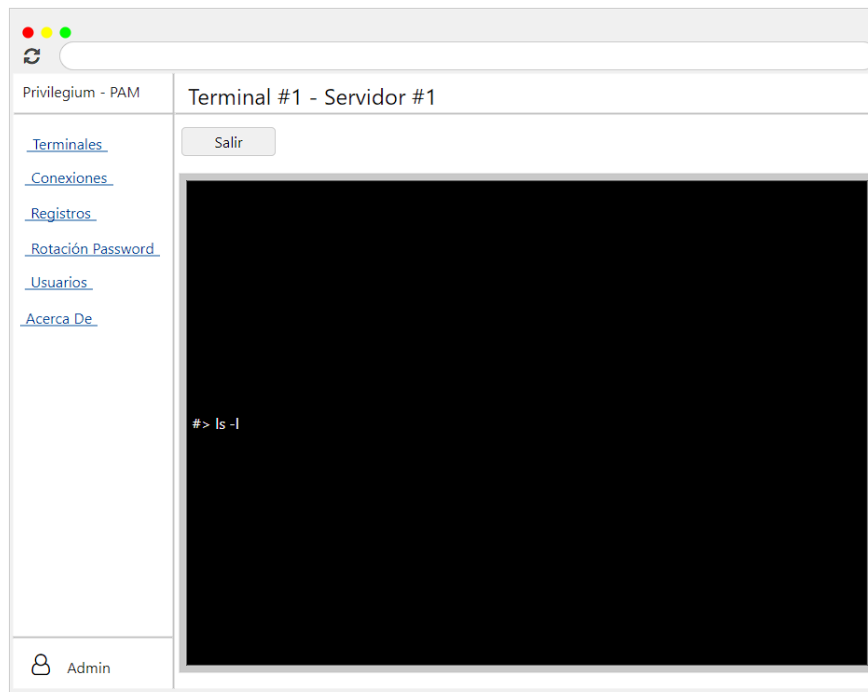


Imagen 9 - Conexión Servidores (Elaboración propia)

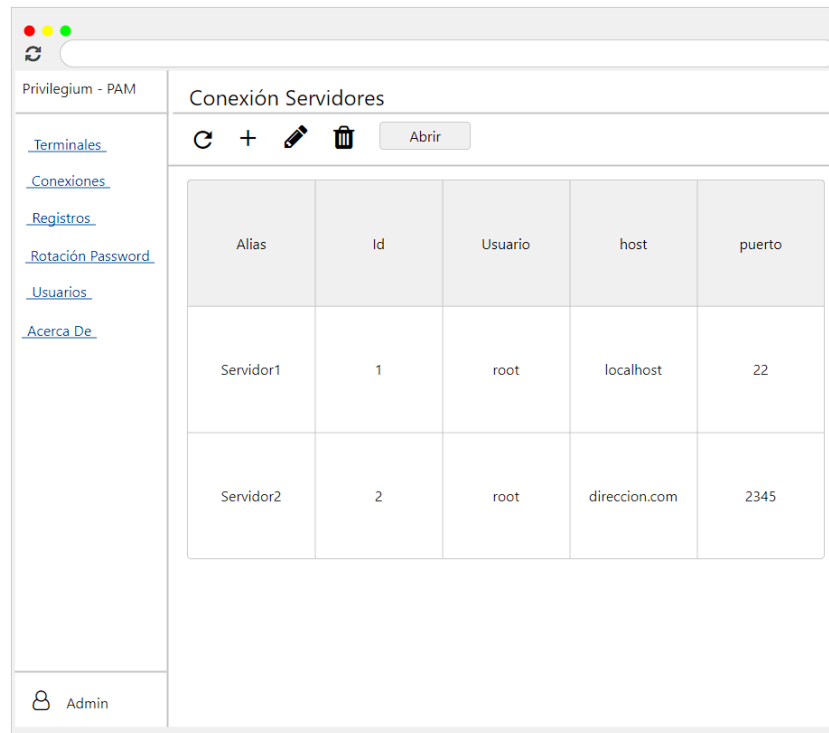


Imagen 10 - Agregar, edición de conexiones (Elaboración propia)

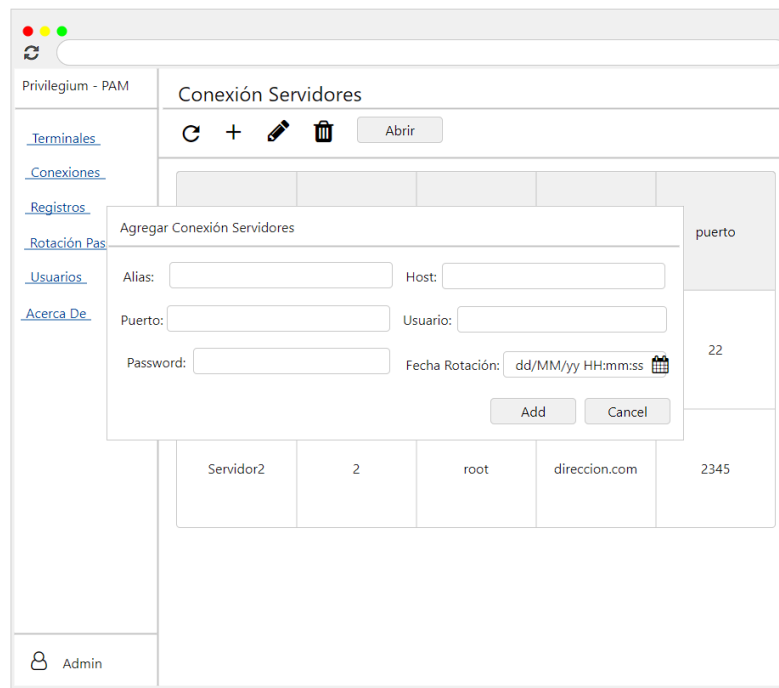
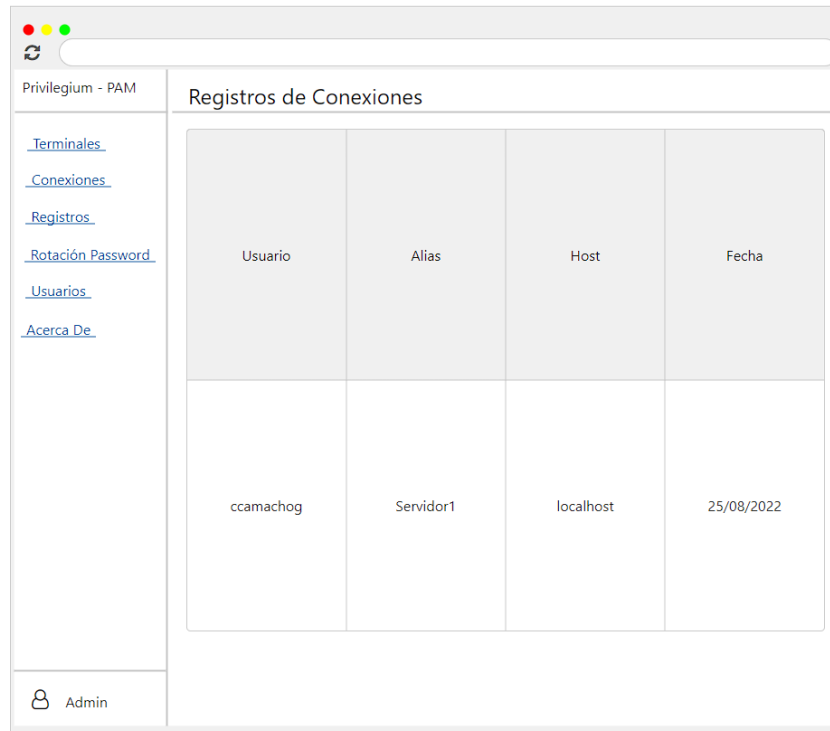
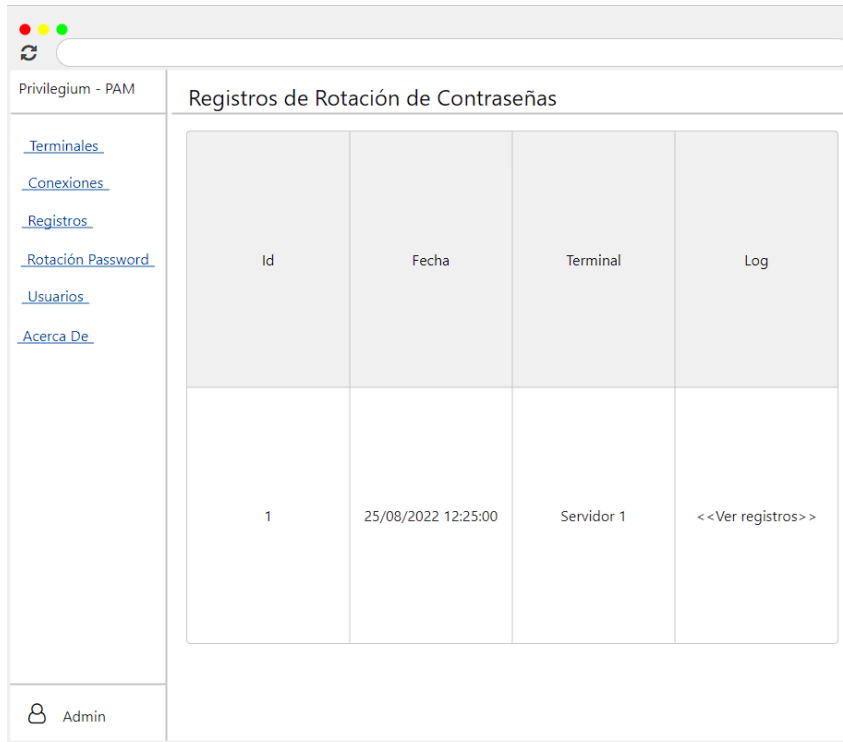


Imagen 11 - Registro de conexiones (Elaboración propia)



