

# BASES DE DATOS

RAQUEL LIMPO MARTÍNEZ



GRUPO MP14

PROYECTO 30

## índice

Introducción y Objetivos .....	3
Contexto del Proyecto .....	3
Objetivos de la Base de Datos .....	3
Análisis y Requisitos .....	3
Requisitos Funcionales .....	3
Requisitos No Funcionales .....	4
Casos de Uso Principales .....	4
Modelado de Datos.....	5
Modelo Conceptual:.....	5
Esquema relacional del sistema WiShield.....	5
Modelo Lógico: .....	8
Modelo Físico:.....	8
Relaciones y restricciones: .....	8
Índices y Optimización: .....	8
Procedimientos Almacenados, Triggers y Vistas.....	9
Procedimientos Almacenados y Funciones: .....	9
Triggers.....	10
Seguridad y Gestión de Usuarios.....	11
Control de Acceso para la base de datos:.....	11
Scripts:.....	11
Pruebas y Validación .....	11
Pruebas de estrés.....	15
Implementación de Encriptación de Contraseñas: .....	17
Recuperación de contraseña.....	19
Encriptación de datos sensibles.....	19
Bloquear accesos desde IPs no autorizadas .....	20
Herramientas y Tecnologías Utilizadas .....	20
Conexión con la Raspberry Pi .....	21
Conexión a PowerBI .....	21
Como conectar de MySQL a PowerBI .....	22
Ejemplo de informe .....	25

Scripts .....	27
Bash.....	27
Python.....	29
SQL.....	30
PHP y HTML .....	40
Glosario:.....	68
Referencias y Recursos: .....	70

## **Introducción y Objetivos**

### **Contexto del Proyecto**

El proyecto WiShield consiste en desarrollar una solución integral para la gestión segura y eficiente de la infraestructura de red inalámbrica de una planta del alojamiento estudiantil de la Vila Universitària de la UAB. La solución no solo cubre aspectos técnicos como la infraestructura física, sino que también contempla la monitorización, detección de vulnerabilidades, gestión del acceso mediante portal cautivo, generación de estadísticas, y análisis de comportamiento de los usuarios. La parte de la base de datos es fundamental, ya que es el núcleo que almacena, procesa y facilita los datos necesarios para la operación técnica, análisis de negocio, y cumplimiento normativo.

### **Objetivos de la Base de Datos**

La base de datos en el contexto de WiShield cumple con los siguientes objetivos principales:

- Almacenar y gestionar información sobre usuarios conectados, dispositivos, puntos de acceso, y eventos generados por la infraestructura de red.
- Registrar accesos y estadísticas de navegación para análisis posteriores sobre patrones de comportamiento y uso de la red.
- Soportar la integración con el portal cautivo, almacenando credenciales de acceso y datos de autenticación.
- Facilitar análisis avanzados mediante integración con Power BI, proporcionando insights que permitan mejorar el rendimiento, seguridad y rentabilidad del servicio.
- Garantizar seguridad y cumplimiento legal mediante el registro exhaustivo de actividades y accesos, favoreciendo auditorías y control sobre la privacidad y protección de datos personales.

## **Análisis y Requisitos**

### **Requisitos Funcionales**

- La base de datos debe almacenar información básica de los usuarios (nombre, correo electrónico, número de teléfono, etc.) recopilada en el portal cautivo.
- Debe registrar información sobre dispositivos conectados (dirección MAC, IP asignada, tipo de dispositivo, hora y duración de conexión).
- Necesita almacenar registros detallados de accesos a la red, incluyendo tiempos de sesión, ancho de banda consumido, y sitios o aplicaciones más visitadas.
- Debe facilitar el almacenamiento de datos generados por sistemas de monitorización de seguridad (por ejemplo, intentos de acceso fallidos, alertas de seguridad generadas por herramientas de escaneo como OpenVAS o Nessus).
- Tiene que integrar información relevante para ofrecer estadísticas comerciales, tales como número de usuarios, horas punta de conexión y segmentación por tipo de usuario (habitual vs. invitado).

### Requisitos No Funcionales

- Seguridad: Debe implementar controles estrictos de acceso y auditoría para proteger la privacidad de los datos personales y garantizar su confidencialidad, integridad y disponibilidad.
- Escalabilidad: La estructura debe soportar incrementos en el volumen de datos sin degradación del rendimiento.
- Rendimiento: Debe responder de manera eficiente a consultas frecuentes (reportes, consultas analíticas, visualización en dashboard).
- Disponibilidad: Es crucial que la base de datos esté disponible el mayor tiempo posible, con procedimientos claros para la recuperación rápida en caso de fallo.
- Integridad: Deben aplicarse restricciones y reglas que aseguren la coherencia y exactitud de los datos almacenados.

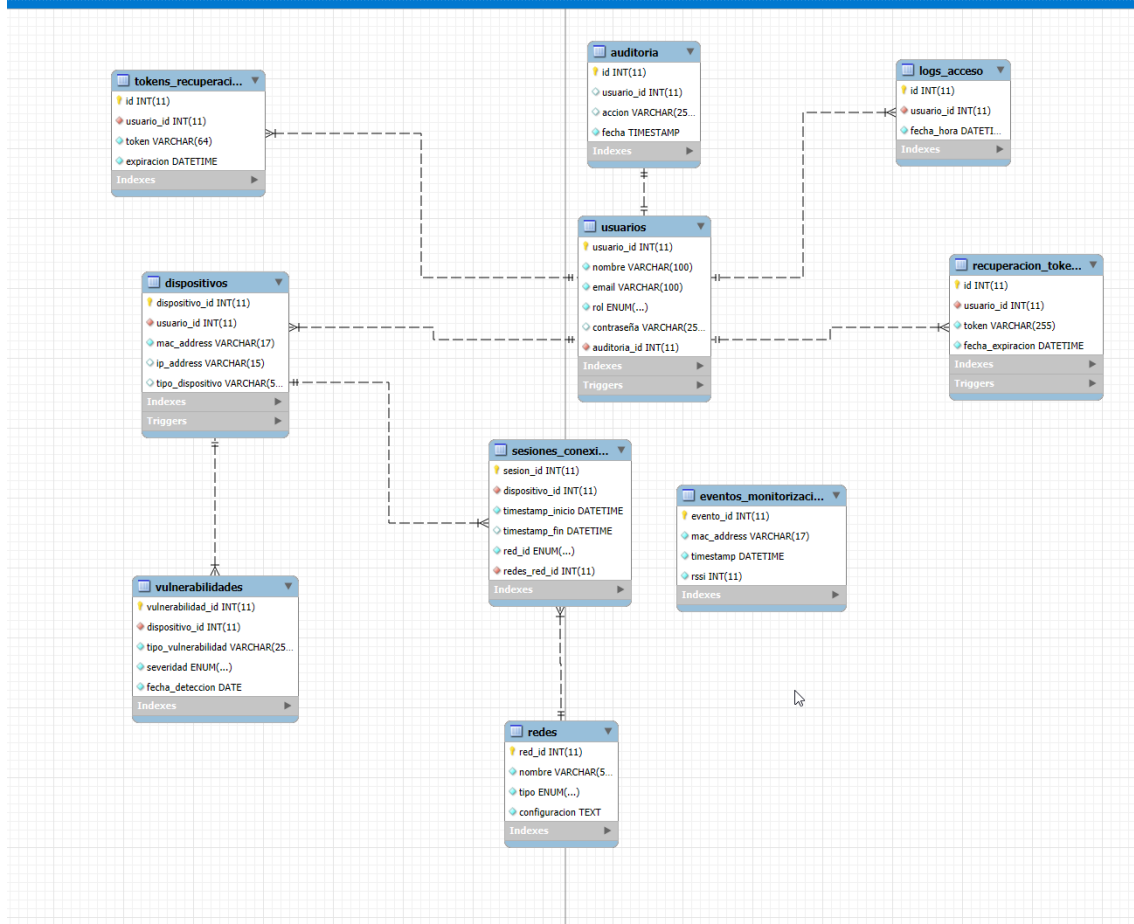
### Casos de Uso Principales

1. **Acceso al Portal Cautivo:** Un usuario nuevo intenta conectarse a la red, rellena sus datos personales en el formulario web, y la información se almacena en la base de datos para futuras autenticaciones y análisis.
2. **Monitorización de Dispositivos:** El sistema registra automáticamente la conexión de dispositivos en tiempo real, almacenando información técnica y temporal para detectar patrones y posibles anomalías.
3. **Generación de Informes de Uso:** El equipo gestor utiliza consultas de la base de datos para generar informes semanales sobre cantidad de usuarios conectados, tiempos medios de sesión, páginas o aplicaciones más consultadas, entre otras métricas clave.
4. **Gestión de Incidentes de Seguridad:** Herramientas automáticas almacenan registros de posibles vulnerabilidades o intentos fallidos de acceso para facilitar la respuesta rápida por parte del equipo de ciberseguridad.

