**PORTADA**



**REDEX**

(Subtítulo, si así se decide)

**Nombre del alumno o de la alumna:**

ScrumMaster(Jesús Ignacio Alonso)

Luis Santiago Gil

David Bernaldo

Hugo Lopez

https://trello.com/b/6W2QDEnD

**Curso académico:**

1ª de ASIR

**Tutora/Tutor del proyecto:**

Carmelo

**ÍNDICE PAGINADO**

1. **Introducción y contexto**
2. **Análisis comparativo (Windows vs Linux)**
3. **Escenarios de uso recomendados**
4. **Instalación paso a paso**
5. **Configuración básica y avanzada**
6. **Seguridad (firewall, políticas, actualizaciones, antivirus, logs)**
7. **Automatización y scripts de mantenimiento**
8. **Gestión de usuarios y permisos**
9. **Documentación técnica**
10. **Conclusiones y propuesta final**
11. **Anexos: capturas, comandos usados, logs, configuración**
12. **JUSTIFICACIÓN DEL PROYECTO**
13. **INTRODUCCIÓN**

Enlace al Github: <https://github.com/ignacio-alonso22/redex>

Primer borrador "esquema" sobre la memoria final a entregar.

**Portada**

* Título del proyecto
* Nombre del módulo: Implantación de Sistemas Operativos – 1º ASIR
* Curso y grupo
* Integrantes del equipo
* Fecha

**1. Introducción y Contexto**

* Objetivo general del proyecto
* Necesidades de la empresa simulada
* Justificación del uso de Windows y Linux
* Metodología de trabajo elegida (Scrum + Kanban)

**2. Análisis Comparativo de Sistemas Operativos**

**2.1 Sistemas Evaluados**

* Windows 11
* Ubuntu, Debian, Rocky Linux

**2.2 Comparativa Técnica**

* Instalación y facilidad de uso
* Interfaz gráfica
* Herramientas administrativas
* Licencias y costes
* Seguridad integrada
* Mantenimiento y actualizaciones

**3. Escenarios de Uso Recomendados**

* Tipos de usuarios: Administrador, Usuario Estándar, Desarrollador
* Sistema operativo más adecuado para cada perfil
* Justificación técnica de las decisiones

**4. Instalación Paso a Paso**

**4.1 Preparación de máquinas virtuales**

* Elección de software (VirtualBox, VMware…)

**4.2 Instalación de sistemas operativos**

* Windows: capturas y comandos clave
* Linux: capturas, opciones de instalación, configuraciones iniciales

**5. Configuración Básica y Avanzada**

* Configuración de red, hostname, usuarios iniciales
* Instalación de software esencial
* Servicios (SSH, RDP, SAMBA, etc.)

**6. Seguridad del Sistema**

* Políticas de actualizaciones
* Configuración de firewall
* Gestión de permisos y usuarios
* Antivirus y logs

**7. Automatización y Mantenimiento**

* Tareas automatizadas con scripts (bash, PowerShell)
* Cron en Linux y tareas programadas en Windows
* Ejemplos de automatización: copia de seguridad, limpieza de logs

**8. Gestión de Usuarios y Permisos**

* Creación de perfiles
* Grupos de usuarios
* Restricciones de acceso
* Herramientas administrativas

**9. Documentación Técnica**

* Comandos utilizados
* Capturas de pantalla relevantes
* Errores frecuentes y soluciones