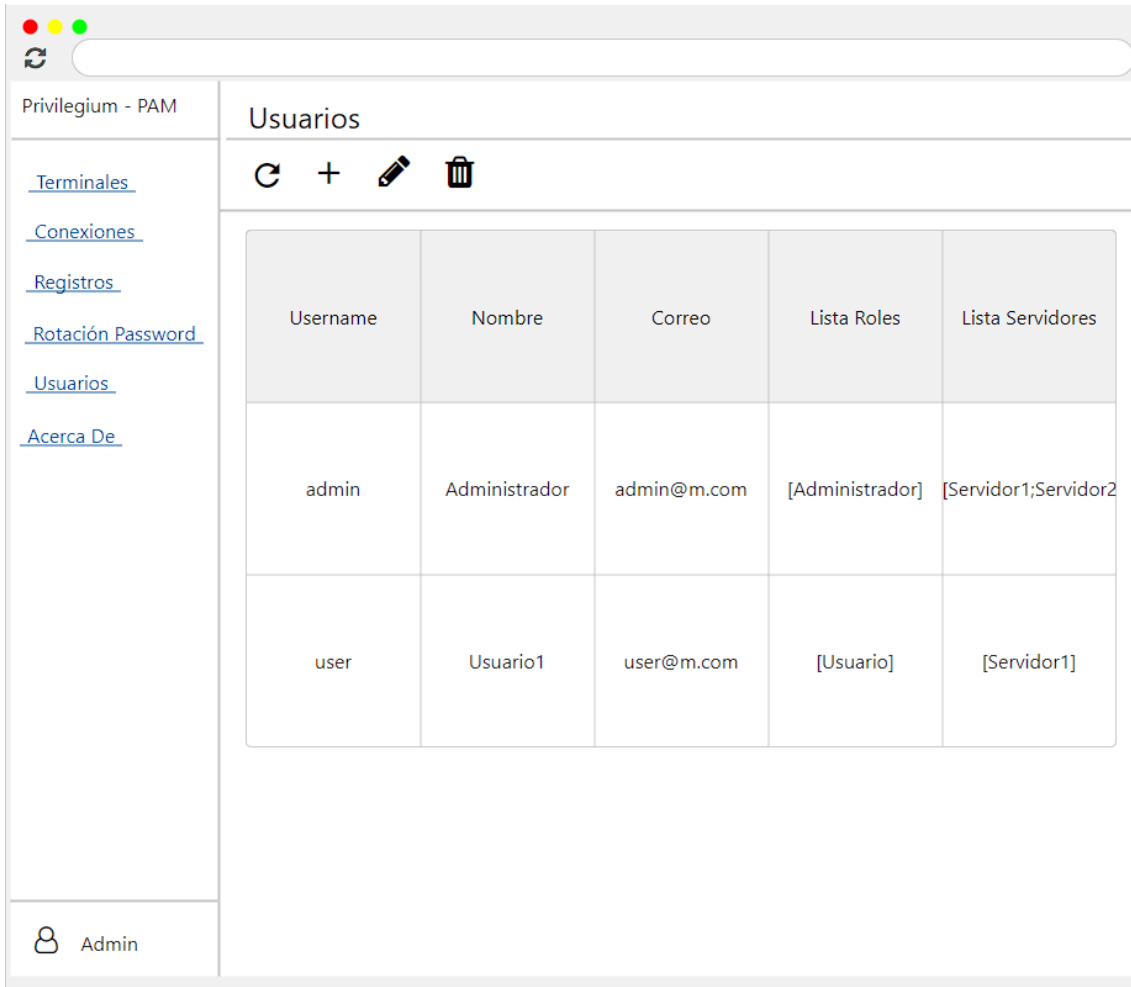
Usuario	Alias	Host	Fecha
ccamachog	Servidor1	localhost	25/08/2022

Imagen 12 - Registro de rotación de contraseñas (Elaboración propia)



Id	Fecha	Terminal	Log
1	25/08/2022 12:25:00	Servidor 1	<<Ver registros>>

Imagen 13 - Usuarios (Elaboración propia)



The screenshot displays the 'Privilegium - PAM' web application. The main section is titled 'Usuarios' and contains a table with the following data:

Username	Nombre	Correo	Lista Roles	Lista Servidores
admin	Administrador	admin@m.com	[Administrador]	[Servidor1;Servidor2]
user	Usuario1	user@m.com	[Usuario]	[Servidor1]

The sidebar on the left includes the following links: [Terminales](#), [Conexiones](#), [Registros](#), [Rotación Password](#), [Usuarios](#), and [Acerca De](#). At the bottom of the sidebar, there is a user profile icon and the text 'Admin'.

6.3. Licenciamiento del proyecto

Con el objetivo de que el desarrollo pudiese ser útil para la comunidad, se ha creado un repositorio público en la plataforma GitHub para ofrecer la posibilidad de accesibilidad, implementación y permitir nuevas mejoras o funcionalidades por parte de la comunidad. El proyecto desarrollado interesa que los usuarios tengan la libertad de ejecutar el programa como lo deseen, estudiar y modificar el código fuente, redistribuir copias exactas y la redistribución de copias modificadas y en caso de utilizar como un servicio mediante la red, las modificaciones deban ser liberadas para provecho de la comunidad. La licencia que engloba ese interés es la licencia pública general de Affero en su versión 3 (en inglés, Affero General Public License, conocida como Affero GPL o AGPL). (GNU, 2007) y es la seleccionada para el proyecto. La licencia utilizada puede ser consultada en la sección de anexo, Tabla 5-Licencia AGPL V3.

6.4. Desarrollo de la aplicación

La aplicación está basada en una arquitectura de Modelo-Vista-Controlador (MVC), utilizando Spring Boot para los procesos gestionados en el servidor y Vaadin Flow como *framework* para la vista del usuario.

La estructura organizativa del proyecto es la presentada en la Imagen 14, mostrando las carpetas y archivos requeridos por los *frameworks* y tecnologías utilizadas en el proyecto. En la Tabla 2, se describe la funcionalidad que tiene cada carpeta o archivo dentro del proyecto.

Imagen 14 - Carpetas y Archivos del Proyecto (Elaboración propia)

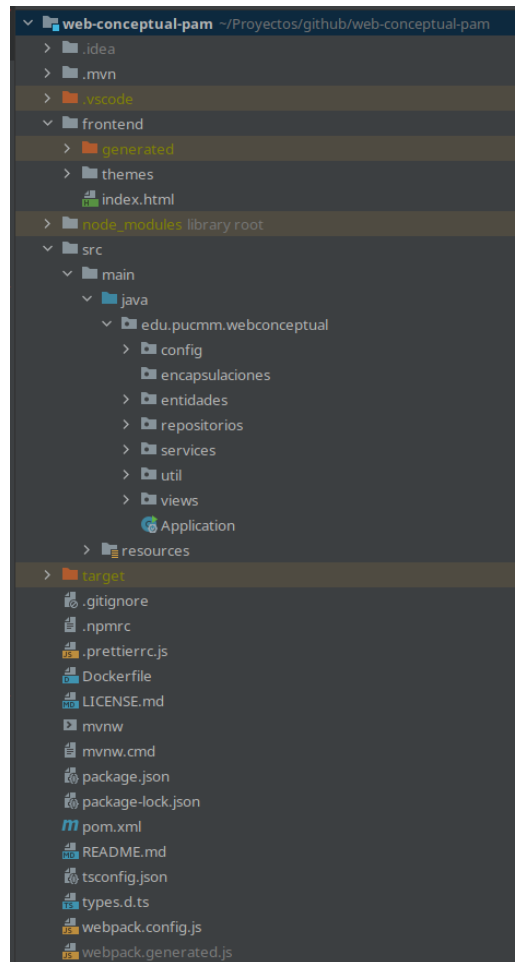


Tabla 2 - Estructura del Proyecto

Estructura del proyecto	
Carpeta / Archivo	Descripción
frontend	Contiene los recursos web (HTML, JS, CSS, entre otros) utilizados por Vaadin Flow para la generación de la vista.
node_modules	Almacena las dependencias instaladas por el <i>framework</i> de Vaadin Flow para los componentes web basados en JavaScript utilizando Node.js y la herramienta de dependencia npm.

src	Carpeta donde se alojan los archivos fuentes en el lenguaje de programación Java utilizando el framework Spring Boot.
target	Carpeta que almacena las clases y binarios generados por el proceso de compilación de la aplicación.
.gitignore	Archivo utilizado para especificar los archivos que estarán participando bajo la gestión del control de versiones bajo el estándar GIT.
Dockerfile	Archivo que permite la definición de imágenes de contenedores de aplicaciones basado en la herramienta Docker.
.dockerignore	Indica los archivos que serán excluidos del proceso de compilación de la imagen Docker.
docker-compose.yml	Archivo que contiene la definición de las instrucciones para construir y crear contenedores de software que son utilizados en el despliegue de la aplicación para el ambiente de demostración
.env.sample	Archivo donde están definidas las variables de ambientes que son utilizadas para la parametrización del Docker Compose.
build.sh	Script para la compilación de la imagen Docker y el arranque del ambiente Docker Compose para la demostración.
.npmrc	Archivo para la configuración de la herramienta npnm utilizado por Vaadin Flow.
.prettierrc.js	Archivo de configuración del <i>framework</i> Prettier, utilizado por Vaadin Flow para la normalización del estilo del código fuente producido.

env-demo.env	Archivo donde están definidas las variables de ambientes que son utilizadas para la parametrización del ambiente de demostración del Docker Compose.
LICENSE.md	Archivo con la definición de la licencia del proyecto.
mvnw y mvnw.cmd	Script para la ejecución del gestor de proyecto Maven utilizado en los sistemas operativos Linux o Mac OS. El .cmd es utilizando en Windows.
package.json y package-lock.json	Archivo con las dependencias JavaScripts utilizadas por el <i>framework</i> Vaadin Flow para el manejador de proyecto npm. El archivo package-lock.json garantiza que las versiones utilizadas por la <i>framework</i> .
package-lock.json	Archivo con las versiones de dependencia
pom.xml	Archivo con la definición de las dependencias de librerías gestionados por el gestor de proyecto Maven.
README.md	Archivo con las generales del proyecto e instrucciones para el compilado y ejecución del proyecto.
tsconfig.json	Archivo para la configuración para el lenguaje TypeScript utilizado por el <i>framework</i> Vaadin Flow.
types.d.ts	Archivo con la definición de tipo de datos para TypeScript utilizado por utilizado por el <i>framework</i> Vaadin Flow.
webpack.config.js	Archivo de configuración del <i>framework</i> WebPack utilizado por utilizado por el <i>framework</i> Vaadin Flow

Las funcionalidades propias del PAM están codificadas en la carpeta de **src** del proyecto, en el paquete **edu.ucjc.privilegium**. En la Imagen 15, podemos visualizar las carpetas y las funcionalidades que están desarrolladas en el proyecto, en la Tabla 3, se describe el contenido de cada carpeta.

