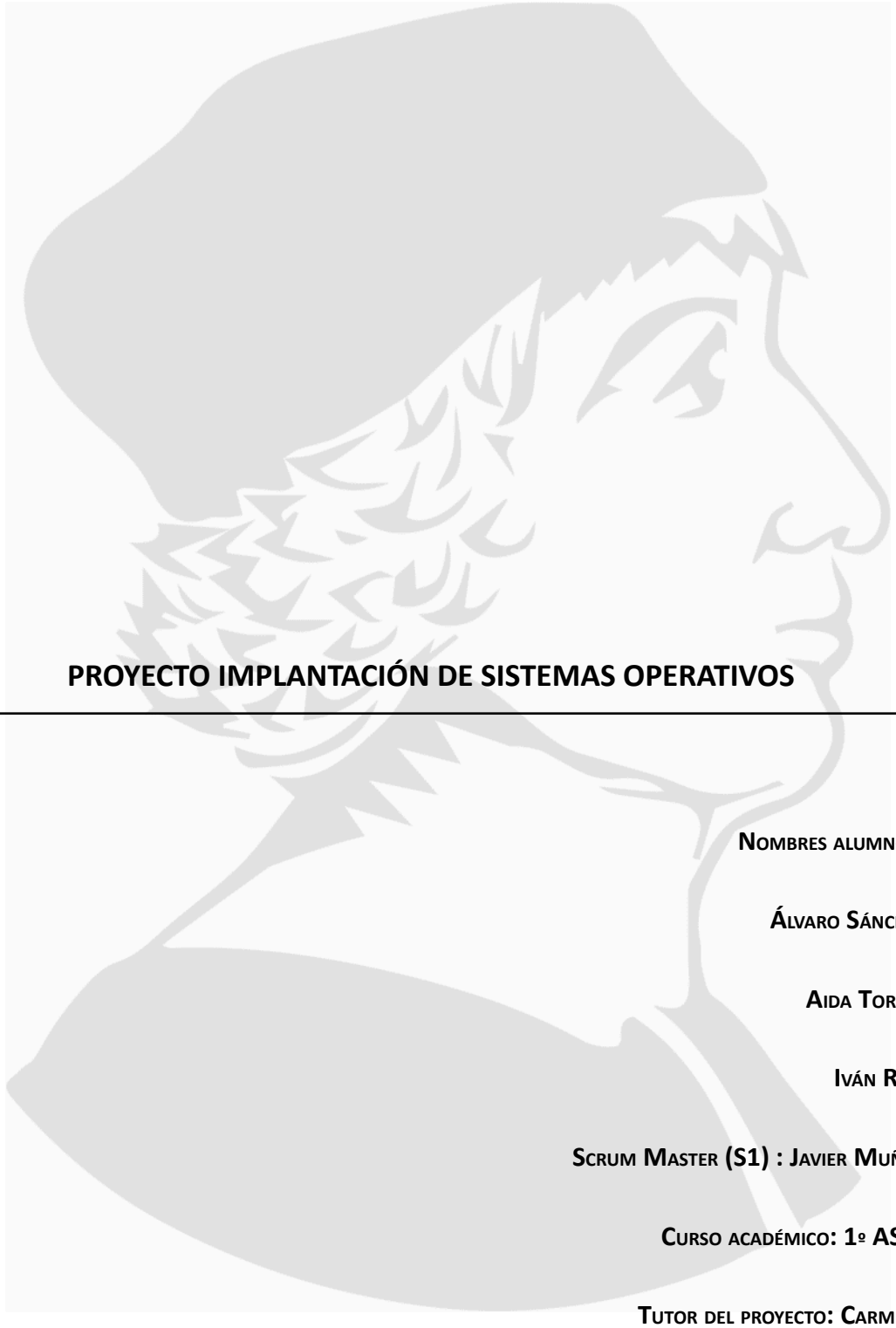
INSTITUTO NEBRIJA | Formación Profesional

TRABAJO FINAL ISO



PROYECTO IMPLANTACIÓN DE SISTEMAS OPERATIVOS

Nombres alumnos:

Álvaro Sánchez

Aida Torres

Iván Ruiz

Scrum Master (S1) : Javier Muñoz

Curso académico: 1º ASIR

Tutor del proyecto: Carmelo

ÍNDICE PAGINADO

¿Quieres que te lo deje en formato Word o PDF también?

1. JUSTIFICACIÓN DEL PROYECTO

SOBRE EL PROYECTO SE PLANTEA UNA SOLUCIÓN TECNOLÓGICA PARA UNA PEQUEÑA EMPRESA EN CRECIMIENTO QUE NECESITA UNA INFRAESTRUCTURA INFORMÁTICA FIABLE, SEGURA Y ADAPTADA A DIFERENTES PERFILES DE USUARIOS.

MUCHAS ORGANIZACIONES REQUIEREN ENTORNOS MIXTOS CON EL USO DE DIFERENTES SISTEMAS OPERATIVOS PROPIETARIOS. WINDOWS EN UN USO COMÚN O SISTEMAS DE CÓDIGO ABIERTO, COMO LINUX. ESTA COMBINACIÓN PERMITE APROVECHAR LO MEJOR DE AMBOS SISTEMAS: COMPATIBILIDAD Y FACILIDAD DE USO DE WINDOWS JUNTO CON LA ESTABILIDAD, SEGURIDAD Y BAJO COSTE DE LAS DISTRIBUCIONES LINUX.

EL OBJETIVO PRINCIPAL DEL PROYECTO ES DISEÑAR, COMPARAR Y DESPLEGAR AMBOS ENTORNOS (WINDOWS Y LINUX) EN UN ENTORNO VIRTUALIZADO QUE UTILICE UNA EMPRESA REAL. A TRAVÉS DEL ANÁLISIS TÉCNICO, LA INSTALACIÓN, CONFIGURACIÓN, AUTOMATIZACIÓN Y DOCUMENTACIÓN DE TAREAS, SE BUSCA OFRECER UNA PROPUESTA DE INFRAESTRUCTURA EQUILIBRADA QUE RESPONDA A LAS NECESIDADES DE DISTINTOS PERFILES: USUARIOS ESTÁNDAR, DESARROLLADORES, Y PERSONAL ADMINISTRATIVO O TÉCNICO.

SE TRABAJARÁ UTILIZANDO METODOLOGÍAS ÁGILES COMO **SCRUM** Y HERRAMIENTAS DE GESTIÓN COMO **KANBAN**, FOMENTANDO LA ORGANIZACIÓN, LA COMUNICACIÓN Y LA COLABORACIÓN DENTRO DEL EQUIPO. TODOS LOS AVANCES SERÁN REGISTRADOS EN UN REPOSITORIO GITHUB, DONDE QUEDARÁN REFLEJADAS LAS TAREAS, SCRIPTS, CONFIGURACIONES Y EVIDENCIAS DEL TRABAJO REALIZADO POR CADA UNO DE LOS MIEMBROS COMPONENTES DEL PROYECTO.

INTEGRAMOS DIFERENTES COMPETENCIAS DEL MÓDULO DE IMPLANTACIÓN DE SISTEMAS OPERATIVOS DEL CICLO FORMATIVO DE PRIMERO DE ASIR.

Primer Borrador del Proyecto final

1. Introducción y contexto

   Justificación de la necesidad: una pequeña empresa en crecimiento necesita sistemas estables, seguros y adaptados a varios perfiles (administrador, usuario estándar, desarrollador). Propuesta: Implantación dual de sistemas Windows y Linux virtualizados.

2. Análisis comparativo

Comparación técnica de: Instalación Interfaz y usabilidad Licencias y costes Herramientas administrativas Seguridad y mantenimiento Distribuciones evaluadas: Windows 11, Ubuntu, Debian, Rocky Linux.

3. Escenarios de uso recomendados

   Windows 11 para usuarios con software específico de oficina o empresarial. Linux (Ubuntu o Debian) para desarrolladores o usuarios generales con necesidades de rendimiento y estabilidad.

4. Instalación paso a paso

Capturas y documentación del proceso de instalación en VirtualBox. Diferencias en la instalación de paquetes, usuarios, drivers, etc.

5. Configuración básica y avanzada

Configuración de red (ping, SSH, RDP). Usuarios, grupos, servicios. Automatización con scripts (Bash, PowerShell) y tareas programadas.

6. Seguridad (detalle más abajo)
7. Automatización y scripts de mantenimiento

Copias de seguridad automatizadas. Limpieza de archivos temporales. Actualizaciones automáticas en ambos sistemas.

8. Gestión de usuarios y permisos

Control de accesos por rol. Grupos y políticas de seguridad en Windows. chmod, chown, sudoers en Linux.

9. Documentación técnica

Todos los pasos registrados con capturas. Anexos: comandos utilizados, logs, archivos de configuración.

10. Conclusiones y propuesta final

Análisis de qué sistema es más adecuado según cada perfil.

3. OBJETIVOS

OBJETIVO GENERAL

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

- Diseñar e implantar ambos sistemas en entornos virtualizados.

- Comparar características clave: instalación, licencias, interfaz, herramientas administrativas, seguridad y mantenimiento.

- Automatizar tareas básicas de mantenimiento (scripts, políticas de grupo, cron, tareas programadas).

- Proponer una arquitectura óptima según el tipo de usuario (administrador, usuario estándar, desarrollador, etc.).

- Gestionar el proyecto utilizando metodología ágil (Scrum + Kanban).

1. FUNDAMENTACIÓN TEÓRICA: SEGURIDAD EN LA INFRAESTRUCTURA DUAL WINDOWS / LINUX

La seguridad es un pilar fundamental en la implantación de cualquier infraestructura de sistemas operativos. En este proyecto, se han aplicado distintas medidas de seguridad adaptadas a cada sistema (Windows y Linux), garantizando la protección de los equipos, usuarios, datos y servicios, tanto a nivel preventivo como reactivo. A continuación, se detallan las acciones y configuraciones realizadas:

---

1. Firewall y control de tráfico

Windows:

- Configuración de Windows Defender Firewall para permitir únicamente tráfico entrante en puertos autorizados (por ejemplo, RDP para administración remota).

- Creación de reglas personalizadas para bloquear conexiones sospechosas.

- Habilitación del registro de eventos del firewall para auditoría.

Linux:

- Uso de ufw (Uncomplicated Firewall) como interfaz de configuración.

- Reglas aplicadas:

  CopiarEditar

  2. `sudo ufw default deny incoming`
  3. `sudo ufw allow ssh`
  4. `sudo ufw allow from 192.168.1.0/24 to any port 22 proto tcp`
  5. `sudo ufw enable`

- Verificación:
  CopiarEditar

  6. `sudo ufw status verbose`

---

2. Gestión de permisos y privilegios

Windows:

- Creación de usuarios separados por roles: administrador, técnico, usuario estándar.

- Aplicación de directivas de seguridad mediante Políticas de Grupo (GPO):
    - Restricción del acceso a configuraciones del sistema.
    - Desactivación del panel de control y la línea de comandos para usuarios no autorizados.
    - Prohibición de uso de dispositivos USB no autorizados.

Linux:

- Control de accesos mediante grupos (`admin`, `sudo`, `users`).
- Edición del archivo `/etc/sudoers` con `visudo` para restringir comandos administrativos.
- Aplicación de permisos con `chmod` y `chown` para proteger archivos críticos:

CopiarEditar

7. `sudo chmod 700 /root`
8. `sudo chown root:root /etc/shadow`

Windows:

- Activación de Windows Update con instalación automática.

- Configuración de política para reinicio automático fuera del horario laboral.

Linux:

- Configuración de actualizaciones automáticas con unattended-upgrades:

  CopiarEditar

  ```
  sudo apt install unattended-upgrades
  sudo dpkg-reconfigure --priority=low unattended-upgrades
  ```

- Automatización mediante cron para notificaciones de parches pendientes.

---

4. Protección antimalware

Windows:

- Uso de Windows Defender Antivirus con protección en tiempo real.

- Configuración de análisis programados semanales y alertas en caso de infección.

Linux:

- Instalación de ClamAV para análisis bajo demanda de archivos descargados o dispositivos externos:

  CopiarEditar

  ```
  sudo apt install clamav
  sudo freshclam
  sudo clamscan -r /home
  ```

- Escaneo periódico mediante cron y envío de logs por correo interno del sistema.

---

5. Cifrado de datos y discos

Windows:

- Activación de BitLocker en unidades sensibles, con cifrado AES.

- Uso de TPM (Trusted Platform Module) para asegurar la integridad del arranque.

Linux:

- Cifrado de particiones de backup y datos sensibles con LUKS:

  CopiarEditar

  14. sudo cryptsetup luksFormat /dev/sdX

  15. sudo cryptsetup luksOpen /dev/sdX secure_data

- Montaje automático seguro con clave cifrada o passphrase.

---

6. Registros y auditoría de seguridad

Windows:

- Activación del Auditor de Seguridad para registrar:
  - Intentos de inicio de sesión fallidos.
  - Cambios en políticas del sistema.
  - Instalación de software.

- Visualización mediante el Visor de Eventos (eventvwr.msc).

Linux:

- Monitoreo de archivos de log críticos:
  /var/log/auth.log, /var/log/syslog, /var/log/fail2ban.log

- Instalación de herramientas como logwatch o logrotate para reportes automáticos y gestión de tamaño de logs.

- Configuración de fail2ban para bloquear IPs tras múltiples intentos fallidos de acceso SSH.

---

7. Medidas adicionales

- Desactivación de servicios innecesarios para reducir superficie de ataque.

- Restricción de puertos abiertos solo a los mínimos necesarios.

- Pruebas básicas de escaneo de vulnerabilidades con herramientas como:

  - nmap

- Lynis
- Microsoft Baseline Security Analyzer (MBSA)

- Concienciación básica al usuario:
  - Políticas de contraseñas seguras.
  - Bloqueo automático de sesión tras inactividad.

16. Materiales y métodos: estrategias de búsqueda, metodología y técnicas utilizadas
17. Resultados y análisis

Enlace repositorio GitHub:

- https://github.com/VictoriaFedericaPatinPiti/iso_proyecto_final

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