



UNIVERSIDAD TECNOLÓGICA DE PUEBLA

Ingeniería en redes Inteligentes y ciberseguridad

Materia: Dirección de Proyectos II

Profesor: NOLASCO HERNANDEZ JAVIER

Alumnos:

García Martínez Luis Daniel

Cortes Alcalá Jessica Banelly

De la Rosa Vázquez Marco Antonio

Rodríguez López Ricardo

Contenido

<i>Introducción:</i>	3
<i>1. Lista de entregables desarrollados.....</i>	4
<i>2. Descripción de cada entregable.</i>	5
<i>3. Informe de Desempeño del Proyecto</i>	7
<i>4. Reporte de actividades de aseguramiento de la calidad.....</i>	9
<i>5. Evaluaciones de desempeño del equipo de trabajo.</i>	12
<i>6. Análisis y evaluación de nuevos riesgos.</i>	13
<i>7. Registro de incidentes del proyecto.....</i>	16
<i>8. Registro de cambios implementados.</i>	16
<i>9. Minutas de reunión con interesados.</i>	18
<i>Ejemplo de Plantilla para minuta de reunión utilizada en nuestro proyecto.....</i>	19
<i>Conclusión:</i>	21

Introducción:

El presente reporte describe el desarrollo y ejecución de un proyecto de seguridad informática titulado Sistema de Detección de Intrusos (IDS). Este proyecto fue elaborado como parte de la asignatura Dirección de Proyectos I, impartida en el marco de la formación profesional en Tecnologías de la Información. Su objetivo principal es el desarrollo de una herramienta digital que permita identificar intrusiones y amenazas en una red de manera sencilla, accesible y efectiva, ayudando a proteger los recursos informáticos de una organización.

El IDS fue diseñado utilizando tecnologías modernas y de código abierto como Python, Flask, Scapy, PyUSB y HTML/CSS/JS. El sistema integra varios módulos que permiten el monitoreo del tráfico de red, escaneo de puertos, verificación de dispositivos USB conectados y análisis de archivos sospechosos. Cada uno de estos componentes se encuentra conectado a través de una interfaz web intuitiva que facilita su uso por parte de administradores o personal encargado de la seguridad informática.

Este proyecto no solo aborda necesidades técnicas específicas de ciberseguridad, sino que también pone en práctica habilidades de dirección de proyectos como planificación, organización de tareas, gestión de riesgos, control de calidad y trabajo colaborativo.

Fue planeado para desarrollarse a lo largo de tres cuatrimestres, siguiendo una metodología en cascada con fases claras de análisis de requerimientos, diseño del sistema, implementación, pruebas, despliegue y mantenimiento. Esta documentación recoge el trabajo realizado durante la primera fase del proyecto y sienta las bases para su continuidad y mejoras futuras.

A continuación, se describen los entregables generados, las actividades de aseguramiento de calidad, el desempeño del equipo y la evaluación de riesgos e incidentes presentados durante este ciclo.

1. Lista de entregables desarrollados.

N. o	Entregable	Descripción breve
1	Documento de requerimientos y análisis de riesgos	Define los objetivos del sistema y evalúa posibles amenazas y riesgos.
2	Diseño de la arquitectura del sistema IDS	Estructura técnica del sistema, componentes y su interacción.
3	Desarrollo del módulo de monitoreo de red	Permite la captura y análisis del tráfico en tiempo real.
4	Desarrollo del módulo de escaneo de puertos	Detecta servicios abiertos y posibles vectores de ataque.
5	Desarrollo del módulo de verificación de USB	Supervisa el uso de dispositivos USB conectados al sistema.
6	Desarrollo del módulo de análisis de archivos	Evalúa archivos sospechosos para detectar posibles amenazas.
7	Interfaz web con navegación entre funciones	Plataforma amigable para gestionar y visualizar las funcionalidades.
8	Pruebas funcionales y corrección de errores	Verificación del correcto funcionamiento y ajuste de fallos detectados.
9	Documentación técnica del proyecto	Manuales, instrucciones y detalles técnicos para instalación y uso.