Imagen 15 - Carpetas del Proyecto (Elaboración propia)

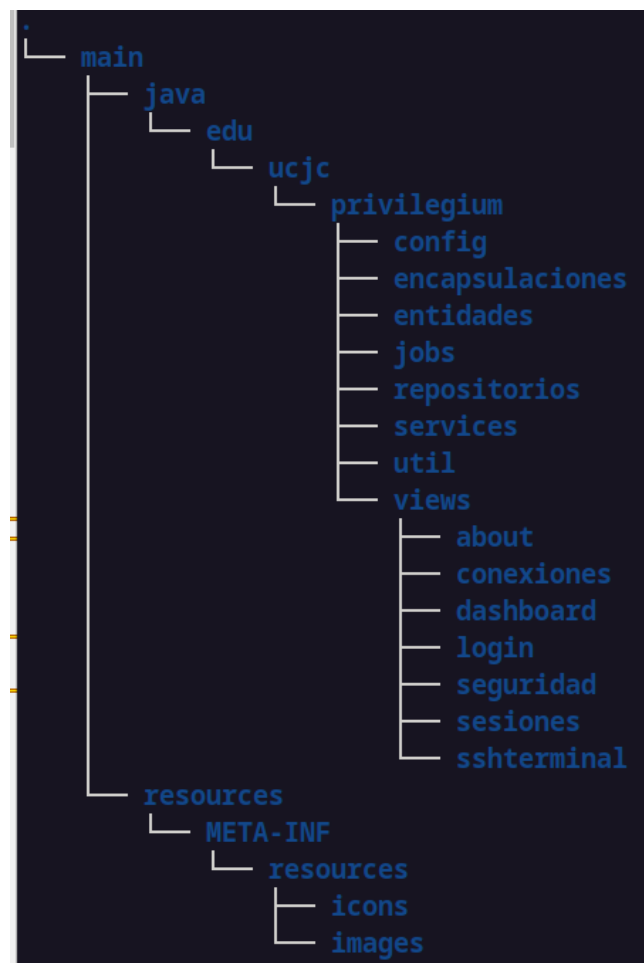


Tabla 3 - Componentes en la tabla src

Componentes en la carpeta src	
Carpeta	Contenido
config	Las clases de configuración del framework Spring Security y las cargas automáticas de información del proyecto.
encapsulaciones	Las clases correspondientes a los objetos para la transferencia de información (Data Transfer Object, DTO).
entidades	Las clases que están asociadas al modelo de datos y al motor de persistencia.
repositorios	Las clases utilizadas por el módulo de acceso al motor de persistencia que integra la gestión de los datos con la aplicación.
util	Las clases con funcionalidades genéricas para el proyecto.
views	Clases que representan las vistas de la aplicación.
services	Clases que manejan las reglas de negocio de la aplicación y controlan la transaccionalidad de los datos con el motor de persistencia.
jobs	Clases que implementa las funcionalidades de calendario (schedule) en el <i>framework</i> Spring boot.
resources	Archivos de configuración propios del frameworks Vaadin Flow, Spring Boot y recursos utilizados en las vistas (imágenes, iconos, entre otros)

La gestión y construcción del proyecto se utiliza la herramienta Maven, encargado de estandarizar la generación del proyecto y gestionar sus dependencias mediante el contenido del archivo pom.xml.

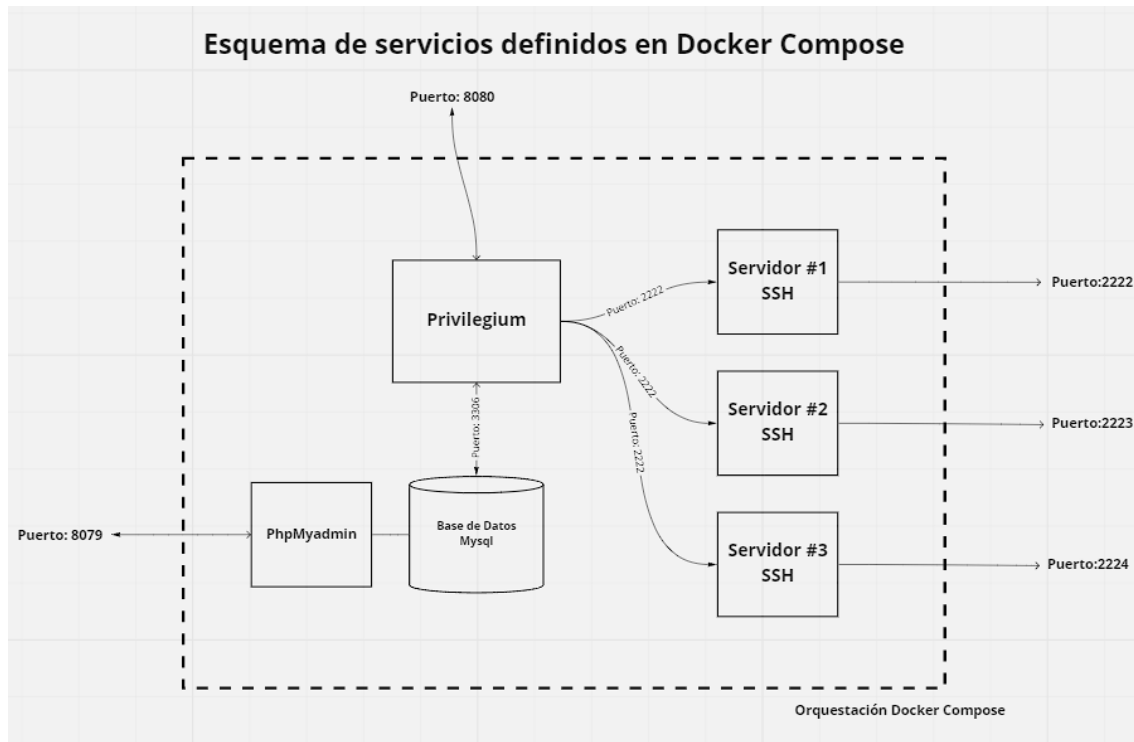
Las dependencias y librerías utilizadas más importantes utilizadas para el desarrollo son presentadas en la tabla 4:

Tabla 4 - Librerías y componentes destacados

Nombre	Tipo	Descripción
xterm-console	Componente Vaadin	Simulador de un terminal para el <i>framework</i> Vaadin Flow en su versión 14 en adelante y la librería xterm.js. Desarrollado por Flowing Code. (XTerm Console Addon, 2022)
Crud UI Add-on	Componente Vaadin	Componente para la generación de los CRUD (Create, Read, Update, Delete) de las entidades utilizadas el proyecto. Desarrollado por Alejandro Duarte. (Crud UI Add-on, 2022)
Spring Security	Librería Spring Boot	Librería para el control de la seguridad web en los diferentes recursos utilizados en el proyecto. (Spring Security, 2022)

Para la ejecución rápida de la aplicación por parte de los usuarios, se añade las configuraciones necesarias para el uso de contenedores de software basados en Docker y la orquestación de los diferentes componentes utilizados mediante Docker Compose. En la Tabla 8 se encuentra el archivo Docker para la creación de la imagen y en la Tabla 9, la definición de los servicios, volúmenes y redes utilizadas para el arranque de todo el sistema. En la Imagen 16, se pueden apreciar los componentes que interactúan en el software.

Imagen 16 - Servicios definidos Docker Compose (Elaboración propia)



Los puertos que están expuesto por defectos desde el servidor anfitrión (*host*) son:

- 8080: Para la aplicación web desarrollada.
- 8079: Aplicación de PhpMyadmin para el acceso remoto de la base de datos.
- 2222: Puerto SSH del Servidor #1 en el ambiente de prueba.
- 2223: Puerto SSH del Servidor #2 en el ambiente de prueba.
- 2224: Puerto SSH del Servidor #3 en el ambiente de prueba.

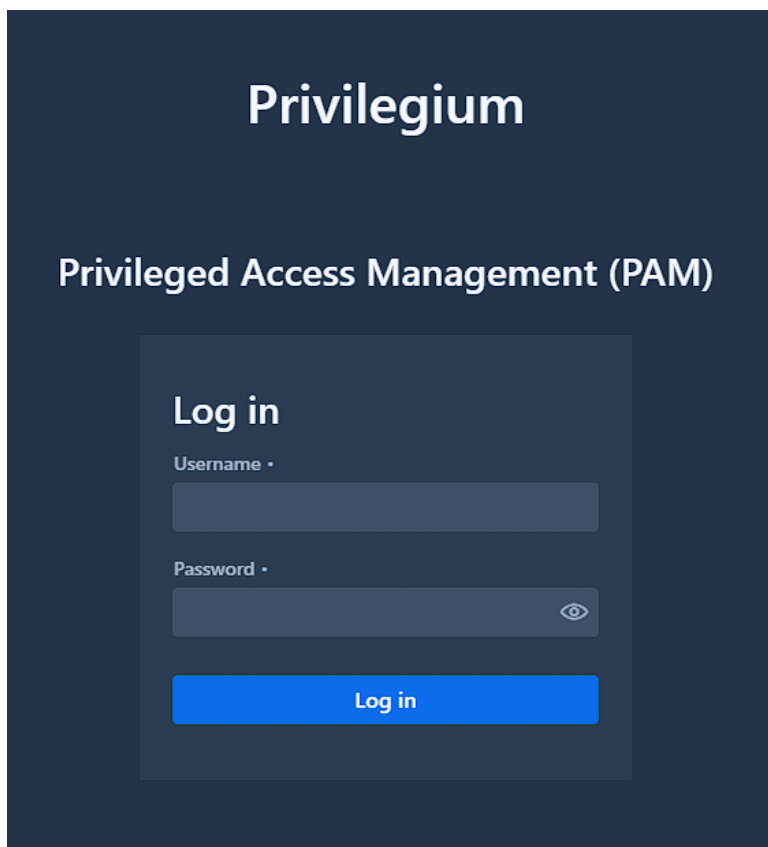
Los puertos de los sistemas dentro del ambiente en Docker Compose, que tienen comunicación directa sin interactuar con el host, son:

- 3306: Base de datos en Mysql.
- 80: Servidor PhpMyAdmin.
- 2222: Todas las instancias de los servidores SSH de prueba.