**10. Conclusiones y Propuesta Final**

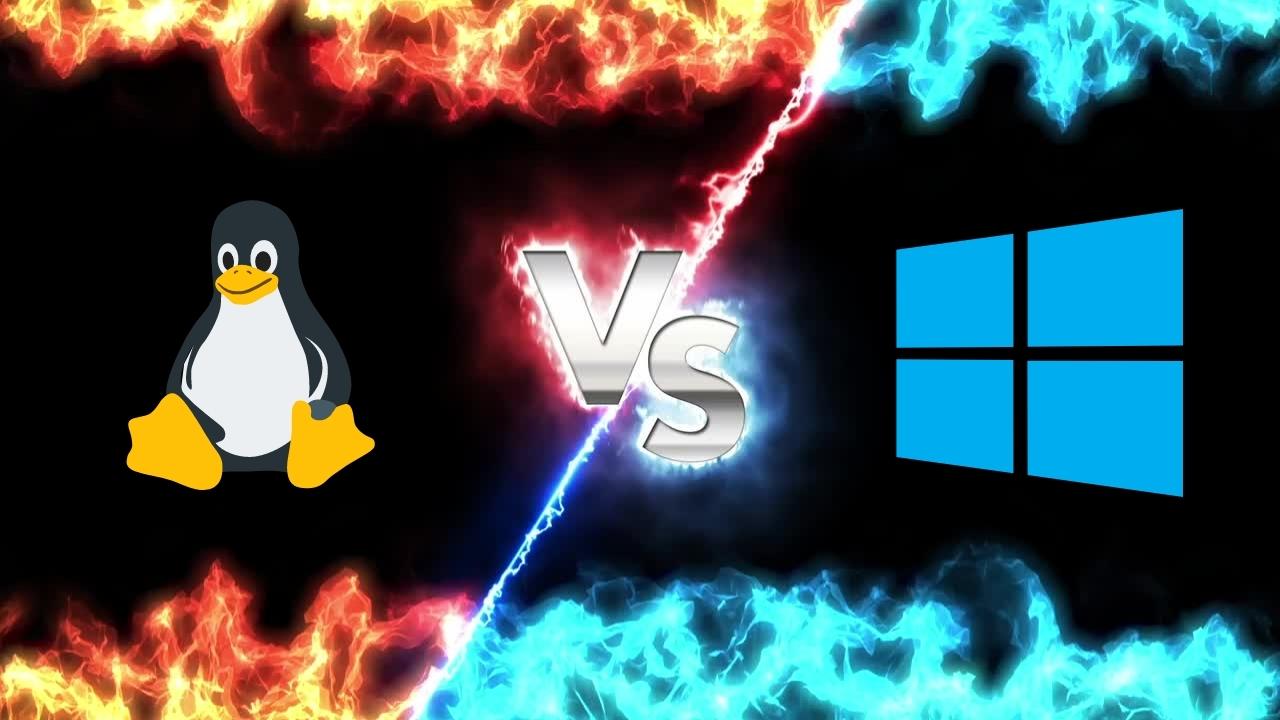
* Resumen de decisiones técnicas
* Propuesta de arquitectura definitiva
* Ventajas e inconvenientes de cada sistema
* Lecciones aprendidas y experiencia del equipo

**11. Anexos**

* Capturas completas de instalación y configuración
* Logs de sistemas
* Scripts completos
* Actas de reuniones
* Tareas completadas en Trello y evidencia en GitHub

1. **OBJETIVOS**
   1. OBJETIVO GENERAL
   2. OBJETIVOS ESPECÍFICOS

Punto 2

**Estudio Comparativo: Windows vs Linux**

**Introducción** El sistema operativo (SO) es un componente esencial del software de un ordenador, ya que gestiona tanto el hardware como otros programas. Windows y Linux son dos de los sistemas operativos más populares, utilizados en una amplia variedad de dispositivos. Este estudio tiene como objetivo comparar ambos sistemas en varios aspectos clave, como la interfaz de usuario, el rendimiento, la seguridad, la compatibilidad, el coste y el uso en distintos entornos.



* + 1. **Historia y desarrollo**

Windows fue desarrollado por Microsoft y lanzado por primera vez en 1985 como una interfaz gráfica para MS-DOS. Desde entonces, ha evolucionado hasta convertirse en un sistema operativo completo, con versiones ampliamente utilizadas como Windows XP, 7, 10 y 11.

Linux, por otro lado, fue creado en 1991 por Linus Torvalds como un sistema operativo de tipo UNIX de código abierto. Su desarrollo está descentralizado y es mantenido por una comunidad global. Existen muchas distribuciones ("distros") de Linux, como Ubuntu, Fedora, Debian o CentOS, adaptadas a diferentes necesidades.