## Modelado de Datos

### Modelo Conceptual:

Durante el diseño del modelo relacional, se ha optado por mantener una estructura normalizada en la que las sesiones (sesiones\_conexion) se vinculan a los dispositivos (dispositivos) mediante dispositivo\_id, y estos a su vez a los usuarios (usuarios) mediante usuario\_id. Esta relación en cascada garantiza integridad referencial y separación lógica de entidades.



### Esquema relacional del sistema WiShield

El sistema de WiShield está diseñado para gestionar la actividad de usuarios y dispositivos conectados a redes Wi-Fi, con un enfoque en la seguridad y la trazabilidad. Este modelo estrella está diseñado para análisis con herramientas de Business Intelligence (como Power BI), permitiendo obtener métricas claves a partir de una tabla de hechos conectada a diversas dimensiones.

**Tabla usuarios:** Es la **tabla principal** del sistema. Contiene la información de los usuarios que acceden a la red WiShield mediante el portal cautivo.

- **usuario\_id:** Clave primaria que identifica a cada usuario.
- **nombre, email:** Información básica del usuario.
- **rol:** Define el tipo de usuario (admin, estudiante, invitado). Determina los permisos dentro del sistema.
- **contraseña:** Encriptada con BCrypt.

**Relacionada con:**



- **dispositivos** (cada usuario puede registrar varios).
- **logs\_acceso**, auditoria (para trazabilidad).
- **recuperacion\_tokens** (gestión de recuperación de cuenta).

**Tabla dispositivos:** Contiene los dispositivos registrados por los usuarios:

- **dispositivo\_id:** Clave primaria.
- **usuario\_id:** Clave foránea a usuarios.
- **mac\_address, ip\_address:** Encriptados con AES.
- **tipo\_dispositivo:** Tablet, Smartphone, Smart TV, etc.
  - **Relacionada con:**
  - **vulnerabilidades:** una o más pueden asociarse a un dispositivo.
  - **sesiones\_conexion:** permite analizar la actividad en red.

**Tabla sesiones\_conexion;** Almacena cada sesión de conexión que inicia un dispositivo contiene los eventos centrales del sistema WiShield: las sesiones activas de los dispositivos. Cada registro representa una sesión individual, con información suficiente para hacer análisis temporales, de red, de usuario y de dispositivos.:

- **sesion\_id:** Identificador de sesión (PK).
- **dispositivo\_id:** FK a dispositivos.
- **timestamp\_inicio / timestamp\_fin:** Marcan la duración.
- **red:** FK a redes.
- Esta tabla actúa como **tabla factual** en el análisis de uso de la red. Permite medir:
  - Uso por dispositivo.
  - Tiempo medio de conexión.
  - Tendencias de uso por día/semana.

**Tabla redes;** Define las redes Wi-Fi activas en el sistema:

- **red\_id:** PK.
- **nombre:** Puede representar el nombre del punto de acceso.
- **tipo,** configuracion: Metainformación técnica.
- Se conecta con sesiones\_conexion para registrar desde dónde se conecta cada usuario.

**Tabla vulnerabilidades:** Se alimenta automáticamente mediante triggers y scripts que simulan detecciones de seguridad.

- **vulnerabilidad\_id:** PK.
- **dispositivo\_id:** FK.
- **tipo\_vulnerabilidad:** Malware, acceso no autorizado, etc.
- **severidad:** Crítica, alta, media, baja.
- **fecha\_deteccion:** Base para análisis mensual.
- Es crucial en el análisis con Power BI para entender la distribución de riesgos.

**Tabla eventos\_monitorizacion:** Generada desde Raspberry Pi mediante Nmap o scripts de escaneo:

- **evento\_id:** PK.

- **mac\_address:** Identifica el dispositivo.
- **timestamp, rssi:** Hora e intensidad de señal (geoposicionamiento indirecto).
- Útil para análisis avanzados o integraciones con IA.

**Tabla logs\_acceso:** Registra los intentos de inicio de sesión al portal web:

- **id:** PK.
- **usuario\_id:** FK.
- **fecha\_hora:** Último acceso.
- Se utiliza para mostrar los últimos accesos en el dashboard y calcular usuarios activos recientes.

**Tabla recuperacion\_tokens:** Tabla auxiliar de seguridad:

- **token:** Enlace único enviado por email al usuario.
- **expiracion:** Permite invalidar tokens automáticamente.
- El trigger `tr_token_reset_cleanup` elimina entradas antiguas cuando se genera un nuevo token.

**Tabla auditoria:** Traza toda la actividad administrativa:

- `usuario_id`, `accion`, `fecha`.
- Guarda acciones como "modificó dispositivo", "eliminó sesión", etc.
- Importante para cumplimiento de políticas de ciberseguridad.

### Conexión con Power BI

Toda esta estructura relacional alimenta el análisis de datos en Power BI, mediante exportaciones automáticas a CSV (por ahora), o bien conexión directa si el entorno lo permite. Los KPIs e informes extraídos incluyen:

- Número de sesiones activas por día o por tipo de red.
- Dispositivos más utilizados y con mayor número de sesiones.
- Vulnerabilidades por severidad, tipo y frecuencia mensual.
- Evolución de usuarios registrados.
- Recuento de roles (admin, estudiante, invitado).
- Últimos accesos al sistema.
- Estos informes permiten monitorizar la red, tomar decisiones de seguridad y planificar mejoras en la infraestructura.

**Justificación del Diseño:** El modelo conceptual se basa en la necesidad de gestionar usuarios, sus dispositivos conectados, sesiones de acceso, eventos de seguridad generados por herramientas como OpenVAS o Nessus, y logs detallados de acceso para análisis estadísticos y de seguridad. Las entidades seleccionadas reflejan claramente los requisitos del proyecto WiShield, asegurando que se cubran aspectos de monitorización, análisis y seguridad.



### Modelo Lógico:

El modelo lógico convierte las entidades del modelo conceptual en tablas relacionales:

El modelo lógico convierte las entidades del modelo conceptual en tablas relacionales:

- **Usuarios** (usuario\_id PK, nombre, email, contraseña, rol, fecha\_creacion)
- **Dispositivos** (dispositivo\_id PK, mac\_address, ip\_address, tipo\_dispositivo, usuario\_id FK)
- **Redes** (red\_id PK, nombre, tipo, configuracion)
- **Sesiones\_Conexion** (sesion\_id PK, dispositivo\_id FK, timestamp\_inicio, timestamp\_fin, red)
- **Eventos\_Monitorizacion** (evento\_id PK, mac\_address, timestamp, rssi)
- **Vulnerabilidades** (vulnerabilidad\_id PK, dispositivo\_id FK, tipo\_vulnerabilidad, severidad, fecha\_deteccion)
- **Logs\_Acceso** (log\_id PK, usuario\_id FK, fecha, hora)
- **Recuperacion\_Tokens** (token\_id PK, usuario\_id FK, token, expira\_en)
- **Auditoria** (id PK, usuario\_id FK, accion, fecha\_hora)

Este modelo está normalizado hasta la tercera forma normal (3FN) para minimizar la redundancia, optimizar el almacenamiento y garantizar la integridad referencial.

### Modelo Físico:

La implementación física se realizará en MySQL, utilizando tipos de datos optimizados (INT, VARCHAR, DATETIME, TEXT), índices específicos sobre columnas frecuentemente consultadas (como claves foráneas, fechas y MACs de dispositivos) para maximizar el rendimiento. No se contemplan particiones inicialmente debido al volumen esperado de datos.

#### Relaciones y restricciones:

- Un usuario puede tener múltiples dispositivos (1:N)
- Un dispositivo puede estar vinculado a múltiples sesiones o eventos (1:N)
- Las IP y MAC están cifradas con AES
- El rol determina el nivel de acceso al sistema
- Las claves primarias son autoincrementadas
- email en usuarios es único
- mac\_address en dispositivos es único
- Se aplican restricciones de unicidad sobre email y MAC, y se establecen reglas de integridad referencial CASCADE para mantener consistencia.

### Índices y Optimización:

Índices en claves primarias, foráneas y columnas de fechaHora e identificadores únicos (email y MAC) para agilizar consultas frecuentes.

## Procedimientos Almacenados, Triggers y Vistas

### **Procedimientos Almacenados y Funciones:**

Un procedimiento almacenado es un bloque de SQL que se guarda en la base de datos y se puede llamar con un simple *CALL nombre\_procedimiento()* tanto desde php como phpMyAdmin.