7. Resultados.

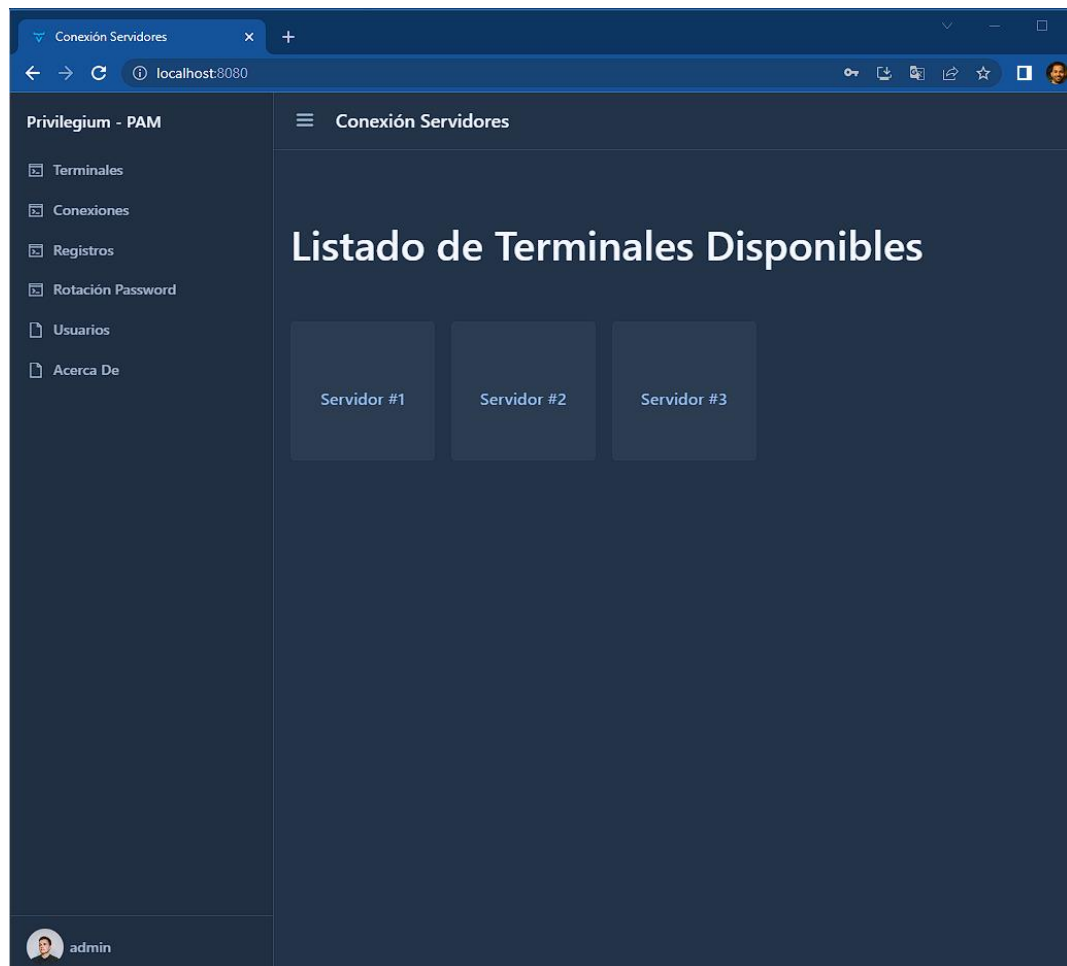
Completado el proceso de desarrollo se presentan las vistas que soportan las diferentes funcionalidades del software que implementan el objetivo general y específicos del proyecto. En la Imagen 17, podemos ver el control de acceso de los usuarios. En caso de no contar con un usuario con los permisos necesarios, el sistema no permitirá el acceso al sistema.

Imagen 17 - Acceso al Sistema (Elaboración propia)

The image shows a login interface for a system named 'Privilegium'. The background is dark blue. At the top, the word 'Privilegium' is written in a large, white, sans-serif font. Below it, 'Privileged Access Management (PAM)' is written in a smaller, white, sans-serif font. In the center, there is a lighter blue rectangular box containing the login form. Inside this box, the text 'Log in' is at the top. Below it are two input fields: 'Username' and 'Password'. The 'Password' field has a small eye icon to its right, indicating a toggle for password visibility. At the bottom of the box is a blue button with the text 'Log in' in white.

Con el acceso al sistema, se presentan las opciones que dispone el sistema mostrado en la Imagen 18, la vista principal, muestra el listado de terminales que tiene acceso al usuario.

Imagen 18 - Entrada del sistema (Elaboración propia)



Es importante indicar que las opciones mostradas al usuario autenticado dependen del rol del usuario asignados, en la Imagen 19, se puede ver el menú para el rol administrador y en la Imagen 20, cuando el usuario tiene asignado el rol usuario. Si un usuario tiene ambos roles asignados, el sistema unificará los permisos asignados a cada rol.

Imagen 19 - Listado de acceso perfil administrador (Elaboración propia)

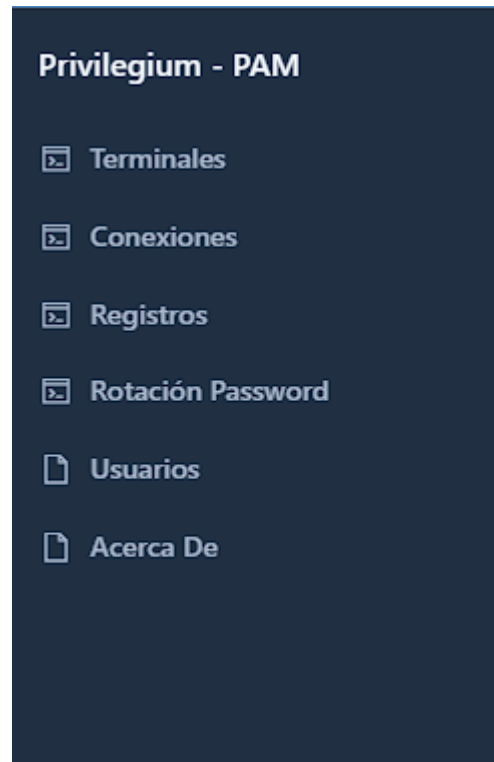
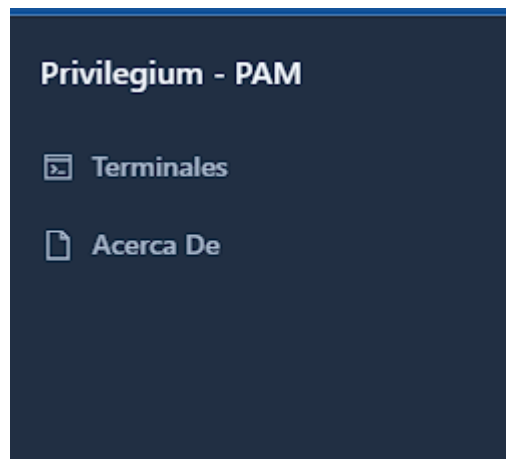


Imagen 20 - Listado de accesos perfil usuario (Elaboración propia)



El acceso a la terminal controlada por el PAM es presentada en la secuencia de imágenes 21 y 22, donde una vez registradas una terminal, el usuario con permisos sobre está, puede acceder desde la opciones de Terminales.

Imagen 21 - Selección de la terminal para el acceso. (Elaboración propia)

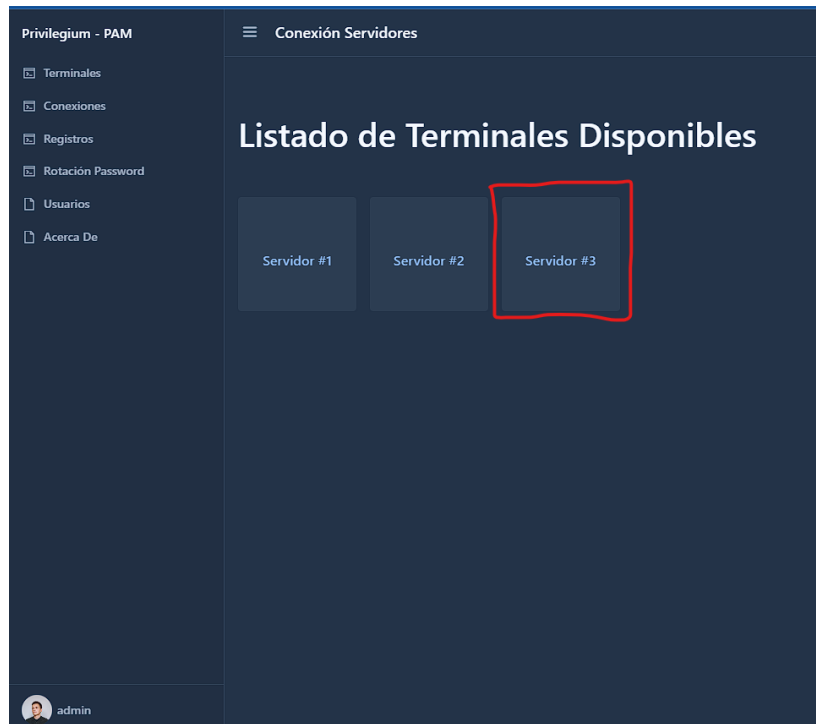


Imagen 22 - Acceso a la terminal por el usuario. (Elaboración propia)

