INSTITUTO NEBRIJA | Formación Profesional

TÍTULO DEL PROYECTO

https://github.com/wheresmyicecream/Trabajo-ISO-/branches

ABSTRACT

Nombre del alumno o de la alumna:Javier Granados (Scrum Master), Daniel Alaez, Marco García, Antonio Medina

Curso académico:1 ASIR

Tutora/Tutor del proyecto: Carmelo

ÍNDICE PAGINADO

1. JUSTIFICACIÓN DEL PROYECTO

El presente proyecto tiene como finalidad la implantación de una infraestructura informática para una pequeña empresa basada en sistemas operativos Linux, ajustada a las necesidades de una pequeña empresa del sector tecnológico. Esta dualidad permite aprovechar las ventajas de cada entorno: estabilidad, seguridad, bajo coste en Linux, y compatibilidad con software comercial y entorno gráfico amigable en Windows.

La empresa, en fase de expansión, requiere soluciones eficientes, seguras y sostenibles y permite cubrir las necesidades tanto de usuarios técnicos como administrativos, garantizando un entorno de trabajo robusto, actualizado y seguro.

Además, la automatización mediante scripts en sistemas Linux mejora significativamente la eficiencia en tareas de mantenimiento, seguridad y administración de usuarios.

2. INTRODUCCIÓN

Este documento presenta una propuesta técnica de infraestructura dual para una organización con 12 empleados, dedicada a servicios informáticos y soporte técnico. Dicha infraestructura se ha diseñado para funcionar en entornos virtualizados, permitiendo la instalación, configuración, pruebas y despliegue de los distintos sistemas operativos.

Durante el desarrollo, se ha utilizado Ubuntu Server para servidores de red .Se han implementado medidas básicas de seguridad, incluyendo firewalls, políticas de actualización, gestión de permisos y monitorización por logs.

Asimismo, se han desarrollado y documentado múltiples scripts Bash para Ubuntu, que permiten automatizar la recolección de logs, la realización de copias de seguridad, la gestión de usuarios y el mantenimiento periódico del sistema. Todo el trabajo ha sido versionado y documentado en GitHub para garantizar la trazabilidad del desarrollo.

3. OBJETIVOS

Diseñar e implantar una infraestructura informática virtualizada basada en sistemas operativos Linux, segura, funcional y adaptada a los distintos perfiles de usuario de una empresa tecnológica, aplicando técnicas de automatización y metodologías ágiles.

B. OBJETIVOS ESPECÍFICOS

- Analizar y comparar diferentes versiones de Windows (10 y 11) y distribuciones de Linux (Ubuntu, Debian, Rocky Linux).

- Instalar ambos entornos operativos en máquinas virtuales mediante VirtualBox y/o VMware.

- Configurar los sistemas operativos con criterios de seguridad, eficiencia y compatibilidad con la red de la empresa.

- Desarrollar scripts Bash para automatizar tareas de mantenimiento, backup, y gestión de usuarios en servidores Linux.

- Gestionar y documentar el proyecto utilizando metodología un tablero Kanban digital.

- Versionar y publicar todos los recursos técnicos (scripts, capturas, configuraciones) en un repositorio GitHub colaborativo.
- Presentación y vídeo explicativo.

4. DESARROLLO

1. FUNDAMENTACIÓN TEÓRICA: lo que vamos a hacer, procedimientos, resolución de la hipótesis o situaciones planteadas, tareas a realizar
2. Materiales y métodos: estrategias de búsqueda, metodología y técnicas utilizadas
3. Resultados y análisis

(Sesión 1: Daniel Análisis comparativo, Antonio; Seguridad, Javier: Github y esquema)

(Sesión 2: Daniel instalacion linux , Antonio: análisis de necesidades e introducción (red team vs blue team), Javier: KanBan, raíces github)

○ Introducción y contexto

Este documento presenta una propuesta para implantar una infraestructura de sistemas operativos mixta en una pequeña empresa en crecimiento, con el fin de garantizar una plataforma estable, segura y rentable, adaptada a los distintos perfiles de usuario existentes en la organización.

Sector de la empresa: Servicios informáticos y soporte técnico

Tamaño: 12 empleados

Modalidad de trabajo: Presencial con opción de teletrabajo parcial

Sistema operativo: Linux (Ubuntu)

Necesidades y perfiles de usuario:

Perfil	Nº Usuarios	Tareas principales
Dirección	1	Gestión, informes, comunicación externa.
Administración	2	Facturación, nóminas, ofimática, correo.
Técnicos de soporte	3	Diagnóstico remoto, configuración de redes, asistencia a clientes.
Desarrolladores	6	Programación, testing, servidores locales, Git.

Ventajas del uso de linux:

Aspecto	Ventajas en Linux
Coste	Sin licencias de sistema operativo ni ofimática.
Seguridad	Menor exposición a malware, actualizaciones constantes y rápidas.
Flexibilidad	Alta personalización del entorno según necesidades del usuario.
Rendimiento	Requiere menos recursos que Windows; ideal para hardware modesto.
Automatización	Fácil integración con scripts, cron, backups automáticos.
Compatibilidad	Compatible con la mayoría de herramientas web, correo y edición.

Implementar exclusivamente Linux (Ubuntu) en esta pequeña empresa es una solución eficiente, segura y económicamente sostenible. Especialmente adecuada para empresas con cierto perfil técnico, Linux permitirá crecer sin depender de licencias, ofreciendo rendimiento y control total sobre los sistemas.

○ Análisis comparativo (Windows vs Linux)

Analizar distintas versiones de Windows y distribuciones Linux (Ubuntu, Debian, Rocky Linux, Windows 11

Windows

Windows 10 es una de las versiones más populares y ampliamente utilizadas. Ofrece una interfaz moderna, soporte para una gran variedad de hardware y software, y es ampliamente adoptado tanto en entornos domésticos como empresariales. Recibe actualizaciones de seguridad periódicas y tiene soporte para aplicaciones legacy y modernas (UWP).

Windows 11

Windows 11 es la versión más reciente (lanzada en 2021). Introduce una interfaz más pulida, mejoras en la gestión de ventanas (Snap Layouts), integración con Microsoft Teams y requisitos de hardware más estrictos (como TPM 2.0 y Secure Boot). Está orientado a mejorar la productividad y la seguridad, pero puede no ser compatible con hardware más antiguo.

Ventajas de Windows (en general):
Compatibilidad con la mayoría del software comercial, facilidad de uso, soporte técnico amplio, integración con servicios de Microsoft.

Desventajas:
Licencia de pago, menos personalizable, más susceptible a malware si no se toman precauciones.

Linux

Ubuntu

Ubuntu es una de las distribuciones más populares y amigables para el usuario. Basada en Debian, se centra en la facilidad de uso, actualizaciones regulares y una gran comunidad. Es ideal tanto para principiantes como para usuarios avanzados. Se utiliza mucho en escritorios, servidores y entornos de desarrollo.

Debian

Debian es conocida por su estabilidad y robustez. Es la base de muchas otras distribuciones (incluyendo Ubuntu). Suele tener paquetes más antiguos pero muy probados, lo que la hace ideal para servidores y sistemas donde la estabilidad es prioritaria sobre la novedad.

Rocky Linux

Rocky Linux es una distribución empresarial, creada como reemplazo de CentOS tras el cambio de enfoque de Red Hat. Es compatible a nivel binario con Red Hat Enterprise Linux (RHEL), lo que la hace ideal para entornos empresariales que requieren estabilidad, soporte a largo plazo y compatibilidad con software de servidor.

Ventajas de Linux (en general):
Gratuito y de código abierto, altamente personalizable, menos susceptible a virus, ideal para servidores y desarrollo, gran variedad de distribuciones para diferentes necesidades.

Desventajas:
Curva de aprendizaje para usuarios nuevos, menor compatibilidad con software comercial (especialmente juegos y aplicaciones profesionales), soporte técnico más comunitario que comercial (excepto en distribuciones empresariales).

Característica	Windows 10/11	Ubuntu	Debian	Rocky Linux
Licencia	Comercial	Libre	Libre	Libre
Facilidad de uso	Muy alta	Alta	Media	Media
Actualizaciones	Automáticas	Regulares	Menos frecuentes	Regulares
Estabilidad	Alta	Alta	Muy alta	Muy alta
Soporte software	Muy amplio	Amplio (open source)	Amplio (open source)	Empresarial
Hardware antiguo	Mejor en Win10	Bueno	Excelente	Bueno
Orientación	Hogar/empresa	Escritorio/servidor	Servidor/infraestructura	Servidor/empresa

Resumen

- **Windows 10/11:** Mejor para usuarios que necesitan compatibilidad con software comercial, juegos o buscan una experiencia lista para usar.

- **Ubuntu**: Ideal para quienes quieren empezar en Linux, desarrolladores o buscan un sistema de escritorio moderno y fácil de usar.

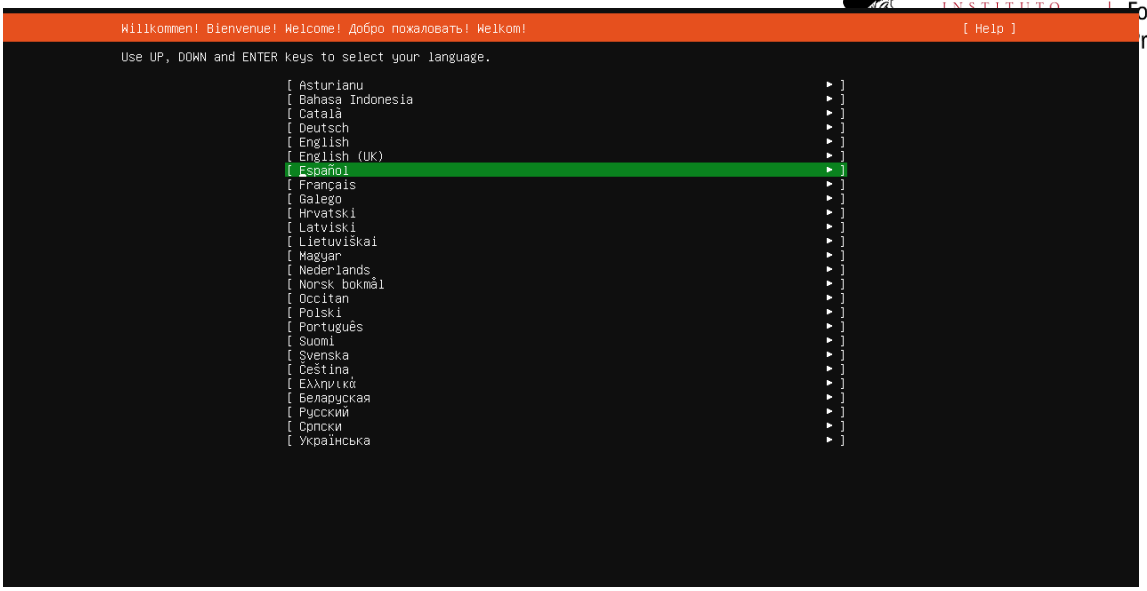
- Debian: Perfecto para servidores o usuarios que priorizan la estabilidad y la robustez.

- Rocky Linux: Recomendado para empresas que buscan una alternativa gratuita y estable a RHEL/CentOS.

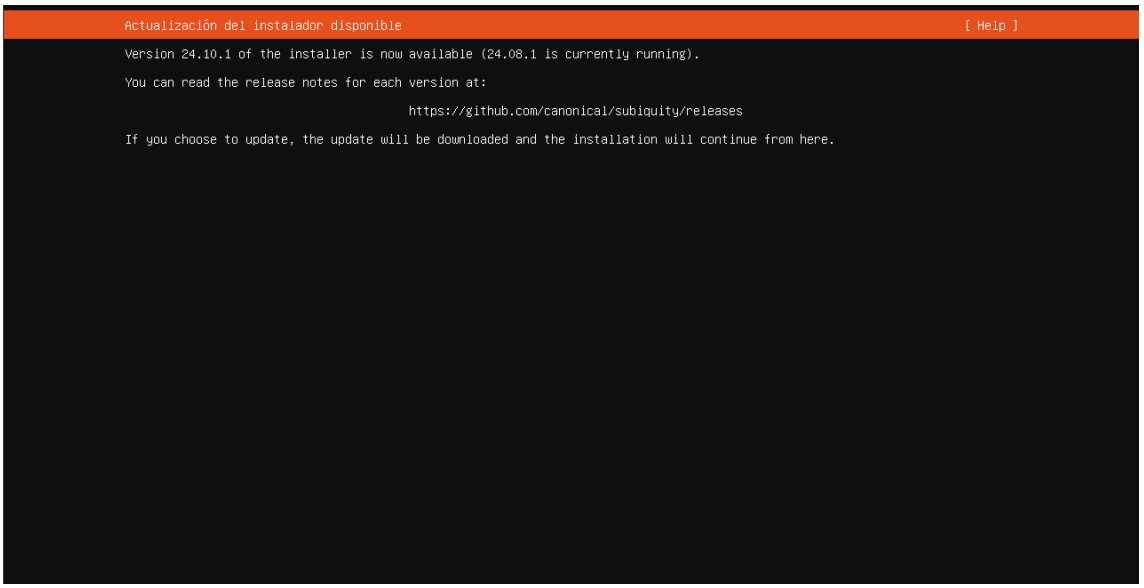
Instalación de Ubuntu (Antonio)

Al iniciarse la máquina nos saldrá para elegir estas 2 opciones, debemos esperar sin elegir nada.

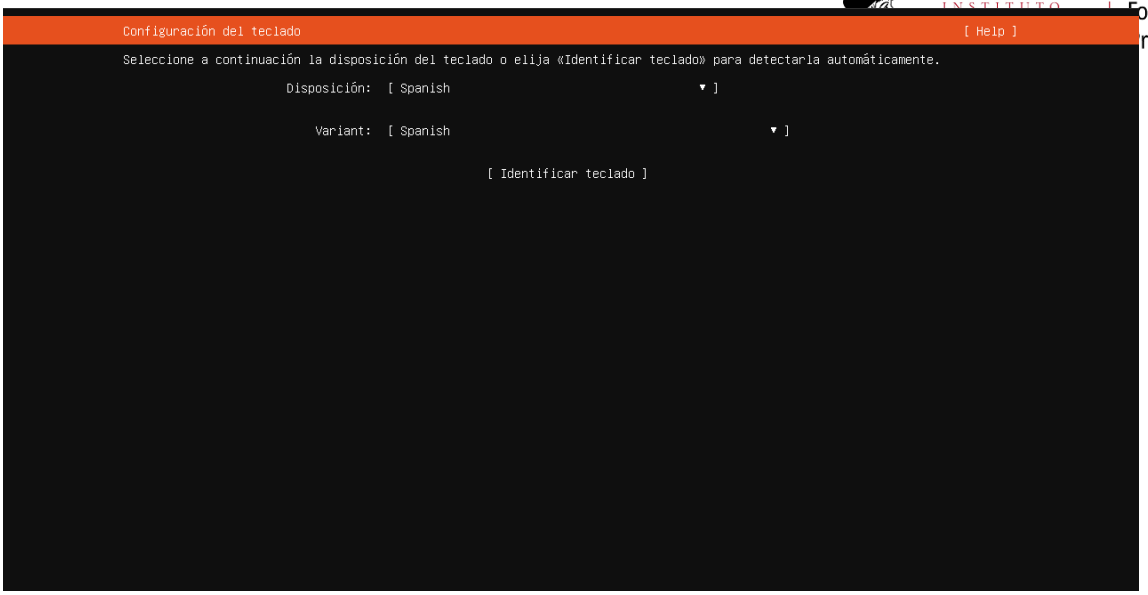
Después, seleccionaremos el idioma “Español” y le daremos a la tecla “enter”.



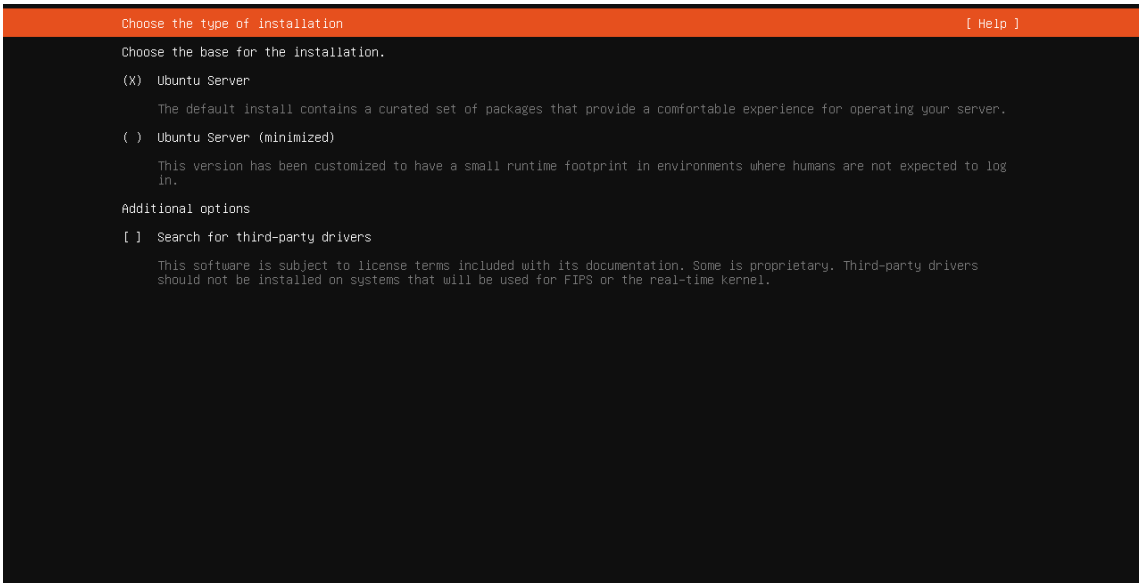
Ahora, le daremos a la tecla enter en la opción de “Continuar sin actualizar”.



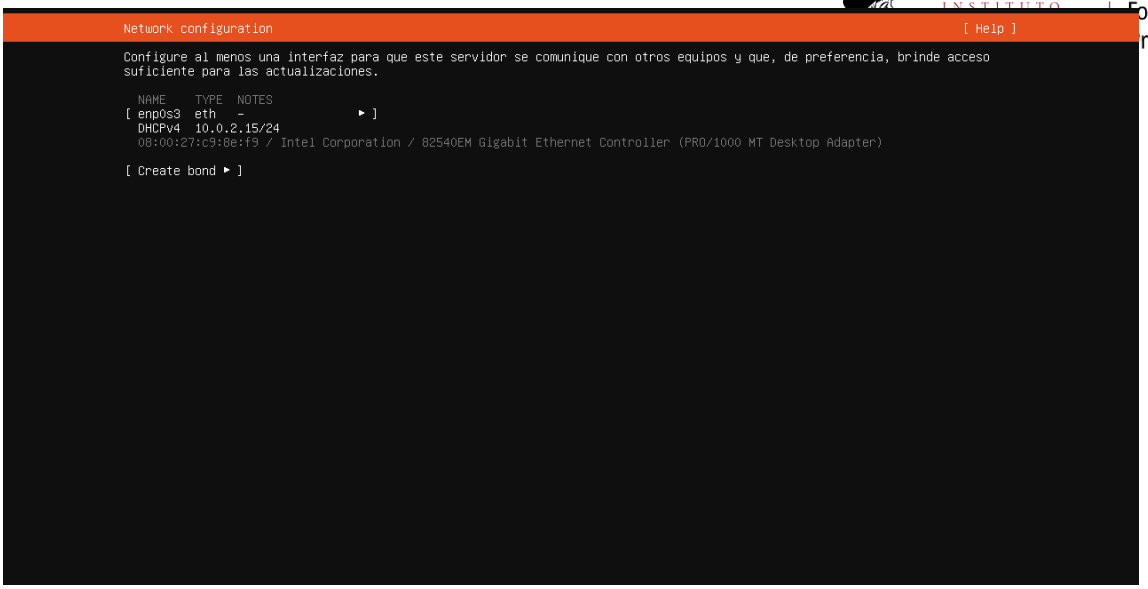
Ahora le daremos a la opción de “Hecho”



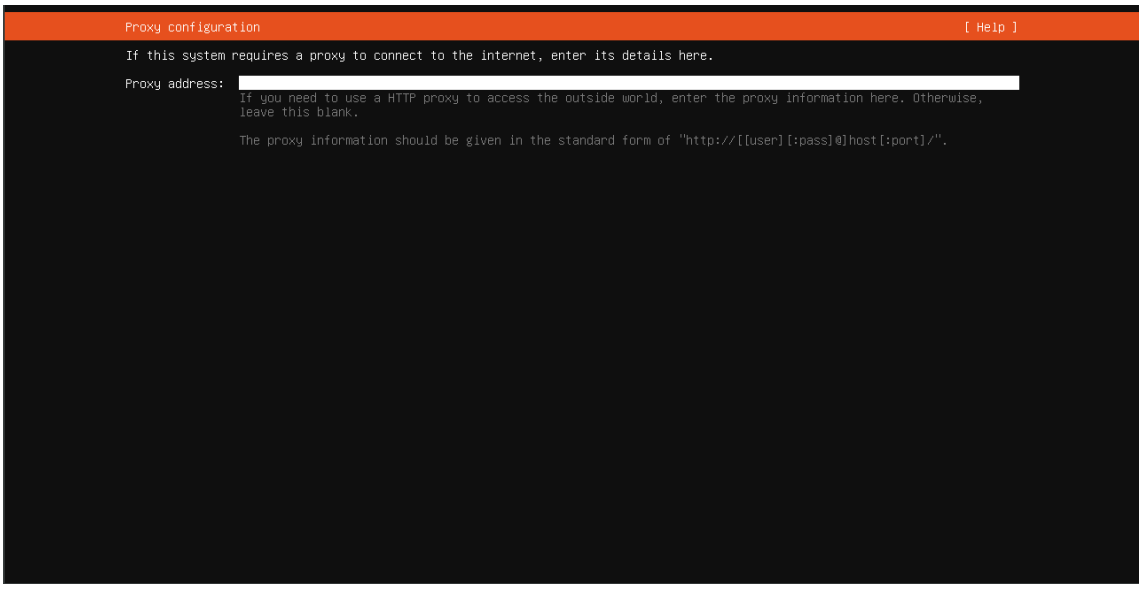
Ahora dejaremos la opción de “Ubuntu Server” y le daremos a “Hecho”.



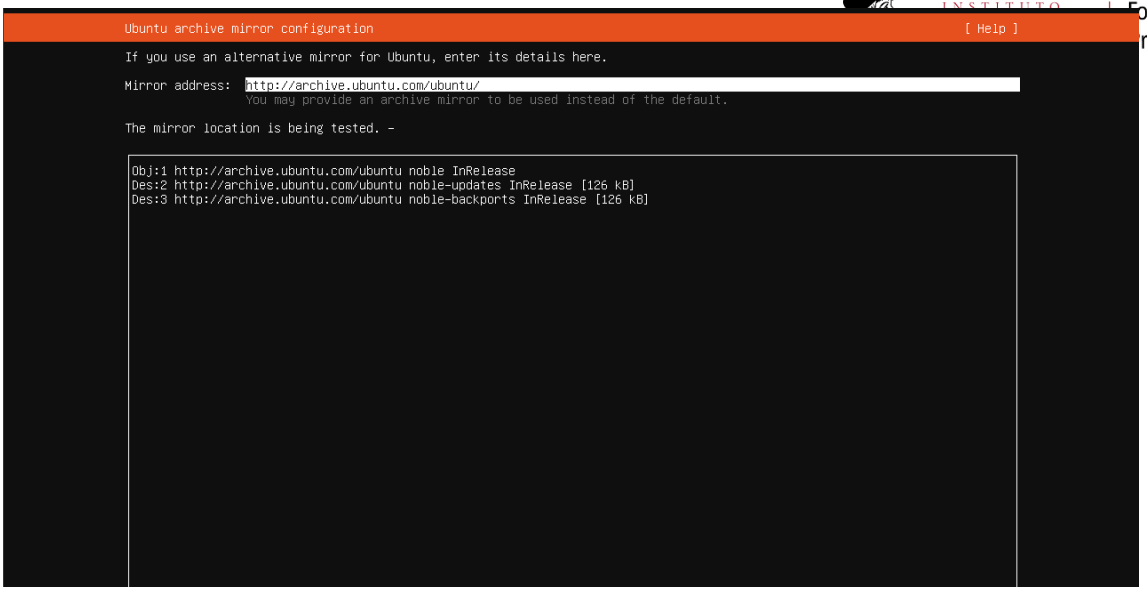
Ahora le volveremos a dar a “Hecho”.



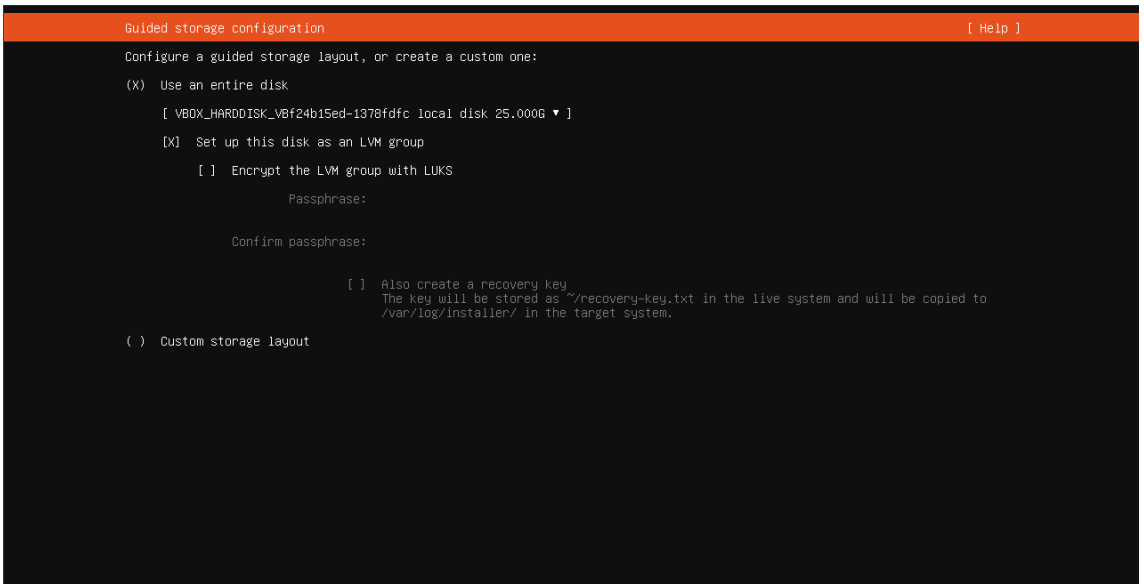
Le volveremos a dar a “Hecho” ya que no tenemos proxy.



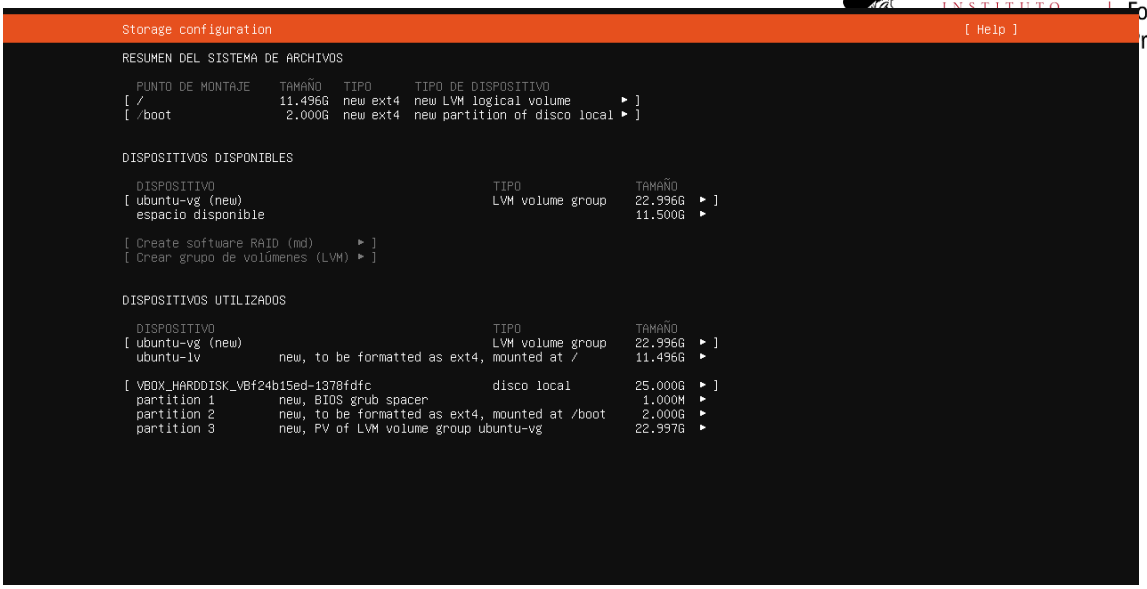
Ahora, le daremos a “Hecho” otra vez.



Dejaremos marcada la opción de “Use entire disk” y le daremos a “Hecho”.

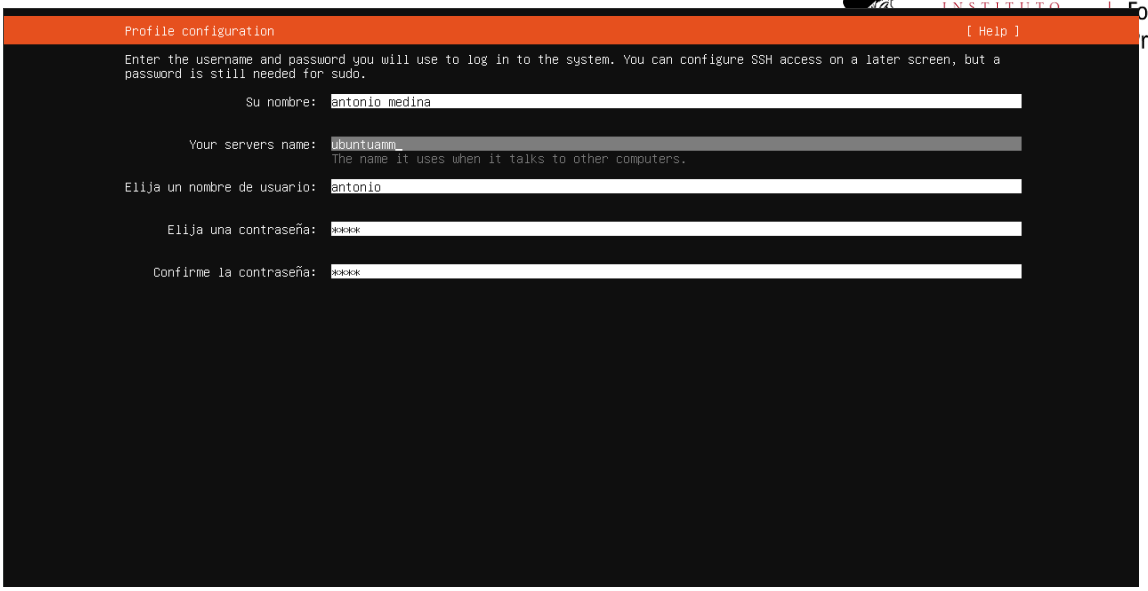


Le daremos a “Hecho”.

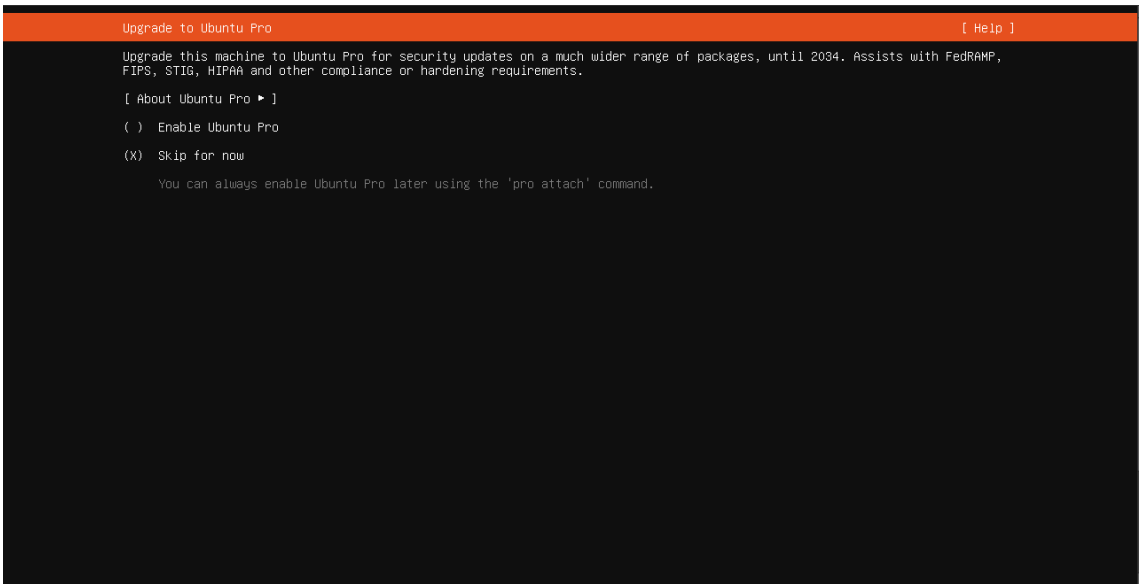


Ahora le daremos a “Continuar”.

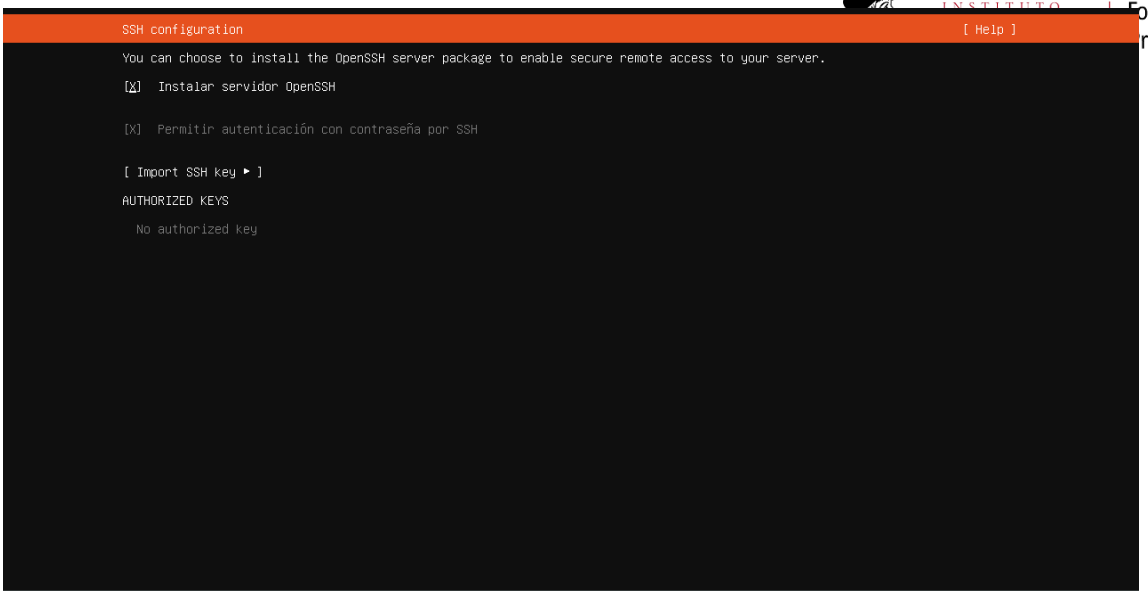
Ahora introduciremos nuestro nombre, el nombre del servidor, el nombre de usuario y la contraseña, después le daremos a “Hecho”.



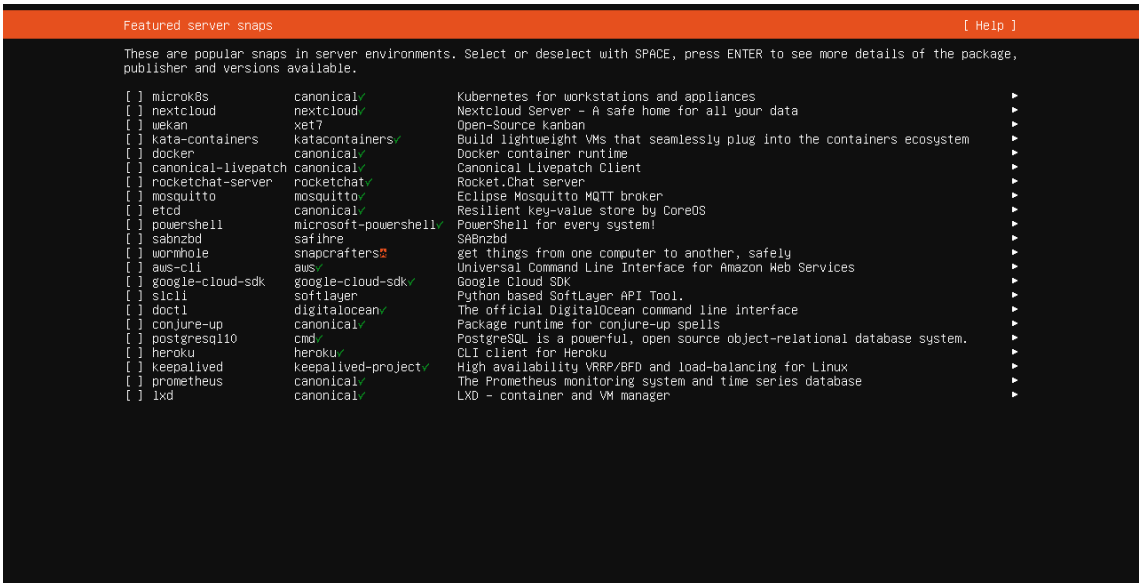
Ahora le daremos a “Continuar”.



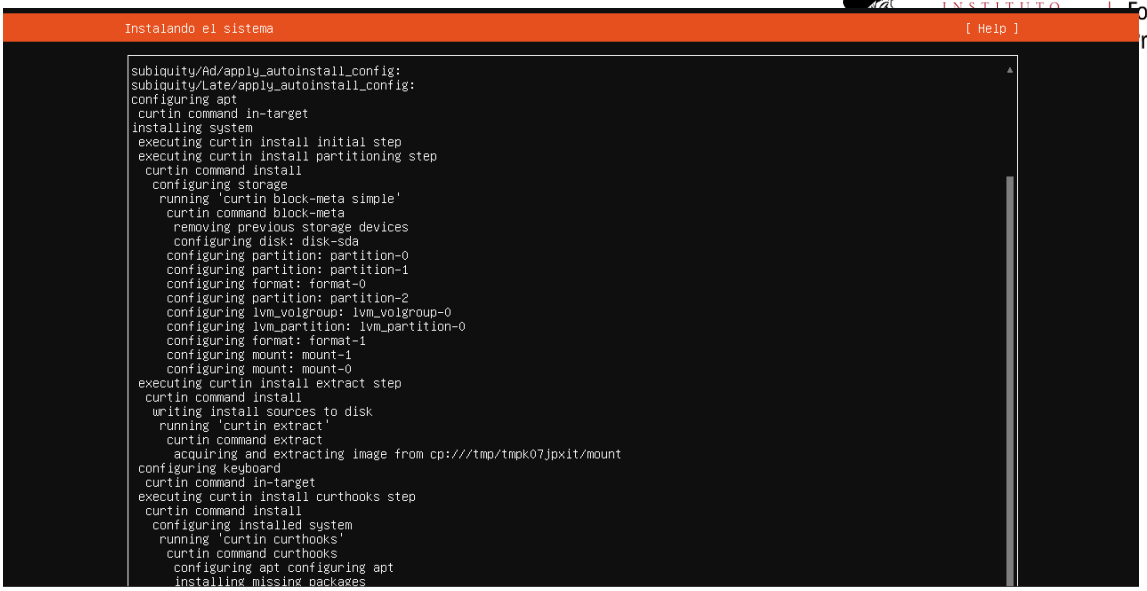
Ahora marcaremos la casilla de “Instalar servidor OpenSSH” y le daremos a “Hecho”.



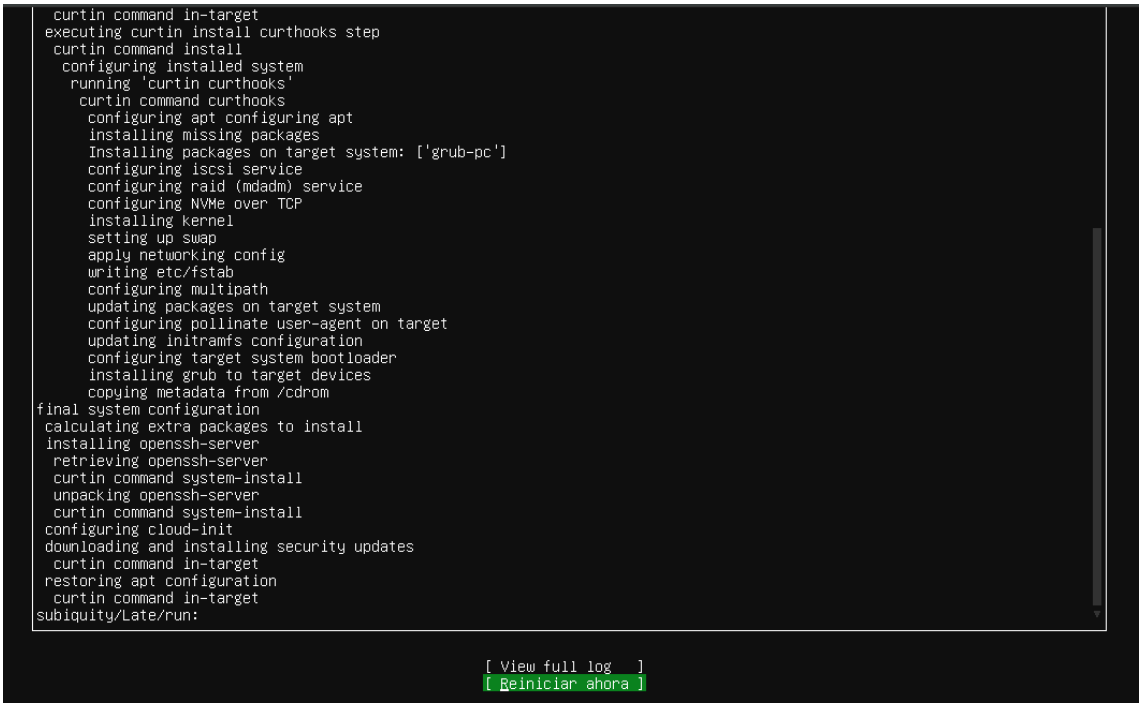
Ahora le daremos a “Hecho”.



Ahora tendremos que esperar y le daremos a



Una vez hayamos terminado le daremos a enter en la opción de “Reiniciar ahora” para reiniciar nuestra máquina.



Ping de windows a máquina virtual (con adaptador puente)

```
javi@javis:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:fd:30:f9 brd ff:ff:ff:ff:ff:ff
    inet 10.66.3.82/16 metric 100 brd 10.66.255.255 scope global dynamic enp0s3
       valid_lft 28788sec preferred_lft 28788sec
    inet6 fe80::a00:27ff:fefd:30f9/64 scope link
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:24:56:10:4d brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
```

Primero comprobamos la ip de la máquina virtual. Ahora desde windows hacemos ping a esa ip en este caso ping 10.66.3.82

```
C:\Users\Javier>ping 10.66.3.82

Haciendo ping a 10.66.3.82 con 32 bytes de datos:
Respuesta desde 10.66.3.82: bytes=32 tiempo=1ms TTL=64
Respuesta desde 10.66.3.82: bytes=32 tiempo<1m TTL=64
Respuesta desde 10.66.3.82: bytes=32 tiempo<1m TTL=64
Respuesta desde 10.66.3.82: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 10.66.3.82:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 1ms, Media = 0ms
```

ping exitoso, está conectado a la red.

○ Seguridad (firewall, políticas, actualizaciones, antivirus, logs) (Teoría Antonio)

Firewall

El firewall es la primera línea de defensa para bloquear accesos no autorizados a la red y a los sistemas.

Windows

Incorpora Windows Defender Firewall, que permite crear reglas de entrada y salida.
Soporta integración con políticas de grupo (GPO) en entornos empresariales.
Interfaz gráfica fácil de gestionar, incluso por personal no experto.

Linux (Ubuntu, Debian, Rocky)

Usa herramientas como UFW (Uncomplicated Firewall) en Ubuntu y firewalld o iptables en Debian y Rocky Linux.

Aunque más técnicas, ofrecen un alto grado de control y personalización.
Muy recomendables en servidores para restringir accesos por puertos y protocolos.

Recomendación

Implementar firewalls en todos los equipos y servidores, con reglas específicas según el rol del sistema (por ejemplo, permitir solo http/https en servidores web).

Políticas de seguridad

Las políticas de seguridad definen cómo deben comportarse los usuarios y el sistema ante diversos escenarios.

Windows

Uso de GPO (Group Policy Objects) en entornos con Active Directory para controlar contraseñas, bloqueo de dispositivos, ejecución de programas, etc.
Políticas locales disponibles para equipos independientes.

Linux

Control de permisos de archivos mediante el sistema de usuarios y grupos.
Uso de herramientas como sudo, SELinux (Rocky Linux) y AppArmor (Ubuntu) para controlar el acceso y limitar privilegios.

Recomendación

Aplicar el principio de mínimos privilegios en ambos entornos. En Linux, evitar el uso del usuario root en tareas diarias. En Windows, usar cuentas estándar para usuarios normales y cuentas con privilegios limitados para tareas administrativas.

Actualizaciones

Las actualizaciones corrigen vulnerabilidades que pueden ser explotadas por atacantes.

Windows

Sistema de actualizaciones automáticas a través de Windows Update.
En empresas, se puede usar WSUS para centralizar y controlar la distribución de parches.

Linux

Actualizaciones mediante gestores de paquetes (apt, dnf, yum).
Posibilidad de automatizar actualizaciones de seguridad mediante cron jobs o herramientas como unattended-upgrades.

Recomendación

Mantener todos los sistemas actualizados automáticamente, especialmente aquellos expuestos a internet (como servidores web o bases de datos). Comprobar periódicamente que no haya fallos en los procesos de actualización.

La protección contra malware es especialmente crítica en sistemas Windows, pero también puede ser útil en Linux.

Windows

Windows 11 incluye Microsoft Defender Antivirus, que ofrece protección en tiempo real, análisis programados y detección de amenazas basada en la nube.

Puede complementarse con soluciones empresariales como ESET, Bitdefender o Kaspersky.

Linux

Aunque menos afectado por malware tradicional, es recomendable usar herramientas como ClamAV en servidores de correo o archivos.

Los sistemas Linux deben protegerse especialmente contra ataques como rootkits o backdoors.

Recomendación

Implementar antivirus al menos en los equipos Windows y escáneres de malware en servidores Linux que gestionen contenido compartido (correos, ftp, etc.).

Registros de actividad (logs)

Los logs permiten detectar anomalías, hacer auditorías y cumplir con normativas.

Windows

Usa el visor de eventos para registrar accesos, errores del sistema, cambios en la configuración y más.

Se puede integrar con soluciones SIEM para análisis centralizado.

Linux

Usa archivos de log como /var/log/syslog, /var/log/auth.log, /var/log/secure.

Herramientas como rsyslog, journald y Logwatch permiten centralizar, monitorear y resumir eventos.

Recomendación

Centralizar los logs en un servidor dedicado, tanto en Windows como en Linux.

Implementar herramientas de monitoreo y alertas para detectar intentos de acceso fallidos, cambios sospechosos, etc.

Debemos combinar correctamente las herramientas y buenas prácticas de ambos entornos operativos. La combinación de firewalls activos, políticas bien definidas, actualizaciones constantes, protección antivirus adecuada y una buena gestión de logs es esencial para garantizar la seguridad a medida que la empresa crece.

Seguridad de nuestra empresa:

Implementación recomendada:

-Firewall: UFW activo en todos los equipos.

-**Actualizaciones automáticas**

-**Antivirus básico**

-**Cuentas de usuario limitadas:** Sin permisos root directo; uso de sudo.

-**Acceso remoto seguro:** VPN y SSH con autenticación por clave.

-**Centralización de logs:** syslog y revisión con Logwatch o scripts periódicos.

Posibles vulnerabilidades:

Vulnerabilidad	Riesgo Principal	Medidas de Corrección Clave
Cuentas mal configuradas	Acceso no autorizado	Crear usuarios sin permisos; controlar uso de `sudo`; activar 2FA
Sistema desactualizado	Exposición a exploits conocidos	Activar `unattended-upgrades`; aplicar parches críticos
Puertos abiertos innecesarios	Ataques remotos por servicios no usados	Configurar `ufw`; escaneo con `nmap`; cerrar puertos no requeridos
Acceso remoto inseguro	Intercepción o acceso no autorizado	Usar SSH con claves; desactivar contraseñas; VPN segura con WireGuard
Software de origen no confiable	Malware o puertas traseras	Usar repos oficiales; verificar firmas de paquetes; auditar software
Permisos incorrectos en archivos	Acceso a datos sensibles	Revisar y ajustar permisos con `chmod`, `chown`, uso de ACLs si es necesario
Logs no supervisados	Actividades maliciosas pasan desapercibidas	Centralizar logs; revisar `/var/log`; configurar alertas
Malware (aunque poco común)	Daños al sistema o fuga de datos	Usar ClamAV; evitar scripts externos; escaneo con `rkhunter`, `chkrootkit`

Red Team VS Blue Team:

Rol / Aspecto	🔴 Red Team (Ofensivo)	🔵 Blue Team (Defensivo)
Objetivo principal	Simular ataques reales para detectar debilidades	Defender la infraestructura y mitigar amenazas
Mentalidad	Atacante (piensa como un hacker)	Defensor (piensa como un guardián)
Actividades clave	- Pentesting - Ingeniería social - Explotación de fallos	- Monitorización - Gestión de incidentes - Aplicación de parches
Herramientas comunes	Metasploit, Nmap, Kali Linux, Burp Suite	SIEM, UFW, fail2ban, ClamAV, rkhunter, Suricata
Conocimientos requeridos	Exploits, redes, evasión de defensas	Análisis de logs, hardening, respuesta ante incidentes
Resultados esperados	Informe de vulnerabilidades y vectores de ataque	Infraestructura robusta, planes de contingencia, alertas
Tipo de pruebas	Intrusiones simuladas, phishing, escaneo	Análisis de tráfico, integridad de sistemas, backups
Interacción interna	Simulan ser un enemigo	Actúan como equipo interno de TI o seguridad
Relación entre ellos	Retan la seguridad existente	Aprenden del ataque y mejoran la protección

Scripts para Seguridad, Mantenimiento y Gestión de Usuarios en Ubuntu Server *(estan subidos a una carpeta comprimida en github- javier granados)*

SEGURIDAD

1. Script de LOGS del sistema (logs.sh)

Este script guarda los principales logs del sistema para auditoría o resolución de problemas. Crea una carpeta donde almacena logs del sistema (journalctl), del kernel (dmesg) y de inicios de sesión (last), usando la fecha actual como parte del nombre.

```
#!/bin/bash
# Guarda logs importantes del sistema en un archivo con fecha

FECHA=$(date +%F)
DESTINO="/var/log/backup_logs"
mkdir -p "$DESTINO"
```

```
journalctl -xe > "$DESTINO/syslog_$FECHA.log"
dmesg > "$DESTINO/dmesg_$FECHA.log"
last > "$DESTINO/lastlog_$FECHA.log"

echo "Logs guardados en $DESTINO"
```

Uso: sudo bash logs.sh

2. Script de BACKUP de /etc y /home (backup.sh)

Este script realiza una copia de seguridad comprimida de los directorios más importantes del sistema: /etc (configuraciones) y /home (datos de usuarios). Guarda los archivos .tar.gz en /var/backups.

```
#!/bin/bash
# Backup comprimido de /etc y /home

DESTINO="/var/backups"
FECHA=$(date +%F)
mkdir -p "$DESTINO"

tar -czf "$DESTINO/etc_backup_$FECHA.tar.gz" /etc
tar -czf "$DESTINO/home_backup_$FECHA.tar.gz" /home

echo "Backups almacenados en $DESTINO"
```

Uso: sudo bash backup.sh

AUTOMATIZACIÓN Y MANTENIMIENTO

3. Script de mantenimiento y actualizaciones (mantenimiento.sh)

Este script actualiza todos los paquetes instalados, elimina los que ya no se usan y limpia el caché. Es útil para mantener el sistema estable y libre de archivos innecesarios.

```
#!/bin/bash
# Actualiza el sistema y limpia paquetes innecesarios
```

```
apt update && apt upgrade -y
apt autoremove -y
apt autoclean

echo "Sistema actualizado y limpio."
```

Uso: sudo bash mantenimiento.sh

4. Script de chequeo de disco y CPU (estado.sh)

Muestra información sobre el uso de la CPU, la memoria RAM y el espacio en disco. También muestra los usuarios conectados. Ideal para un control rápido del estado del servidor.

```
#!/bin/bash
# Muestra estado de CPU, memoria y disco

echo "=== USO DE CPU ==="
top -b -n1 | grep "Cpu(s)"

echo -e "\n=== MEMORIA ==="
free -h

echo -e "\n=== DISCO ==="
df -h

echo -e "\n=== USUARIOS CONECTADOS ==="
who
```

Uso: bash estado.sh

GESTIÓN DE USUARIOS Y PERMISOS

5. Script de creación de usuarios en lote (crear_usuarios.sh)

Este script crea automáticamente usuarios desde un archivo de texto. También les asigna una contraseña inicial.

Ejemplo del archivo usuarios.txt:

```
nginx
```

ana
pedro
lucas

Script:

```
#!/bin/bash
# Crea usuarios desde un archivo de texto

USUARIOS="usuarios.txt"
while read usuario; do
   useradd -m "$usuario" -s /bin/bash
   echo "$usuario:Contraseña123" | chpasswd
   echo "Usuario $usuario creado."
done < "$USUARIOS"
```

Uso: sudo bash crear_usuarios.sh

6. Script para asignar grupo y permisos (permisos.sh)

Este script crea un grupo llamado empresa, asigna los usuarios a ese grupo y cambia los permisos de sus carpetas personales para que solo el usuario y el grupo puedan acceder.

```
#!/bin/bash
# Asigna grupo "empresa" a todos los usuarios y cambia permisos

groupadd empresa

for user in $(cut -d: -f1 /etc/passwd | grep -E '^ana|^pedro|^lucas'); do
   usermod -aG empresa "$user"
   chown -R "$user:empresa" /home/$user
   chmod 750 /home/$user
   echo "Permisos aplicados a $user"
done
```

Uso: sudo bash permisos.sh

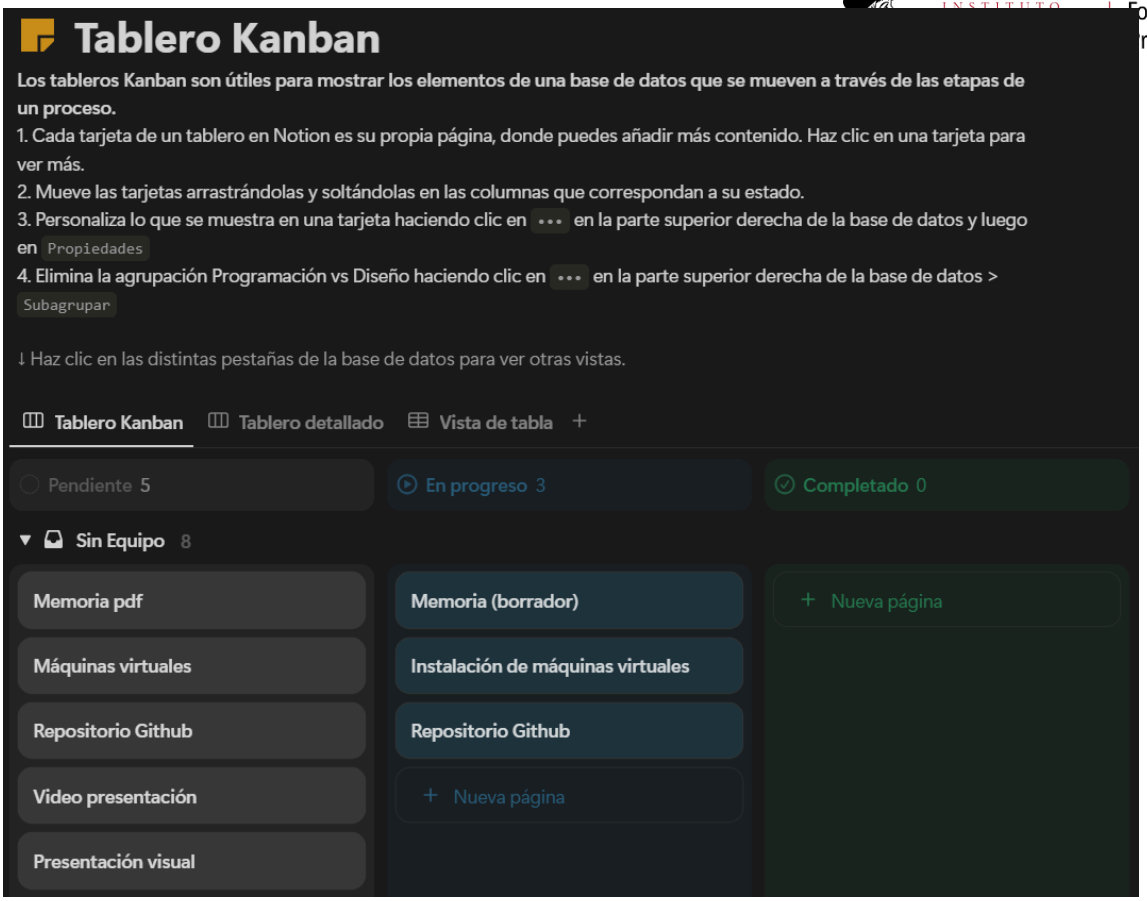
7. Script para desactivar usuarios inactivos (desactivar_inactivos.sh)

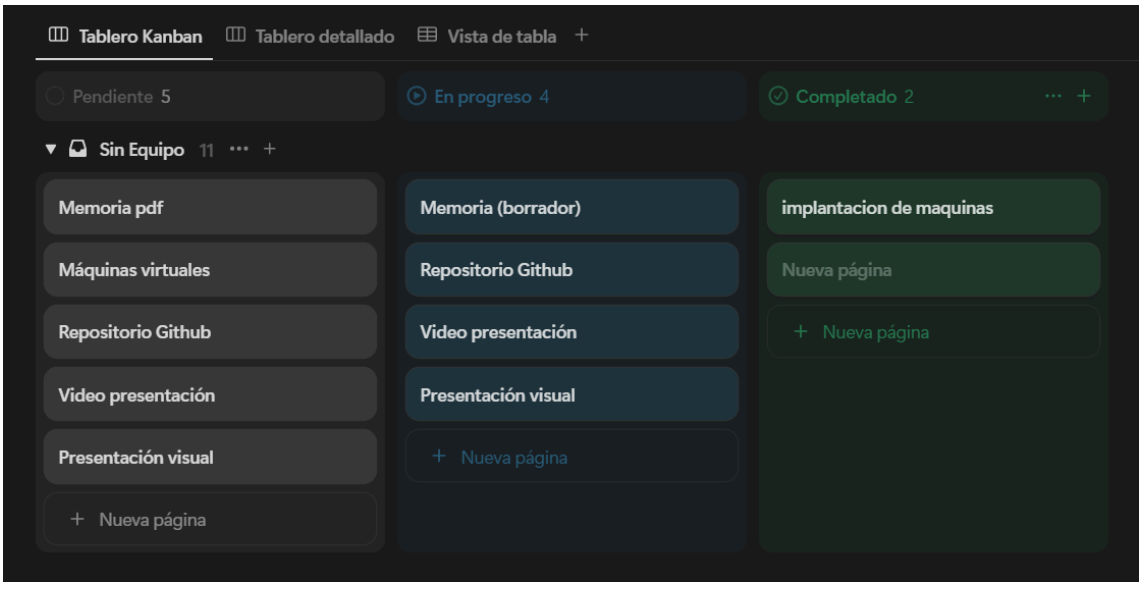
Este script revisa usuarios que no han iniciado sesión en más de 30 días y los bloquea usando usermod -L.

```
#!/bin/bash
# Desactiva usuarios inactivos por más de 30 días

dias=30
usuarios_inactivos=$(lastlog -b $dias | awk 'NR>1 && $NF!="**Never logged in**" {print $1}')

for u in $usuarios_inactivos; do
   usermod -L "$u"
   echo "Usuario $u bloqueado por inactividad"
done
```

Uso: sudo bash desactivar_inactivos.sh

Tablero KanBan (también subido a imágenes github) (primera sesión)





(segunda sesión)

5. CONCLUSIONES

El desarrollo de este proyecto nos ha permitido comprender en profundidad las ventajas y desafíos de implantar una infraestructura dual que combine sistemas operativos Windows y Linux dentro de una organización. A través de un análisis técnico y práctico, hemos demostrado que ambos entornos pueden coexistir de manera eficiente, aportando lo mejor de cada uno en función del perfil y las necesidades específicas de los usuarios.

En el caso de Windows, destacan su interfaz gráfica amigable, la compatibilidad con una amplia variedad de aplicaciones comerciales y la facilidad de administración mediante herramientas como las políticas de grupo. Estas características lo convierten en una opción especialmente adecuada para usuarios administrativos o de oficina, donde la productividad y la integración con software propietario son fundamentales.

Por su parte, Linux en distribuciones como Ubuntu, Debian o Rocky Linux ofrece un entorno altamente configurable, seguro y eficiente, con la ventaja añadida de no requerir licencias de pago. Esto lo convierte en una solución ideal para entornos de desarrollo, servidores o usuarios técnicos que necesitan un control total sobre el sistema.

La virtualización ha sido una herramienta clave en la fase de implementación y pruebas, ya que nos ha permitido simular escenarios reales, probar configuraciones, aplicar medidas de seguridad y automatizar tareas mediante scripts. Asimismo, el uso de tableros Kanban, ha favorecido una gestión eficiente del proyecto, promoviendo la colaboración, la responsabilidad compartida y la mejora continua.

En lo referente a seguridad y mantenimiento, ambos sistemas ofrecen soluciones sólidas, aunque con enfoques distintos. Mientras que Windows se apoya en mecanismos centralizados como Active Directory y antivirus integrados, Linux permite una configuración más granular del firewall, los permisos y las tareas programadas, lo que resulta especialmente útil en entornos que requieren un alto grado de personalización.

6. LÍNEAS DE INVESTIGACIÓN FUTURAS

(No son obligatorios, pero pueden aparecer)

7. BIBLIOGRAFÍA

Ubuntu Documentation. (2024). *Ubuntu Server Guide*. Canonical Ltd.
Disponible en: https://ubuntu.com/server/docs

Open Web Tutorials. (2023). *Bash Scripting Basics and Automation for Linux*.
Disponible en: https://ryanstutorials.net/bash-scripting-tutorial/

Linux Handbook. (2024). *UFW – Uncomplicated Firewall Tutorial*.
Disponible en: https://linuxhandbook.com/ufw-guide/

FreeCodeCamp. (2023). *Crontab – Automating Commands with Cron Jobs on Linux*.
Disponible en: https://www.freecodecamp.org/news/schedule-tasks-with-cron/

https://chatgpt.com

GitHub. (2024). *Repositorio del proyecto - Infraestructura Dual*
https://github.com/wheresmyicecream/Trabajo-ISO-

8. ANEXOS

9. OTROS PUNTOS

(No son obligatorios, pero pueden aparecer)

- Aportaciones personales
- Retos profesionales
- Restos personales
- Agradecimientos