El código de insertar usuario, por ejemplo, sería el siguiente:

*DELIMITER //*

```
CREATE PROCEDURE sp_insertar_usuario (  
    IN p_nombre VARCHAR(100),  
    IN p_email VARCHAR(100),  
    IN p_contraseña VARCHAR(255),  
    IN p_rol VARCHAR(50),  
    IN p_mac_address VARCHAR(100),  
    IN p_ip_address VARCHAR(100),  
    IN p_tipo_dispositivo VARCHAR(50),  
    IN p_clave_encryption VARCHAR(255)  
)  
BEGIN  
    DECLARE uid INT;  
    INSERT INTO usuarios (nombre, email, contraseña, rol)  
    VALUES (p_nombre, p_email, p_contraseña, p_rol);  
  
    SET uid = LAST_INSERT_ID();  
    INSERT INTO Dispositivos (usuario_id, mac_address, ip_address,  
    tipo_dispositivo)  
    VALUES (  
        uid,  
        AES_ENCRYPT(p_mac_address, p_clave_encryption),  
        AES_ENCRYPT(p_ip_address, p_clave_encryption),  
        p_tipo_dispositivo  
    );  
MySQL  
    SELECT uid AS nuevo_usuario_id;  
END //  
DELIMITER ;
```

Y luego se llamaría desde php con:

```
$res_usuarios = $conexion->query("CALL sp_total_por_rol()");
```

Los procedimientos almacenados realizados son los siguientes:

1. **sp\_insertar\_usuario:** Insertar un nuevo usuario con su dispositivo cifrado.

#### **Parámetros:**

p\_nombre (VARCHAR)

p\_email (VARCHAR)

p\_rol (VARCHAR)  
p\_contraseña (VARCHAR cifrada con BCRYPT)  
p\_mac\_address (VARCHAR, cifrada con AES)  
p\_ip\_address (VARCHAR, cifrada con AES)  
p\_tipo\_dispositivo (VARCHAR)

2. **sp\_total\_vulnerabilidades:** Obtiene un conteo de vulnerabilidades agrupadas por severidad y crea una tabla
3. **sp\_total\_por\_rol;** Cuenta usuarios agrupados por su rol y crea una tabla
4. **sp\_logs\_por\_usuario:** Obtiene el último acceso de cada usuario. Y muestra una tabla con : usuario\_id, nombre, email, ultima\_conexion
5. **sp\_actividad\_por\_fecha:** Muestra la cantidad de sesiones iniciadas por día de la última semana y muestra una tabla con sesiones y fecha.
6. **sp\_dispositivos\_por\_tipo:** Obtiene la cantidad de dispositivos agrupados por tipo y crea una tabla.
7. **sp\_crearSesion**(idDispositivo, idAP): Automático al conectar dispositivo.
8. **sp\_cerrarSesion**(idSesion): Invocado al desconectar el dispositivo.

### Triggers

Para finalizar con este apretado he creado unos triggers que mejoran la seguridad y la automatización de la base de datos sin necesidad de código PHP adicional.

1. **tr\_log\_acceso\_usuario:** Diseñado como base para auditar accesos. Actualmente no realiza acciones adicionales.  
**Evento:** AFTER INSERT en logs\_acceso
2. **tr\_prevent\_admin\_delete:** Previene eliminar un usuario con rol admin. Lanza error SQL si se intenta.  
**Evento:** BEFORE DELETE en usuarios
3. **tr\_auto\_revisar\_vuln:** Inserta vulnerabilidades automáticamente según el tipo de dispositivo:  
Smart TV → "Fuga de datos detectada" (crítica)  
Smartphone → "Intento de acceso no autorizado" (alta)  
**Evento:** AFTER INSERT en Dispositivos
4. **tr\_token\_reset\_cleanup:** Elimina cualquier token anterior existente para el mismo usuario\_id antes de insertar uno nuevo.  
**Evento:** BEFORE INSERT en recuperacion\_tokens
5. **tr\_afterInsertSesion:** Al insertar una sesión, actualiza estado de PuntosDeAcceso.

### Vistas:

- vw\_sesionesActivas: Muestra las sesiones activas en tiempo real.

## Seguridad y Gestión de Usuarios

### Control de Acceso para la base de datos:

- Roles definidos para administradores, analistas y operadores, con permisos específicos de lectura, escritura y gestión de usuarios y logs.

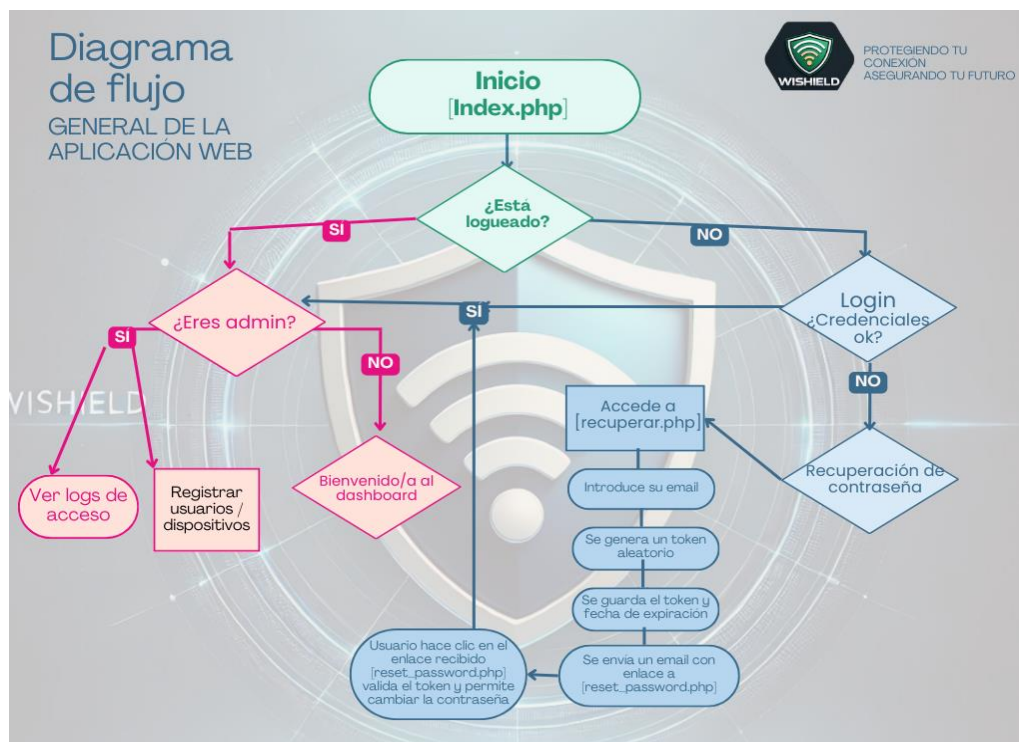
### Scripts:

Estos scripts automatizan tareas clave en el servidor:

- **Acceso\_fallido.bash:** detecta accesos fallidos y envía alertas por correo,
- **Backup\_wishield:** realiza copias de seguridad programadas de la base de datos
- **Restaura\_backups:** permite restaurar backups fácilmente desde la consola
- **Carga\_csv:** carga usuarios desde un archivo CSV
- **EscaneaYmanda.bash:** Utiliza nmap para escanear la red local, detecta dispositivos conectados y los registra automáticamente en la base de datos con IP y MAC cifradas.
- **Envio\_pi.py:** permite a la Raspberry Pi conectarse directamente a la base de datos WiShield e insertar un nuevo dispositivo detectado usando cifrado AES

## Pruebas y Validación

**Modulo de Gestión Web:** Para poder hacer las pruebas de estrés de producción he creado una aplicación web sencilla que permite la gestión de usuarios y sus dispositivos dentro del sistema WiShield, orientado a la monitorización y control de conexiones en entornos residenciales universitarios. Incluye funcionalidades de registro, visualización, búsqueda y dashboard de monitorización. Para ello, habrá que diseñarla antes.

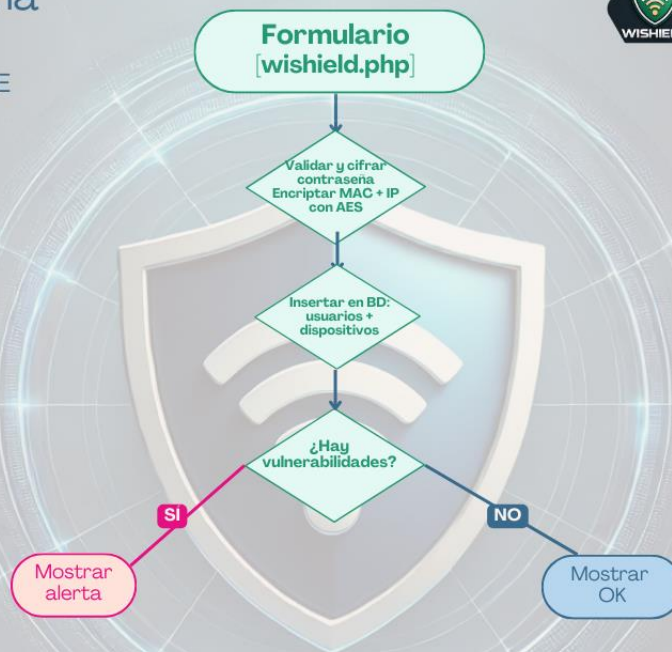


## Diagrama de flujo REGISTRO DE USUARIOS



PROTEGIENDO TU  
CONEXION  
ASEGURANDO TU FUTURO

WISHIELD



## Diagrama de flujo DEL DASHBOARD

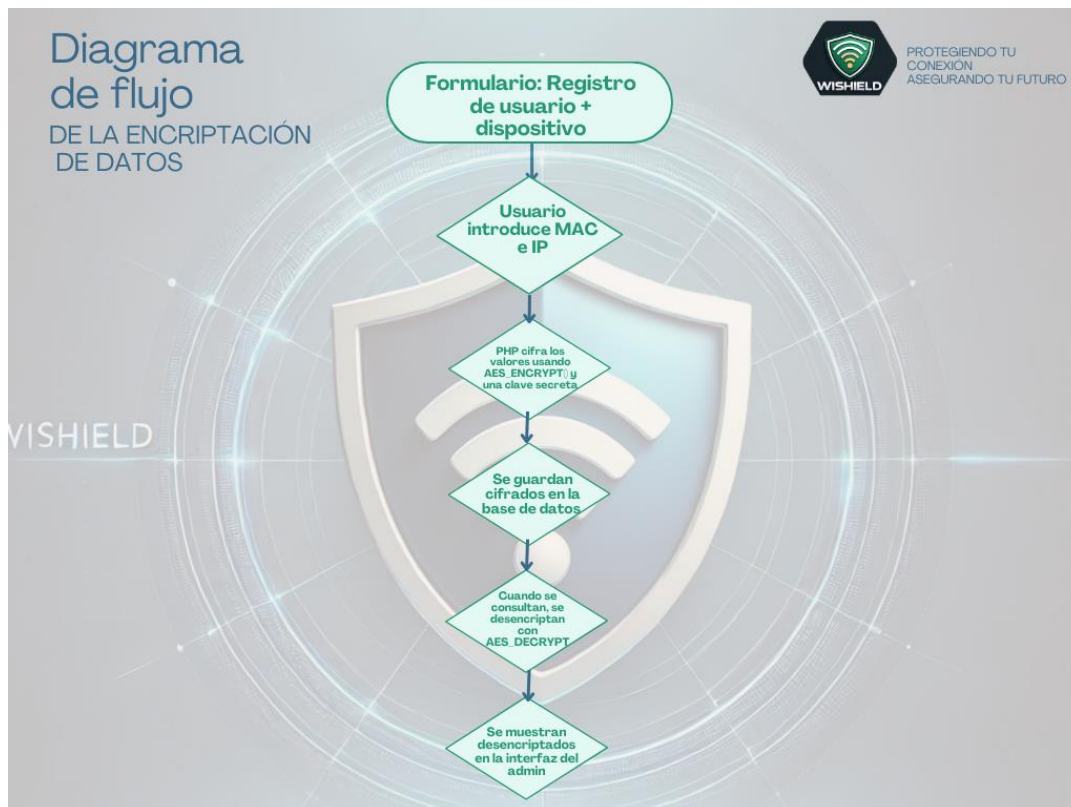


PROTEGIENDO TU  
CONEXION  
ASEGURANDO TU FUTURO

WISHIELD







He empezado subiendo la base de datos de wishield creada con mysqlworkbench a phpMyAdmin y también una carpeta en htdocs de xampp.

usuario_id	nombre	email	rol
1	Juan Pérez	juan.perez@correo.com	estudiante
2	Maria López	maria.lopez@correo.com	invitado
3	Carlos Gutiérrez	carlos.gutierrez@correo.com	admin
4	Ana Torres	ana.torres@correo.com	estudiante
5	Luis Fernández	luis.fernandez@correo.com	invitado
6	Amy Stake	amy.stake@correo.com	invitado
7	Barb Dwyer	barb.dwyer@correo.com	invitado
8	Chris P Bacon	chris.p.bacon@correo.com	invitado
9	Chris P Baker	chris.p.baker@correo.com	invitado
10	Doug Graves	doug.graves@correo.com	invitado
11	Ella Vader	ella.vader@correo.com	invitado
12	Emma Roids	emma.roids@correo.com	invitado
13	Jacqueline Hyde	jacqueline.hyde@correo.com	invitado
14	Jed I Caballero	jed.i.caballero@correo.com	invitado
15	Laura Lynn Hardy	laura.lynn.hardy@correo.com	invitado
16	Lee King	lee.king@correo.com	invitado
17	Ofelia Pane	ofelia.pane@correo.com	invitado
18	Paige Turner	paige.turner@correo.com	invitado
19	Paul Bearer	paul.bearer@correo.com	invitado
20	Philpa Bucket	philpa.bucket@correo.com	invitado
21	Rhoda Wolff	rhoda.wolff@correo.com	invitado
22	Robyn Banks	robyn.banks@correo.com	invitado
23	Sue Flay	sue.flay@correo.com	invitado
24	Sum Ting Wong	sum.ting.wong@correo.com	invitado
25	Teresa Brown	teresa.brown@correo.com	invitado
26	Teresa Crowd	teresa.crowd@correo.com	invitado

Después he abierto Brackets y he creado los archivos de wishield.php (Página de registro y gestión de usuarios y dispositivos) y dashboard.php (Visualización

analítica del sistema en tiempo real).

RegistroDashboardphpMyAdminAdmin Temporal (admin)Cerrar sesión

Buscar por nombre o rol.

Buscar

Agregar nuevo usuario + dispositivo

Nombre:

Email:

actividad@

Contraseña:

Rol:

Invitado

Tipo de dispositivo:

Laptop

MAC Address:

IP Address:

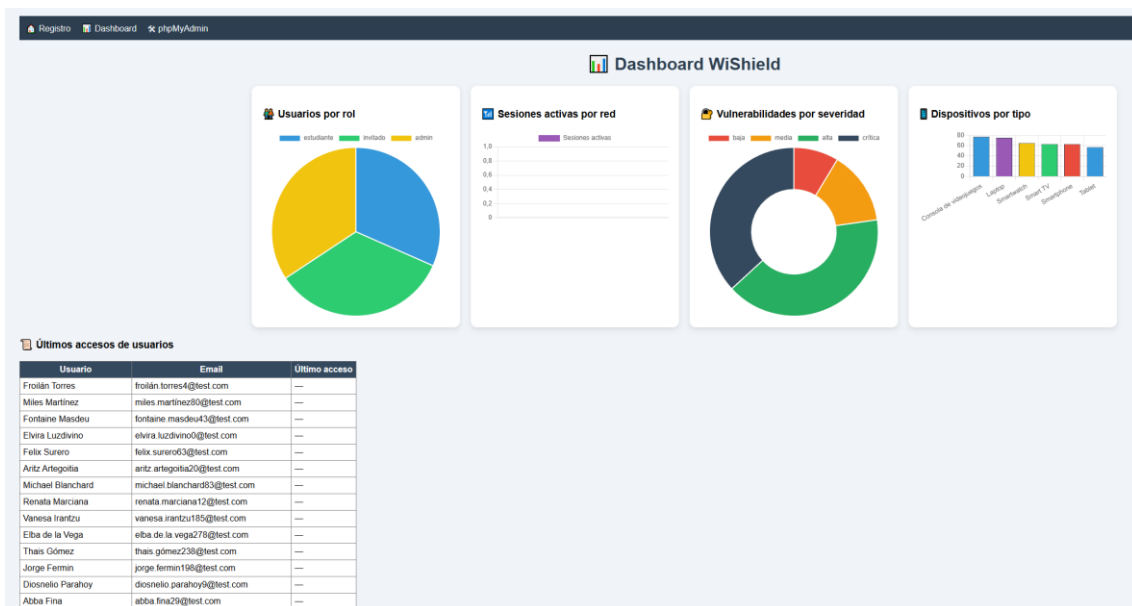
Agregar Usuario

Usuarios registrados y sus dispositivos

Nombre	Email	Rol	Tipo de Dispositivo	MAC Address	IP Address
Hierónides Armandez	hierónides.armandez99@test.com	estudiante	Consola de videojuegos	CC:25:38:6E:C9:DA	192.168.2.164
Lidia Kermit	lidia.kermit98@test.com	admin	Smartphone	25:C4:97:23:0B:A6	192.168.3.69
Masamune Dhu	masamune.dhu97@test.com	admin	Tablet	77:C0:7E:99:4B:A2	192.168.3.178
Protasio Faure	protasio.faure96@test.com	invitado	Consola de videojuegos	34:4A:82:27:67:67	192.168.2.167
Seitaro Bonifacio	seitaro.bonifacio95@test.com	estudiante	Smartphone	F0:C3:25:21:9D:A5	192.168.2.96
Hierónides Gordo	hierónides.gordo94@test.com	invitado	Smartwatch	A2:78:B9:40:5E:7D	192.168.1.145
Estanislada Delano	estanislada.delano93@test.com	admin	Laptop	EE:36:4C:74:81:D0	192.168.3.210
Andrea Mingo	andrea.mingo92@test.com	invitado	Smart TV	7D:1B:26:CD:0F:B7	192.168.3.74
Robert Mihura	robert.mihura91@test.com	estudiante	Consola de videojuegos	D9:D7:58:1D:25:86	192.168.2.188
Shinosuke Zas	shinosuke.zas90@test.com	admin	Consola de videojuegos	B3:E9:00:58:01:10	192.168.3.71
Fulgencia Suzuki	fulgencia.suzuki89@test.com	admin	Consola de videojuegos	8C:45:2F:35:78:FC	192.168.3.168
Akihiro Martínez	akihiro.martínez88@test.com	estudiante	Smartwatch	DB:8F:B5:D3:5D:3D	192.168.1.16
Hiroshi Ishikawa	hiroshi.ishikawa87@test.com	invitado	Tablet	61:1B:2F:5B:A5:88	192.168.1.168
Peter de Covadonga	peter.de.covadonga86@test.com	invitado	Smart TV	95:B7:2C:02:EB:82	192.168.2.210
Renata Jurado	renata.jurado85@test.com	admin	Smartphone	DF:99:55:80:37:60	192.168.2.44
Estanislada Lambert	estanislada.lambert84@test.com	admin	Smartwatch	A2:77:51:02:D8:76	192.168.3.178
Oier de Covadonga	oier.de.covadonga83@test.com	estudiante	Consola de videojuegos	68:AB:85:F1:4F:F8	192.168.3.102
Masaru Yoshida	masaru.yoshida82@test.com	invitado	Smart TV	A0:C3:C8:EB:0D:18	192.168.2.155
Renata Mori	renata.mori81@test.com	admin	Consola de videojuegos	C6:E8:62:B8:3B:42	192.168.3.229
Kagome Minami	kagome.minami80@test.com	invitado	Smart TV	ED:B1:87:8F:32:8B	192.168.2.227

1234567891011121314151617181920Siguiente >





### Tecnologías usadas:

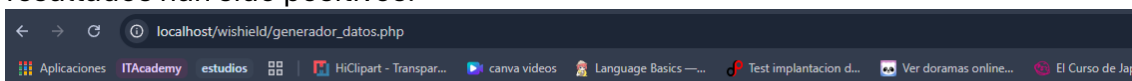
- Frontend: HTML + CSS (estilo simple)
- Backend: PHP con conexión a MySQL (mysqli + prepared statements)
- Visualización: Chart.js (en dashboard.php)
- Gestión de datos: phpMyAdmin

### Pruebas de estrés

Ahora que ya tenemos la aplicación web creada, podemos ponernos manos a la obra y evaluar a capacidad del sistema WiShield para manejar una gran cantidad de registros de usuarios, dispositivos, sesiones de conexión y vulnerabilidades, midiendo la estabilidad, rendimiento y tiempos de respuesta.

Primero he probado a crear usuarios directamente desde la web, sin problema. El dashboard y la tabla reaccionaban bien.

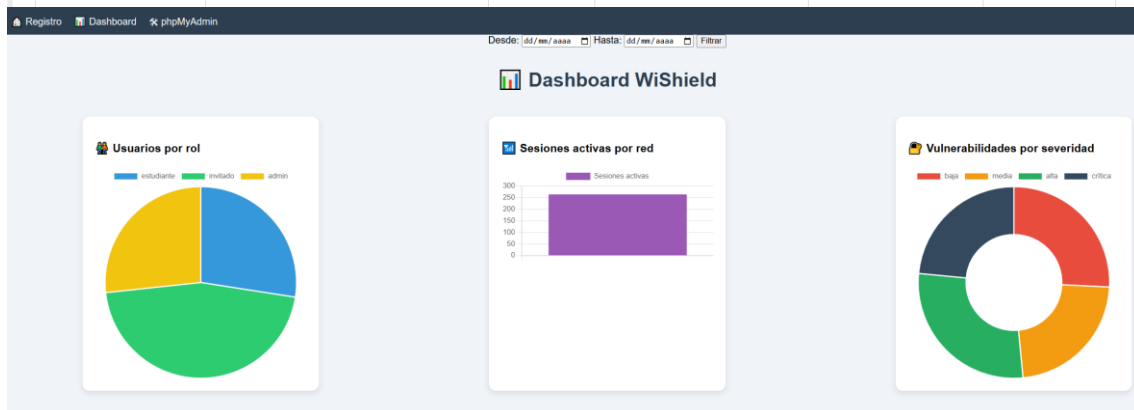
Pero con solo unos pocos usuarios no demostramos su eficacia, así que he generado un script nuevo en php. El generador\_datos.php inserta automáticamente una cantidad configurable de usuarios (100, 500, 1000...) y asocia a cada usuario un dispositivo con MAC e IP aleatorias, una sesión activa o finalizada y la posibilidad de una vulnerabilidad aleatoria (25% de los dispositivos). He hecho una comprobación rápida con un total de 500 usuarios de golpe y los resultados han sido positivos.



✓ **500 usuarios insertados con sus dispositivos, sesiones y vulnerabilidades aleatorias.**

### 📄 Usuarios registrados y sus dispositivos

Nombre	Email	Rol	Tipo de Dispositivo	MAC Address	IP Address
Ana Torres	ana.torres499@test.com	estudiante	Smartphone	A0:B0:BF:94:65:1B	192.168.2.162
Raúl López	raúl.lopez498@test.com	admin	Smart TV	8C:EC:1D:59:C5:93	192.168.2.92
Lucía Torres	lucía.torres497@test.com	estudiante	Tablet	53:78:C2:3B:6D:DC	192.168.1.14
Sergio Gómez	sergio.gómez496@test.com	estudiante	Tablet	03:1A:5D:73:71:41	192.168.3.19
Sergio Gómez	sergio.gómez495@test.com	invitado	Smart TV	B9:2E:EB:EE:F6:F3	192.168.1.161
Sergio Vega	sergio.vega494@test.com	invitado	Smartwatch	DD:05:2A:F9:A2:3E	192.168.1.22
Lucía Torres	lucía.torres493@test.com	invitado	Tablet	2C:18:10:E7:B2:A8	192.168.2.30
Carlos Sánchez	carlos.sánchez492@test.com	estudiante	Smartphone	62:BD:14:01:F1:4A	192.168.3.173
Sergio García	sergio.garcía491@test.com	estudiante	Smart TV	B4:F3:E2:65:70:79	192.168.1.60
Ana García	ana.garcía490@test.com	invitado	Laptop	2A:F4:5F:AF:D2:AA	192.168.1.40
Paula Martínez	paula.martínez489@test.com	estudiante	Consola de videojuegos	CD:26:0E:DD:62:EA	192.168.1.137
Jorge Martínez	jorge.martínez488@test.com	estudiante	Smartphone	1E:6D:7B:D8:42:9A	192.168.2.119
Paula Ruiz	paula.ruiz487@test.com	admin	Smart TV	1B:93:80:F2:C1:AC	192.168.1.203
Valentina Sánchez	valentina.sánchez486@test.com	invitado	Smartphone	1A:47:62:7E:85:E7	192.168.3.93
Valentina Díaz	valentina.díaz485@test.com	admin	Tablet	B4:1D:A5:76:8F:A5	192.168.1.56
David Pérez	david.pérez484@test.com	invitado	Smartphone	92:8A:1F:51:5B:74	192.168.2.139
David Pérez	david.pérez483@test.com	estudiante	Laptop	66:AD:55:97:41:FB	192.168.2.233
Ana Sánchez	ana.sánchez482@test.com	admin	Smartphone	8D:31:77:CA:0A:DC	192.168.1.48



Como se puede ver, la aplicación web ha respondido apropiadamente, aumentando en tiempo real los usuarios y modificando los gráficos sin retrasos perceptibles. Más adelante lo he comprobado con 750 y con mil usuarios nuevos y sigue funcionando.

### Implementación de Encriptación de Contraseñas:

Ahora que ya hemos comprobado que aguanta bien el estrés, vamos a reforzar la seguridad añadiendo contraseñas y encriptándolas con bcrypt.

Empezamos añadiendo a la tabla de usuarios l campo de contraseña:

- ALTER TABLE usuarios ADD contraseña VARCHAR(255);

Luego retocamos el archivo wishield.php:

<label>Contraseña:

<input type="password" name="password" required>

</label>

`$password = password_hash($_POST['password'], PASSWORD_BCRYPT);`

Tambien hemos añadido este código al generador\_datos.php para que genere contraseñas aleatorias

`$csv = fopen("usuarios_generados.csv", "w");`

`fputcsv($csv, ["nombre", "email", "rol", "contraseña_plana"]);`

`for ($i = 0; $i < $TOTAL; $i++) {`

`$nombre = nombreFalso();`

`$email = strtolower(str_replace(' ', '', $nombre)) . $i . '@test.com';`

`$rol = $roles[array_rand($roles)];`

`$pass_plana =`

`substr(str_shuffle('abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'), 0, 8);`

`$pass_hash = password_hash($pass_plana, PASSWORD_BCRYPT);`

`$stmt = $conexion->prepare("INSERT INTO usuarios (nombre, email, rol, contraseña) VALUES (?, ?, ?, ?)");`

`$stmt->bind_param("ssss", $nombre, $email, $rol, $pass_hash);`

`$stmt->execute();`

`$usuario_id = $conexion->insert_id;`

`$stmt->close();`

`fputcsv($csv, [$nombre, $email, $rol, $pass_plana]);`


`}`

`fclose($csv);`


Así conseguimos una base de datos más completa


Opciones extra						
		usuario_id	nombre	email	rol	contraseña
<input type="checkbox"/>	Editar	150	Raúl Sánchez	raul.sanchez49@test.com	admin	\$2y\$10\$ZrIMWdXiqNqP193CGcgCCOnTQW5wluqTzks3ucbyGi...
<input type="checkbox"/>	Editar	1212	Alejandro Armulfo	alejandro.armulfo281@test.com	admin	\$2y\$10\$ZxIPQq75QrKniaS8WmS27u8ZtyW0wEnin4qFyF3Snu...
<input type="checkbox"/>	Editar	1398	Fermin del Bosque	fermin.del.bosque297@test.com	estudiante	\$2y\$10\$ZxhR0XED5IZAcPMdekgtQuYSHqu1yGYSVTMa5Ygia0b...
<input type="checkbox"/>	Editar	1238	Mercé Sánchez	merce.sanchez294@test.com	estudiante	\$2y\$10\$ZvylLuS.g2Y0BfoLjHrHuPLVeBAZsdgDfjqT4ByaP...
<input type="checkbox"/>	Editar	1160	Georgina Pieplano	georgina.pieplano255@test.com	admin	\$2y\$10\$ZVeBKRdegQd8.VPjYyYnzuYzJy7IP1aKCA9C2njclLR...
<input type="checkbox"/>	Editar	1152	Covadonga Aritz	covadonga.aritz251@test.com	estudiante	\$2y\$10\$ZughT0Gn0Juijyan2C4QkUthWnqR8P/QI1aRGvGPhn...
<input type="checkbox"/>	Editar	134	Rúben Raven	ruben.raven33@test.com	invitado	\$2y\$10\$ZrOjE4Lnud/H.lqZU1an6ebigXhbkG6.joj.kR.9Ja...
<input type="checkbox"/>	Editar	1036	Montse de la Repolla	montse.de.la.repolla193@test.com	estudiante	\$2y\$10\$Zr2HmYNQI2c01b.h4y7keUpI/G7xXL.jv8JL0hNcNXE...
<input type="checkbox"/>	Editar	1037	Abba Alcoholado	abba.alcoholado42@test.com	estudiante	\$2y\$10\$Zq0bi7jET7PqrGoPynLuOLQfscCmTFNsH1ehSZAe...
<input type="checkbox"/>	Editar	182	Rosa Aine	rosa.aine81@test.com	estudiante	\$2y\$10\$ZPuGAnkbM.plhV1fAfHnCeasjxum00JnFmMpkpHE5P...
<input type="checkbox"/>	Editar	1043	Duncan Leona	duncan.leona45@test.com	invitado	\$2y\$10\$ZPngVx43bgnJMuJ2jg70OfauF8Io08AZ15g7EDJCe...
<input type="checkbox"/>	Editar	1095	Drac Hercules	drac.hercules71@test.com	admin	\$2y\$10\$ZmnuXYa7RWIO9ChX530kMoygKPsUZWCUksEV7aFDbXc...
<input type="checkbox"/>	Editar	1108	Thais Patel	thais.patel229@test.com	admin	\$2y\$10\$Zj1gIX9N6NjgSncMUuVeuSY6kO4I6Aht4CI0Ia1Xa...
<input type="checkbox"/>	Editar	1218	Fermin de la Vega	fermin.de.la.vega284@test.com	invitado	\$2y\$10\$ZHzRaRaAmF63NhX08ikEeldT9MIEcb8U21cU2XlktR...
<input type="checkbox"/>	Editar	599	Miles Martínez	miles.martinez80@test.com	invitado	\$2y\$10\$ZHykv9HuVAHEiWRstFY106pvSOpn9KAUJIACsXwM0C...
<input type="checkbox"/>	Editar	1130	Irantzu Mogollón	irantzu.mogollon240@test.com	invitado	\$2y\$10\$ZHGOK7SMYmIvSbxFGtQo.jexmuTk1Y.p0SqtK2Jn1...
<input type="checkbox"/>	Editar	806	Froilana Eiba	froilana.elba121@test.com	admin	\$2y\$10\$Zc20cLYNoKEibMRf5g00PX5xCdICCHR.vETdu1qG...
<input type="checkbox"/>	Editar	1190	Eire Valentino	eire.valentino270@test.com	admin	\$2y\$10\$ZbWkZWZWWvUjCE0ZNeKzBxcRGea8IWTZYCTXqbM...
<input type="checkbox"/>	Editar	518	Arnulfo Sandro	arnulfo.sandro17@test.com	invitado	\$2y\$10\$Zam227G2hT8VzGZGNuFVmOCau2rsGaF4.Nh5UKRMPVd...
<input type="checkbox"/>	Editar	1223	Leiona Diogenes	leiona.diogenes135@test.com	admin	\$2y\$10\$ZAdiqmimU8stHX9VPP5eE2.3uTocAZQbCrsUX8ZrK1m...
<input type="checkbox"/>	Editar	1256	Rúben Japón	ruben.japon155@test.com	invitado	\$2y\$10\$Z80T92cCoY3X3g96lDlqluVjOeZjcaYim.adMkSWbC/...
<input type="checkbox"/>	Editar	959	Ben Aritz	ben.aritz3@test.com	invitado	\$2y\$10\$Z6RILG6GOiUsiAHg8g3C0ci0chrcRkTQ8fPrp6FioL...
<input type="checkbox"/>	Editar	1195	Argi Botelli	argi.botelli121@test.com	estudiante	\$2y\$10\$Z3EE/8Al.FtggSuN.6k.J.GZE9nIEKHhWOTI4WoFT...
<input type="checkbox"/>	Editar	1270	Louis Paramí	louis.parami169@test.com	invitado	\$2y\$10\$Z2iF56BcPapf6CdylafmeDoadf88HL746PK0Nh1ilr...
<input type="checkbox"/>	Editar	989	Brian García	brian.garcia18@test.com	estudiante	\$2y\$10\$Zr1A/vI.vvtS7SKv6fDfCIPMwvSTWVFTmsIY.IeiuRwRCV...

Y para comprobar que funciona, creamos el archivo login.php

 **Login WiShield**

Comprobación realizada

 **Isabel Becerra (estudiante)**

 [Cerrar sesión](#)

Vamos a aprovechar las modificaciones para añadir un control de acceso solo para administradores en la parte de la tabla de usuarios.

## Recuperación de contraseña

He incluido una funcionalidad segura de recuperación de contraseñas mediante el envío de enlaces personalizados por correo electrónico. Esta funcionalidad se ha implementado con **PHPMailer sin Composer** y consta de tres fases principales:

### 1. Solicitud de recuperación (recuperar.php)

El usuario introduce su email. Si se encuentra en la base de datos, se genera un **token aleatorio de 64 caracteres** con una expiración de 1 hora. Este token se guarda en una tabla llamada `tokens_recuperacion` junto al `usuario_id`.

### 2. Envío de correo (enviar\_token.php)

Se utiliza la biblioteca **PHPMailer** para enviar un correo a la dirección del usuario. El email contiene un enlace que apunta a `reset_password.php` con el token como parámetro (`?token=abc123...`). Este enlace solo es válido mientras el token no haya expirado.

### 3. Restablecimiento de contraseña (reset\_password.php)



Cuando el usuario accede al enlace, se valida el token. Si es válido y no ha caducado, se muestra un formulario para establecer una nueva contraseña. Esta contraseña se **cifra usando `password_hash()` con `bcrypt`** y se actualiza en la base de datos. Finalmente, el token es eliminado para evitar reutilizaciones.

## Seguridad

- Los tokens están limitados por tiempo (expiran en 1 hora)
- No se muestra si el email existe (protege contra enumeration)
- Las contraseñas nunca se envían por correo ni se almacenan sin cifrar
- Cada token es único y se invalida tras su uso

## Encriptación de datos sensibles

Para proteger los datos sensibles de los usuarios, se ha implementado un mecanismo de **cifrado simétrico** usando la función `AES_ENCRYPT()` de MySQL. Esta se utiliza específicamente en los campos de la tabla `Dispositivos` que contienen:

-  Dirección MAC (`mac_address`)
-  Dirección IP (`ip_address`)

El cifrado y descifrado se realiza mediante una **clave secreta definida en el archivo `config.php`** del sistema. Esta clave debe mantenerse en secreto y no debe compartirse ni subirse a ningún repositorio público. Es imprescindible que esta clave **permanezca constante** para poder descifrar los datos correctamente. En el momento del registro de un usuario, los campos se cifran con la siguiente instrucción SQL:

```
AES_ENCRYPT(valor_original, 'CLAVE_SECRETA')
```

Esto se aplica al registrar el dispositivo con:

```
$sql = "INSERT INTO Dispositivos (usuario_id, mac_address, ip_address,
tipo_dispositivo)
VALUES (?, AES_ENCRYPT(?, ?), AES_ENCRYPT(?, ?), ?)";
```

### Desencriptado al consultar

Cuando se muestra la tabla de dispositivos en wishield.php, los campos se recuperan con:

```
CAST(AES_DECRYPT(mac_address, 'CLAVE_SECRETA') AS CHAR)
```

Esto permite visualizar los valores originales sin modificar la lógica del resto del sistema. El desencriptado se realiza en las consultas SQL directamente.

### Bloquear accesos desde IPs no autorizadas

Vamos a añadir un fragmento de código extra para evitar que accedan a la aplicación web las IP que no estén en nuestra lista. Muestra un mensaje y corta la ejecución.

Copiaré este fragmento al inicio de archivos sensibles (wishield.php, dashboard.php):

```
<?php
$ips_autorizadas = ['127.0.0.1', '::1', '192.168.1.100']; // Agregamos aquí las
IPs que se permiten
if (!in_array($_SERVER['REMOTE_ADDR'], $ips_autorizadas)) {
    header("HTTP/1.1 403 Forbidden");
    echo "🚫 Acceso no autorizado desde " . $_SERVER['REMOTE_ADDR'];
    exit;
}
?>
```

### Herramientas y Tecnologías Utilizadas

- MySQL Workbench
- Brackets
- phpMyAdmin

## Conexión con la Raspberry Pi

La Raspberry Pi está integrada en el sistema WiShield como nodo de recolección, análisis y mantenimiento auxiliar. Se conecta directamente a la base de datos MySQL del sistema para insertar, consultar y actualizar información de sesiones, dispositivos y eventos de red.

La conexión se realiza mediante scripts programados en Python y Bash, con usuarios que tienen permisos limitados para mantener la seguridad del sistema. Esta arquitectura distribuida permite que la Raspberry Pi actúe como punto de monitoreo dentro de la red de la Vila Universitària.

**envio\_pi.py:** Este script fue creado para enviar periódicamente sesiones simuladas o reales desde la Raspberry Pi a la base de datos de WiShield. Es el núcleo de la integración entre la Pi y el backend del sistema.

- Se conecta a la base de datos wishield usando mysql.connector.
- Inserta registros en la tabla sesiones\_conexion simulando:
  - El inicio y fin de una sesión.
  - El identificador del dispositivo conectado.
  - El identificador de la red utilizada.
- Puede ejecutarse manualmente o de forma automatizada desde crontab.

**Scripts Bash utilizados en Raspberry Pi:** Además de envio\_pi.py, se han desarrollado varios scripts en Bash que ayudan a mantener los datos actualizados y seguros dentro de la Raspberry Pi:

- **acceso\_fallido.bash:** Detecta intentos fallidos de conexión a MySQL y envía una alerta por correo al administrador.
- **backup\_wishield.bash:** Genera backups automáticos de la base de datos wishield, los guarda con fecha y borra los más antiguos de 7 días.
- **carga\_csv.bash:** Importa archivos CSV desde /home/pi/datos/ directamente a la base de datos.
- **restaura\_backup.bash:** Permite restaurar manualmente un backup .sql o .gz en caso de fallo o corrupción de datos.
- **escaneaYmanda.bash**
  - Usa nmap para escanear dispositivos conectados a la red.
  - Extrae IPs y MACs.
  - Inserta los datos cifrados con AES en la tabla Dispositivos.
  - Automatizable desde crontab para escaneo horario.

## Conexión a PowerBI

Power BI se utiliza en el proyecto WiShield para analizar y visualizar grandes volúmenes de datos generados por el portal web y la base de datos MySQL. Permite transformar los registros de sesiones, vulnerabilidades, usuarios y dispositivos en gráficos e indicadores clave que aportan valor estratégico.



Los datos exportados desde la base de datos MySQL incluyen:

- Usuarios registrados y roles
- Dispositivos conectados (tipo, uso)
- Sesiones iniciadas por día
- Vulnerabilidades detectadas (por tipo y severidad)

En Power BI, se prepararon los datos (formateo de fechas, unión de tablas por claves foráneas, columnas derivadas como mes o semana, etc.).

Y finalment se crearon visuales interactivos como:

1. Recuento de usuarios por rol
2. Dispositivos más comunes
3. Evolución de sesiones diarias
4. Vulnerabilidades por severidad y tipo
5. Indicadores dinámicos: nuevos usuarios este mes, sesiones activas, vulnerabilidades detectadas

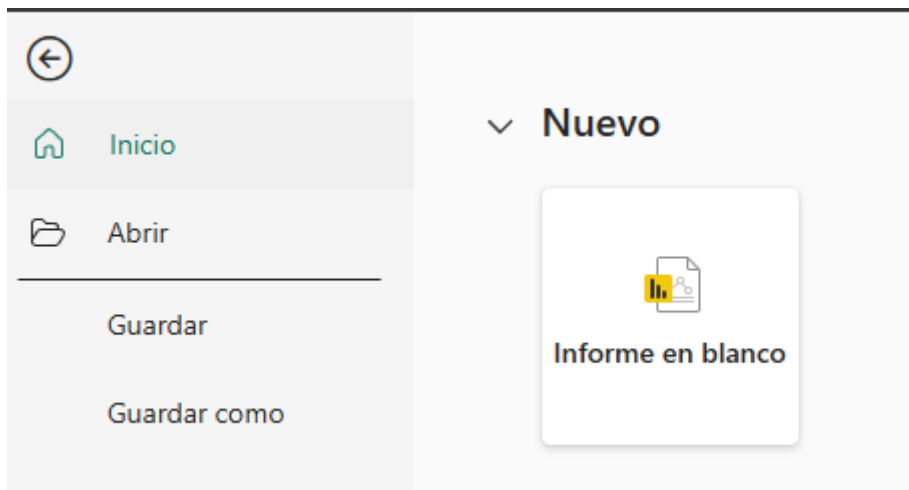
Power BI actúa como una capa de análisis superior sobre la base de datos gestionada por el portal web WiShield. Las acciones realizadas por los usuarios y administradores generan datos en MySQL que luego se analizan para apoyar la toma de decisiones.