1. **Interfaz de usuario**

Windows ofrece una interfaz gráfica muy estándar y fácil de usar, con un menú de inicio, iconos de escritorio y ventanas. Es coherente en todas sus versiones y está diseñada para usuarios sin conocimientos técnicos.

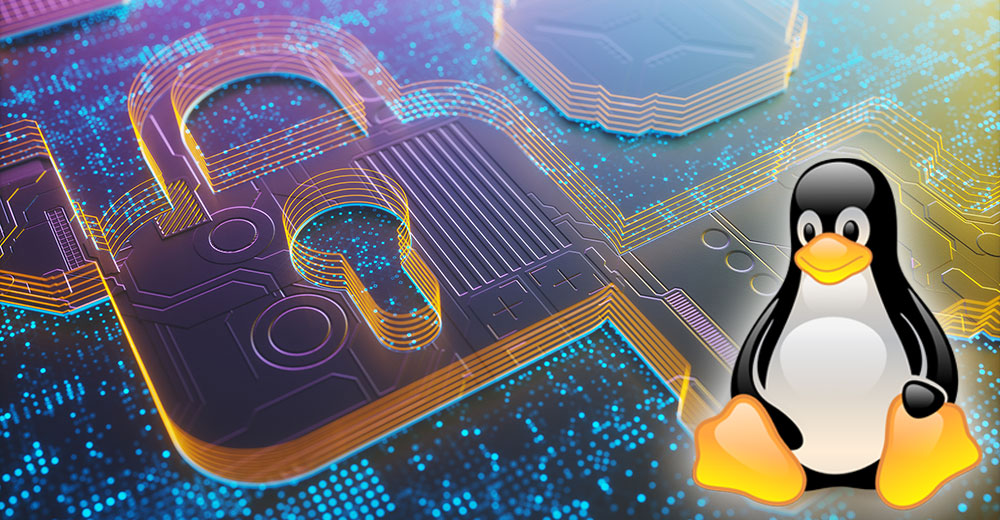
Linux, dependiendo de la distribución y el entorno de escritorio (GNOME, KDE, XFCE), puede tener interfaces muy variadas. Esto ofrece una gran personalización, pero puede resultar más complejo para usuarios nuevos. Sin embargo, distros como Ubuntu ofrecen una experiencia muy accesible.



1. **Rendimiento y eficiencia**

Linux es conocido por su eficiencia en el uso de recursos. Puede funcionar en hardware antiguo o de bajas prestaciones, lo que lo hace ideal para servidores o sistemas embebidos.

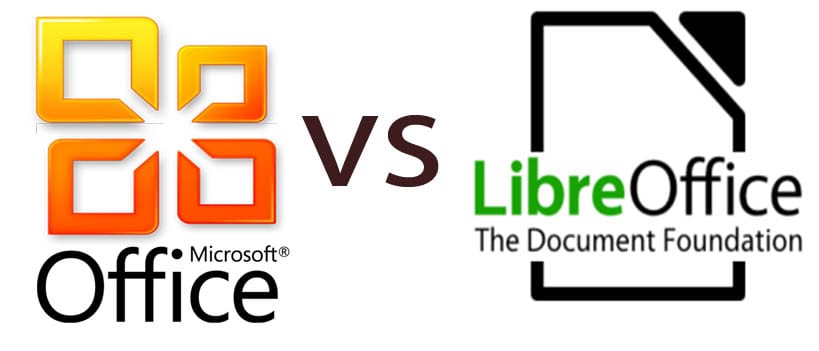
Windows, aunque ha mejorado en eficiencia, requiere generalmente hardware más potente para ofrecer un buen rendimiento. Sin embargo, para tareas gráficas o de ofimática intensiva, ofrece buena compatibilidad con software optimizado.



1. **Seguridad**

Linux es considerado más seguro por diseño. Al ser de código abierto, las vulnerabilidades pueden ser detectadas y corregidas rápidamente por la comunidad. Además, los permisos de usuario están más restringidos por defecto.