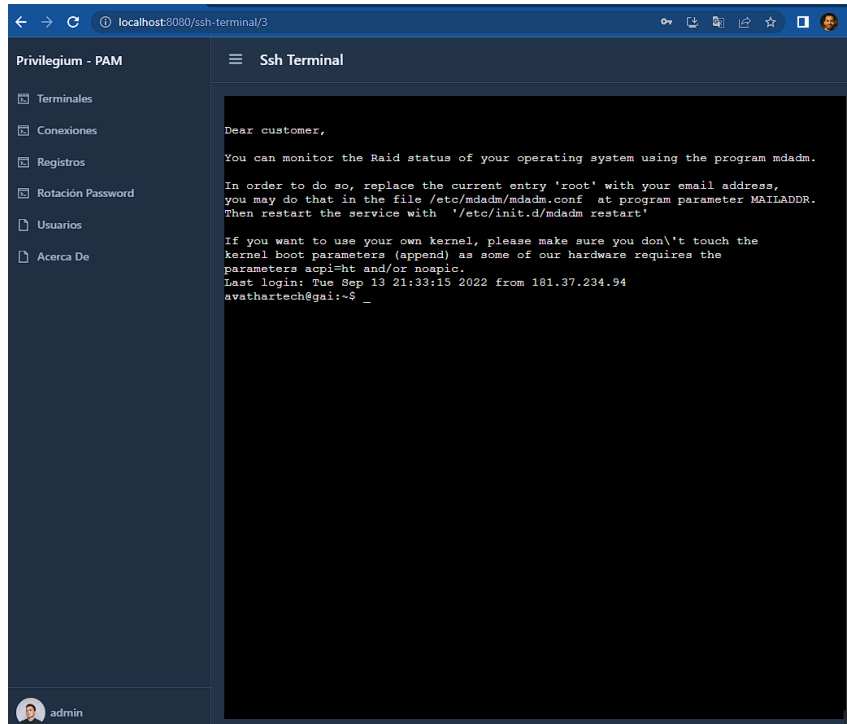


Imagen 23 - Registro de conexiones de los usuarios (Elaboración propia)

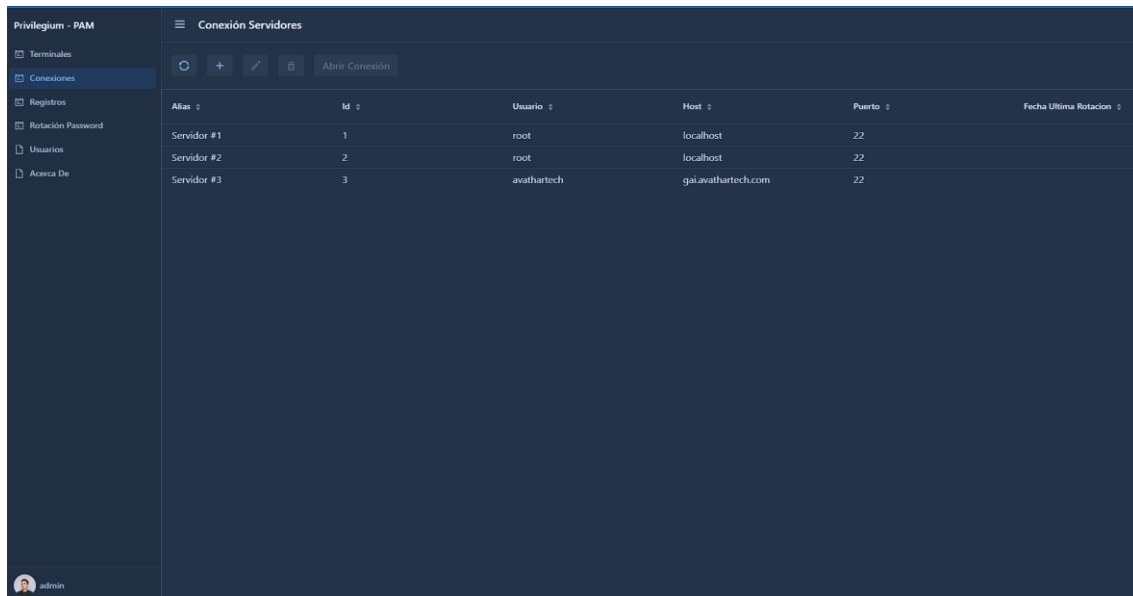
42

Imagen 24 - Registro de los comandos y respuesta de la sesión. (Elaboración propia)

Registro de Eventos			
ID: 1 - Usuario: admin - Terminal: Servidor #3 - Fecha: 2022-09-13 22:13:25.802			
Id	Fecha Creacion	Tipo Registro	log
1	2022-09-13 22:13:28.533	RESPUESTA	<p>if you want to use your own kernel, please make sure you don't touch the kernel boot parameters (append) as some of our hardware requires the parameters acpi=ht and/or noapic. Last login: Tue Sep 13 21:33:15 2022 from 181.37.234.94</p>

Para la gestión de las terminales, los usuarios con el rol administrador pueden realizar esa operación. En la Imagen 25, se presenta la vista para la creación, edición, visualización y eliminación de las terminales.

Imagen 25 - Administración de terminales (Elaboración propia)



En la Imagen 26, podemos resaltar que el registro de la contraseña por parte de los administradores puede ser visualizada en caso de que sea requerido acceder de forma directa sin utilizar el software PAM o si la funcionalidad de rotación de contraseña fue ejecutada por el sistema. La rotación de la contraseña es realizada cuando se cumple la fecha programada y es reprogramada cada semana una vez cambiada.

Imagen 26 - Ventana de edición de las terminales (Elaboración propia)

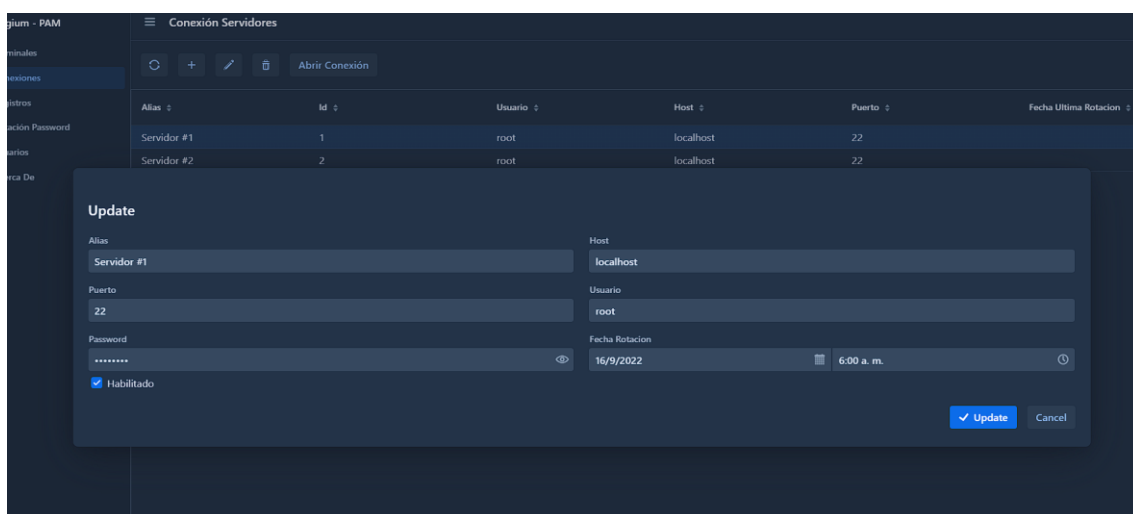
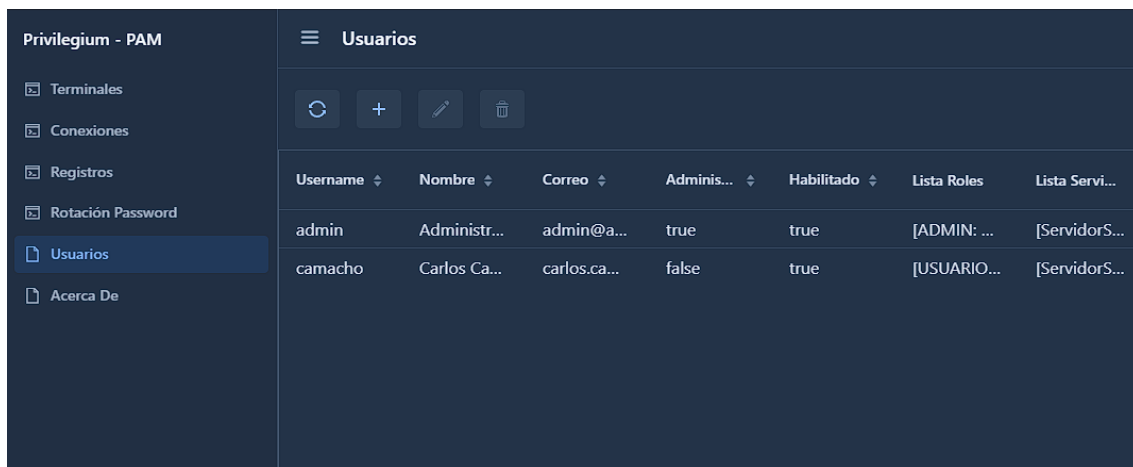