2. Descripción de cada entregable.

Nº	Entregable	Descripción
1	Documento de requerimientos y análisis de riesgos	Define los objetivos del sistema, especificaciones funcionales (como monitoreo de red, escaneo de puertos, verificación de USB, etc.) y no funcionales (seguridad, facilidad de uso, portabilidad). También contempla las restricciones tecnológicas (por ejemplo, debe usarse Python y Flask) y los criterios de aceptación. Sirvió como base para la planeación y la validación del alcance del proyecto.
2	Diseño de la arquitectura del sistema IDS	Presenta la estructura del sistema, los módulos desarrollados, su interacción y el flujo de datos entre ellos. Este diseño fue clave para mantener una codificación organizada. Además, se definió la jerarquía de carpetas, archivos estáticos y templates, así como el comportamiento de cada ruta o función del sistema.
3	Módulo de monitoreo de red	Implementado con la biblioteca Scapy captura y analiza el tráfico de red en tiempo real para detectar comportamientos sospechosos o anómalos. Se integraron alertas que indican IP de origen, IP de destino, tipo de ataque.
4	Módulo de escaneo de puertos	Desarrollado usando Nmap. Identifica puertos abiertos y servicios activos en la red, evaluando posibles vulnerabilidades. Se agregaron validaciones para evitar errores de sintaxis en IPs y mostrar advertencias en caso de fallos en la conexión.
5	Módulo de verificación de USB	Este componente usa PyUSB. Supervisa dispositivos USB conectados, detectando accesos no autorizados o potencialmente peligrosos. Compara sus IDs con una lista segura almacenada localmente. Si el USB no es reconocido, genera una alerta indicando que puede tratarse de un dispositivo no autorizado o malicioso.

6	Módulo de análisis de archivos	Examina archivos en busca de firmas o patrones sospechosos que puedan indicar malware o amenazas. Se realizaron pruebas con archivos de texto, ejecutables y archivos de imagen con resultados positivos. El análisis incluye estructura interna del archivo. Se muestran mensajes como “archivo seguro” o “posible amenaza”.
7	Interfaz web con navegación entre funciones	Desarrollada con HTML5 y CSS usando Flask como backend. Permite a los usuarios acceder y utilizar los módulos a través de una interfaz amigable y funcional.
8	Pruebas funcionales y corrección de errores	Verifica el correcto funcionamiento de los módulos, detecta fallos y documenta las correcciones aplicadas.
9	Documentación técnica del proyecto	Incluye manuales de instalación, uso, mantenimiento y referencias técnicas para facilitar futuras mejoras.

3. Informe de Desempeño del Proyecto

3.1 Cumplimiento de Objetivos

Se cumplieron todos los objetivos definidos en la etapa de planificación. El sistema IDS fue desarrollado de forma modular, integrando funcionalidades clave como monitoreo de red, escaneo de puertos, control de dispositivos USB y análisis de archivos. Todos los módulos fueron operativos y se integraron mediante una interfaz web que permite la navegación entre funciones clave:

- Monitoreo de tráfico de red mediante captura de paquetes con Scapy.
- Escaneo de puertos a través de socket y/o Nmap.
- Verificación de dispositivos USB conectados utilizando PyUSB.
- Análisis de archivos subidos desde la interfaz para evaluar su integridad y seguridad.

Todos los módulos fueron probados individualmente y luego integrados en una interfaz web amigable desarrollada con Flask, que permite la navegación entre funciones. Esto permitió validar la viabilidad técnica del sistema y sentar las bases para la ampliación futura del proyecto.

3.2 Gestión del Tiempo

El proyecto se ejecutó conforme al cronograma previsto. Las actividades se dividieron en fases (análisis, diseño, desarrollo, pruebas y documentación), y cada fase se completó en los tiempos estipulados. La gestión de tiempos fue efectiva, permitiendo incluso margen para pruebas adicionales y ajustes finales.

Fase	Fecha programada	Fecha real	Estado
Análisis y requerimientos	15 febrero 2025	17 febrero 2025	Completado
Diseño de arquitectura	22 febrero 2025	25 febrero 2025	Completado
Desarrollo de módulos	01 marzo 2025	22 abril 2025	Completado
Pruebas y correcciones	23 abril 2025	30 abril 2025	Completado
Documentación final	01 mayo 2025	04 mayo 2025	Completado

3.3 Calidad de los Entregables

Los entregables fueron evaluados mediante pruebas funcionales y revisiones internas. Cada módulo presentó un desempeño conforme a las especificaciones técnicas. La interfaz de usuario fue probada por usuarios internos y cumplió con criterios básicos de usabilidad.

Se aplicaron correcciones menores a lo largo del proceso de pruebas, lo que contribuyó a una mejora continua y a un producto final estable.

3.4 Desempeño del Equipo de Trabajo

El equipo de desarrollo mostró un alto nivel de compromiso y colaboración. Las tareas fueron distribuidas equitativamente, favoreciendo el trabajo paralelo. La comunicación constante entre los integrantes permitió una integración eficiente de los módulos y una rápida resolución de incidencias. Cada integrante asumió con responsabilidad los roles asignados, como desarrollo backend, diseño de interfaz, pruebas o documentación.

- Las tareas fueron distribuidas de forma equitativa, lo que permitió trabajo en paralelo y redujo tiempos muertos.
- Se realizaron reuniones periódicas para dar seguimiento, resolver dudas y reasignar tareas en caso de necesidad.
- Las decisiones técnicas se tomaron por consenso, garantizando participación activa y sentido de pertenencia.
- El liderazgo técnico y la capacidad de adaptación del grupo fueron clave para resolver problemas sin generar conflictos ni retrasos significativos.

El equipo logró desarrollar un producto funcional, colaborar de manera efectiva y cumplir los plazos establecidos con un alto nivel de profesionalismo.

Resultados esperados para el siguiente cuatrimestre:

- Optimización del rendimiento del sistema bajo condiciones de tráfico real.
- Implementación de autenticación de usuarios para mayor seguridad.
- Mejora en la gestión de la lista de dispositivos USB confiables.
- Desarrollo de un registro histórico de alertas y actividades del sistema (log de eventos).

4. Reporte de actividades de aseguramiento de la calidad.

¿Qué es el aseguramiento de la calidad en proyectos?

El aseguramiento de la calidad (Quality Assurance, QA) es un conjunto de actividades planificadas y sistemáticas que se implementan durante el ciclo de vida del proyecto con el objetivo de garantizar que los productos o servicios cumplan con los estándares de calidad definidos, así como con las expectativas de los usuarios. A diferencia del control de calidad, que se enfoca en detectar errores una vez que el producto está terminado, el aseguramiento de la calidad busca prevenir errores antes de que ocurran mediante revisiones, pruebas y auditorías constantes.

¿Para qué sirve este reporte?

Este reporte tiene como finalidad:

- Documentar todas las actividades realizadas para asegurar que los entregables del sistema IDS cumplan con los requisitos técnicos y funcionales definidos.
- Mostrar evidencia del seguimiento a estándares de desarrollo y buenas prácticas.
- Servir como respaldo en auditorías, evaluaciones o revisiones posteriores.
- Ayudar a identificar áreas de mejora y retroalimentar futuras iteraciones del sistema.

Aplicación del Aseguramiento de la Calidad en el Proyecto IDS

El proyecto IDS (Sistema de Detección de Intrusos) implementó un enfoque de calidad enfocado en la confiabilidad, seguridad, usabilidad y mantenibilidad del sistema. A continuación, se describen las actividades clave realizadas para asegurar estos aspectos:

Actividades Realizadas

4.1. Planificación de la calidad

Desde la etapa de diseño, el equipo definió los siguientes criterios de calidad:

- Precisión en la detección de paquetes de red.
- Tiempo de respuesta óptimo (<10s en escaneos).
- Detección automática de dispositivos USB.
- Manejo seguro de archivos subidos por el usuario.
- Interfaz clara, funcional y segura.

Estos criterios guiaron todas las fases de desarrollo y prueba.

4.2. Revisión de código

Cada módulo fue revisado por un miembro del equipo distinto al desarrollador original, evaluando:

- Cumplimiento de estándares PEP8 para Python.
- Legibilidad y documentación interna del código.
- Manejo adecuado de errores y excepciones.

Esta práctica ayudó a detectar inconsistencias lógicas antes de realizar pruebas funcionales.

4.3. Pruebas unitarias

Se crearon pruebas unitarias para validar individualmente cada función clave. Por ejemplo:

- Comprobación de detección de paquetes ICMP con Scapy.
- Verificación del análisis de archivos según su extensión y contenido.
- Validación de detección y desconexión de dispositivos USB.

Las pruebas fueron repetidas múltiples veces bajo diferentes condiciones.

4.4. Pruebas funcionales

Desde la interfaz web, se probó el funcionamiento completo del sistema:

- Subida y análisis de archivos.
- Escaneo de red desde diferentes dispositivos.
- Detección de intrusos o tráfico no usual.

Se usaron escenarios reales de prueba con usuarios no involucrados directamente en el desarrollo.

4.5. Control de errores y bitácora

Se mantuvo una bitácora donde se registraron errores detectados y acciones correctivas implementadas. Esto permitió hacer seguimiento y asegurar que no se repitieran fallos en versiones futuras.

4.6 Ejemplos de Plantillas Utilizadas para el Reporte de Aseguramiento de Calidad

Plantilla 1: Registro de Pruebas Unitarias

ID de Prueba	Módulo Evaluado	Función Probada	Entrada	Resultado Esperado	Resultado Obtenido	Estado	Observaciones
PU-001	Tráfico de red	Captura de paquetes con Scapy	Paquetes ICMP	Detecta y lista paquetes ICMP	Detecta y lista paquetes ICMP	Aprobado	-
PU-002	Escaneo de puertos	Escaneo con Nmap	IP: 192.168.0.1	Lista puertos abiertos	Lista puertos abiertos	Aprobado	-
PU-003	USB conectado	Detección con PyUSB	USB Kingston	Muestra ID del dispositivo	No detecta en primer intento	Fallido	Se solucionó reiniciando hilo
PU-004	Análisis de archivos	Archivo .exe sospechoso	archivo .exe	Detecta posible riesgo o alerta	Muestra alerta de archivo sospechoso	Aprobado	-

Plantilla 2: Lista de Verificación de Calidad (Checklist QA)

Criterio de Calidad	Cumple (Sí/No)	Observaciones
La interfaz carga correctamente en todos los navegadores	Sí	Probado en Chrome, Firefox y Edge
El sistema detecta dispositivos USB externos	Sí	Compatible con dispositivos FAT32 y NTFS
El escaneo de puertos responde en menos de 10 segundos	Sí	Tiempo promedio: 7.2 segundos
El análisis de archivos rechaza formatos no permitidos	Sí	Se bloqueó carga de archivos .bat y .dll
El sistema genera errores controlados (try/except)	Sí	Se documentaron logs en archivo error.log

Plantilla 3: Bitácora de Corrección de Errores

ID de Error	Fecha	Módulo	Descripción del Error	Solución Aplicada	Responsable	Estado
ERR-01	20/04/2025	USB Detection	No se detectaban dispositivos conectados	Se reinició hilo de escucha PyUSB		Solucionado

ERR-02	21/04/2025	Análisis de archivos	Archivo .pdf válido era rechazado	Se ajustó validación de MIME types		Solucionado
ERR-03	23/04/2025	Interfaz web	No cargaban estilos en algunos navegadores	Se corrigieron rutas de archivos estáticos		Solucionado

5. Evaluaciones de desempeño del equipo de trabajo.

La evaluación del desempeño del equipo de trabajo tiene como objetivo identificar fortalezas, áreas de mejora y el nivel de colaboración alcanzado durante el desarrollo del proyecto IDS .

La evaluación se realizó con base en los siguientes criterios:

- Cumplimiento de responsabilidades asignadas
- Colaboración y trabajo en equipo
- Comunicación efectiva
- Calidad del trabajo entregado
- Resolución de problemas y actitud proactiva

La escala de evaluación fue del 1 al 5:

- 1 = Muy deficiente
- 2 = Deficiente
- 3 = Aceptable
- 4 = Bueno
- 5 = Excelente

Tabla: Evaluación General del Desempeño

Integrante	Cumplimiento de tareas	Colaboración	Comunicación	Calidad del trabajo	Actitud proactiva	Promedio Final
Marco Antonio De la Rosa	5	5	5	5	4	4.8
Luis Daniel García Martínez	5	5	5	5	4	4.8
Jessica Banelly Cortez Alcalá	4	5	4	5	5	4.6

Ricardo Rodríguez López	5	4	4	5	4	4.4
-------------------------	---	---	---	---	---	-----

Análisis de Resultados

- **Marco Antonio De la Rosa** mostró liderazgo técnico y organizacional en la gestión del módulo de detección de dispositivos USB y escaneo de red. Su participación fue constante y aportó soluciones eficientes ante problemas técnicos.
- **Luis Daniel García Martínez** destacó por su compromiso con el desarrollo backend en Flask. Mostró buena disposición para colaborar y se comunicó de manera efectiva con sus compañeros, aunque con oportunidades de mejora en la entrega puntual de tareas.
- **Jessica Banelly Cortez Alcalá** tuvo una participación muy activa en la parte de pruebas, documentación y diseño de interfaz. Su enfoque detallista y actitud proactiva contribuyeron a mantener la calidad general del sistema.
- **Ricardo Rodríguez López** se encargó del análisis de archivos y apoyo en las pruebas de red. Mostró una alta calidad técnica en sus entregables y mantuvo una actitud colaborativa durante todas las fases del proyecto.

El equipo demostró un alto nivel de compromiso y trabajo colaborativo. La combinación de habilidades técnicas, comunicación constante y responsabilidad individual permitió alcanzar los objetivos del proyecto IDS con eficiencia. Las evaluaciones muestran un equilibrio en el desempeño del grupo, con una sinergia positiva que fortaleció el desarrollo del sistema.

6. Análisis y evaluación de nuevos riesgos.

Durante el desarrollo del proyecto se identificaron diversos riesgos emergentes que podrían afectar la funcionalidad, estabilidad y seguridad del Sistema de Detección de Intrusos (IDS). Estos riesgos fueron evaluados considerando su probabilidad de ocurrencia, impacto potencial y estrategias de mitigación. A continuación, se detallan los principales riesgos detectados y las acciones tomadas o previstas para abordarlos:

Sobrecarga del sistema: En redes con alto tráfico, el monitoreo continuo de paquetes puede provocar lentitud o fallos en el sistema. Esto se detectó durante pruebas con simuladores de tráfico pesado. Para mitigarlo, se propuso el uso de filtros específicos que analicen únicamente protocolos seleccionados, así como la limitación de paquetes por segundo para mantener el rendimiento del sistema estable.

```

0 # Monitorear tráfico de red en una interfaz específica
1 def sniff_traffic_iface(iface):
2     print(f"\n[🔍] Monitoreando tráfico de red en la interfaz {iface}...\n")
3     scapy.sniff(filter="", prn=process_packet, store=False, iface=iface, timeout=10) # Filtrado sin restricciones, captura
4
5 # Procesar cada paquete y pasarlo a la detección
6 def process_packet(packet):
7     detect_port_scan(packet) # Detectar escaneo de puertos
8     detect_ping(packet) # Detectar pings ICMP
9
10 # Función para monitorear tráfico de red automáticamente en ambas interfaces
11 def start_network_monitoring():
12     # Iniciar monitoreo en dos hilos separados (Ethernet y Wi-Fi)
13     ethernet_thread = threading.Thread(target=sniff_traffic_iface, args=("Ethernet",))
14     ethernet_thread.start()
15
16     wifi_thread = threading.Thread(target=sniff_traffic_iface, args=("Wi-Fi",))
17     wifi_thread.start()
18
19     # Espera 10 segundos mientras se monitorean ambas interfaces
20     time.sleep(10)
21     ethernet_thread.join()
22     wifi_thread.join()

```

Los 10 segundos se pusieron para evitar que la captura de paquetes se vuelva demasiado pesada o saturante para el sistema.

```

.. \n")
e, timeout=10) #

```

Por qué 10 segundos y no otro valor:

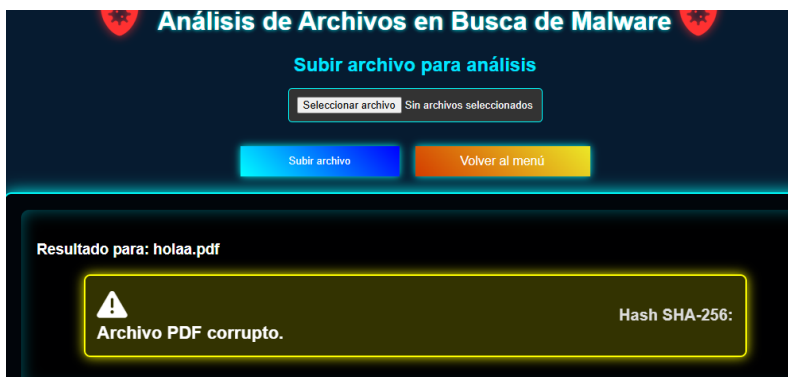
Los tiempos comunes para el timeout según la carga y el contexto:

- 5 segundos: Muy corto, para pruebas rápidas o para capturas muy puntuales.
- 10 segundos: Buen balance para pruebas cortas sin saturar el sistema, recomendado para ambientes con tráfico moderado.
- 30 segundos: Para capturas más extensas sin detener el programa.
- 60 segundos o más: Para monitoreo más prolongado, pero puede generar mucha carga y datos.
- Sin timeout: Para monitoreo indefinido, pero puede saturar el sistema y llenar la memoria si no se controla.

Compatibilidad con sistemas operativos: Algunas bibliotecas utilizadas (por ejemplo, PyUSB o Scapy) presentan comportamientos diferentes según el sistema operativo. En distribuciones de Linux con restricciones de permisos o en versiones de Windows sin drivers adecuados, el sistema podría no funcionar como se espera. Para ello, se están desarrollando scripts adaptativos y recomendaciones específicas en la documentación técnica para cada sistema.

Falsos positivos en el análisis de archivos: Durante las pruebas del módulo de análisis de archivos, algunos documentos legítimos fueron marcados como sospechosos debido a patrones genéricos mal definidos. Este riesgo puede generar desconfianza en el sistema. La mitigación

propuesta es mejorar el algoritmo de análisis, incorporar validación por hash y, en fases futuras, aplicar bibliotecas de análisis con firmas digitales o aprendizaje automático.



Mal uso del sistema por usuarios no capacitados: Existe el riesgo de que usuarios sin conocimientos técnicos utilicen erróneamente los módulos, provocando sobrecarga o interpretaciones incorrectas de los resultados. Se planea mitigar este riesgo mediante la creación de una guía de usuario clara, mensajes de ayuda dentro de la interfaz y restricciones sobre el uso de ciertas funciones avanzadas.

Problemas en la detección de dispositivos USB no registrados: En algunos casos, dispositivos legítimos pueden no ser reconocidos si no están bien registrados en la lista segura. Esto genera alertas innecesarias. Se propuso crear un módulo de administración de USBs confiables accesible desde la interfaz para facilitar su gestión por parte de administradores.

```
# Lista de dispositivos autorizados (nombre del dispositivo)
known_usb_devices = [
    "KingstonDataTraveler_2.01",
    "USBSTOR\\DiskKingstonDataTraveler_2.01.00",
    "USBSTOR\\DiskKingstonDataTraveler_2.0",
    "USBSTOR\\KingstonDataTraveler_2.01",
    "USB\\VID_0951&PID_1665\\60A44C426518F0A0363335C1",
    "USB\\VID_04F2&PID_B1D6\\6&2C9DDD91&0&4",
    "USB\\VID_8087&PID_0024\\5&31C9D4A9&0&1",
    "USB\\VID_1EA7&PID_0066\\6&2C9DDD91&0&1",
    "USB\\ROOT_HUB20\\4&125BC64D&0",
    "USB\\ROOT_HUB20\\4&17684393&0",
    "USB\\VID_0BDA&PID_0138\\20090516388200000",
    "USB\\VID_1EA7&PID_0066\\6&2C9DDD91&0&1",
    "USB\\VID_258A&PID_0016\\5&10BE35C7&0&5"
]
```

Durante las pruebas del módulo de detección de dispositivos USB, se identificó un fallo recurrente en la validación de dispositivos confiables: algunos dispositivos legítimos no eran reconocidos como válidos si su ruta no incluía la doble diagonal invertida (\\).

Estos riesgos se documentan formalmente en un registro y se actualizarán conforme avance el desarrollo. La identificación temprana de riesgos y la implementación de medidas preventivas aseguran una mayor robustez y fiabilidad del sistema conforme se integre en entornos reales.

7. Registro de incidentes del proyecto.

Durante el desarrollo del proyecto, se presentaron diversos incidentes técnicos y organizacionales que requirieron atención inmediata y resolución oportuna. A continuación, se describen los principales incidentes registrados, junto con las acciones correctivas implementadas:

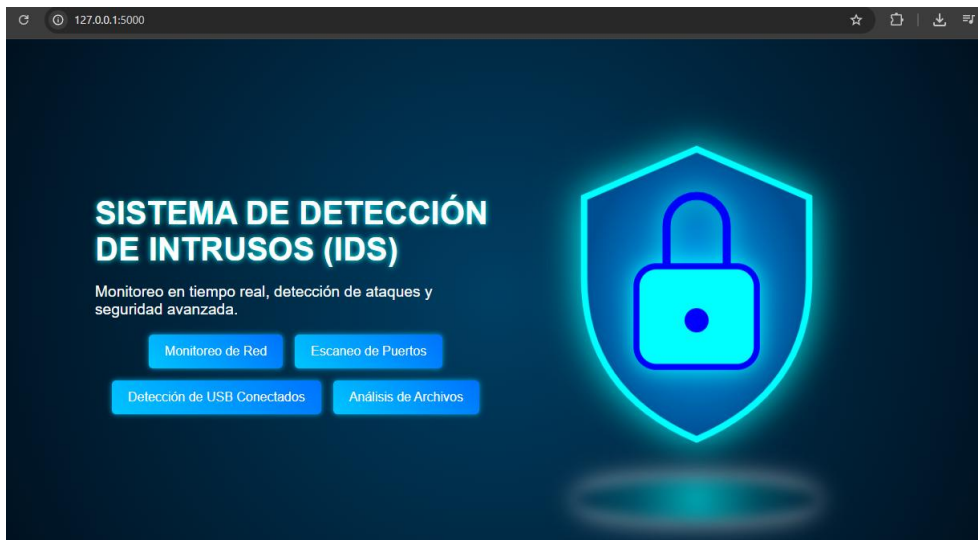
- **Desincronización entre los botones de la interfaz y las rutas en Flask:** Algunos botones de la interfaz web no ejecutaban correctamente sus funciones por errores en la configuración de las rutas. Se revisó y reorganizó el archivo `app.py`, verificando la conexión entre backend y frontend.
- **Problemas de codificación en la carga de archivos:** Durante las pruebas del módulo de análisis de archivos, algunos archivos `.docx` y `.pdf` generaban errores por codificaciones no reconocidas. Se amplió la validación y se integró un filtro de tipos de archivo aceptados.
- **Comando incorrecto para instalación de librerías:** En algunas ocasiones, el entorno de desarrollo falló debido a comandos mal ejecutados al instalar dependencias (por ejemplo, errores en la sintaxis de `pip` o instalación de versiones incompatibles).
- **Fallos del módulo Scapy en diferentes interfaces de red:** El módulo de monitoreo no funcionaba correctamente al alternar entre conexiones Ethernet y WiFi. En ciertos equipos, Scapy no detectaba tráfico al cambiar la interfaz activa. Se solucionó incorporando lógica para detección automática de la interfaz activa y se añadieron mensajes de advertencia al usuario cuando no se detectaba tráfico.

Cada incidente fue documentado en una bitácora interna del proyecto, descripción del problema, responsables de resolución y solución aplicada. Esta práctica permitió mejorar el control de calidad y prevenir errores similares en fases futuras.

8. Registro de cambios implementados.

Durante el desarrollo del Sistema de Detección de Intrusos (IDS), se realizaron diversos ajustes, optimizaciones y mejoras al sistema como parte del proceso iterativo de desarrollo, así como en respuesta a resultados de pruebas técnicas, retroalimentación de usuarios internos y resolución de incidentes. Estos cambios fueron clave para garantizar la funcionalidad, estabilidad y facilidad de uso del sistema. A continuación, se detalla de forma más amplia cada uno de los principales cambios implementados durante esta primera fase del proyecto:

Optimización de la interfaz de usuario: Se reorganizó el diseño visual y funcional de la interfaz web con el objetivo de hacerla más intuitiva para el usuario. Se aplicaron estilos CSS mejorados, se ajustó el contraste de colores, se agregaron botones de regreso al menú principal en cada módulo, y se reorganizó la estructura de navegación. Estas mejoras contribuyeron a una experiencia de usuario más fluida y comprensible.



Corrección de rutas en Flask: En la etapa de integración, se detectaron inconsistencias entre las rutas definidas en Flask y las acciones esperadas en la interfaz web. Estas rutas fueron revisadas, depuradas y modificadas para asegurar que cada botón o acción en la interfaz estuviera correctamente enlazada con su correspondiente función backend. Esto permitió el funcionamiento correcto de cada módulo desde la plataforma web.

Mejora en el manejo de archivos cargados: El módulo de análisis de archivos fue reforzado mediante validaciones adicionales para evitar errores al procesar archivos corruptos, vacíos o de tipos no soportados. Se añadieron filtros por extensión y validación MIME, así como mensajes personalizados que informan claramente al usuario si el archivo es seguro, sospechoso o presenta problemas. Esta mejora fortaleció la confiabilidad del módulo de análisis.

Inclusión de validaciones para entradas de IP: En el módulo de escaneo de puertos, se incluyó una función para validar la estructura de la dirección IP antes de ejecutar el escaneo. También se integró una verificación de conexión mediante ping previo para asegurar que el host esté activo, reduciendo errores durante el proceso y mejorando la eficiencia del sistema.

Documentación técnica mejorada: Se reforzó la documentación técnica del sistema, añadiendo una guía detallada paso a paso sobre la instalación en distintos sistemas operativos (Windows), instalación de dependencias y ejecución del servidor. También se incluyeron capturas de pantalla de la interfaz y ejemplos de uso práctico para facilitar el despliegue por parte de otros usuarios.

Todos estos cambios fueron registrados en una bitácora de cambios que acompaña al proyecto y que será actualizada continuamente en las siguientes fases. Esto garantiza trazabilidad y control sobre la evolución del sistema.

9. Minutas de reunión con interesados.

¿Qué son las minutas de reunión?

Las minutas de reunión son documentos breves y estructurados que registran de forma oficial lo ocurrido en una reunión, incluyendo: fecha, participantes, temas tratados, acuerdos, tareas asignadas, responsables y fechas límite. Funcionan como evidencia del seguimiento y avance del proyecto.

En el contexto de la gestión de proyectos, las minutas reflejan la comunicación efectiva entre el equipo de trabajo y los interesados.

¿Para qué sirven las minutas?

1. Registrar acuerdos importantes: Todo lo que se decide en una reunión queda documentado para evitar malentendidos.
2. Asignar y dar seguimiento a tareas: Se especifica quién es responsable de cada acción y su fecha de entrega.
3. Facilitar la continuidad del proyecto: Ayuda a que cualquier miembro nuevo o ausente pueda ponerse al día fácilmente.
4. Respaldar la toma de decisiones: Sirven como evidencia en caso de que se necesite justificar una acción o cambio.
5. Medir avances: Comparando minutas pasadas, se puede verificar el cumplimiento de objetivos parciales.


¿Por qué son importantes las minutas en este proyecto?

En el desarrollo del sistema IDS, las minutas fueron claves para:

- Comunicar avances técnicos a los interesados, como el docente asesor.
- Obtener retroalimentación sobre la interfaz, funcionalidades o prácticas seguras.
- Formalizar decisiones sobre cambios en el alcance del proyecto o herramientas utilizadas.
- Registrar incidentes técnicos o demoras y proponer soluciones conjuntas.

Ejemplo de Plantilla para minuta de reunión utilizada en nuestro proyecto

Las minutas son una herramienta esencial en cualquier proyecto, ya que garantizan la transparencia, trazabilidad y organización del trabajo colaborativo. En el desarrollo del sistema IDS, permitieron una comunicación clara con los interesados y facilitaron la toma de decisiones fundamentadas. Gracias a ellas, el equipo pudo mantener el rumbo del proyecto alineado a los objetivos y tiempos establecidos.

