



Tecnológico Nacional de México Campus Querétaro

Reporte Final de Residencia Profesional

Creación de sistema de gestión para la planificación de recursos

Que presenta:

Daniel Ignacio Arellano Ochoa

Estudiante de la carrera:

Ingeniería en Sistemas Computacionales

Asesor:

Guillermo Fernandez Romero

Periodo:

Enero-Junio 2024

EVALUACIÓN DE RESIDENCIA PROFESIONAL

Área:	DEPARTAMENTO DE SISTEMAS Y COMPUTACION		Expediente:	DSyC-RP-1-024/2024	
Residente:	DANIEL IGNACIO ARELLANO OCHOA				
Carrera:	Ingeniería en Sistemas Computacionales		No. de Control:	19141118	
Proyecto:	Creación de sistema de gestión para la planificación de recursos				
Periodo de realización:	Inicio:	12/Feb/2024	Término:	31/May/2024	
Empresa:	MC MICROCOMPUTACION SA DE CV				
CRITERIOS A EVALUAR			PUNTUACIÓN		
En qué medida el Residente cumple con lo siguiente:			Máxima	Asignada	
Asesor Externo	1. Asiste puntualmente con el horario establecido.		5	5	
	2. Trabaja en equipo.		10	10	
	3. Tiene iniciativa para ayudar en las actividades encomendadas.		10	10	
	4. Organiza su tiempo y trabaja sin necesidad de una supervisión estrecha.		5	5	
	5. Realiza mejoras al proyecto.		10	10	
	6. Cumple con los objetivos correspondientes al proyecto.		10	10	
Asesor Interno	1. Mostró responsabilidad y compromiso en la residencia profesional.		5	5	
	2. Realizó un trabajo innovador en su área de desempeño.		10	10	
	3. Aplica las competencias para la realización del proyecto.		10	10	
	4. Es dedicado y proactivo en los trabajos encomendados.		10	10	
	5. Cumple con los objetivos correspondientes al proyecto.		10	10	
	6. Entrega en tiempo y forma el informe técnico.		5	5	
CALIFICACIÓN FINAL			100		
NIVEL DE DESEMPEÑO:		Excelente			
OBSERVACIONES:					

Tabla de Valoración		
Desempeno	Niveles de Desempeno	Valoración Numérica
Competencia alcanzada	Excelente	De 95 a 100
	Notable	De 85 a 94
	Bueno	De 75 a 84
	Suficiente	De 70 a 74
Competencia No alcanzada	Insuficiente	NA (No Alcanzada)

GUILLERMO FERNANDEZ ROMERO
ASESOR INTERNO

ING. DAVID ISAÍAS ROSALES CASILLAS
ASESOR EXTERNO

MC
MICROCOMPUTACIÓN

MC MICROCOMPUTACION,
S.A. DE C.V.
MCM8906284HA

Original: Coordinador de carrera, Residente, Asesor Interno, Servicios Escolares, Departamento Académico

ITQ-AC-PO-007-07

Rev. 1



Av. Tecnológico s/n esq. Mariano Escobedo, Col. Centro, C.P. 76000, Querétaro, Querétaro.
Plantel Centro tel. 01(442) 2274400 ext. 4404 y Plantel Norte tel. 01(442) 2435554
tecnm.mx | queretaro.tecnm.mx





Índice general

Introducción	1
Capítulo 1. Generalidades de la empresa	2
1.1 Datos generales.....	2
1.2 Reseña histórica de la empresa.....	2
1.3 Organigrama de la empresa.	3
1.4 Misión, Visión y Política.	3
1.5 Principales productos y clientes.....	4
1.6 Premios y certificaciones.	6
1.7 Caracterización del área de Open Source.	6
1.8 Ubicación de la residencia	8
Capítulo 2. Planteamiento del problema	9
2.1 Antecedentes y definición del problema.....	9
2.2 Objetivos.....	9
2.2.1 Generales.	9
2.2.2 Específicos.	10
2.3 Justificación.	11
2.4 Alcance.	11
Capítulo 3. Marco Teórico	12
Código Abierto	12
Red Hat.....	12
CentOS.....	13
Fedora	13
Red Hat Subscription Manager.....	14
Network Manager.....	14
SELinux.....	15
Partición de Disco.....	16
Esquema de partición MBR	16



<i>Esquema de partición GPT</i>	17
<i>LVM</i>	17
<i>Dispositivos físicos</i>	18
Volúmenes físicos (PV)	18
Grupos de volúmenes (VG)	18
Volúmenes lógicos (LV)	18
<i>SCSI</i>	19
<i>NFS</i>	19
<i>RAID</i>	20
Nivel 0	21
Nivel 1	21
Nivel 4	21
Nivel 5	22
<i>Virtualización</i>	22
<i>Contenedor</i>	23
<i>Imagen de Contenedor</i>	23
<i>Docker y Podman</i>	24
<i>Script Shell Bash</i>	24
<i>Ansible</i>	25
<i>SSH</i>	26
<i>SFTP</i>	27
<i>LDAP</i>	27
<i>DHCP</i>	28
<i>PXE</i>	28
<i>SCRUM</i>	29
<i>MVC (Modelo-Vista-Controlador)</i>	31
<i>SOA</i>	31
Capítulo 4. Desarrollo del proyecto	33
4.1 Módulo “Checador”	33
4.2 Taiga	34



4.3 Mudanza de servidor	37
4.4 Scripts.....	40
4.5 Servidor PXE y DHCP.....	44
4.6 Tareas extras.....	46
Capítulo 5. Resultados.....	47
5.1 Módulo "Checador"	48
5.2 Taiga.....	48
5.3 Mudanza de servidor	49
5.4 Scripts.....	49
5.5 Servidor PXE y DHCP.....	50
Capítulo 6. Competencias desarrolladas y/o aplicadas.....	51
6.1 Competencias generales	51
6.2 Competencias específicas	52
Conclusiones.....	53
Recomendaciones.....	53
Referencias.....	54



Índice de Ilustraciones.

Ilustración 1: Organigrama de MC.	3
Ilustración 2: Premios ganados por MC.	6
Ilustración 3: Pantalla del módulo "Checador"	34
Ilustración 4: Pantalla de inicio de Taiga	35
Ilustración 5: Captura de los distintos estados del proyecto de Intranet.	35
Ilustración 6: Ejecución de Taiga por medio de contenedores.	36
Ilustración 7: Dockerfile para integrar LDAP al servidor Taiga.	36
Ilustración 8: Opciones de configuración adicional para el LDAP en Taiga.	37
Ilustración 9: Pantalla de inicio del hipervisor Ovirt.	38
Ilustración 10: Parte de un Ansible playbook.	39
Ilustración 11: Script para configurar un bond.	41
Ilustración 12: Script en Python.	43
Ilustración 13: Configuración del menú del servidor PXE.	44
Ilustración 14: Cliente PXE cargando la configuración del servidor.	45
Ilustración 15: Vista de selección configurada en el servidor PXE.	45
Ilustración 16: Monitoreo del servidor DHCP y PXE.	46



Introducción

En este informe se presentará el trabajo realizado durante el periodo de residencia profesional en la empresa MC Micro Computación, centrado en la creación de la intranet de la empresa. El proyecto incluyó la implementación de diversas características, destacando una aplicación web que permite al área de recursos humanos obtener informes completos sobre las asistencias de los empleados elegibles para el bono de puntualidad. Además, se instalaron varios servidores nuevos y se configuraron servidores existentes para adaptarse a las nuevas implementaciones, ofreciendo así servicios que mejoran el rendimiento de los empleados en distintas áreas de la empresa. Paralelamente, se llevaron a cabo investigaciones y se documentaron las mejores prácticas en codificación y configuración de servidores.

El propósito del informe es proporcionar evidencia clara del trabajo realizado durante la residencia profesional en MC Micro Computación, detallando los antecedentes del problema, los objetivos generales y específicos, la justificación, el alcance, el desarrollo y los resultados del proyecto. Se describe cada tarea realizada, especificando los protocolos, herramientas, comandos y procesos empleados para lograr una ejecución óptima y eficiente.



Capítulo 1. Generalidades de la empresa.

1.1 Datos generales.

- Razón social: MC Micro computación S.A C. V
- Ubicación: Avenida Universidad 332A-Pte, Centro, 76000 Santiago de Querétaro, Qro.
- Giro Organizacional: Servicio
- Tamaño de la empresa: Mediana
- Rama: Informática

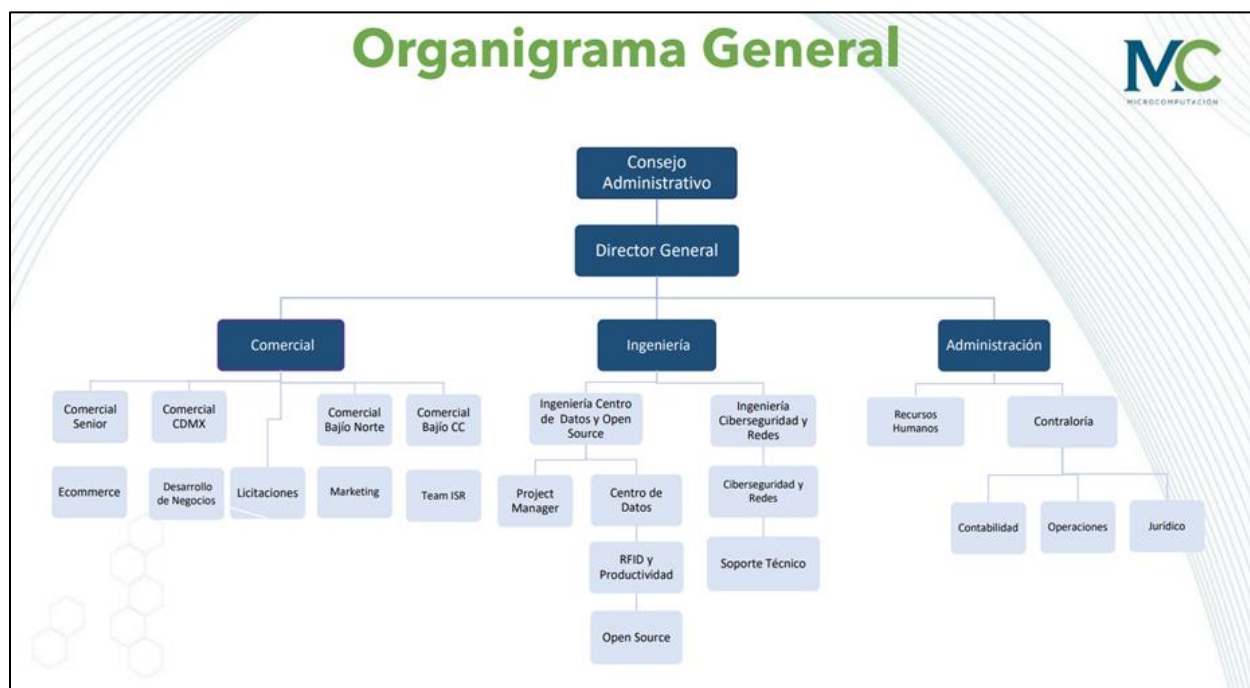
1.2 Reseña histórica de la empresa.

MC Micro computación inició operaciones el 28 de junio de 1989 con domicilio en Querétaro, fundada por un grupo de emprendedores apasionados por las nuevas tecnologías de la informática. Comenzó brindando soluciones tecnológicas innovadoras tanto para empresas de sector privado como organizaciones gubernamentales.

Actualmente cuenta con casi 35 años atendiendo toda la República Mexicana principalmente en los mercados de Querétaro, Ciudad de México, San Luis Potosí, Guanajuato, Morelia, Jalisco, Monterrey, Tabasco. Brindando soluciones avanzadas de tecnologías de la Información, servicios de mantenimiento y soporte técnico especializado, volviéndose un importante distribuidor de bienes y servicios en el área de redes, comunicaciones, tecnologías de educación, seguridad y otras afines.