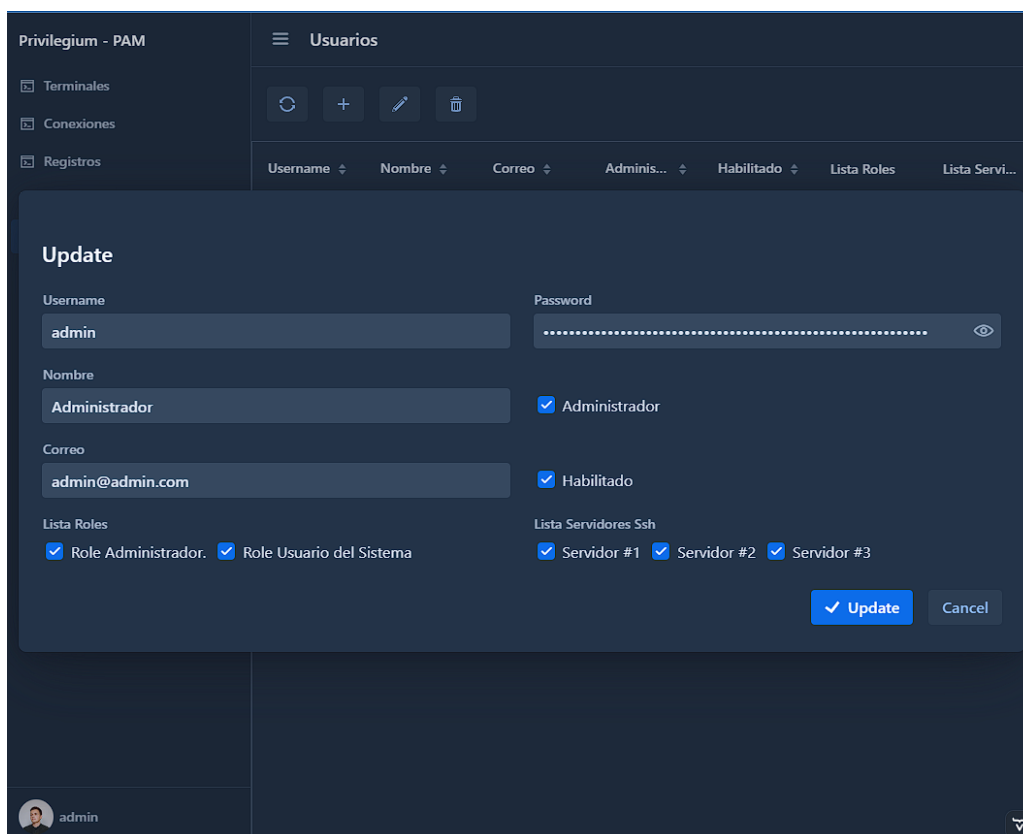
Imagen 28 - Gestión de los usuarios (Elaboración propia)



Username	Nombre	Correo	Adminis...	Habilitado	Lista Roles	Lista Servi...
admin	Administr...	admin@a...	true	true	[ADMIN: ...	[ServidorS...
camacho	Carlos Ca...	carlos.ca...	false	true	[USUARIO...	[ServidorS...

La ventana de edición, presentada en la Imagen 29, nos permite gestionar las informaciones relevantes de un usuario, la asignación de los roles y los servidores que tendrá acceso.

Imagen 29 - Ventana de edición de los usuarios (Elaboración propia)



Privilegium - PAM

Terminales
Conexiones
Registros
Rotación Password
Usuarios
Acerca De

Usuarios

+

Username	Nombre	Correo	Adminis...	Habilitado	Lista Roles	Lista Servi...
admin	Administrador	admin@admin.com				

Update

Username

admin

Nombre

Administrador

Correo

admin@admin.com

Lista Roles

☒ Role Administrador.
☒ Role Usuario del Sistema

Lista Servidores Ssh

☒ Servidor #1
☒ Servidor #2
☒ Servidor #3

Password

.....

☒ Administrador
☒ Habilitado

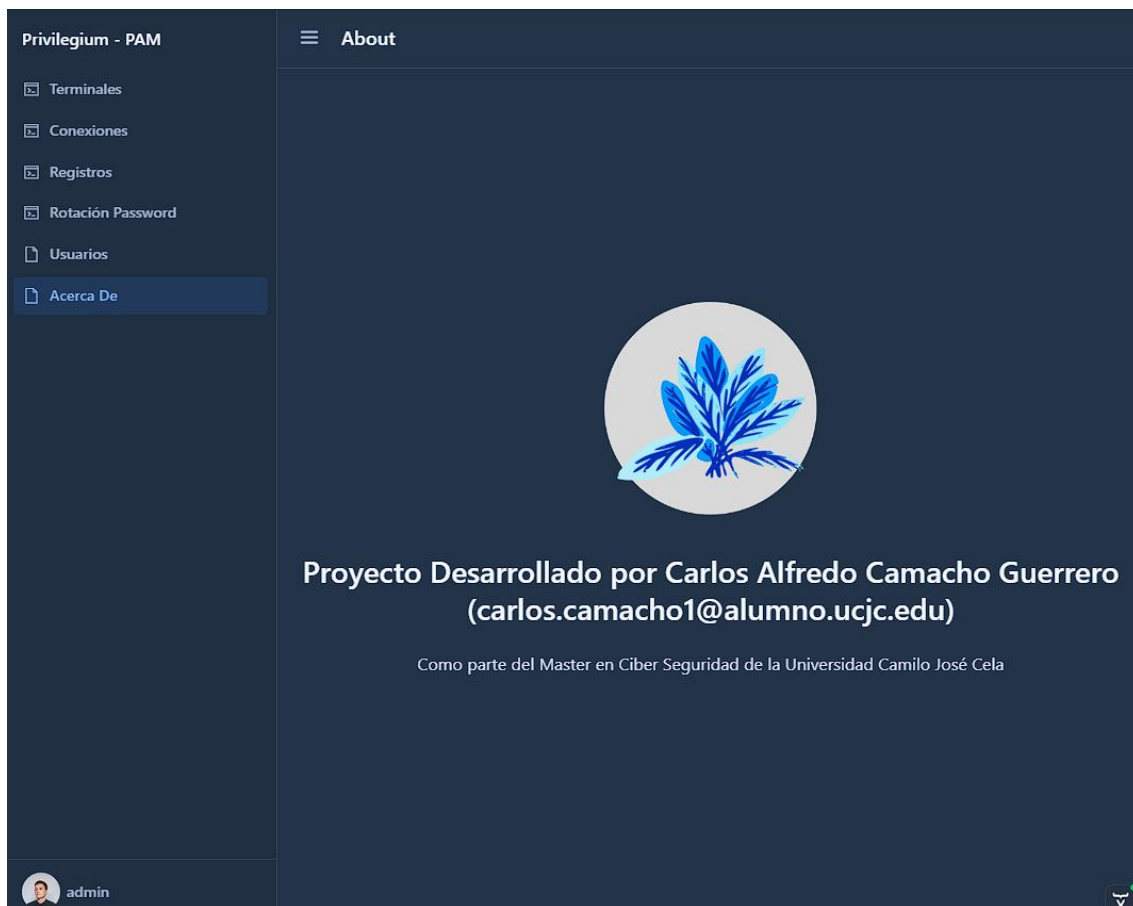
☒ Update

admin

46

La vista de Acerca De, presenta la información de autoría del proyecto, ver Imagen 30.

Imagen 30 - Vista Acerca De (Elaboración propia)



El software está diseñado para ser responsivo, adaptándose a la vista del dispositivo que utilice el usuario para acceder (teléfono inteligente, tableta, PC o televisor). En las imágenes 31, 32, 33 y 34 podemos visualizar algunas de las vistas adaptadas al uso del teléfono móvil.

Imagen 31 - Acceso software utilizando teléfono inteligente (Elaboración propia)

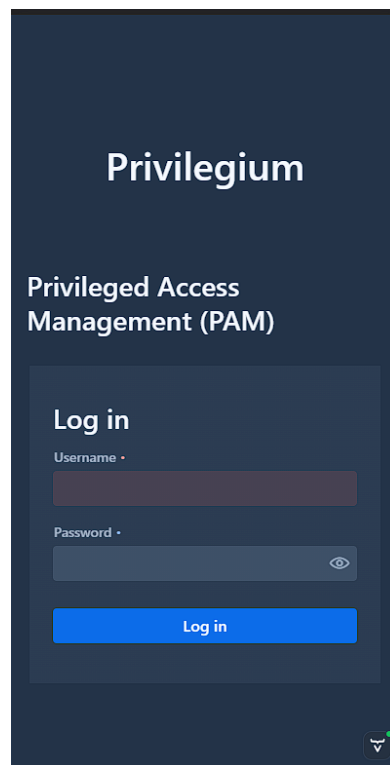


Imagen 32 - Vista de inicio utilizando teléfono inteligente (Elaboración propia)



Imagen 33 - Menú de las funcionales utilizando teléfono inteligente (Elaboración propia)

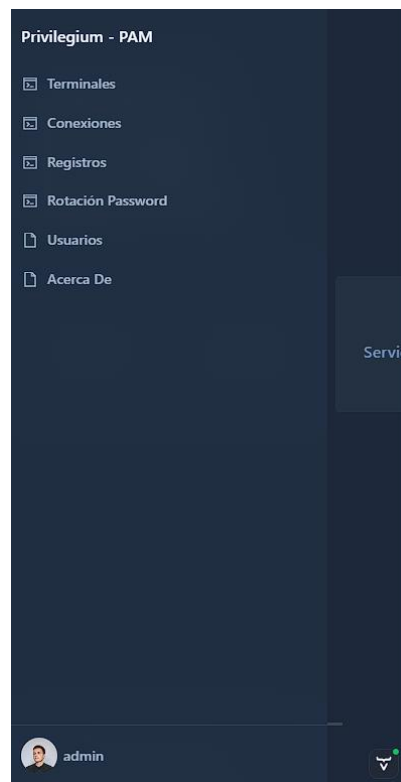


Imagen 34 - Vista de gestión de terminales utilizando teléfono inteligente (Elaboración propia)

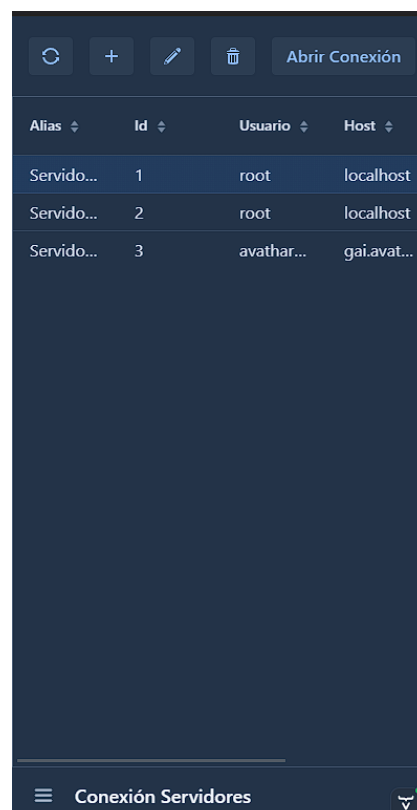


Imagen 35 - Actualización de terminales utilizando teléfonos inteligentes(Elaboración propia)



The image shows a mobile application interface for managing server connections. At the top, there is a header bar with icons for refresh, add, edit, and delete, along with a button labeled "Abrir Conexión". Below this is a form titled "Update" with the following fields:

- Alias:** A text input field containing "Servidor #1".
- Host:** A text input field containing "localhost".
- Puerto:** A text input field containing "22".
- Usuario:** A text input field containing "root".
- Password:** A password input field with a masked password "....." and a toggle icon for visibility.
- Fecha Rotacion:** A date and time selection field showing "15/9/2022" and "3:00 a. m." with calendar and clock icons.
- Habilitado:** A checkbox that is checked, labeled "Habilitado".