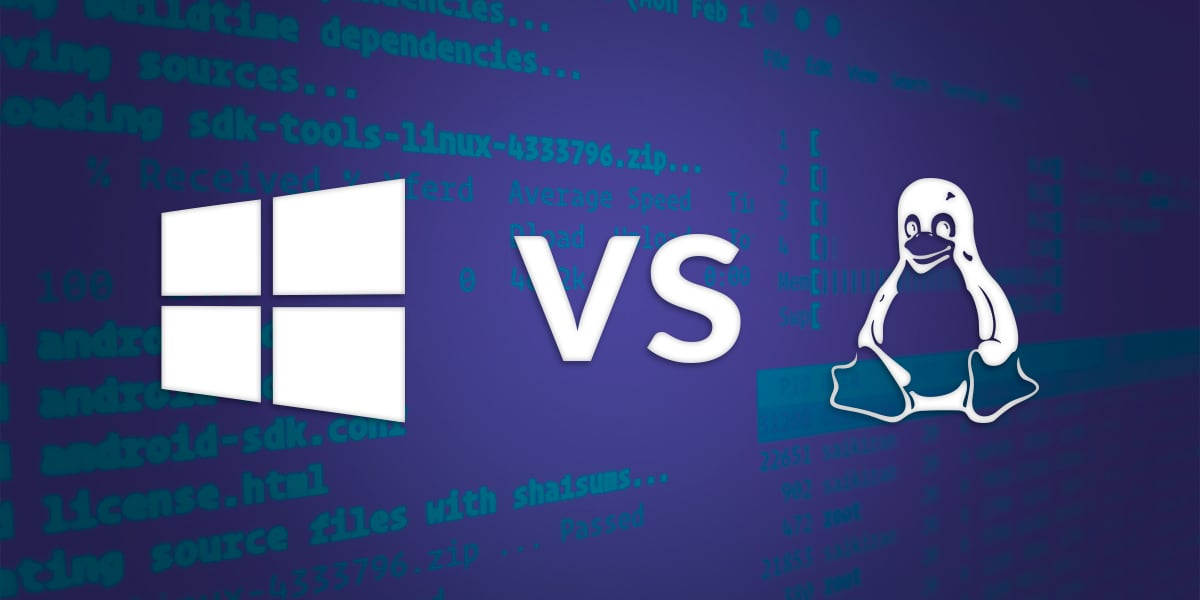
Windows, al ser el sistema más utilizado a nivel mundial, es también el más atacado. Aunque Microsoft ha hecho grandes avances en seguridad, sigue siendo objetivo de malware y virus. Es común el uso de antivirus en Windows.



1. **Compatibilidad y software disponible**

Windows tiene una mayor compatibilidad con software comercial, especialmente en videojuegos, programas de diseño y aplicaciones empresariales como Microsoft Office.

Linux ha mejorado mucho en este aspecto. Existen alternativas libres (LibreOffice, GIMP, etc.) y mediante herramientas como Wine o máquinas virtuales, se pueden ejecutar programas de Windows. No obstante, algunos programas específicos no están disponibles.



1. **Coste y licencias**

Linux es gratuito y de código abierto. Cualquier usuario puede descargar, modificar y distribuir una distro sin coste alguno. Esto lo hace atractivo para usuarios particulares, desarrolladores y empresas.

Windows es un sistema propietario y requiere licencia. Aunque muchas veces viene preinstalado en ordenadores nuevos, su uso en entornos empresariales o educativos implica un coste por licencia.



1. **Uso en distintos entornos**

En el ámbito empresarial y de escritorio, Windows domina por su facilidad de uso y compatibilidad. En servidores, supercomputadoras y dispositivos IoT, Linux es el preferido por su estabilidad, seguridad y bajo consumo de recursos.



**Conclusión**

Tanto Windows como Linux tienen ventajas y desventajas. La elección depende del contexto y necesidades del usuario. Para usuarios que buscan simplicidad y compatibilidad, Windows es una buena opción. Para quienes valoran la seguridad, el control y la personalización, Linux ofrece un entorno ideal.

Ambos sistemas están en constante evolución y es posible que en el futuro las diferencias se reduzcan aún más, promoviendo la interoperabilidad y la libertad de elección.