Acta de reunión 	
ASUNTO: KICKOFF DEL PROYECTO IDS: DEFINICIÓN DE ALCANCE, ROLES Y PLAN INICIAL	
Fecha: 10/02/25	Hora: 16:00 a 17:00 Hrs
Lista de asistentes	Resumen de discusiones
<ul style="list-style-type: none">• Marco Antonio De la Rosa Vázquez• Luis Daniel García Martínez• Jessica Banelly Cortez Alcalá• Ricardo Rodríguez López	<p>Presentación del proyecto:</p> <p>Se explicó el objetivo del IDS: monitoreo de red, escaneo de puertos, detección de USB y análisis de archivos.</p> <p>Tecnologías confirmadas: Python, Flask, Scapy, Nmap, PyUSB.</p> <p>Definición de roles:</p> <p>Marco Antonio: Módulos de red (Scapy) y puertos (Nmap).</p> <p>Luis Daniel: Interfaz web (HTML/CSS/Flask).</p> <p>Jessica: Pruebas de calidad y documentación.</p> <p>Ricardo: Coordinación general y comunicación con el profesor.</p> <p>Alcance y cronograma preliminar:</p> <p>Semana 1: Investigación técnica (Scapy/PyUSB).</p> <p>Semana 2: Desarrollo de módulos básicos.</p> <p>Semana 3: Integración con interfaz web.</p>
Agenda	
<ol style="list-style-type: none">1. Presentación del proyecto2. Asignación de roles .3. Planificación inicial4. Discusión de riesgos .5. Acuerdos y próximos pasos	
Responsabilidades y plazos	Observaciones
<ul style="list-style-type: none">• Investigación Scapy/Nmap• Boceto interfaz web• Documentar requisitos• Enviar avance al profesor	<ul style="list-style-type: none">• Prioridad: Completar la investigación técnica antes de codificar.• Próxima reunión: 17/02/2025 para revisar avances.

Acta de reunión



ASUNTO: REVISIÓN DE AVANCES Y CORRECCIÓN DE MÓDULOS IDS

Fecha: 15/04/25

Hora: 16:00 a 17:00 Hrs

Lista de asistentes

- Marco Antonio De la Rosa Vázquez
- Luis Daniel García Martínez
- Jessica Banelly Cortez Alcalá
- Ricardo Rodríguez López

Agenda

Presentación del estado actual de proyecto.
Reporte de errores en la detección de dispositivos USB.
Comentarios y sugerencias del asesor sobre la interfaz web.
Planeación de la fase de pruebas de calidad.

Responsabilidades y plazos

- Solucionar errores de PyUSB o migrar a pyudev
- Mejorar la interfaz web en Flask (estilo y colores)
- Diseñar y compartir el checklist para pruebas QA
- Enviar informe de reunión y avances-

Resumen de discusiones

Se presentó el funcionamiento del módulo de escaneo de red y se mostraron resultados preliminares.

Luis reportó fallos intermitentes al detectar ciertos dispositivos USB. Se sugirió evaluar el uso de la librería pyudev.

El asesor indicó mejorar el diseño de la interfaz web y unificar estilos CSS.

Se acordó comenzar la fase de pruebas funcionales el 23 de abril bajo una lista de verificación previamente diseñada.

Observaciones

Buena comunicación entre los miembros.
El equipo ha cumplido con la mayoría de entregables previstos.
Se recomendó hacer un respaldo del proyecto en GitHub al finalizar cada módulo.

Conclusión:

El desarrollo del **Sistema de Detección de Intrusos (IDS)** representó una experiencia integral que permitió aplicar conocimientos técnicos, habilidades de gestión de proyectos y trabajo colaborativo. Durante este primer cuatrimestre, se logró materializar una versión funcional del sistema, cumpliendo con los objetivos establecidos desde la planificación, como la implementación de los módulos de monitoreo de red, escaneo de puertos, verificación de dispositivos USB y análisis de archivos. El proyecto avanzó dentro del cronograma previsto, enfrentando y resolviendo diversos desafíos técnicos, como incompatibilidades de librerías, problemas de permisos y adaptación del sistema a distintos entornos. Estas situaciones fortalecieron la capacidad del equipo para buscar soluciones efectivas, mejorar continuamente el sistema y documentar adecuadamente cada fase del proceso.

La coordinación entre los integrantes fue clave para lograr una distribución equitativa de tareas, favoreciendo la eficiencia y el cumplimiento de plazos. Además, la retroalimentación obtenida durante las pruebas permitió realizar mejoras significativas en usabilidad, rendimiento y seguridad.

Este proyecto no solo cumplió su propósito como producto académico, sino que también demostró su potencial como una herramienta real para entornos educativos o institucionales que deseen fortalecer su seguridad informática. En las próximas etapas se espera refinar los módulos existentes, implementar autenticación, ampliar la capacidad de análisis y documentar eventos para una gestión más completa de las amenazas.

En conclusión, el IDS es una solución viable, escalable y con amplias posibilidades de mejora, fruto de un esfuerzo conjunto bien estructurado desde la dirección de proyectos hasta su desarrollo técnico y funcional.