At the bottom of the form, there are two buttons: a blue "✓ Update" button and a grey "Cancel" button. The bottom of the screen features a navigation bar with a menu icon, the text "Conexión Servidores", and a status icon.

7.1. Publicación del código fuente

El repositorio de control de versiones del proyecto se encuentra alojado en la plataforma de GitHub, donde puede ser accedido mediante la dirección web <https://github.com/vacax/privilegium>. El proyecto cuenta con la documentación técnica necesaria para la compilación y ejecución del software, en la Imagen 36.

Imagen 36 - Documentación del repositorio de control de versiones del proyecto (Elaboración propia).



El software permite la ejecución mediante uso de contenedores de software basado en Docker y dispone del script de inicialización de los servicios utilizando Docker Compose (ver Tabla 8). Para el arranque de todos los servicios que interactúan con el software, pueden ejecutar los comandos presentados en la Tabla 5, contando como requisito la instalación y ejecución del motor de Docker y Docker Compose en el equipo anfitrión.

Tabla 5 - Comando ejecución Docker Compose

```
docker-compose up -d
```

8. Conclusiones

El uso de un sistema de gestión de acceso privilegiado (PAM, por sus siglas en inglés) en las empresas, permite el control de los accesos a los servicios que son administrados por la solución, brinda una trazabilidad de las operaciones que realizan los usuarios, que en la mayoría de los casos operan con cuentas con alto nivel de privilegios y cualquier error por parte del usuario puede provocar un alto impacto en los sistemas que operan.

El uso del PAM permite que las políticas de rotación de contraseñas y confidencialidad de estas sean gestionadas por un número reducido de usuario, minimizando los accesos no autorizados por los usuarios operativos. Cada sesión abierta por los usuarios es auditada por la solución, permitiendo un análisis en caso de algún uso indebido por parte de los usuarios administradores o brindado la posibilidad de bloquear en línea los comandos que puedan no estar permitidos su ejecución.

El PAM desarrollado permite el acceso SSH a servidores Linux vía el componente de emulación de terminal web. Las interacciones entre el servidor y el usuario son registradas en la base de datos, discriminando los comandos enviados por el usuario y las respuestas de la sesión abierta.

Las contraseñas para la conexión con la terminal son gestionadas por la aplicación y visualizadas por los usuarios administradores. La solución permite el rotado de contraseñas de forma automática, definiendo un tiempo para aplicar el rotado de la aplicación por parte de los usuarios administradores. Los usuarios finales nunca tienen conocimiento de la contraseña utilizada para acceder a la terminal a los cuales tienen permisos, permitiendo la confidencialidad de las contraseñas.

El software permite el acceso vía web para la gestión administrativa y de operación de los usuarios. Permite la adaptación de las vistas en función del dispositivo utilizado. Para fines de implementación fue incluido la creación de imágenes de Docker y el archivo de ejecución basado en Docker Compose para facilitar la puesta en producción o prueba de la solución, demostrando la viabilidad de la solución con el uso de contenedores de software.

El objetivo general y los específicos indicados en el trabajo realizado fueron cubiertos y desarrollados en la solución PAM. El software es publicado software libre

como un aporte a la comunidad científica y permitir el uso de terceros o la incorporación de nuevas funcionalidades por la comunidad.

9. Limitaciones y Líneas futuras

9.1. Limitaciones

Entre las limitaciones encontradas a la hora de elaborar el TFM fue encontrar un componente de software o librería que se integre directamente con el *framework* Vaadin Flow para interpretar la comunicación del protocolo SSH y la sesión abierta con el equipo conectado a la hora de realizar un flujo de datos. Es necesario desarrollar un componente que especifique que aborde las diferentes casuísticas presentadas para contar con terminal real del protocolo SSH.

9.2. Líneas futuras de Investigación

La plataforma desarrollada cubre los aspectos fundamentales definidos en un PAM, pero puede beneficiarse con la incorporación de las siguientes funcionalidades:

- Grabación de la sesión SSH abierta por el usuario.
- Bloqueo de comandos no autorizados por el administrador al usuario.
- Dashboard presentando información de estadística de los usuarios.
- Controlar el acceso a base de datos.
- Control el acceso a sistemas Remote Desktop Protocol (RDP).
- Habilitar el acceso mediante la autenticación de dos factores (A2F)
- Habilitar control de horario para el acceso de los usuarios a servicios en específicos.

10. Glosario de términos

1. **Maven:** Apache Maven es una herramienta de comprensión y gestión de proyectos de software. Basado en el concepto de un modelo de objetos de proyecto (POM), Maven puede administrar la creación, los informes y la documentación de un proyecto desde una pieza central de información. Apache Maven (2022).
2. **Control de versiones:** son software que ayudan a realizar un seguimiento de los cambios realizados en el código a lo largo del tiempo. Microsoft (2022).
3. **Máquina Virtual (VM):** Es un sistema de simulación de software que proporciona un entorno de simulación para la ejecución simultánea de varias computadoras dentro de otra. Ciset (2022).
4. **Protocolo:** Es un conjunto formal de reglas y normas. Estos controlan tanto la coordinación como la interacción entre los diversos dispositivos en la red o sistema de comunicación. El propósito es que puedan transmitirse datos entre sí, Imagar Solution Company (2021).
5. **Javascript:** Es un lenguaje de programación o scripting que permite realizar funciones complejas en páginas web, Mozilla (S.f.)
6. **Github:** Es una empresa sin fines de lucro que brinda servicios de almacenamiento en la nube, Kinsta (2020).
7. **Interfaz de programación de aplicaciones (API, por sus siglas en inglés):** Middleware que permite que las aplicaciones se comuniquen entre sí, Mulesoft (S.f).
8. **Script:** Un programa o serie de instrucciones que es interpretado o ejecutado por otro programa en lugar de un procesador de computadora, Techtarget (2021).
9. **Bash:** es una interfaz legítima para su computadora, y no es solo para administradores de servidores y programadores. Puede ser su escritorio, su procesador de textos, su aplicación de edición de gráficos y mucho más, Red Hat (2022).
10. **User Datagram Protocol (UDP):** es un protocolo muy popular para el tráfico de voz y video, Cloudflare (2022).
11. **Aplicación web:** es un software de aplicación alojado en un servidor remoto y distribuido a través de una interfaz de navegador, TechTarget (2019).

12. **Modelo-Vista-Controlador (MVC):** es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. (Universidad Alicante, 2022)

11. Referencias

1. Glosario de CyberArk. Gestión del Acceso con Privilegios (PAM). CyberArk. Recuperado el 20 de marzo del 2022. <https://www.cyberark.com/es/what-is/privileged-access-management/>
2. Morey J. Haber, Darran Rolls. (2022) Identity Attack Vectors: Implementing an Effective Identity and Access Management Solution, ISBN 978-1-4842-5164-5.
3. Jason Garbis, Jerry W. Chapman. (2021). Zero Trust Security: An Enterprise Guide. ISBN 978-1-4842-6701-1.
4. Listado de PAM por Gartner. Recuperado el 15 de mayo del 2022. <https://www.gartner.com/reviews/market/privileged-access-management>.
5. Solución CyberArk en AWS Marketplace. Recuperado el 15 de mayo del 2022. <https://aws.amazon.com/marketplace/pp/prodview-dbkqcf5ilt26>.
6. Listado de proyectos de código abierto sobre PAM publicados en la plataforma de Github. Recuperado el 22 de agosto del 2022. <https://github.com/search?q=Privileged+Access+Management&type=repositories>.
7. Proyecto de código abierto sobre un sistema PAM llamado Vault Project. Recuperado el 22 de agosto del 2022. <https://www.vaultproject.io/>.
8. Proyecto de código abierto sobre un sistema PAM llamado Teleport. Recuperado el 22 de agosto del 2022. <https://goteleport.com/>.
9. Proyecto de código abierto sobre un sistema PAM llamado OpenAKC. Recuperado el 22 de agosto del 2022. <https://github.com/netlore/OpenAKC>.
10. Jon Byous. (1998) JAVA TECHNOLOGY: THE EARLY YEARS. Recuperado el 22 de agosto del 2022. <https://web.archive.org/web/20050420081440/http://java.sun.com/features/1998/05/birthday.html>.
11. El País (2009, abril, 20). Oracle adquiere Sun Microsystems por 5.710 millones. https://elpais.com/tecnologia/2009/04/20/actualidad/1240216080_850215.html
12. Lindholm, Yellin, Bracha y Buckley. (2014). The Java® Virtual Machine Specification, Java SE 8 Edition. ISBN 978-0-13-390590-8.