**6. Seguridad del Sistema**

La seguridad es un aspecto esencial en cualquier infraestructura de sistemas operativos. En este apartado analizamos y aplicamos medidas de seguridad en ambos entornos, Windows y Linux, adaptadas a los diferentes perfiles de usuario y al entorno virtualizado propuesto.

**6.1. Políticas de Actualizaciones**

**En Linux**

* Uso de herramientas como apt, dnf, unattended-upgrades.
* Configuración de actualizaciones automáticas o programadas.
* Verificación de repositorios seguros.

**En Windows**

* Configuración de **Windows Update** desde el Panel de control o PowerShell.
* Uso de directivas de grupo (GPO) para definir:
  + Horario de instalación
  + Qué tipo de actualizaciones aplicar
  + Control de reinicios automáticos

**6.2. Configuración del Cortafuegos**

**En Linux**

* Uso de ufw, firewalld o directamente iptables.
* Definición de reglas básicas:
  + Permitir SSH solo desde ciertas IPs
  + Denegar puertos innecesarios
  + Habilitar puertos específicos para servicios críticos

**En Windows**

* Uso del Firewall de Windows con configuración avanzada.
* Creación de reglas de entrada y salida para servicios concretos (RDP, SMB, etc.).
* Asociación de perfiles de red (pública/privada) según tipo de interfaz.

**6.3. Gestión de Permisos y Cuentas**

**En Linux**

* Uso de comandos chmod, chown, groups.
* Separación de privilegios entre usuarios y root.
* Configuración de sudo para controlar qué usuarios pueden ejecutar comandos administrativos.
* Control de acceso a archivos sensibles (/etc/passwd, /etc/shadow, logs, etc.).

**En Windows**

* Creación y gestión de cuentas mediante MMC (Usuarios y grupos locales) o net user.
* Uso de ACLs (Listas de Control de Acceso) para definir permisos sobre carpetas y archivos.
* Configuración de derechos de usuario en políticas locales.

**6.4. Antivirus y Protección contra Malware**

**En Linux**

* Instalación y configuración de herramientas como:
  + ClamAV
  + rkhunter (para rootkits)
  + chkrootkit
* Análisis programado de directorios críticos y logs.

**En Windows**

* Configuración de **Microsoft Defender**.
* Escaneos automáticos y programados.
* Activación de protección en tiempo real y en red.
* Integración con SmartScreen y control de aplicaciones.

**6.5. Gestión de Logs y Auditoría**

**En Linux**

* Revisión y protección de:
  + /var/log/auth.log
  + /var/log/syslog
* Uso de logrotate para la gestión automática del tamaño y retención de logs.
* Configuración básica de auditd para seguimiento de cambios importantes.

**En Windows**

* Análisis del Visor de Eventos (eventvwr).
* Activación de auditoría en políticas locales:
  + Inicio de sesión
  + Acceso a archivos
  + Cambios en cuentas
* Uso de herramientas como gpedit.msc para configurar políticas de registro.

**6.6. Buenas Prácticas Adicionales**

* **Cambio de puertos por defecto** en servicios críticos (ej. SSH en Linux).
* **Bloqueo tras intentos fallidos** con fail2ban (Linux) o directiva de cuenta (Windows).
* Uso de contraseñas seguras o autenticación de dos factores cuando sea posible.
* Desactivación de servicios innecesarios.

**Evidencias a Incluir en la Memoria**

* Capturas de reglas de firewall, logs, antivirus y permisos aplicados.
* Fragmentos de configuración (ufw, GPO, sudoers, etc.).
* Logs de ejecución de antivirus y escaneos.
* Resultados de pruebas de acceso entre máquinas.
* Scripts de automatización de seguridad (opcional).

**6. Seguridad y Ciberseguridad**

En este proyecto hemos trabajado con la premisa de que cualquier sistema operativo, independientemente de su robustez, puede ser vulnerable si no se configura ni mantiene adecuadamente. Por ello, además de aplicar buenas prácticas de seguridad básicas, realizamos un análisis de riesgos y simulamos una estrategia tipo **Red Team vs Blue Team** para detectar posibles vulnerabilidades y aplicar medidas correctivas.

**6.1. Red Team vs Blue Team: Enfoque en el Proyecto**

* **Red Team (equipo atacante)**: simula ataques y explora posibles debilidades de las configuraciones, como puertos abiertos, servicios innecesarios activos o contraseñas débiles.
* **Blue Team (equipo defensor)**: aplica medidas de protección, monitoreo y respuesta, fortaleciendo el sistema contra ataques detectados o simulados.

En nuestro caso, el enfoque Red Team lo usamos para revisar configuraciones desde una visión ofensiva, mientras que el Blue Team se centró en reforzarlas y monitorizar los sistemas.

**6.2. Posibles Vulnerabilidades Detectadas**

Durante la implantación inicial de los entornos Windows y Linux, detectamos las siguientes vulnerabilidades potenciales:

| **Vulnerabilidad** | **Sistema** | **Impacto Potencial** | **Detectado por** | **Medida Correctiva (Blue Team)** |
| --- | --- | --- | --- | --- |
| Puertos abiertos innecesarios (ej. FTP, Telnet) | Ambos | Medio | Red Team | Bloqueo en firewall (ufw, Firewall de Windows) |
| Servicio SSH sin restricción de IP ni autenticación | Linux | Alto | Red Team | Limitar acceso por IP y usar claves públicas |
| Contraseñas por defecto o débiles | Ambos | Alto | Red Team | Políticas de contraseñas seguras |
| Logs accesibles por usuarios no privilegiados | Linux | Medio | Red Team | Cambios de permisos (chmod, chown) en /var/log |
| Actualizaciones deshabilitadas o manuales | Windows | Alto | Red Team | Activación forzosa de Windows Update con GPO |
| Falta de antivirus activo en Linux | Linux | Medio | Red Team | Instalación de ClamAV + escaneos programados |
| Permisos excesivos a usuarios estándar | Windows | Alto | Red Team | Revisión de permisos y uso de GPO |
| Tareas programadas sin control de privilegios | Ambos | Medio | Red Team | Validación de scripts y sus permisos de ejecución |

**6.3. Herramientas utilizadas para la auditoría y defensa**

**Red Team**

* nmap: escaneo de puertos y detección de servicios abiertos.
* hydra: intentos de fuerza bruta en servicios expuestos.
* netcat, telnet: comprobación de conexiones abiertas y servicios obsoletos.

**Blue Team**

* ufw y iptables para el control de tráfico entrante/saliente.
* fail2ban para detección de intentos de login fallidos y bloqueo automático.
* auditd en Linux y Visor de Eventos en Windows para registrar acciones sospechosas.
* ClamAV, rkhunter, Microsoft Defender como soluciones antimalware.

**6.4. Buenas prácticas aplicadas como resultado**

* Cierre de todos los puertos no utilizados.
* Autenticación reforzada en servicios SSH y RDP.
* Políticas estrictas de contraseñas (longitud mínima, expiración).
* Actualizaciones automatizadas habilitadas.
* Revisión periódica de logs y alertas.
* Reducción de la superficie de ataque eliminando software innecesario.
* Monitorización continua de integridad del sistema y accesos sospechosos.

**6.5. Conclusión**

La ciberseguridad no debe ser tratada como una fase única del proyecto, sino como un proceso continuo. Gracias a la aplicación de una estrategia básica de Red Team vs Blue Team, hemos podido mejorar sustancialmente la postura de seguridad inicial, reduciendo los riesgos de accesos no autorizados, fuga de datos o compromisos del sistema.

En un entorno real, esto debería complementarse con auditorías externas, pentesting profesional y formación constante del personal.

**5. CONCLUSIONES**

**6. LÍNEAS DE INVESTIGACIÓN FUTURAS**

**(No son obligatorios, pero pueden aparecer)**

**7. BIBLIOGRAFÍA**

**8. ANEXOS**

**9. OTROS PUNTOS**

**(No son obligatorios, pero pueden aparecer)**

* Aportaciones personales
* Retos profesionales
* Restos personales
* Agradecimientos