1.3 Organigrama de la empresa.

Ilustración 1: Organigrama de MC.



1.4 Misión, Visión y Política.

Misión: Somos consultores y proveedores en soluciones de Tecnología de la información, respondemos ágilmente y somos el socio más confiable con clientes, proveedores, colaboradores y la Sociedad.

Visión: Ser una empresa líder en todas las regiones en las que participamos, reconocidos como un equipo experto con un profundo conocimiento de la industria. Ofreciendo soluciones más eficientes y reflejando nuestros valores en cada proyecto



Políticas:

- Política de Integridad
- Política de viáticos
- Política de vacaciones
- Política de Regalos y Obsequios

Valores y principios:

Confianza: Cumplimos todo lo que prometemos.

Ética de trabajo: Trabajamos en equipo con responsabilidad, sentido de urgencia y damos los mejores resultados. Es la única forma de cumplir nuestro propósito.

Integridad: Tomamos la responsabilidad de nuestras acciones y hacemos lo correcto.

Flexibilidad: Nos adaptamos a las circunstancias, respetando las normas y políticas.

Respeto: Reconocemos y valoramos cada idea y cada uno de nuestros compañeros.

1.5 Principales productos y clientes.

Algunos de los clientes son:

- Secretaria de planeación y finanzas del estado de Querétaro
- Gobierno de CDMX
- BIG BOLA
- AGRO ASEMEX
- SAMSUNG
- Cecoban
- REDPACK
- HITACHI
- Kelloggs
- PROMEX



- TRAXION
- AVERY DENNISON
- SCANIA
- KALTEX
- Caja de Ahorro
- Michael Page
- ANDREA
- CONCAMIN
- PPG Industries
- YUTAKA
- ABC Technologies
- JAFRA
- AUTOCOM
- AXD
- HARMAN
- Instituto Tecnológico de San Luis Potosí
- NEATON AUTO MEXICANA
- Instituto Mexicano de Transporte

Principales productos:

- Soluciones en Data Center.
- Soluciones en Desarrollo de Aplicaciones.
- Soluciones de Servidores.
- Soluciones de infraestructura y su implementación.
- Soluciones de logística.
- Soluciones de Respaldos.
- Soluciones de Modularidad.
- Soluciones en equipos de cómputo.

1.6 Premios y certificaciones.

Soles Mayores Ventas Sucursal Qro. - 2022

Mejor TeamMate León – 2022

Ilustración 2: Premios ganados por MC.



1.7 Caracterización del área de Open Source.

Descripción del área

El área de Open Source es la encargada de la evaluación, selección e implementación de software de código abierto para cubrir las necesidades de la empresa o de los clientes de esta. Esto incluye sistemas operativos como Red Hat Enterprise, base de datos como PostgreSQL, herramientas de desarrollo como Git y Gitlab, herramientas de gestión de proyectos como Taiga, herramientas de gestión de infraestructura y virtualización como ManagerIQ u Ovirt, automatización de procesos de infraestructura como Ansible, además del manejo de contenedores y Dockerfiles.



Actividades del área:

- Selección y evaluación de tecnologías: Implica en la investigación de diferentes tecnologías y herramientas de código abierto que puedan solucionar necesidades de la empresa, al evaluarlas se busca determinar cuáles son las más adecuadas para la empresa.
- Implementación de Tecnologías: Tras realizar la investigación y evaluación, se buscará poner en funcionamiento la tecnología más adecuada a las necesidades de la empresa.
- Monitoreo y revisión de Software: Constantes evaluaciones y monitoreos del software implementado para detectar posibles fallos, actualizaciones del software o posibles mejoras.
- Configuración de procesos automatizados: Desarrollo de Ansible playbook para la automatización de procesos en relación con la infraestructura de tanto equipos físicos como virtuales.

Funciones del área:

- Gestión estratégica al evaluar, desarrollar y ejecutar la implementación de tecnologías de código abierto en la infraestructura disponible, para encontrar una mejor optimización y automatización de procesos dentro de la empresa y con sus clientes.
- Monitoreo constante de los distintos servidores para verificar su buen funcionamiento y acceso por parte de los usuarios requeridos.
- Documentación constante acerca de las nuevas tecnologías para poder evaluar posibles actualizaciones en servicios o tecnologías ya establecidas en la empresa.
- Comunicación constante con las distintas áreas de la empresa para conseguir solucionar problemáticas de estas.



1.8 Ubicación de la residencia.

El proyecto se desarrolla en el área de Open Source en las oficinas de MC ciudad Santiago de Querétaro en el lugar de trabajo asignado.



Capítulo 2. Planteamiento del problema.

2.1 Antecedentes y definición del problema.

En el entorno actual donde las empresas buscan adentrarse en el mundo tecnológico para conseguir mayor productividad y alcanzar sus objetivos, la empresa MC Micro computación detectó una gran oportunidad para agilizar procesos repetitivos que consumen una gran cantidad de tiempo de los colaboradores, además de encontrar soluciones óptimas para conseguir una mejor organización en la realización de proyectos.

Para conseguir lo anterior se ha buscado optimizar procesos para disminuir el tiempo que se emplea en estos mismos, con el fin agilizar los procesos además de poder utilizar el tiempo ahorrado en nuevas actividades que permitan alcanzar nuevos objetivos. La empresa ha encontrado en el uso de la tecnología la manera de mejorar estos procesos mediante la automatización, uso de los lenguajes de programación y nuevas tecnologías como lo son contenedores y herramientas de orquestación de sistemas como Ansible, utilizando en su conjunto softwares de servicios ejecutados desde servidores locales permitiendo tener una intranet más completa. Con estos puntos se pretende poder automatizar las tareas repetitivas, así como procesos complejos dentro de la institución.

2.2 Objetivos.

2.2.1 Generales.

- Detectar procesos y tareas repetitivas dentro de la empresa que puedan ser automatizadas disminuyendo como mínimo el 50% del tiempo invertido.
- Creación y desarrollo de servidores que brindan servicios a la intranet.
- Configuración de servicios ya establecidos para adaptarse a la implementación de nuevas tecnologías.



2.2.2 Específicos.

- Analizar los procesos que realizan los diferentes departamentos que pudieran ser automatizados.
- Logística y dirección, para poder entender los flujos de trabajo que se realizan en esas áreas.
- Procesar la información obtenida del análisis para cotejar las tareas repetitivas y procesos complejos.
- Generar una solución optimizada para resolver los problemas encontrados en el análisis y procesamiento de datos a través de la programación de microservicios.
- Generar una intranet que contendrá las soluciones de los procesos.
- Hacer uso de las tecnologías actuales como contenedores y virtualización para la solución de los procesos.
- Configurar de mejor manera los servicios para adaptarse a las necesidades de la empresa y usuarios.
- Reconfigurar procesos y servicios ya establecidos dentro de la empresa para adaptarse a las nuevas implementaciones.
- Mantener estándares de seguridad en todo lo realizado, tanto en programación como en la creación de servidores.



2.3 Justificación.

Actualmente el mercado de las empresas es cada vez más competitivo, por lo que las empresas buscan nuevas estrategias, optimización de procesos e innovaciones para el mercado, para conseguir esto las empresas deben realizar una gestión adecuada a sus recursos, especialmente el tiempo. Bajo este contexto, el proyecto presentado en este reporte surge como respuesta a la necesidad de optimizar el tiempo utilizado para realizar procesos y tareas, a través de las nuevas tecnologías, que permitan de manera directa automatizar gran parte de tareas y procesos, como también de brindar servicios que consigan una mayor eficiencia en procesos de planeación en proyectos de desarrollo.

2.4 Alcance.

El proyecto está planteado para solucionar problemas dentro de empresa por lo que todo será montado dentro de la intranet de esta, abarcando áreas como Recursos Humanos e Ingeniería, facilitando tareas de rutina dando la posibilidad de realizar procesos más complejos en menor tiempo.



Capítulo 3. Marco Teórico.

Código Abierto

“El código abierto es un modelo de producción descentralizada que permite a cualquier persona modificar y compartir tecnología debido a que su diseño es accesible de manera pública.” (Red Hat, 2022)

Actualmente “código abierto” es designado como el conjunto de principios de intercambio libre de información, creación rápida de prototipos y el desarrollo colaborativo.

Red Hat

“Red Hat es el proveedor líder de soluciones de software de código abierto, que ha adoptado un enfoque basado en la comunidad para proporcionar tecnologías fiables y de alto rendimiento de nube, Linux, middleware, almacenamiento y virtualización.” (Red Hat, 2022)

El Sistema operativo de Red Hat es basado principalmente en versiones de Fedora, con la diferencia que Red Hat decide qué paquetes incluir, qué mejoras adicionales realizar y tomar decisiones de configuración que permiten satisfacer las necesidades de sus clientes

Red Hat Enterprise Linux (RHEL) usa un modelo de distribución basado en suscripciones, respetando los lineamientos del software de código abierto, por lo que esto no es un cargo por licencia, por lo que realmente el cargo es por el soporte, el mantenimiento, las actualizaciones, los parches de seguridad, acceso al portal de clientes de Red Hat, certificaciones, entre otros servicios y ventajas. En otras palabras, el cliente paga por recibir soporte, experiencia, conocimiento, compromiso y asistencia para el momento que lo requiera.



CentOS

“CentOS es una distribución de Linux impulsada por la comunidad y derivada en gran parte de la base de código de Red Hat Enterprise Linux de código abierto y de otras fuentes.” (Red Hat, 2022)

Este sistema operativo es gratuita, fácil de instalar y dispone de una comunidad activa de usuarios voluntarios que opera de manera independiente a Red Hat, para brindar y cubrir las necesidades de los usuarios.

Fedora

“Fedora es un proyecto comunitario que produce y lanza un sistema operativo completo y gratuito basado en Linux. Red Hat patrocina a la comunidad y trabaja con sus representantes para integrar el último software ascendente en una distribución segura y de rápida actualización.” (Red Hat, 2022)

Cualquier persona puede participar en la contribución de Fedora, este por sí mismo se centra en la innovación y no en la estabilidad a largo plazo, por lo que las nuevas actualizaciones importantes ocurren cada seis meses ofreciendo cambios significativos. La comunidad de Fedora solamente brinda soporte a las versiones por un año aproximadamente, esto hace que Fedora sea un sistema operativo poco práctico para el uso empresarial. Siendo así Red Hat la mejor versión de entre Fedora, CentOS y Red Hat.



Red Hat Subscription Manager

“Red Hat Subscription Manager realiza un seguimiento de los productos Red Hat que ha adquirido su organización y de los sistemas en los que están instalados los productos. Subscription Manager establece la relación entre las suscripciones de productos que están disponibles para el sistema y los elementos de infraestructura de su negocio donde se asignan esas suscripciones.” (Red Hat, 2022)

Como se mencionó antes Red Hat funciona a través de una licencia pública GNU, sin embargo, Red Hat respalda sus productos por medio de una licencia basada en suscripción, la cual brinda soporte que incluye:

- Contenido descargable y actualizaciones.
- Acceso a la base de conocimientos.
- Soporte para su producto

Network Manager

En RHEL el servicio de red lo proporciona Network Manager, este es un “daemon” de configuración y control de red dinámico permitiendo manejar los dispositivos y conexiones de red. Algunas características importantes de Network Manager son:

Facilitar la gestión de la red: Garantiza que la conectividad de la red funcione cuando detecta que no hay configuración de red en un sistema, pero hay dispositivos de red crea conexiones temporales para proporcionar conectividad.

Fácil configuración de conexión al usuario: Ofrece gestión a través de diferentes herramientas (GUI, nmtui, nmcli).

Flexibilidad de configuración. Network Manager puede configurar alias de red, direcciones IP, rutas estáticas, información DNS y conexiones VPN, así como muchos parámetros específicos de la conexión.



SELinux

“Security-Enhanced Linux (SELinux) es una arquitectura de seguridad para sistemas Linux que permite a los administradores tener más control sobre quién puede acceder al sistema. Fue desarrollado originalmente por la Agencia de Seguridad Nacional de los Estados Unidos (NSA) como una serie de parches para el kernel de Linux utilizando Módulos de seguridad de Linux (LSM).” (Red Hat, 2022)

El objetivo principal de SELinux es proteger los datos del usuario del uso inadecuado por parte de empresas comprometidas, para lograr esto SELinux define controles de acceso para las aplicaciones, procesos y archivos de un sistema, para esto SELinux utiliza políticas de seguridad, estos son conjuntos de reglas que le indican a SELinux a qué se puede acceder o no.

Para verificar el acceso a un recurso del sistema por parte de una aplicación o proceso, conocido como sujeto, SELinux verifica con un caché de vector de acceso (AVC), donde se almacenan en caché los permisos para sujetos y objetos, después SELinux envía la solicitud al servidor de seguridad, el cual comprueba el contexto de seguridad de la aplicación o proceso y el recurso. El contexto de seguridad se aplica desde la base de datos de políticas de SELinux. Luego se concede o deniega el permiso.

SELinux utiliza políticas las cuales son reglas de seguridad que definen cómo procesos específicos que acceden a archivos relevantes, directorios y puertos. Cada entidad de recurso, como un archivo, proceso, directorio o puerto, tiene una etiqueta llamada contexto SELinux. La etiqueta de contexto coincide con una regla de política SELinux definida para permitir a una aplicación o proceso acceder al recurso etiquetado. Las etiquetas son una forma lógica de agrupar cosas, usando el formato “usuario: rol: tipo: nivel”.



Dentro del contexto de SELinux existen los booleanos que son configuraciones de activación/desactivación de funciones en SELinux, existen cientos de configuraciones que pueden activar o desactivar las capacidades de SELinux, y muchas ya están predefinidas.

Partición de Disco

La partición del disco consiste en dividir un disco duro en múltiples partes de almacenamiento lógico. Estas divisiones proporcionan numerosos beneficios:

- Limitar el espacio disponible para aplicaciones o usuarios.
- Separar los archivos del sistema operativo y del programa de los archivos del usuario.
- Crear un área separada para el intercambio de memoria.
- Limitar el uso del espacio en disco para mejorar el rendimiento de las herramientas de diagnóstico y las imágenes de respaldo.

Al realizar una partición de discos podemos seleccionar un esquema de partición, el cual podemos elegir entre MBR y GPT.

Esquema de partición MBR

El esquema de partición Master Boot Record (MBR) es el estándar en sistemas que ejecutan BIOS firmware. Algunas características de este esquema son:

- Admite un máximo de cuatro particiones primarias y 15 particiones secundarias.
- Cada partición puede tener un tamaño de hasta 2 TiB.
- Asigna 32 bits para las direcciones de bloques lógicos.



Esquema de partición GPT

Para sistemas que ejecutan firmware de interfaz de firmware extensible unificada (UEFI), GPT es el estándar para particionar discos y aborda las limitaciones del esquema MBR. Algunas características de GPT son:

- Proporciona un máximo de 128 particiones.
- Asigna 64 bits para direcciones de bloques lógicos
- Admite particiones y discos de hasta ocho zebibytes (ZiB).
- Permite que la tabla de particiones sea redundante
- Utiliza una suma de verificación para detectar errores en el encabezado GPT y la tabla de particiones.

LVM

Dentro del sistema de Red Hat, Logical Volume Manager (LVM) permite crear volúmenes de almacenamiento lógicos como una copia de almacenamiento físico, permitiendo así mayor flexibilidad que el uso físico.

El uso de volúmenes lógicos ofrece las siguientes ventajas sobre el uso directo del almacenamiento físico:

- Capacidad flexible: Cuando se utilizan volúmenes lógicos, los sistemas de archivos pueden extenderse a través de varios discos.
- Grupos de almacenamiento redimensionables: Puede ampliar los volúmenes lógicos o reducir su tamaño con simples comandos de software, sin necesidad de formatear y particionar los dispositivos de disco.
- Una cómoda denominación de los dispositivos: Los volúmenes de almacenamiento lógico se pueden gestionar en grupos definidos por el usuario y con nombres personalizados.



- Separación de discos: Permite aumentar drásticamente el rendimiento, repartiendo los datos entre dos o más discos.
- Volúmenes en espejo: El uso de volúmenes lógicos proporcionan una forma conveniente de configurar respaldos para sus datos.
- Instantáneas de volumen: Puede tomar instantáneas de dispositivos para realizar copias de seguridad consistentes.

Dispositivos físicos

Los volúmenes lógicos utilizan dispositivos físicos para almacenar datos, estos dispositivos pueden ser particiones de disco, discos completos, matrices RAID o discos SAN.

Volúmenes físicos (PV)

Para poder utilizar un dispositivo físico dentro de un sistema LVM, se debe inicializar el dispositivo como volumen físico LVM, utilizando todo el almacenamiento del dispositivo físico.

Grupos de volúmenes (VG)

Los grupos de volúmenes son grupos de almacenamiento formados por uno o más PV. Se puede ver como el equivalente funcional de un disco completo en comparación con el almacenamiento físico. Es importante señalar que un PV solo puede pertenecer a únicamente un solo VG.

Volúmenes lógicos (LV)

Los volúmenes lógicos se crean a partir de extensiones físicas libres en un VG y se proporcionan como dispositivo de almacenamiento para aplicaciones, usuarios y sistemas operativos. Los LV son una colección de extensiones lógicas.



SCSI/

El protocolo SCSI (Small Computer System Interface) original se diseñó para compartir datos a través de una red de área local (LAN). El protocolo iSCSI se desarrolló a finales de la década de 1990 para permitir el protocolo SCSI a través de una red TCP/IP.

Es un protocolo de capa de transporte diseñado para proporcionar un acceso sin interrupciones a los dispositivos de almacenamiento a través de una red. El nombre iSCSI se utilizó para indicar que el protocolo original se modificó y encapsula los comandos SCSI en paquetes TCP/IP.

La arquitectura iSCSI es de la categoría cliente-servidor. El cliente se conoce como iniciador y el servidor se conoce como destino iSCSI. El dispositivo de almacenamiento en bloques se conoce como una unidad lógica y un destino iSCSI puede tener muchas unidades lógicas. Cada uno tiene un número de unidad lógica (LUN) designado.

El protocolo iSCSI funciona de la siguiente manera:

- El iniciador se conecta a un destino mediante el protocolo de autenticación por desafío mutuo (CHAP).
- Tras la conexión, el dispositivo de almacenamiento aparece como una unidad de disco local en el cliente.

NFS

Tanto el sistema de archivos de red (NFS) como la interfaz de sistemas informáticos pequeños de Internet (iSCSI) se utilizan para compartir datos en una relación cliente-servidor a través de una red o red virtual. Estos han sido protocolos populares en las comunicaciones empresariales remotas.

El protocolo NFS se diseñó como un protocolo de intercambio de archivos cliente-servidor para sistemas Unix en la década de 1980. Permanece activo a través de varias actualizaciones, de las cuales la más reciente es la versión 4 de NFS. Es un protocolo popular para un sistema de archivos distribuidos.



El protocolo NFS funciona de la siguiente manera:

- El cliente solicita acceso a un recurso en un servidor NFS remoto.
- El servidor monta el recurso de forma remota en el cliente.
- Aparece el almacén de datos NFS y actúa como un recurso local en el cliente.
- Los recursos de lectura se almacenan en la memoria caché del sistema de archivos del cliente para un acceso rápido.

El acceso al recurso, como un archivo o un directorio, se comparte a través de una conexión virtual. Utiliza llamadas a procedimientos remotos (RPC) como tecnología de comunicación subyacente.

RAID

“Una matriz redundante de discos económicos (RAID) es una tecnología de virtualización de almacenamiento que crea grandes volúmenes lógicos de múltiples componentes de dispositivos de bloques físicos o virtuales. Diferentes formas de los volúmenes RAID ofrecen redundancia de datos, mejora del rendimiento o ambas, mediante la implementación de diseños de reflejo o rayas. LVM admite los niveles RAID 0, 1, 4, 5, 6 y 10”. (Red Hat, 2022)

De manera simple RAID combina múltiples unidades de disco pequeñas y por lo tanto más económicas, en una matriz para así lograr objetivos de rendimiento o redundancia que no se podría conseguir con unidades de discos grandes o costosas. De esta manera el computador identifica a este conjunto de unidades como una única unidad.



Algunos beneficios del uso de RAID para los administradores de sistemas son:

- Mejora de velocidad
- Aumento de la capacidad de almacenamiento al utilizar un único disco virtual
- Minimizar el fallo del disco

Existen distintos niveles RAID, cada uno con distinto enfoque, algunas de estas configuraciones son:

Nivel 0

El nivel RAID 0, a menudo llamado "striping", se enfoca en un mapeo de datos seleccionado para conseguir el mayor rendimiento de E/S a un bajo costo inherente pero desafortunadamente este nivel no proporciona redundancia.

Nivel 1

El nivel RAID 1, o "duplicación", este nivel ofrece redundancia al escribir los datos idénticos en cada disco miembro de la matriz, creando así una copia reflejada en cada disco, permitiendo a su vez la alta disponibilidad de datos. Este nivel funciona con dos o más discos que usan acceso paralelo para altas tasas de transferencia de datos durante la lectura, aunque normalmente en este nivel los discos funcionan de forma independiente para brindar altas tasas de transacciones de E/S.

Nivel 4

El nivel RAID 4 maneja una paridad concentrada en una única unidad de disco para proteger los datos, este nivel es el más adecuado para las transacciones de E/S que, para transferencias de archivos grandes, esto por consecuencia de que el disco de paridad dedicado se maneja como un cuello de botella inherente.



Nivel 5

El nivel RAID 5 es más común y consiste en distribuir la paridad entre algunas o todas las unidades de disco miembros de una matriz, este nivel a diferencia del nivel 4, elimina el cuello de botella de escritura inherente, siendo el único cuello de botella el de rendimiento en el proceso de cálculo de paridad, aunque con las nuevas tecnologías esto no suele ser un gran problema. Este nivel se utiliza constantemente con el almacenamiento en caché de reescritura para reducir la asimetría.

Virtualización

“La virtualización es una característica que permite dividir una única máquina física en múltiples máquinas virtuales, cada una de las cuales puede ejecutar un sistema operativo independiente.” (Red Hat, 2022)

Red Hat Enterprise Linux admite KVM (máquina virtual basada en kernel), esta herramienta puede ejecutar múltiples sistemas operativos de Linux y Windows. Una de las ventajas de KVM al igual que otras herramientas de Linux, es que puede ser manejada desde la línea de comandos con el comando “virsh” o también de manera gráfica con la consola web.

Algunas características de las máquinas virtuales son:

- Permite que varios sistemas operativos se ejecuten simultáneamente en una única plataforma de hardware.
- Utiliza un hipervisor para dividir el hardware en múltiples sistemas de hardware virtuales.
- Requiere un entorno de sistema operativo completo para soportar la aplicación



Contenedor

Un contenedor es un conjunto de uno o más procesos que están aislados del resto del sistema los cuales proporcionan una forma de empaquetar aplicaciones y simplificar su implementación y gestión. Una gran ventaja del uso de contenedores es debido a que el contenido de un contenedor está aislado del contenido de otros, imposibilitando que se afecten entre ellos.

Red Hat Enterprise Linux admite contenedores mediante el uso de las siguientes tecnologías principales:

- Grupos de Control (cgroups) para la gestión de recursos.
- Espacios de nombres para aislamiento de procesos.
- SELinux y Seccomp (modo de Computación Segura) para hacer cumplir los límites de seguridad.