13. Índice TIOBE de lenguajes de programación. Recuperado el 17 de mayo del 2022.
<https://www.tiobe.com/tiobe-index/>
14. Apache Mina SSHD. Recuperado el 23 de agosto del 2022.
<https://mina.apache.org/sshd-project/>
15. Java Secure Channel (JSch). Recuperado el 23 de agosto del 2022.
<http://www.jcraft.com/jsch/>
16. SSHJ - SSHv2 library for Java. Recuperado el 23 de agosto del 2022.
<https://github.com/hieronymus/sshj>
17. Edix.com (2022, julio, 26). Framework.
<https://www.edix.com/es/instituto/framework/#:~:text=Un%20framework%20es%20un%20esquema,organizaci%C3%B3n%20y%20desarrollo%20de%20software>. Recuperado el 25 de agosto del 2022.
18. IONOS (2019, junio, 3). Spring: el framework para aplicaciones Java complejas.
<https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/spring-framework-la-columna-vertebral-de-java/>. Recuperado el 25 de agosto del 2022.
19. IBM Cloud Education (2020, marzo, 25). Java Spring Boot.
[https://www.ibm.com/cloud/learn/java-spring-boot#:~:text=Java%20Spring%20Boot%20\(Spring%20Boot,ability%20to%20create%20standalone%20applications](https://www.ibm.com/cloud/learn/java-spring-boot#:~:text=Java%20Spring%20Boot%20(Spring%20Boot,ability%20to%20create%20standalone%20applications). Recuperado el 25 de agosto del 2022.
20. Vaadin Team. (2019). Book of Vaadin. <https://vaadin.com/book>. Recuperado el 25 de agosto del 2022.
21. GNU (2007, noviembre, 19). GNU AFFERO GENERAL PUBLIC LICENSE.
<https://www.gnu.org/licenses/agpl-3.0.html>. Recuperado el 26 de agosto del 2022.
22. XTerm Console Addon. <https://vaadin.com/directory/component/xterm-console-addon/overview>. Recuperado el 26 de agosto del 2022.
23. Crud UI Add-on. <https://vaadin.com/directory-beta/addon/crud-ui-add-on>. Recuperado el 26 de agosto del 2022.
24. Spring Security. <https://docs.spring.io/spring-security/reference/index.html>. Recuperado el 26 de agosto del 2022.

25. Spring Framework Componentes (2022). [Imagen]. Spring.
<https://docs.spring.io/spring-framework/docs/5.0.0.RC2/spring-framework-reference/images/spring-overview.png>
26. Vaadin Flow con los componentes instanciado en el servidor (2022). [Imagen]. Vaadin.
<https://vaadin.com/docs/v22/static/721031cfafedc3513135c78aa5a6ced2/2ad7f/application-architecture.webp>
27. Azure (2022). ¿Qué es un contenedor?. <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-a-container/#overview>. Recuperado el 11 de septiembre del 2022.
28. AWS (2022). ¿Qué es Docker?. <https://aws.amazon.com/es/docker/>. Recuperado el 11 de septiembre del 2022.
29. Red Hat (2022). ¿Qué es Docker?. <https://www.redhat.com/es/topics/containers/what-is-docker>. Recuperado el 11 de septiembre del 2022.
30. Keepcoding (2022). ¿Qué es Docker Compose?. <https://keepcoding.io/blog/que-es-docker-compose/>. Recuperado el 11 de septiembre del 2022.
31. Gestor, C. (2022). Ciset. Centro de Innovación. Ciset.es. <https://www.ciset.es/glosario/492-vmware-virtualizacion>. Recuperado el 11 de septiembre del 2022
32. Mozilla. (s. f.). ¿Qué es JavaScript? Mozilla.org. Recuperado 30 de junio de 2022, https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript. Recuperado el 11 de septiembre del 2022
33. Imagar Solutions Company. (2021). ¿Qué es el protocolo en informática? Imagar Solutions Company. <https://www.imagar.com/blog-desarrollo-web/que-es-el-protocolo-eninformatica/>. Recuperado el 11 de septiembre del 2022.
34. Kinsta, Inc. (2020). ¿Qué es GitHub? Una Guía para Principiantes sobre GitHub. <https://kinsta.com/es/base-deconocimiento/que-es-github/>. Recuperado el 11 de septiembre del 2022
35. MuleSoft. (s. f.). ¿Qué es una API? (Interfaz de programación de aplicaciones). MuleSoft. <https://www.mulesoft.com/es/resources/api/what-is-an-api>. Recuperado el 11 de septiembre del 2022

36. Techtarget. (2021). Script. Techtarget.com; TechTarget
<https://www.techtarget.com/whatis/definition/script>. Recuperado el 11 de septiembre del 2022
37. Red Hat, (2022). What is bash? Opensource.com.
<https://opensource.com/resources/what-bash>. Recuperado el 11 de septiembre del 2022
38. TechTarget. (2019). What is Web Application (Web Apps) and its Benefits. SearchSoftwareQuality; TechTarget.
<https://www.techtarget.com/searchsoftwarequality/definition/Webapplication-Web-app>. Recuperado el 11 de septiembre del 2022
39. Cloudflare. (2022). What is HTTP? Cloudflare.com.
<https://www.cloudflare.com/learning/ddos/glossary/hypertexttransfer-protocol-http/>. Recuperado el 11 de septiembre del 2022.
40. Apache Maven. (2022). Apache Maven. <https://maven.apache.org/>. Recuperado el 11 de septiembre del 2022.
41. Microsoft. (2022). ¿Qué es el control de versiones?.
<https://docs.microsoft.com/es-es/devops/develop/git/what-is-version-control>.
Recuperado el 11 de septiembre del 2022.
42. Universidad Alicante. (2022). Modelo vista controlador (MVC).
<https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. Recuperado el 11 de septiembre del 2022.
43. Redeszone. (2022). Protocolos de redes: la guía completa con todos los protocolos básicos. <https://www.redeszone.net/tutoriales/internet/protocolos-basicos-redes/>.
Recuperado el 11 de septiembre del 2022.

12. Anexos

Tabla 6 - Licencia AGPL V3

Tabla licencia utiliza en el Software – AGPL v3						
Privilegium	-	Privileged	access	management	for	web access
Copyright	(C)	2022,	Carlos	Alfredo	Camacho	Guerrero.
<carlos.camacho1@alumno.ucjc.edu>						
<p>This program is free software: you can redistribute it and/or modify it under the terms of the GNU Affero General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.</p> <p>This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Affero General Public License for more details.</p> <p>You should have received a copy of the GNU Affero General Public License along with this program. If not, see <http://www.gnu.org/licenses/agpl-3.0.html>.</p>						

Tabla 7 - Archivo Docker para la creación de la imagen

Archivo Docker
<pre># Incluyendo la característica de Stage Builds para la compilación de las dependencias FROM maven:3-openjdk-17-slim as build RUN curl -sL https://deb.nodesource.com/setup_14.x bash - RUN apt-get update -qq && apt-get install -qq --no-install-recommends nodejs # Cambiando el usuario de ejecución en el contenedor para evitar temas de seguridad. RUN useradd -m myuser WORKDIR /usr/src/app/ RUN chown myuser:myuser /usr/src/app/ USER myuser # Copiando del archivo pom.xml para obtener la dependencia de la aplicación en la etapa de compilación. COPY --chown=myuser pom.xml ./ RUN mvn dependency:go-offline -Pproduction # Copiando todos los archivos necesarios del proyecto con el usuario sin privilegios. COPY --chown=myuser:myuser src src COPY --chown=myuser:myuser frontend frontend COPY --chown=myuser:myuser package.json ./ # Copiando los archivos propios del framework de Vaadin COPY --chown=myuser:myuser package-lock.json* pnpm-lock.yaml* webpack.config.js* ./ # Creando el paquete de producción RUN mvn clean package -DskipTests -Pproduction</pre>

```
# Creando la etapa de ejecución con el archivo compilado.
FROM openjdk:17-jdk-slim
COPY --from=build /usr/src/app/target/*.jar /usr/app/app.jar
RUN useradd -m myuser
USER myuser
EXPOSE 8080
CMD java -jar /usr/app/app.jar
```

Tabla 8 - Archivo Docker Compose

Archivo Docker Compose
<pre>version: "3.7" services: # Base de datos. base-datos: restart: always image: mysql:8.0.30 env_file: - env-demo.env environment: - MYSQL_ROOT_PASSWORD=\$MYSQL_PASSWORD - MYSQL_USER=\$MYSQL_USER - MYSQL_DATABASE=\$MYSQL_DB volumes: - datos_pam:/var/lib/mysql # Aplicación web para gestionar la base de datos. phpmyadmin: image: phpmyadmin:5.2.0-apache env_file: - env-demo.env ports:</pre>

```
- ${PUERTO_PHPMYADMIN}:80
environment:
  - PMA_ARBITRARY=1
depends_on: #indica que primero debe subir los servicios indicados.
  - base-datos
# Aplicación web - PAM
app-web:
  restart: always
  image: web-pam
  env_file:
    - env-demo.env
  environment:
    - SPRING_PROFILES_ACTIVE=${AMBIENTE}
  ports:
    - "${PUERTO_APP}:8080"
  depends_on:
    - base-datos
# Imagenes con SSH para prueba
ssh-servidor-1:
  restart: always
  image: linuxserver/openssh-server:version-8.8_p1-r1
  hostname: ssh-server-1
  environment:
    - USER_PASSWORD=12345678
    - USER_NAME=camacho
    - SUDO_ACCESS=true
    - PASSWORD_ACCESS=true
  ports:
    - "2222:2222"
ssh-servidor-2:
  restart: always
```


image: linuxserver/openssh-server:version-8.8_p1-r1

hostname: ssh-server-2

environment:

- USER_PASSWORD=12345678
- USER_NAME=carlos
- SUDO_ACCESS=true
- PASSWORD_ACCESS=true

ports:

- "2223:2222"

ssh-servidor-3:

restart: always

image: linuxserver/openssh-server:version-8.8_p1-r1

hostname: ssh-server-3

environment:

- USER_PASSWORD=12345678
- USER_NAME=ucjc
- SUDO_ACCESS=true
- PASSWORD_ACCESS=true

ports:

- "2224:2222"

volumes:

datos_pam: