Universidad Politécnica de Texcoco

MANUAL DE USUARIO DEL SISTEMA DE MONITOREO Y PROTECCIÓN PARA UAVS

Leal Sánchez Juan Marcos

Carventes Garduño Alan Daniel

INTRODUCCIÓN

Este manual de usuario tiene como propósito proporcionar una guía detallada para la correcta utilización del sistema de monitoreo y protección de datos en UAVs desarrollado como parte del proyecto "Diseño del proceso de protección de datos y comunicaciones en UAV usando la normativa PCI DSS, PCI PIN e ISO 27001".

El sistema, basado en una arquitectura segura y modular, permite simular un entorno real de monitoreo utilizando un ESP32 como intermediario de red, y una plataforma web que brinda visualización en tiempo real, análisis de eventos, autenticación básica y simulación de amenazas. Este manual está dirigido a usuarios técnicos y no técnicos, como ingenieros, estudiantes y profesionales en formación.

2 COMPONENTES DEL SISTEMA

2.1. Hardware requerido

- UAV modelo LYZRC L900 Pro con doble cámara 8K.
- Módulo ESP32 con capacidad WiFi (AP + STA).
- Laptop o estación de monitoreo con navegador actualizado (Chrome, Firefox).

2.2. Software requerido

- Navegador con soporte JavaScript y HTML5.
- Servidor local Apache (AppServ o XAMPP).
- Archivos PHP (monitoreo.php, guardar_log.php, registrar_evento.php, etc.)
- Scripts de cliente (monitoreo_uav.js, visualizacion_uav.js, controlador_uav.js, tabs_dinamicos.js, script.js).
- Hojas de estilo style.css para diseño responsivo y alertas visuales.
- Estructura HTML (index.html) como punto de inicio para navegación de la solución.

3. ACCESO Y PRIMER USO

3.1. Conexión inicial

1. Encender el dron.

2. Alimentar el ESP32 (batería o USB).

3. Conectarse desde la laptop al AP "Dron_Seguro" generado por el ESP32.

4. Abrir el navegador y acceder a `http://192.168.4.1/tesis/index.php`.

3.2. Ingreso al sistema

- Ingresar el ID del UAV y contraseña registrados en la base de datos.

- Si las credenciales son válidas, se redirige al tablero principal. (dashboard.php).

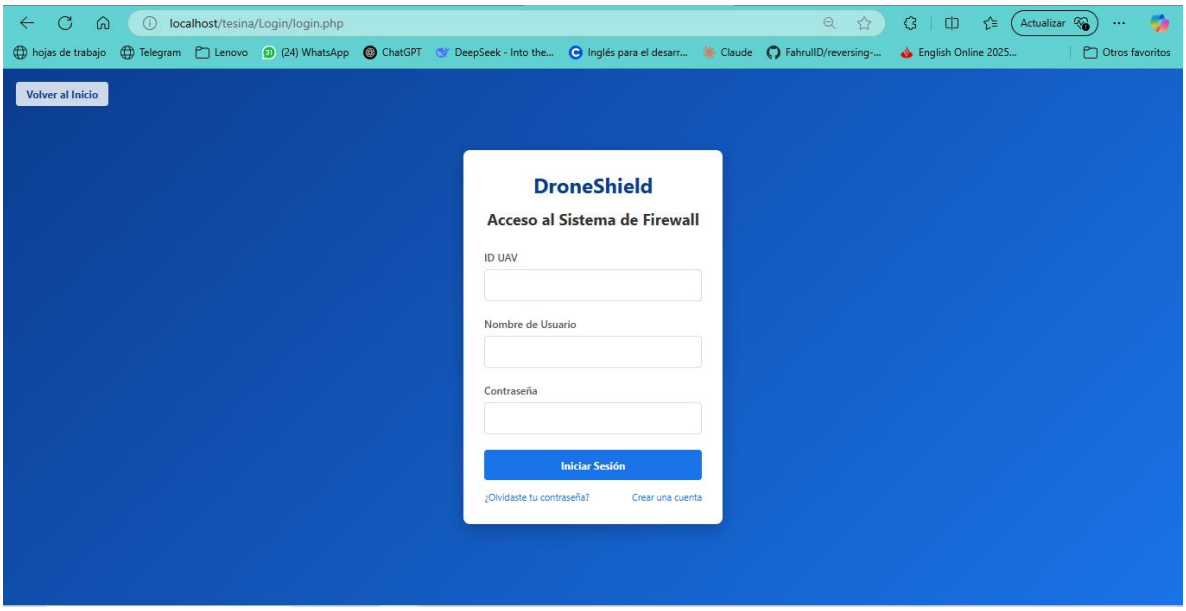

Imagen 1 – Pantalla de inicio de sesión del sistema de monitoreo. Permite ingresar con ID del UAV y contraseña para acceder al sistema.

4. FUNCIONALIDADES DEL SISTEMA

4.1. Visualización de métricas (tabs_dinamicos.js)

- **Señal (dBm):** indica la potencia de la conexión inalámbrica.

- **Velocidad estimada (Mbps):** tráfico de datos entre el UAV y el servidor.

- **Latencia (ms):** tiempo de respuesta del sistema.

- **Pérdida de paquetes (%):** pérdida en la transmisión.

Los datos se actualizan automáticamente cada 4 segundos.

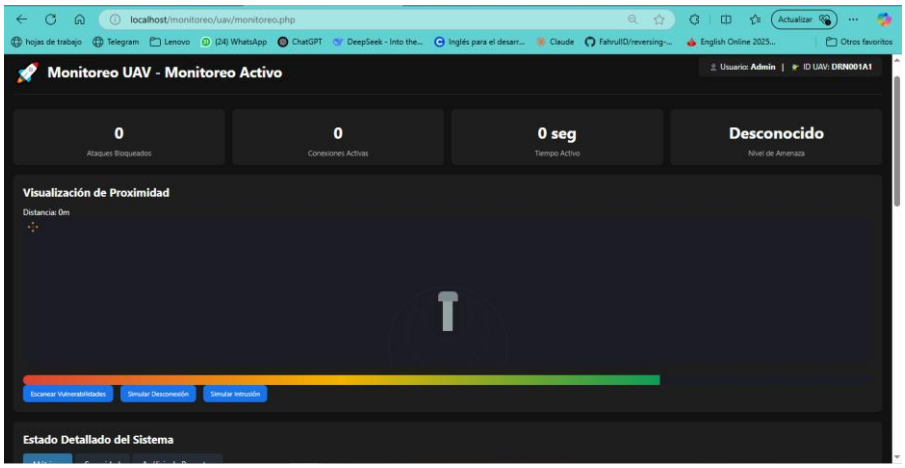

Imagen 2 – Vista del panel de métricas simuladas. Se visualizan en tarjetas informativas con colores que indican el nivel de calidad

4.2. Representación gráfica del UAV (visualizacion_uav.js)

- Animación del dron con movimiento simulado.
- Cálculo de distancia a la antena.
- Visualización de intensidad de señal.

- Simulaciones:

- **Escaneo:** emula un análisis sin amenazas.
- **Intrusión:** muestra alerta crítica en pantalla.
- **Desconexión:** corta la señal y simula pérdida de enlace.

Imagen 3 – Área de visualización gráfica del dron. Se muestra un modelo animado del UAV desplazándose respecto a una antena base.

4.3. Monitoreo de estado y amenazas (monitoreo_uav.js)

- Variables activas:

- Uptime.
- Número de conexiones.
- Nivel de amenaza (bajo, medio, alto).
- Número de ataques bloqueados.

- Detección automática:

- Alertas visuales y sonoras ante múltiples conexiones sospechosas.

- Actualización de la gráfica de uptime (Chart.js).

- Reportes en base de datos vía `guardar_log.php`.

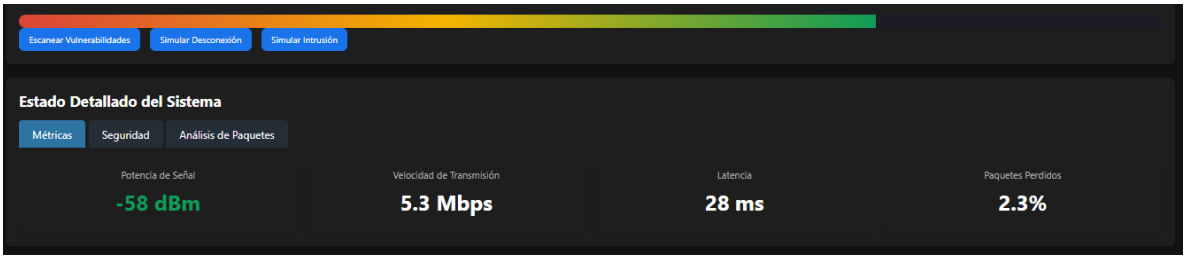

Imagen 4 – Sección de estado del sistema y amenazas. Se representan con colores e iconos según el nivel de riesgo.

4.4. Control de conexiones (controlador_uav.js)

- Listado de MACs conectadas al ESP32.
- Botón para desconectar individualmente cada dispositivo.
- Confirmación previa de acción.
- Actualización automática de tabla cada 30 segundos.

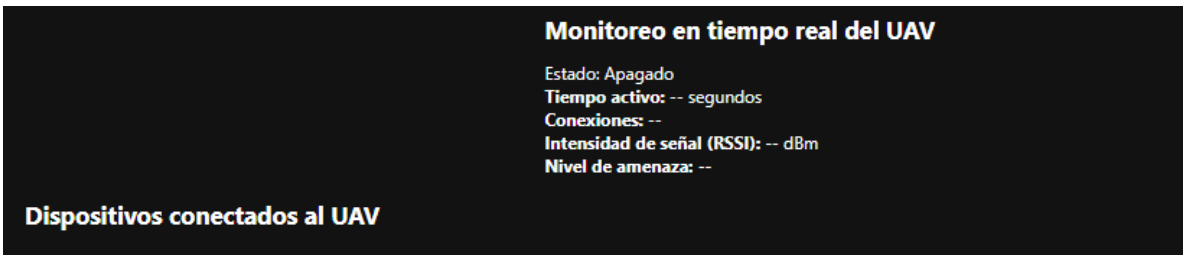

Imagen 5 – Tabla de clientes conectados. Permite observar las direcciones MAC activas y desconectarlas manualmente.

4.5. Registro y visualización de eventos

- Eventos de tipo "Intrusión", "Desconexión", etc., son almacenados en MySQL.
- Se visualizan en una tabla dinámica.
- Utiliza `obtener_eventos.php` y `registrar_evento.php` como backend.

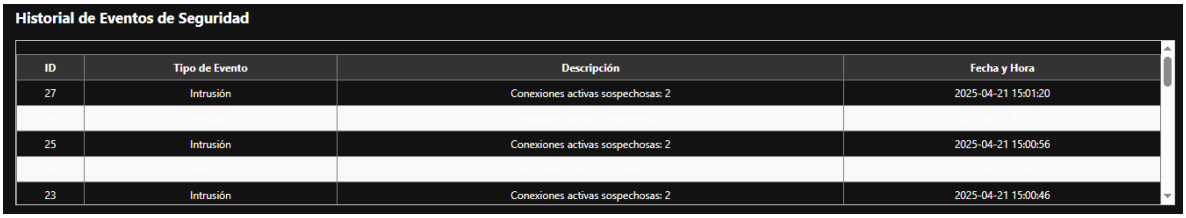

Imagen 6 – Tabla histórica de eventos de seguridad. Incluye columnas como tipo de evento, fecha, y descripción.

4.6. Simulación Avanzada del Firewall (script.js, style.css, index.html)

- Esta sección representa la interfaz inicial de bienvenida (index.html) donde se expone un entorno profesional con menús de navegación hacia "Características", "Certificaciones", "Beneficios", "Testimonios" y una CTA de contacto.
- El botón "Solicitar Demo" redirige a login.php, desde donde se ingresa al sistema real.

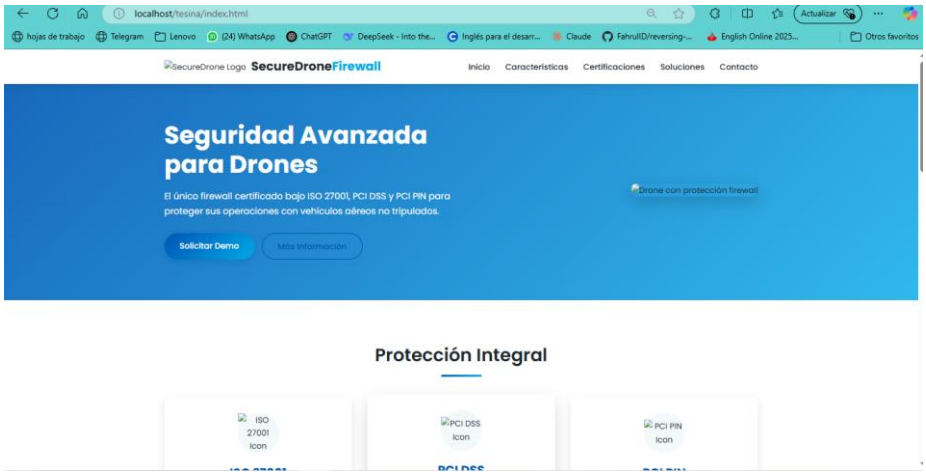

Imagen 8 – Página de bienvenida del sistema SecureDrone Firewall con presentación del producto y acceso al demo.

Funciones adicionales integradas desde script.js:

- **Mapa de amenazas dinámico:** Crea puntos de amenaza visuales en un mapa de fondo.
- **Gráfica de tráfico:** Compara conexiones legítimas e intentos bloqueados (usando Chart.js).
- **Nivel de amenaza:** Controlado mediante menú desplegable.
- **Modo emergencia:** Cambia el estado visual del sistema a rojo oscuro, simulando un incidente.
- **Escaneo de vulnerabilidades:** Simula búsqueda de amenazas y genera notificaciones.
- **Simulación constante de tráfico:** Añade eventos aleatorios al log de seguridad con IPs ficticias y tipo de amenaza (SQLi, fuerza bruta, puerto escaneado, etc.).
- **Control de logs:** Los eventos en lista pueden investigarse (🔍) o bloquearse (🚫), lo cual genera retroalimentación visual.

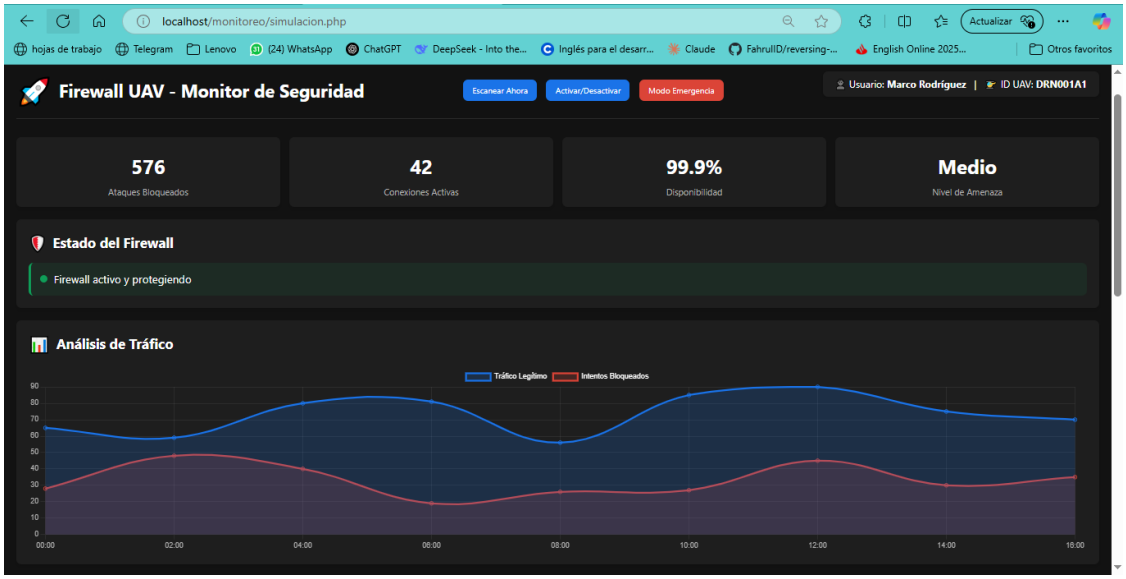

Imagen 9 – Interfaz de simulación con opciones de escaneo, firewall, modo emergencia, y gráfica en tiempo real.

4.7. Estilo visual y notificaciones (style.css)

- Se utilizan variables CSS (:root) para controlar colores de estado, fondo, animaciones.
- Se incluye sistema de notificaciones flotantes que aparece al ejecutar acciones (activación del firewall, escaneo, alertas críticas, etc.).
- Controles de seguridad como toggle-switch para activar o desactivar funciones.

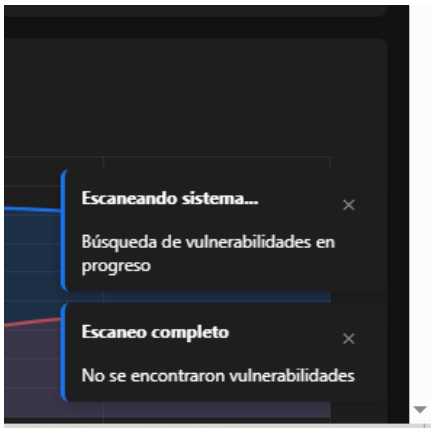

Imagen 10 – Notificaciones emergentes configurables en la parte inferior derecha del sistema. Permiten alertar sobre amenazas detectadas.

5. INTERFAZ Y USABILIDAD

5.1. Estructura visual (style.css)

- Diseño en modo oscuro para menor fatiga visual.
- Colores de estado (verde = seguro, naranja = alerta, rojo = crítico).
- Animaciones suaves y efectos responsivos.

- Secciones:

- Panel de control.
- Estado del firewall.
- Gráficas.
- Eventos y logs.
- Configuraciones y simulaciones.

5.2. Interacción del usuario

- Cambios reflejados sin recargar la página (AJAX).
- Alertas emergentes ante actividades anómalas.
- Botones accesibles y confirmaciones en acciones críticas.

6. ESCENARIOS DE USO

Escenario 1: Monitoreo en tiempo real

- El usuario visualiza la posición simulada del dron, la calidad de la señal y los eventos generados.
- La gráfica muestra el tiempo activo desde que inició la sesión.

Escenario 2: Simulación de amenaza

- Se presiona "Simular intrusión".
- El sistema muestra una alerta crítica.
- Se registra el evento en la base de datos automáticamente.

Escenario 3: Revisión de historial de eventos

- El usuario accede a la tabla "Historial de eventos".
- Puede consultar fecha, tipo y descripción de cada incidente registrado.

7. RECOMENDACIONES DE USO

- Utilizar el sistema en red cerrada local (sin conexión a Internet).
- Evitar múltiples sesiones abiertas desde diferentes dispositivos.
- No apagar el ESP32 durante una simulación activa.
- Validar los datos en la tabla de eventos regularmente.
- Usar credenciales únicas por UAV para mejorar trazabilidad.

8. LIMITACIONES

- El sistema está diseñado para simulación en entorno educativo.
- No incluye cifrado real TLS/SSL ni autenticación multifactor avanzada.
- La detección de amenazas depende de parámetros simulados.

9. MANTENIMIENTO BÁSICO

- Actualizar periódicamente los scripts (`.js`, `.php`) si se detectan errores.
- Respaldar la base de datos de eventos al menos semanalmente.
- Revisar el funcionamiento del ESP32 (recalibrar si pierde conexión).

10. CONTACTO Y SOPORTE

Desarrolladores:

- Carventes Garduño Alan Daniel
- Leal Sánchez Juan Marcos

Universidad Politécnica de Texcoco

Ingeniería en Sistemas Computacionales – Abril 2025

Para más detalles técnicos y normativos, consultar el documento completo de tesis.