Un contenedor tiene las siguientes características:

- Se ejecuta directamente en el sistema operativo host y comparte recursos con todos los contenedores del sistema.
- Comparte el kernel del host, pero aísla los procesos de la aplicación del resto del sistema.
- Requiere muchos menos recursos de hardware que las máquinas virtuales.
- Incluye todas las dependencias para la ejecución de los programas.

Imagen de Contenedor

Para ejecutar contenedores, debe utilizar una imagen de contenedor. Una imagen contenedora es un archivo estático que contiene pasos codificados y sirve como modelo para crear contenedores. Las imágenes del contenedor empaquetan una aplicación con todas sus dependencias, como sus bibliotecas del sistema, tiempos de ejecución del lenguaje de programación y bibliotecas, entre otros.



Las imágenes de contenedores se crean de acuerdo con especificaciones, como la Open Container Initiative (OCI). Estas especificaciones también definen el formato de las imágenes del contenedor.

Docker y Podman

Tanto Podman como Docker son herramientas para desarrollar, gestionar y ejecutar contenedores, pero existen sus diferencias entre ellos, siendo la principal que Podman da mejores funciones de seguridad y desarrollo.

Podman utiliza un enfoque de microservicios, creando una red con muchos otros productos nativos de la nube, como Buildah y Skopeo, para construir e impulsar contenedores. Esto convierte a Podman en una aplicación más ligera y rápida que Docker.

Otra característica importante de Podman es la facilidad en la creación de grupos lo que le permite combinar contenedores en ejecución para crear pods, siendo tan simple crearlo tanto en GUI como en CLI.

El principal motivo por el cual Podman es más seguro que Docker es debido a su modelo sin root, esto significa que el sistema de contenedores, de forma predeterminada, lo ejecuta un usuario no root. Incluso si un usuario está dentro del contenedor es el usuario root. Esto conlleva a que Podman opte por ejecutar contenedores con espacios de nombres de usuario y contenedores SELinux, de modo que, si un contenedor es atacado, el atacante sólo tiene acceso a los archivos del contenedor. No pueden alterar el host ni otros contenedores debido a permisos incorrectos.

Script Shell Bash

Un script de shell Bash es un archivo ejecutable que contiene una lista de comandos para controlar la toma de decisiones en la tarea general. Puede utilizar scripts de shell para mejorar la eficiencia y precisión de la finalización de tareas de rutina.



Ansible

“Ansible es una plataforma de automatización de código abierto. Es un lenguaje de automatización simple que puede describir con precisión una infraestructura de aplicaciones de TI en Ansible Playbook”. (Red Hat, 2022)

Una gran ventaja de Ansible es su simpleza para aprenderlo permitiendo así gestionar potentes tareas de automatización adaptándose a varios flujos de trabajo y ambientes.

Ansible utiliza playbook la cual es una herramienta de automatización fáciles de leer, comprender y cambiar para los humanos. Los playbook ejecutan tareas en orden, su diseño simple los hace compatibles para todos los equipos.

Una característica importante de Ansible es su arquitectura sin agentes, esto significa que Ansible se conecta a los hosts que gestiona mediante OpenSSH o WinRM ejecutando tareas, a menudo sacando pequeñas programas llamados módulos, estos programas se utilizan para poner el sistema en un estado deseado específico. Todos los módulos que se insertan se eliminan cuando Ansible ha finalizado sus tareas. El hecho de que no hay agentes ni infraestructura de seguridad personalizada adicional, Ansible da más eficiencia y seguridad que otras tecnologías.

Las características principales de Ansible son:

- Soporte multiplataforma: Ansible proporciona soporte sin agentes para Linux, Windows, UNIX y dispositivos de red, en entornos físicos, virtuales, de nube y de contenedores
- Automatización legible por humanos: Los playbook de Ansible, escritos como archivos de texto YAML, son fáciles de leer y ayudan a garantizar que todos comprendan lo que hacen.
- Descripciones precisas de las aplicaciones: cada cambio se puede realizar mediante Ansible playbook y cada aspecto del entorno de su aplicación se pueden describir y documentar.



- Fácil de administrar en control de versiones: los playbook y proyectos de Ansible son texto sin formato. Ellos pueden ser tratados como código fuente y colocados en su sistema de control de versiones, como Git.
- Soporte para inventarios dinámicos: la lista de máquinas que administra Ansible puede ser dinámicamente actualizada desde fuentes externas para capturar la lista correcta y actual de todos los servidores administrados, independientemente de la infraestructura o ubicación.
- Orquestación que se integra fácilmente con otros sistemas: HP SA, Puppet, Jenkins, Red Hat

La funcionalidad de Ansible consiste en ser un motor de estado deseado, lo que significa que aborda el problema de cómo automatizar las implementaciones de TI expresándose en términos del estado en el que desea que estén sus sistemas, realizando sólo los cambios que sean necesarios.

A diferencia de otras herramientas, Ansible combina la orquestación con la gestión de la configuración, aprovisionamiento e implementación de aplicaciones en una plataforma fácil de usar.

SSH

Secure Shell (SSH) es un protocolo de seguridad de red que utiliza mecanismos de cifrado y autenticación para implementar servicios como acceso seguro y transferencia de archivos. Existen otros protocolos o métodos de inicio de sesión remoto y transferencia de archivos, como Telnet y FTP, pero estos transmiten datos en texto sin cifrar, lo cual es inseguro, por el contrario, SSH proporciona servicios de red seguros en un entorno de red inseguro mediante el cifrado y la autenticación de datos de la red.



SFTP

El Protocolo seguro de transferencia de archivos (SFTP) es un protocolo de red basado en SSH para acceder, transferir y administrar de forma segura archivos y datos confidenciales entre dispositivos.

LDAP

El protocolo ligero de acceso a directorios (LDAP) es un protocolo que ayuda a los usuarios a encontrar datos sobre organizaciones, personas y más. LDAP tiene dos objetivos principales: almacenar datos en el directorio LDAP y autenticar a los usuarios para acceder al directorio. También proporciona el lenguaje de comunicación que las aplicaciones requieren para enviar y recibir información de los servicios de directorio. Un servicio de directorio proporciona acceso a dónde se encuentra la información sobre organizaciones, individuos y otros datos dentro de una red.

El caso de uso más común de LDAP es proporcionar una ubicación central para acceder y administrar servicios de directorio. LDAP permite a las organizaciones almacenar, administrar y proteger información sobre la organización, sus usuarios y activos, como nombres de usuarios y contraseñas. Esto ayuda a simplificar el acceso al almacenamiento al proporcionar una estructura jerárquica de información y puede ser fundamental para las corporaciones a medida que crecen y adquieren más datos y activos de usuarios.

LDAP también funciona como una solución de gestión de identidad y acceso (IAM) dirigida a la autenticación de usuarios, incluida la compatibilidad con Kerberos y el inicio de sesión único (SSO), la capa de seguridad de autenticación simple (SASL) y la capa de sockets seguros (SSL).



DHCP

“El Protocolo de configuración dinámica de host (DHCP) es un protocolo cliente servidor que proporciona automáticamente un host de protocolo de Internet (IP) con su dirección IP y otra información de configuración relacionada, como la máscara de subred y la puerta de enlace de predeterminada” (Microsoft, 2023)

El protocolo DHCP funciona a partir de cuatro tipos de paquetes:

DHCPDISCOVER: El cliente emite un mensaje broadcast para lograr encontrar y conectar con cualquier servidor DHCP disponible en la red.

DHCPOFFER: Enviado por el servidor DHCP como respuesta al DHCPDISCOVER, varios servidores DHCP pueden enviar una oferta al mismo cliente.

DHCPREQUEST: Enviado por el cliente, este mensaje indicará cuál oferta fue aceptada por el cliente

DHCPPACK: A modo de respuesta del DHCPREQUEST, el servidor confirma y finaliza la asignación de ip.

PXE

PXE o Network Booting es una tecnología que permite al BIOS descargar y arrancar un sistema operativo (OS) a través de la red. Cuando un cliente PXE arranca siguiendo el orden de arranque, suponiendo que el siguiente modo de arranque en PXE, el cliente solicita una dirección DHCP, una vez asignada la IP con todas las características configuradas en el servidor DHCP, seguido de esto el cliente PXE descarga el especificado en la respuesta DHCP y lo ejecuta para que el sistema operativo se cargue en el servidor.



SCRUM

“Scrum es un marco de administración que los equipos utilizan para organizarse por cuenta propia y trabajar en aras de alcanzar un objetivo común. Describe un conjunto de reuniones, herramientas y funciones para entregar proyectos de forma eficiente.” (Amazon, 2023)

El método SCRUM funciona a partir de tres principios fundamentales:

Transparencia, consiste en trabajar bajo un entorno en que todos son conscientes de los retos que pueden estar llevando los demás, para conseguir esto se realizan conversaciones periódicas entre los distintos miembros del equipo.

Reflexión, se realizan puntos de reflexión frecuentes en el marco para conseguir que los miembros del equipo sean conscientes de los avances logrados y de los que aún faltan por conseguir, esto se hace con el fin de realizar estimaciones y planificaciones para el futuro del proyecto, permitiendo así llevar los proyectos de manera eficiente, con los recursos disponibles.

Adaptación, consiste en priorizar tareas en función de los requisitos cambiantes de los clientes.

Existen varios roles dentro del métodos SCRUM, estos son:

Propietario del producto:

- Suministran al equipo de trabajo orientación clara sobre qué características se deben entregar a continuación.
- Tienen un puente entre lo que desea el negocio y lo que el equipo de trabajo entiende.
- Deciden en qué momento y con qué frecuencia realizar los lanzamientos.



Líder de SCRUM:

- Programar los recursos necesarios para cada *Sprint*.
- Organizar otros eventos del Sprint y las reuniones del equipo de trabajo.
- Liderar la transformación digital dentro del equipo de trabajo.
- Organizar las formaciones dirigidas al equipo al adoptar nuevas tecnologías.
- Comunicarse con grupos externos para resolver cualquier reto al que se enfrente el equipo en su conjunto.

Equipo de desarrollo SCRUM:

- Trabajar de forma colaborativa para garantizar que el Sprint se complete con éxito.
- Promover prácticas de desarrollo sostenible.
- Organizarse por cuenta propia y abordar los proyectos con una actitud fraternal y solidaria evidente.
- Controlar la planificación y estimar qué tanto trabajo se requiere para completar con éxito cada *Sprint*.

Dentro del método SCRUM existen eventos o pasos los cuales son un conjunto de reuniones secuenciales que los equipos realizan de manera periódica:

Planificación del Sprint: Se realiza la estimación del trabajo que se debe completar durante el siguiente Sprint, definiendo objetivos y cuidando que sean específicos, medibles y viables.

Sprint: Es el tiempo que el equipo de SCRUM trabaja conjuntamente para la consecución de un incremento, usualmente dura dos semanas.



Reunión diaria: Sesión breve en la que cada participante del equipo se reporta y planifica el día, expresando las conclusiones y dificultades para lograr el objetivo del Sprint. Se busca que la reunión sea lo más breve posible.

Revisión del Sprint: Al final el Sprint se presenta el proyecto ante las partes interesadas.

Retrospectiva del Sprint: Reunión con el fin de documentar y hablar sobre los fallos y éxitos durante el Sprint, esto con el fin de mejorar en los siguientes Sprint.

MVC (Modelo-Vista-Controlador)

El Modelo Vista Controlador separa una aplicación en tres componentes:

Modelos: Son las partes de la aplicación que implementan la lógica del dominio de datos de la aplicación. Estos recuperan y almacenan el estado del modelo en una base de datos.

Vistas: Son los componentes que muestra la interfaz de usuario de la aplicación.

Controladores: Controlan la interacción del usuario, trabajan con el modelo y finalmente seleccionan una vista para representar la interfaz de usuarios. En resumidas palabras controla y responde a la interacción y los datos que introduce el usuario.

SOA

“La arquitectura orientada a servicios (SOA, por sus siglas en inglés) es un método de desarrollo de software que utiliza componentes de software llamados servicios para crear aplicaciones empresariales. Cada uno de estos servicios brinda una capacidad empresarial y, además, pueden comunicarse también con el resto de los servicios mediante diferentes plataformas y lenguajes. Los desarrolladores usan SOA para reutilizar servicios en diferentes sistemas o combinar varios servicios independientes para realizar tareas complejas.” (Amazon, 2023)

Dentro de esta metodología existen cuatro componentes:



Servicio: Componentes básicos, pueden ser privados o públicos, cada uno cumple tres características principales.

Implementación de servicios: Código que crea la lógica para realizar la función de servicio específico.

Contrato del servicio: Define la naturaleza del servicio y sus condiciones, son como los prerequisites del servicio.

Interfaz del servicio: Manera en la que se puede invocar el servicio.

Proveedor de servicios: Es quien crea, mantiene y proporciona uno o más servicios que otros podrán utilizar, pueden ser tanto propios como adquiridos de manera externa.

Consumidor de servicios: Es quien solicita al proveedor un servicio, especificando las reglas que el proveedor y el consumidor deben seguir al momento de la interacción. Al igual que los proveedores, los consumidores pueden pertenecer a departamentos, organizaciones o sectores diferentes.

Registro de servicios: Es un directorio de servicios accesibles a través de redes. Se encarga de almacenar documentos descriptivos sobre el servicio que pertenecen a los proveedores de servicios, dando información acerca del servicio y cómo comunicarse con él.



Capítulo 4. Desarrollo del proyecto.

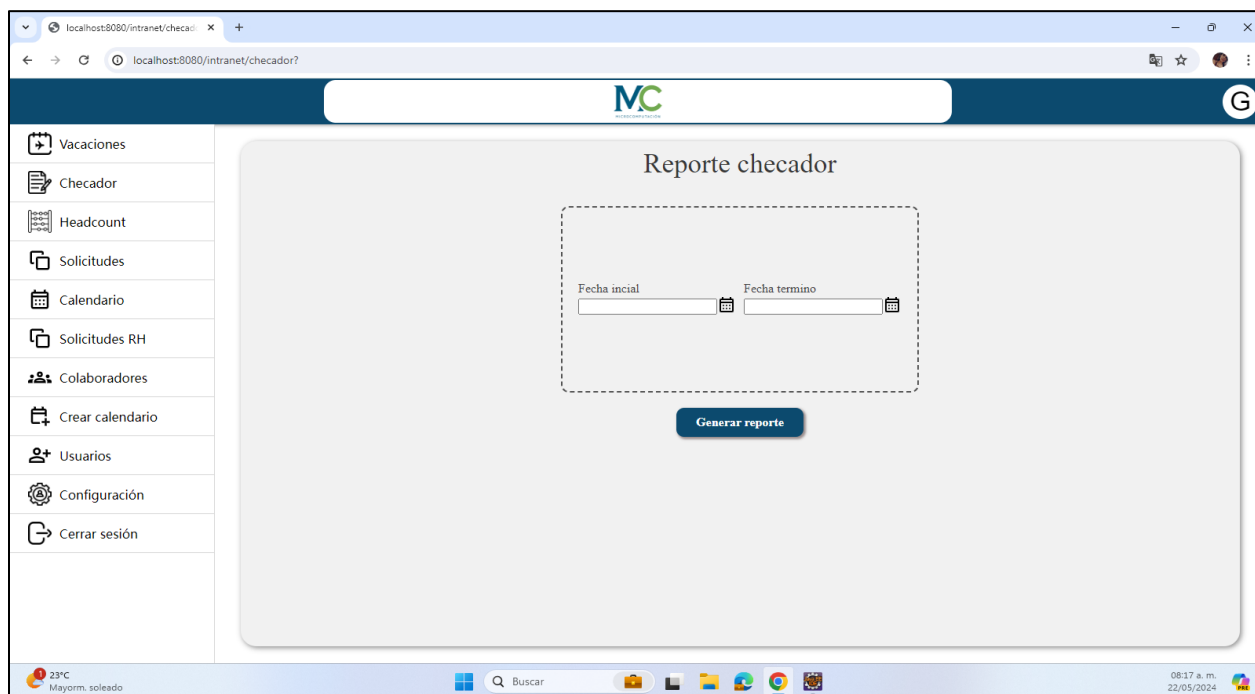
Para este proyecto se realizaron varias tareas que en su conjunto ayudaron a la intranet, ya sea para brindar un nuevo servicio, para automatizar algún proceso o realizar mudanza de datos o procesos. A continuación, se listan las distintas tareas realizadas durante el proyecto:

4.1 Módulo “Checador”

Primeramente, se realizó un módulo de la página web usada de manera interna en MC, el módulo realizado fue “Checador” el cual tiene como objetivo automatizar el proceso realizado por el departamento de Recursos Humanos para crear semanalmente un reporte en Excel el cual muestra quienes derecho a un bono de puntualidad basado en los registros de un checador. Antes de implementar el módulo, la persona encargada de este proceso demoraba hasta 4 horas cada semana, una vez implementado el módulo el reporte se realiza de manera casi instantánea al dar clic en un botón, seleccionando antes el rango de fechas para realizar el reporte.

Este proceso se realizó primeramente configurando el hardware del checador y la base de datos para que en ambos los IDs y nombres coincidan. Después con una API en C# se consiguió obtener desde el checador un archivo CSV para finalmente mandar este archivo al servicio programado en Java con Spring Boot, el microservicio procesa el CSV para así crear un total de tres tablas que dan un resumen de las asistencias de los empleados de MC, dichas tablas se colocan en un archivo Excel para ser descargado de manera automática en la computadora del usuario.

Ilustración 3: Pantalla del módulo "Checador".

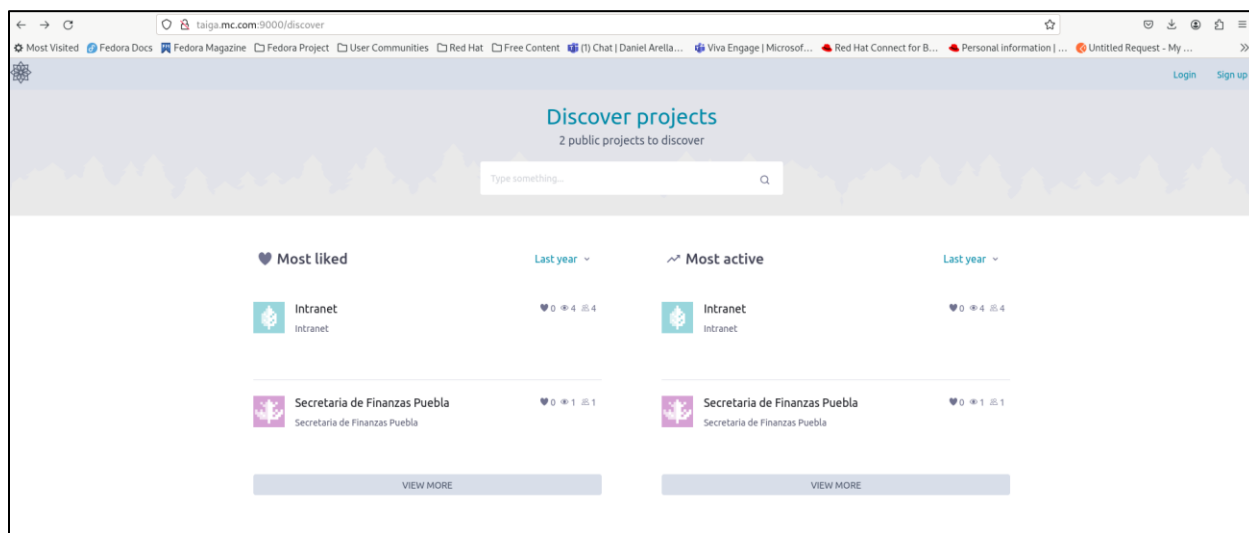


Como extra a esta tarea, se configuró una tarea programable en el servidor Linux de TomCat que permite eliminar los archivos almacenados en otros módulos de la intranet, utilizando el comando “crontab -e”.

4.2 Taiga

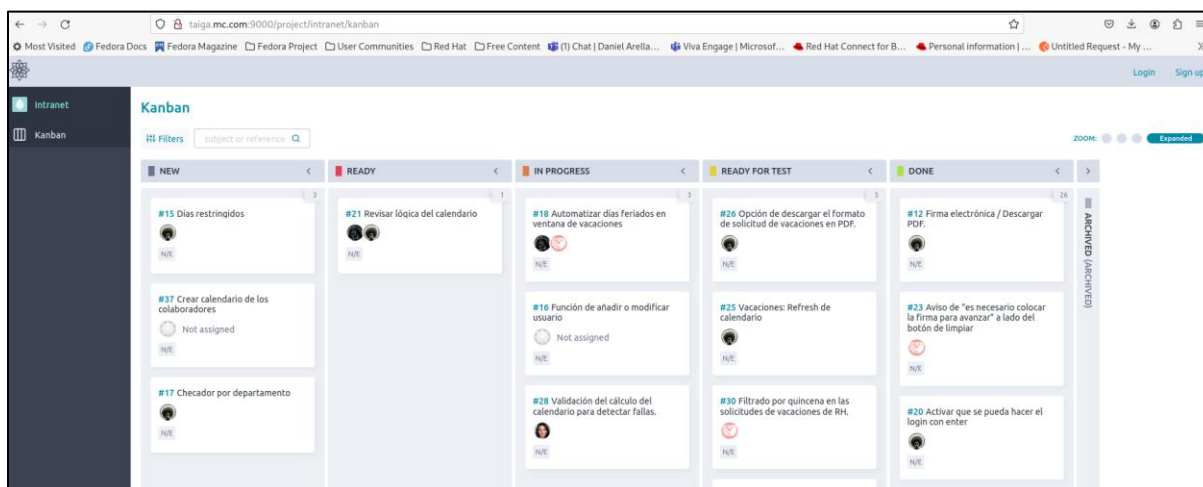
Taiga es una plataforma de gestión de proyectos para desarrolladores y diseñadores ágiles que desean una herramienta sencilla para seguir metodologías de planeación como el modelo SCRUM.

Ilustración 4: Pantalla de inicio de Taiga.



Esta herramienta fue solicitada al departamento de Open Source para poder gestionar de mejor manera los proyectos del departamento de proyectos. La herramienta tenía que ser funcional y accesible desde un servidor local de MC, para esto se instaló en una máquina virtual con sistema operativo de Red Hat 9, utilizando a su vez contenedores enfocados a distintas tareas como la base de datos que permite que al haber un corte de luz los datos se mantengan estables hasta volver alzar el servicio.

Ilustración 5: Captura de los distintos estados del proyecto de Intranet.



Como tarea extra, se solicitó que la herramienta de taiga fuera únicamente accesible a los empleados del área de proyectos, para esto se utilizó el protocolo LDAP gestionado por Red Hat Identity Management, para esta tarea únicamente se tuvo que realizar una configuración extra al Dockerfile de las imágenes de contenedor.

Ilustración 6: Ejecución de Taiga por medio de contenedores.

```
[taiga@taiga ~]$ podman ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
32dca94888e5	docker.io/library/postgres:12.3	postgres	4 weeks ago	Up 2 hours (healthy)		taiga-docker_taiga-db_1
d57addd31a99	docker.io/library/rabbitmq:3.8-management-alpine	rabbitmq-server	4 weeks ago	Up 2 hours		taiga-docker_taiga-async-rabbitmq_1
b2964334a9ad	docker.io/taigaio/taiga-front:latest	nginx -g daemon o...	4 weeks ago	Up 2 hours		taiga-docker_taiga-front_1
08bbf02fec02	docker.io/library/rabbitmq:3.8-management-alpine	rabbitmq-server	4 weeks ago	Up 2 hours		taiga-docker_taiga-events-rabbitmq_1
fc8bf86e9318	docker.io/taigaio/taiga-protected:latest		4 weeks ago	Up 2 hours		taiga-docker_taiga-protected_1
82e929464e21	docker.io/taigaio/taiga-events:latest		4 weeks ago	Up 2 hours		taiga-docker_taiga-events_1
aa29d0d2489d	localhost/taiga-docker_taiga-back:latest		4 weeks ago	Up 2 hours		taiga-docker_taiga-back_1
79b5993f2710	docker.io/taigaio/taiga-back:latest		4 weeks ago	Up 2 hours		taiga-docker_taiga-back_1
2e2a97228a7a	docker.io/library/nginx:1.19-alpine	nginx -g daemon o...	4 weeks ago	Up 2 hours	0.0.0.0:9000->80/tcp	taiga-docker_taiga-gateway_1

```
[taiga@taiga ~]$
```

Para poder realizar todo lo anterior se utilizaron herramientas como Docker-compose y Podman-compose, a través de la configuración archivos como son docker-compose.yml, Docket Files y archivos de configuración adicionales.

Ilustración 7: Dockerfile para integrar LDAP al servidor Taiga.

```
FROM taigaio/taiga-back:latest

# Insert custom configuration into the taiga configuration file
COPY config.append.py /taiga-back/settings
RUN cat /taiga-back/settings/config.append.py >> /taiga-back/settings/config.py && rm /taiga-back/settings/config.append.py

RUN apt-get update \
    && apt-get install -y git \
    && rm -rf /var/lib/apt/lists/*

RUN pip install git+https://github.com/Monogrammm/taiga-contrib-ldap-auth-ext.git

RUN apt-get purge -y git \
    && apt-get autoremove -y
```

El código anterior es un ejemplo de un Dockerfile necesario para poder configurar el servidor de taiga con la implementación del servidor LDAP de la empresa.

Ilustración 8: Opciones de configuración adicional para el LDAP en Taiga.

```
INSTALLED_APPS += ["taiga_contrib_ldap_auth_ext"]

# Multiple LDAP servers are currently not supported, see
# https://github.com/Monogrammm/taiga-contrib-ldap-auth-ext/issues/16
LDAP_SERVER = "ldap://xxxx.com"
LDAP_PORT = xxx

LDAP_BIND_DN = 'uid=xxxx,cn=xxxx,cn=xxxx,dc=xxxx,dc=xxxx'
LDAP_BIND_PASSWORD = 'xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'

LDAP_SEARCH_BASE = 'cn=xxxxxx,cn=xxxxxx,dc=xxxx,dc=xxxx'

LDAP_USERNAME_ATTRIBUTE = "uid"
LDAP_EMAIL_ATTRIBUTE = "mail"
LDAP_FULL_NAME_ATTRIBUTE = "givenName"

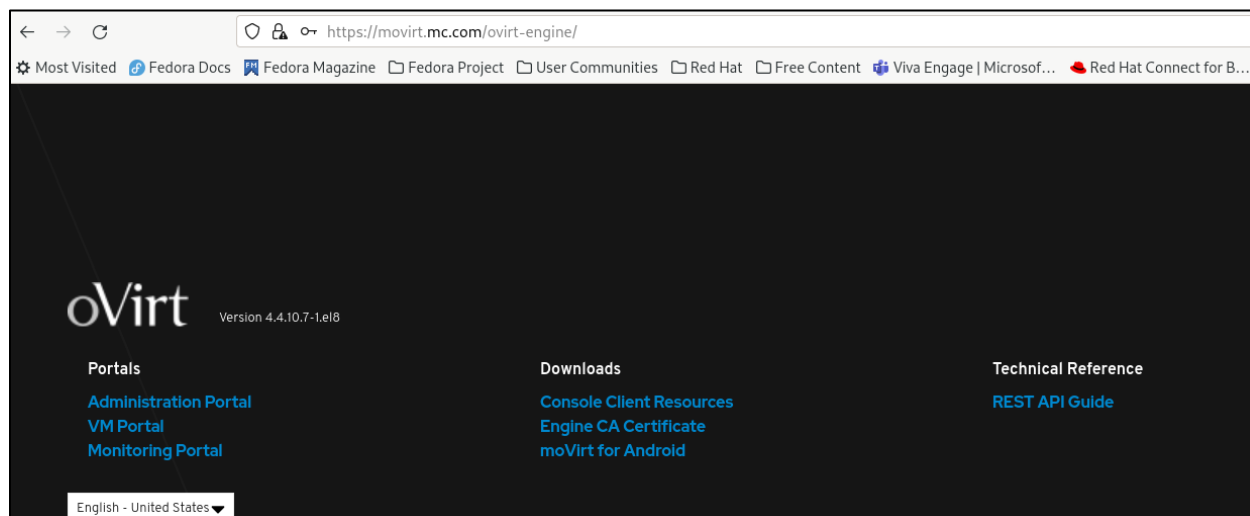
LDAP_SAVE_LOGIN_PASSWORD = False

LDAP_MAP_USERNAME_TO_UID = None
```

4.3 Mudanza de servidor

Durante las prácticas uno de los servidores donde se encontraban varias máquinas virtuales con distintos servicios corriendo, entre ellos algunos utilizados para la intranet, sufría una falla de energía, provocando así que en varias ocasiones los servicios cayeran y tardaran en levantarse. Para solucionar esto se me solicitó realizar una mudanza de las máquinas virtuales alojadas en el servidor Ovirt a otro servidor en mejores condiciones, he de mencionar que el nuevo servidor utiliza KVM como hipervisor.

Ilustración 9: Pantalla de inicio del hipervisor Ovirt.



Primeramente, se tomó registro de las características de las máquinas virtuales que alojaban los distintos servicios además del id de los discos virtuales.

Tanto en el servidor de Ovirt como en el nuevo servidor, se montó un directorio NFS, conectado a un tercer servidor que servirá de intermediario, esto debido a que posee el módulo de Ovirt necesario para ejecutar ciertas tareas del ansible playbook. Para realizar esto se utilizó el siguiente comando “mount <dirección ip del servidor intermediario>:Kaspersky /mount”

Se creó un playbook de Ansible para poder descargar los discos virtuales utilizando módulos de Ovirt.

Ilustración 10: Parte de un Ansible playbook.

```
pre_tasks:
- name: Login to RHV
  ovirt_auth:
    url: "{{ engine_url }}"
    username: "{{ engine_user }}"
    password: "{{ engine_password }}"
    ca_file: "{{ engine_cafile | default(omit) }}"
    insecure: "{{ engine_insecure | default(true) }}"
  tags:
    - always

tasks:
- name: download image
  ovirt_disk:
    auth: "{{ ovirt_auth }}"
    id: "{{ ovirt_id_disk }}"
    download_image_path: "/mnt/taiga_disk.raw"

post_tasks:
- name: Logout from RHV
  ovirt_auth:
    state: absent
    ovirt_auth: "{{ ovirt_auth }}"
  tags:
    - alwaysovi
```

En el playbook anterior se muestran tres secciones cada una con sus respectivas tareas, siendo la primera “Login to RHV” necesaria para hacer inicio de sesión al hipervisor de Ovirt y acceder a sus archivos, la segunda tarea “download image” se ocupa para poder descargar en el host líder la imagen del disco virtual, por medio del id, finalmente la última tarea es para cerrar sesión y evitar fallos futuros además de ser una medida de seguridad ante posibles ciberataques.

Después se informó al departamento de proyectos sobre la mudanza para llegar a un acuerdo del momento idóneo de realiza la tarea

Una vez establecida la fecha y hora, se ejecutó el playbook creado anteriormente, monitoreando que se descargara de manera correcta.



A continuación, con el comando “rsyn <origen> <destino>” se trasladaron los discos virtuales a la dirección correcta dentro del nuevo servidor.

Finalmente se crearon nuevas máquinas virtuales con las características registradas en el primer paso, asignándoles los discos correspondientes, logrando así levantar los servicios nuevamente en el nuevo servidor.

Tras realizar esta tarea hasta el momento de realizar este reporte, los servicios no se han visto afectados como en el anterior servidor, dando así una mejor estabilidad y disponibilidad a la intranet.

El motivo por el cual se optó por descargar los discos utilizando ansible, fue debido a que el servidor Ovirt no permitía descargar de manera local los discos utilizando su interfaz web, el motivo de este error se desconoce, sin embargo, al ser un servidor el cual iba a ser eliminado después se consideró no relevante el solucionar el problema directo y mejor se optó por utilizar ansible como alternativa para la mudanza.

4.4 Scripts

Durante las prácticas se solicitaron scripts que permitieran automatizar procesos, facilitando y agilizando procesos más complejos de otras áreas o de la misma área de Open Source. Un ejemplo de estos scripts son los siguiente:

Ilustración 11: Script para configurar un bond.

```
#Colores
greenColor="\e[0;32m\033[1m"
endColor="\e[0;33[0m\e[0m"
redColor="\e[0;31m\033[1m"
blueColor="\e[0;34m\033[1m"
yellowColor="\e[0;33m\033[1m"
purpleColor="\e[0;35m\033[1m"
magentaColor="\e[0;36m\033[1m"
grayColor="\e[0;37m\033[1m"

#Network specs
DNS=XXX.XXX.XXX.XXX
GW=XXX.XXX.XXX.XXX
MASTER_IP=XXX.XXX.X

#MACs
MAC_1=xx:xx:xx:xx:xx:xx
MAC_2=xx:xx:xx:xx:xx:xx
MAC_3=xx:xx:xx:xx:xx:xx
MAC_4=xx:xx:xx:xx:xx:xx

#GET INTERFACES
echo -e "\n${redColor}Get interfaces \n${endColor}"
ifn_1=$(python3 obtener.py $MAC_1)
ifn_2=$(python3 obtener.py $MAC_2)
ifn_3=$(python3 obtener.py $MAC_3)
ifn_4=$(python3 obtener.py $MAC_4)

echo -e "\n${greenColor}Setup bonding 0\n${endColor}"

nmcli connection add type bond con-name bond0 ifname bond0 ipv4.address $MASTER_IP ipv4.gateway $GW ipv4.dns $DNS ipv4.method manual ipv6.method ignore mode 802.3ad

echo -e "\n${greenColor}Setup Slaves bond0\n${endColor}"
nmcli connection add type ethernet slave-type bond con-name bond0-part1 ifname $ifn_1 master bond0
nmcli connection add type ethernet slave-type bond con-name bond0-part3 ifname $ifn_3 master bond0
nmcli connection add type ethernet slave-type bond con-name bond0-part2 ifname $ifn_2 master bond0
nmcli connection add type ethernet slave-type bond con-name bond0-part4 ifname $ifn_4 master bond0

echo -e "\n${greenColor}Auto negotiate off bond0\n${endColor}"
nmcli connection modify bond0-part1 802-3-ethernet.auto-negotiate no 802-3-ethernet.speed 0 802-3-ethernet.duplex ""
nmcli connection modify bond0-part3 802-3-ethernet.auto-negotiate no 802-3-ethernet.speed 0 802-3-ethernet.duplex ""
nmcli connection modify bond0-part2 802-3-ethernet.auto-negotiate no 802-3-ethernet.speed 0 802-3-ethernet.duplex ""
nmcli connection modify bond0-part4 802-3-ethernet.auto-negotiate no 802-3-ethernet.speed 0 802-3-ethernet.duplex ""
```

El anterior Script automatiza la creación de conexiones bridge en un sistema operativo Linux de la familia Red Hat, utilizando únicamente la dirección MAC de las interfaces, esto debido a que sin importar el sistema operativo estas direcciones nunca cambian por lo que es mejor utilizarla en vez del nombre de las interfaces. La creación de las conexiones se realiza utilizando la herramienta de Network Manage con el comando nmcli.



- `nmcli connection add type bond con-name bond0 ifname bond0 ipv4.address $MASTER1_IP ipv4.gateway $GW ipv4.dns $DNS ipv4.method manual ipv6.method ignore mode 802.3ad`
 - “nmcli connection add”: Utilizado para añadir una nueva conexión
 - “type bond”: Se especifica el tipo de conexión
 - “con-name bond0”: Se asigna un nombre a la conexión
 - “ifname bond0”: Se asigna una interfaz, en este caso se crea una nueva de manera virtual
 - “ipv4.address \$MASTER1_IP ipv4.gateway \$GW ipv4.dns \$DNS ipv4.method manual”: Se configura la dirección ipv4 de manera estática, colocando el gateway y el DNS, todo a través de variables.
 - “ipv6.method ignore”: Se ignora la configuración ipv6
 - “mode active-backup”: Se asigna el tipo de bond.
- `nmcli connection add type ethernet slave-type bond con-name bond0-port1 ifname $ifn_3 master bond0`
 - “nmcli connection add”: Utilizado para añadir una nueva conexión
 - “type ethernet”: Se especifica el tipo de conexión.
 - “slave-type bond”: Especifica que este será parte de una conexión bond
 - “con-name bond0-port1”: Se asigna un nombre a la conexión
 - “ifname \$ifn_3”: Se asigna la interfaz, siendo esta el valor almacenado en una variable
 - “master bond0”: Se asigna el bond

Ilustración 12: Script en Python.

```
import glob
import sys

mac = sys.argv[1]

# Ruta con el patrón de comodín
ruta = '/sys/class/net/*/address'

# Utiliza glob para buscar archivos que coincidan con el patrón
archivos = glob.glob(ruta)

for a in archivos:
    with open(a, 'r') as f:
        contenido = f.read()
        if contenido.strip() == mac:
            print(a[15:-8])
```

El segundo Script se encuentra programado en Python y es utilizado por el primero el cual permite a través de una ruta conocida y dirección MAC proporcionada al momento de ejecutar el script, muestra el nombre de la interfaz asignada a la dirección MAC.

El motivo por el cual se optó por utilizar la dirección MAC para esta tarea es por el hecho que el nombre de las interfaces puede llegar a cambiar por varios factores, por el contrario, las direcciones MAC nunca cambian.



4.5 Servidor PXE y DHCP

Para el momento de realizar el reporte esta fue la última tarea asignada, la cual consistió en crear un servidor PXE, para lo cual primeramente se necesitaba crear un servidor DHCP, para asignar direcciones ip al momento de solicitar servicio al servidor PXE. El objetivo de crear el servidor PXE era permitir que los usuarios puedan instalar sistemas operativos recurrentes dentro de la empresa, como son la familia Red Hat, sin necesidad de ocupar métodos poco seguros como son las memorias booteables o métodos poco eficientes como son descargar varias veces las isos de los sistemas operativos cada vez que se requieran, este último método puede sonar factible, pero debemos considerar que descargar un archivo de gran peso puede tardar bastante según sea la velocidad de internet, además de afectar a la misma durante la descarga. Con el servidor PXE se busca eliminar estos problemas para poder instalar sistemas operativos únicamente usando la misma red de la empresa siendo un métodos más rápido y seguro.

Ilustración 13: Configuración del menú del servidor PXE.

```
default vesamenu.c32
prompt 1
timeout 120

display boot.msg

label linux
  menu label ^Install Red Hat 8.6 with PXE server MC
  menu default
  kernel S0/rhel-8.6/images/pxeboot/vmlinuz
  append initrd=S0/rhel-8.6/images/pxeboot/initrd.img ip=dhcp inst.repo=xxxx://xxx.xxx.xxx.xxx/rhel-8.6
label vesa
  menu label Install Red Hat 8.6 with ^basic video driver
  kernel S0/rhel-8.6/images/pxeboot/vmlinuz
  append initrd=S0/rhel-8.6/images/pxeboot/initrd.img ip=dhcp inst.xdriver=vesa nomodeset inst.repo=xxxx://xxx.xxx.xxx.xxx/rhel-8.6
label rescue
  menu label ^Rescue installed system
  kernel centos-st9/vmlinuz
  append initrd=centos-st9/initrd.img rescue
label local
  menu label Boot from ^local drive
  localboot 0xffff
```

Ilustración 14: Cliente PXE cargando la configuración del servidor.

```
iPXE (http://ipxe.org) 00:0B.0 CB80 PCI2.10 PnP PMM BFF94380 BFED4380 CB80

iPXE (http://ipxe.org) 00:0C.0 CC80 PCI2.10 PnP PMM BFF94380 BFED4380 CC80

Press ESC for boot menu.

Booting from ROM...
iPXE (PCI 00:03.0) starting execution...ok
iPXE initialising devices...ok

iPXE 1.0.0+ (133f4c) -- Open Source Network Boot Firmware -- http://ipxe.org
Features: DNS HTTP iSCSI TFTP VLAN AoE ELF MBOOT PXE bzImage Menu PXEXT

net0: 52:54:00:71:16:ad using virtio-net on 0000:00:03.0 (open)
[Link:up, TX:0 TXE:0 RX:0 RXE:0]
Configuring (net0 52:54:00:71:16:ad).....
```

Ilustración 15: Vista de selección configurada en el servidor PXE.

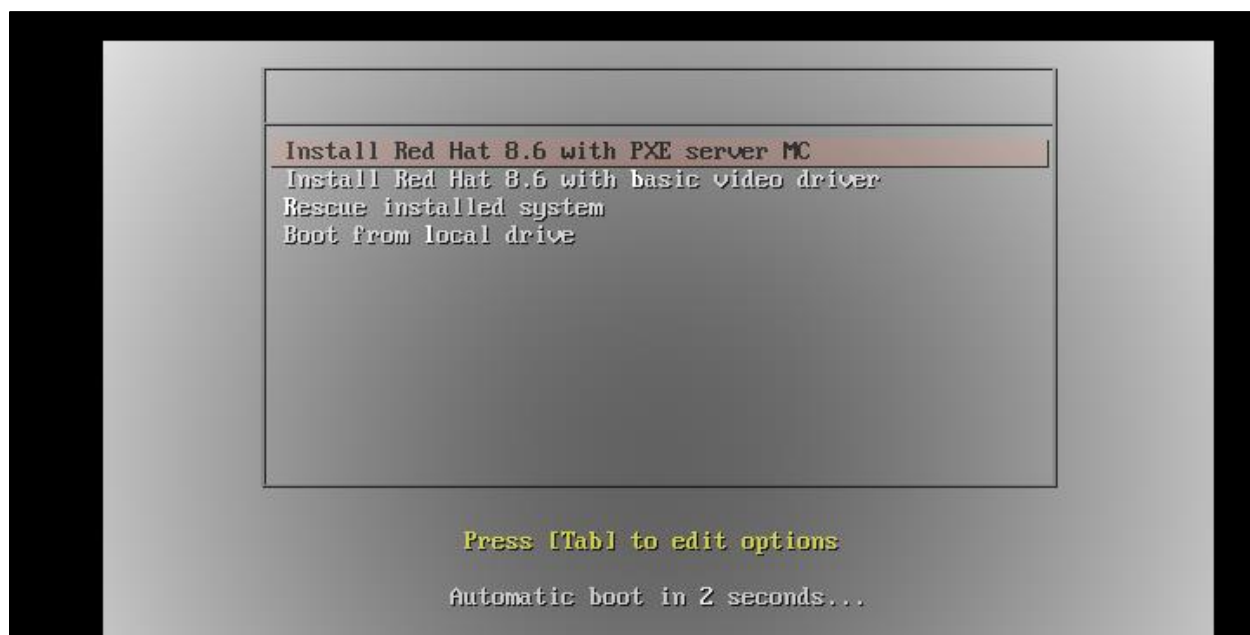


Ilustración 16: Monitoreo del servidor DHCP y PXE.

```
May 28 11:29:30 PXE dhcpd[2859]: DHCPDISCOVER from 52:54:00:71:16:ad via ens3
May 28 11:29:30 PXE dhcpd[2859]: DHCPOFFER on 192.168.100.92 to 52:54:00:71:16:ad via ens3
May 28 11:29:30 PXE dhcpd[2859]: DHCPREQUEST for 192.168.100.92 (192.168.100.90) from 52:54:00:71:16:ad via ens3
May 28 11:29:30 PXE dhcpd[2859]: DHCPACK on 192.168.100.92 to 52:54:00:71:16:ad via ens3
May 28 11:29:30 PXE systemd[1]: Started Tftp Server.
May 28 11:29:30 PXE in.tftpd[3281]: Client ::ffff:192.168.100.92 finished pxelinux.0
May 28 11:29:30 PXE in.tftpd[3282]: Client ::ffff:192.168.100.92 finished ldlinux.c32
May 28 11:29:30 PXE in.tftpd[3293]: Client ::ffff:192.168.100.92 finished pxelinux.cfg/default
May 28 11:29:42 PXE in.tftpd[3295]: Client ::ffff:192.168.100.92 finished vesamenu.c32
May 28 11:29:42 PXE in.tftpd[3296]: Client ::ffff:192.168.100.92 finished libcom32.c32
May 28 11:29:42 PXE in.tftpd[3297]: Client ::ffff:192.168.100.92 finished libutil.c32
May 28 11:29:42 PXE in.tftpd[3298]: Client ::ffff:192.168.100.92 finished pxelinux.cfg/default
May 28 11:29:57 PXE in.tftpd[3299]: Client ::ffff:192.168.100.92 finished S0/rhel-8.6/images/pxeboot/vmlinuz
May 28 11:30:12 PXE in.tftpd[3300]: Client ::ffff:192.168.100.92 finished S0/rhel-8.6/images/pxeboot/initrd.img
```

En las capturas anteriores se logra observar el funcionamiento completo del servidor PXE junto con DHCP, siendo la última captura una vista del monitoreo de los servidores en el que se logra observar la asignación de IP por parte del servidor DHCP, para después comentar con el servicio de PXE.

4.6 Tareas extras

Es importante mencionar que durante las prácticas, me fue brindado capacitación a través de cursos, los cuales duraron aproximadamente dos meses, estos cursos fueron:

- Red Hat Enterprise Linux Automation with Ansible
- Red Hat System Administration I
- Red Hat System Administration II



Capítulo 5. Resultados.

Al cabo de mi estancia en la empresa en el lapso de las prácticas, finalicé y conseguí analizar diferentes procesos, realizados por diferentes áreas como sería Data center y Recursos humanos, que pudieran ser automatizados para así brindar ayuda en dichas áreas para mejorar la productividad de la empresa

Para conseguir dicha tarea se tuvo que entender de manera específica los flujos de trabajo relacionados a las tareas y procesos, además de analizar a profundidad la información que la empresa y área poseen en cada proceso para así poder aprovechar la información de mejor manera.

Se consiguió generar soluciones lo más óptimas posibles para resolver los problemas de los análisis anteriores, utilizando conocimiento tanto en lenguajes de programación como Java y Python como también la automatización de tareas en sistemas Linux. Según fuera la tarea fue agregada a la página oficial de la intranet de MC.

Se crearon varias máquinas virtuales además de contenedores para poder ejecutar nuevos servicios que cumplieran una o varias necesidades de la empresa, además de configurarlos de la manera óptima para adaptarse a las necesidades de los usuarios, coordinando así tanto las nuevas implementaciones como las ya establecidas con anterioridad.

Finalmente se debe mencionar que desde la primera tarea hasta la última se realizaron utilizando la mayor cantidad de pautas y protocolos de seguridad en la creación de código como en la implementación de servidores, cuidando el uso de contraseñas seguras además del uso de protocolos seguros como son https y SSH.

A continuación, se listará las diferencias del antes y después de haber implementado las tareas asignadas durante las prácticas laborales.



5.1 Módulo "Checador"

Antes de implementar el módulo, el encargado de recursos humanos tenía que descargar con el programa nativo del checador un archivo Excel donde cada línea mostraba cada registro dentro del margen de fechas establecido, después tenía que verificar por cada empleado su hora de entrada, salida y sus horas de comida tanto entrada como salida, ya que en ocasiones los empleados registran más de una vez ya sea por error o por asegurarse que el registro se realizó correctamente. Para finalmente crear un reporte con los datos procesados, todo este proceso se tiene que hacer cada quincena, y lleva aproximadamente 3 horas realizarlo en su totalidad.

Después de haber implementado el módulo el encargado de recursos humano únicamente tiene que iniciar sesión a la intranet, entrar a la sección de "Checador", seleccionar el rango de fechas, y dar clic en el botón "Generar reporte", con esto logra obtener el reporte final con los datos limpios de manera casi instantánea. Por lo que el tiempo fue reducido casi en su totalidad.

5.2 Taiga

Antes de crear el servidor de Taiga, los registros, información y progreso de los proyectos eran colocados en un archivo Excel compartido a través de Onedrive entre los miembros del equipo, esto era muy poco práctico por las mismas limitaciones de Excel al ser un programa no destinado a esta tarea, en consecuencia, la tarea de realizar estos registros conllevaba varias horas de trabajo rebasando las 6 u 8 horas.

Después con la creación e implementación de Taiga, el tiempo de realizar los registros y progresos de los proyectos se volvió tanto más cómodo como eficiente al ser una plataforma dedicada para esta tarea, reduciendo así el tiempo empleado a tan solo unos minutos o por máximo un par de horas. Por lo tanto, el tiempo se redujo en un 75%.



5.3 Mudanza de servidor

Antes de realizar esta tarea, como se explicó en capítulo 4, las máquinas virtuales y servicios se encontraban alojadas en un servidor con fallos eléctricos, dichos fallos se detectaron cuando después de varios apagones de luz, el servidor se apaga sin motivo aparente y al reiniciarse, tardaba en volver a estar operativo. Este ciclo se repetía a las pocas horas, afectando la estabilidad de los servicios, incluida la intranet, que solo estaban disponibles el 76% del tiempo laboral de la empresa.

Después de realizar la mudanza de las máquinas virtuales y servicios al nuevo servidor no se ha presentado ninguna falla en la disponibilidad de los servicios. Con lo anterior podemos decir que actualmente la disponibilidad de los servicios es del 100%.

5.4 Scripts

Durante el tiempo de las prácticas, varios compañeros tuvieron que ir a otros estados a realizar instalaciones y mantenimiento a distintos clientes, una de las configuraciones que tienen que realizar es la creación de Bonds en las distintas interfaces de los dispositivos, esto puede ser una gran problema debido a que en ocasiones durante la instalación de nuevos dispositivos o migraciones de sistemas operativos provocan que el nombre de las interfaces detectadas por el sistema operativo cambien, provocando a su vez errores de conexión, y dado que para corregirlos tienen que detectar direcciones Mac y nombres de interfaces de varias conexiones, terminan llevándose más tiempo de lo pensado para el proyecto, llegando incluso a tardar una o dos horas, dependiendo del número de conexiones, siendo crucial debido a cuestiones de gastos y contratos.

Pero ahora con ayuda de los scripts creados se pretende disminuir gran parte del tiempo invertido en esta tarea, ya que el script detecta de manera automática el nombre de interfaces con solamente la dirección MAC, por lo que si se tiene un registro de las direcciones MAC de cada dispositivo esta tarea puede ser realizada en cuestión de entre 5 a 15 minutos, disminuyendo así el tiempo en un 85% aproximadamente.



5.5 Servidor PXE y DHCP

Antes de la existencia del servidor, dentro de MC la instalación de sistemas operativos era utilizando memorias booteables o descargando los isos de un servidor para después adjuntarlas a las máquinas virtuales, conllevando a varios problemas, principalmente de seguridad. Al utilizar memorias aumenta el riesgo de varias máquinas se contaminen de algún virus, comprometiendo datos sensibles tanto de la empresa como de los clientes. Por otro lado, utilizar isos almacenados en un servidor conlleva a tener que dar acceso al servidor a cualquier usuario que necesitara algún iso, teniendo así problemas de control con quién accede al servidor.

Ahora con el servidor PXE se busca centralizar todos los sistemas operativos utilizados por la empresa, de esta manera se podrán instalar sistemas operativos sin recurrir a descargar isos o utilizar memorias, además de tener un mejor control de quienes ocupan el servidor como cliente gracias al servidor DHCP y DNS de la empresa.



Capítulo 6. Competencias desarrolladas y/o aplicadas.

6.1 Competencias generales

Durante el desarrollo del proyecto se lograron aplicar y adquirir competencias específicas cruciales para lograr los objetivos planteados. El conocimiento y uso de lenguajes de programación como son Python, Java y JavaScript fueron necesarios para adaptarse a los módulos de la intranet y lograr crear e implementar con éxitos los nuevos módulos, realizando las mejoras prácticas de programación, consiguiendo así un código legible y fácil de mantener.

El conocimiento y práctica del manejo de sistemas operativos de Linux fue fundamental para cumplir con la mayoría de las tareas, debido a que en su mayoría los servidores se encuentran en sistemas operativos de la familia Red Hat, por lo que era necesario conocer la mayor cantidad de comandos, configuración de servicios o daemon, sistema de archivos, configuraciones de conectividad y manejo de seguridad por medio del firewall del mismo sistema operativo, todo con el fin de mantener un servidor estable y seguro. Además, se utilizó la codificación de script en Bash para poder administrar de manera eficiente y óptima los servidores.

Creación e implementación de aplicaciones en contenedores, además de su configuración de imágenes por medio de Docker compose y Dockerfile, permitiendo desplegar con éxito aplicación de código abierto adaptándolas a las necesidades de la empresa, siempre manejando normativas de seguridad como la creación de un usuario sin privilegios para desplegar los contenedores además del uso preferencial de Podman en vez de Docker.

Por último, se aprendió y desarrollaron las habilidades para manejar y codificar Ansible playbook para la realización de tareas de manera automatizada, permitiendo así poder desempeñar otras tareas mientras se ejecutan y monitorean los playbook en distintos dispositivos.



6.2 Competencias específicas

De manera paralela se aplicaron y desarrollaron competencias generales siendo la más importante la comunicación verbal y escrita, para poder transmitir ideas y problemáticas de manera concisa a otros compañeros de trabajos en distintas áreas.

Se aplicaron técnicas de solución de problemas, primeramente, identificándose, analizarlos, plantearlos y finalmente encontrarles una solución óptima y concisa aplicando la solución en el menor tiempo posible. Solucionando problemas de manera sistemática y creativa.

La toma de decisión y gestión de tiempo fue otra competencia crucial para conseguir entregar las tareas de manera eficiente, priorizando siempre aquellas tareas que conlleva más tiempo o afectarán a más individuos, decidiendo también los momentos óptimos para realizar las tareas para en algunos casos afectar a la menor cantidad de usuarios posibles.



Conclusiones.

Los resultados esperados fueron conseguidos de manera exitosa, consiguiendo en algunos mayores beneficios que los esperados, logrando cumplir con los objetivos planteados de manera general y específica, brindando una intranet más sólida y completa para el uso de las distintas áreas de la empresa, siempre cumpliendo con los estándares y protocolos de seguridad.

Recomendaciones.

En varias ocasiones se presentó la situación de interactuar con dispositivos físicos siendo switches o servidores, y al tener poca experiencia manipulando estos dispositivos, se recomienda buscar alternativas para tener acercamiento con total seguridad para futuros proyectos, debido a que en muchos proyectos es necesario manipular dispositivos para conectarlos, configurarlos o incluso instalarlos por lo que la seguridad en el manejo de estos es necesario.



Referencias.

Red Hat. (2022). *Red Hat System Administration I* (2.^a ed.).

Red Hat. (2022). *Red Hat System Administration II* (2.^a ed.).

Red Hat. (2022). *Red Hat Enterprise Linux Automation with Ansible* (2.^a ed.).

Chapter 7. Using Red Hat Subscription Manager Subscription Central 1-Latest / Red Hat

Customer Portal. (s. f.). Red Hat Customer Portal.

https://access.redhat.com/documentation/en-us/subscription_central/1-latest/html/getting_started_with_rhel_system_registration/adv-reg-rhel-using-rhsm_

Chapter 2. Getting Started with NetworkManager Red Hat Enterprise Linux 7 / Red Hat

Customer Portal. (s. f.). Red Hat Customer Portal.

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/networking_guide/getting_started_with_networkmanager

Chapter 6. Redundant Array of Independent Disks (RAID) Red Hat Enterprise Linux 5 / Red Hat

Customer Portal. (s. f.). Red Hat Customer Portal.

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/5/html/deployment_guide/ch-raid

Comprensión y resolución de problemas del arranque PXE en el fabric ACI. (2022, 11 marzo).

Cisco. https://www.cisco.com/c/es_mx/support/docs/cloud-systems-management/application-policy-infrastructure-controller-apic/212041-Understanding-and-Troubleshooting-PXE-bo.html



¿En qué consiste Scrum? - Explicación sobre la metodología Scrum - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/scrum/>

Gillis, A. S. (2022, 13 octubre). *Secure File Transfer Protocol (SSH File transfer Protocol).* Content Management.

[https://www.techtarget.com/searchcontentmanagement/definition/Secure-File-Transfer-Protocol-SSH-File-Transfer-Protocol#:~:text=Secure%20File%20Transfer%20Protocol%20\(SFTP\)%20is%20a%20network%20protocol%20for,of%20files%20over%20a%20network.](https://www.techtarget.com/searchcontentmanagement/definition/Secure-File-Transfer-Protocol-SSH-File-Transfer-Protocol#:~:text=Secure%20File%20Transfer%20Protocol%20(SFTP)%20is%20a%20network%20protocol%20for,of%20files%20over%20a%20network.)

Información sobre la empresa - Red Hat. (s. f.). <https://www.redhat.com/es/about/company>

JasonGerend. (2023, 9 marzo). *Protocolo de configuración dinámica de host (DHCP).* Microsoft Learn. <https://learn.microsoft.com/es-es/windows-server/networking/technologies/dhcp/dhcp-top>

Keenan, A. (2024, 29 enero). *Discover 3 advantages of Podman over Docker | Red Hat Developer.* Red Hat Developer. https://developers.redhat.com/articles/2023/08/03/3-advantages-docker-podman#_3__better_tools_and_extensions

NFS e iSCSI: diferencias entre los protocolos de intercambio de datos - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/compare/the-difference-between-nfs-and-iscsi/>

¿Qué es AOS? - Explicación sobre la arquitectura orientada a servicios - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/service-oriented-architecture/>

¿Qué es el código abierto? - Explicación del código abierto - AWS. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/open->



source/#:~:text=Se%20denomina%20c%C3%B3digo%20abierto%20a,es%20accesible%20de%20manera%20p%C3%BAblica.

Rick-Anderson. (2024, 19 marzo). *Introducción a MVC de ASP.NET*. Microsoft Learn.

<https://learn.microsoft.com/es-es/aspnet/mvc/overview/older-versions-1/overview/asp-net-mvc-overview>

The CentOS project. (s. f.). <https://www.centos.org/>

Tingting, G. P. G. (2024, 12 marzo). *What is SSH? How does SSH work? - Huawei*. Huawei.

<https://info.support.huawei.com/info-finder/encyclopedia/en/SSH.html>

What is lightweight directory access protocol (LDAP) authentication? (s. f.).

<https://www.redhat.com/en/topics/security/what-is-ldap-authentication>

What is SELinux? (s. f.). <https://www.redhat.com/en/topics/linux/what-is-selinux>