Por ejemplo:

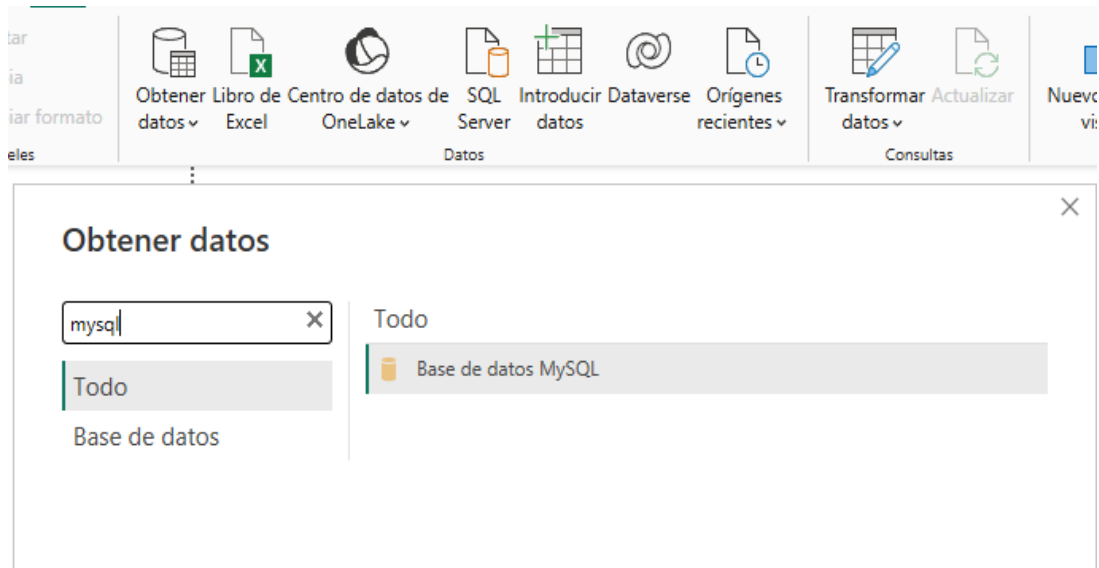
- El administrador puede identificar rápidamente los días con más sesiones o detectar picos de vulnerabilidades.
- Se pueden detectar patrones de comportamiento según los roles (invitado, estudiante, admin).
- El rendimiento de las redes y dispositivos se puede evaluar en tiempo real o con análisis históricos.

### **Como conectar de MySQL a PowerBI**

Para importar los datos primero hemos de instalar, si no está hecho ya, un conector como este: <https://dev.mysql.com/downloads/file/?id=534712>. Posteriormente, abrimos PowerBi y le pedimos que cree un nuevo informe.



Para cargar los datos vamos a elegir Obtener datos, ya que es desde donde se puede cargar directamente desde MySQL. Tras seleccionar *Base de datos MySQL*, le damos al botón *Conectar*.



Nos aparecerá otra ventana donde tenemos que poner nuestro servidor (por defecto es el 3306) y la base de datos. Le damos a *Aceptar*.



Tras esta ventana, aparecerá otra, a la que solo tenemos que darle a *Conectar*.

Base de datos MySQL

localhost:3308;wishield

Use sus credenciales de Windows para obtener acceso a esta base de datos.

☒ Usar mis credenciales actuales  
☐ Usar credenciales alternativas

Nombre de usuario

Contraseña

Seleccionar en qué nivel hay que aplicar esta configuración

localhost:3308

Atrás Conectar Cancelar

Si no hay fallos ya nos aparecerá otra ventana donde seleccionamos que tablas se quieren cargar y tras esa, ya podemos empezar a trabajar.

## Ejemplo de informe



Durante el mes de abril, se han monitorizado e integrado los datos recopilados por la plataforma WiShield, permitiendo obtener una visión clara de la actividad de red, dispositivos conectados, sesiones activas, creación de usuarios y estado de la ciberseguridad dentro del complejo residencial de la Vila Universitària.

### Estado de la Seguridad

- Se han detectado 266 vulnerabilidades en total, destacando como principales amenazas intentos de acceso no autorizado y fugas de datos
- La severidad predominante ha sido crítica y alta, lo cual requiere atención continuada y ajustes en las políticas de acceso de red.
- Los triggers de monitorización automática y el sistema de logging implementado han sido clave para detectar estas amenazas en tiempo real.

### Actividad de Usuarios

- 1763 usuarios registrados en el sistema, con una distribución equilibrada entre roles de administrador, estudiante e invitado.
- Se han registrado casi 500 sesiones iniciadas solo en abril, lo que muestra un uso activo de la red por parte de la comunidad.
- Este mes se ha notado un ligero incremento en la creación de cuentas, especialmente en el rol "invitado", lo cual coincide con el periodo de puertas abiertas.

### Dispositivos y Conexiones

- Los dispositivos más comunes han sido consolas, laptops y smartwatches, demostrando una alta variedad tecnológica entre los usuarios.
- En cuanto a sesiones, el top 5 de dispositivos más activos se asocia principalmente a usuarios del rol estudiante, lo cual refuerza la necesidad de mantener balance entre flexibilidad de acceso y control.
- Estos datos alimentan nuestra estrategia de segmentación por tipo de dispositivo para aplicar controles más granulares.

### Conclusiones y Recomendaciones

- La red se está utilizando intensamente, lo cual es positivo, pero también incrementa la superficie de exposición a amenazas.
- Recomendamos reforzar la política de detección de anomalías, así como realizar campañas de concienciación en ciberseguridad para los nuevos usuarios.
- En futuras versiones, se propondrá una segmentación por zonas y una red de invitados más restrictiva.

## Scripts

### Bash

*Acceso\_fallido.bash:*

```
#!/bin/bash
# Nombre: acceso_fallido.sh
# Descripción: detecta intentos fallidos y avisa
LOG_FILE="/var/log/mysql/error.log"
ADMIN_EMAIL="admin@wishield.com"
ATTEMPTS=$(grep "Access denied" $LOG_FILE | tail -n 10)
if [ ! -z "$ATTEMPTS" ]; then
    echo " ⚠ Se han detectado intentos de acceso fallidos en la base de datos:"
    echo "$ATTEMPTS"
    echo -e "Asunto: ⚠ Alerta de Acceso Fallido en MySQL\n" \
        "Se han detectado intentos de acceso fallidos en la base de datos.\n\n" \
        "$ATTEMPTS" | sendmail -v "$ADMIN_EMAIL"
    echo " ⚠ Se ha enviado una alerta a $ADMIN_EMAIL con los intentos de acceso fallidos."
fi
```

*Backup\_wishield.bash:*

```
#!/bin/bash
# Script: backup_wishield.sh
# Descripción: Realiza un backup automático de la base de datos WiShield y lo almacena con fecha.
DB_NAME="wishield"
BACKUP_DIR="/backups"
DATE=$(date +"%Y-%m-%d_%H-%M-%S")
BACKUP_FILE="$BACKUP_DIR/wishield_backup_$DATE.sql"
MYSQL_USER="root"
MYSQL_PASSWORD="root"
mkdir -p $BACKUP_DIR
mysqldump -u$MYSQL_USER -p$MYSQL_PASSWORD $DB_NAME > $BACKUP_FILE
if [ $? -eq 0 ]; then
    echo "Backup realizado con éxito: $BACKUP_FILE"
else
    echo "Error en el backup." >&2
    exit 1
fi
find $BACKUP_DIR -type f -name "wishield_backup_*.sql" -mtime +7 -exec rm {} \;
exit 0
# Para que se ejecute en crontab crontab -e
# 0 2 * * * /home/pi/scripts/backup_proyecto.sh
```

### *Carga\_csv.bash*

```
#!/bin/bash
# Variables
DB_NAME="wishield"
DB_USER="root"
DB_PASSWORD="root"
CSV_FILE="/home/pi/datos/datos_prueba.csv"
TABLE_NAME="usuarios"
# Importar datos
mysql -u$DB_USER -p$DB_PASSWORD -e "
LOAD DATA INFILE '$CSV_FILE'
INTO TABLE $DB_NAME.$TABLE_NAME
FIELDS TERMINATED BY ';'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS;"
echo "Carga de datos completada desde $CSV_FILE"
```

### *restaura\_backup.bash*

```
#!/bin/bash
# nombre: restaurar_backup.sh
#Descripción: restaura un backup especifico
DB_NAME="wishield"
DB_USER="root"
DB_PASSWORD="root"
BACKUP_FILE="$1"
if [ -z "$BACKUP_FILE" ]; then
    echo "Uso: $0 <archivo_backup.sql.gz>"
    exit 1
fi
if [[ $BACKUP_FILE == *.gz ]]; then
    gunzip -c $BACKUP_FILE | mysql -u$DB_USER -p$DB_PASSWORD $DB_NAME
else
    mysql -u$DB_USER -p$DB_PASSWORD $DB_NAME < $BACKUP_FILE
fi
echo "Restauración completada desde $BACKUP_FILE"
#para usarlo ./restore_backup.sh /home/pi/backups/proyecto-2024-03-19_02-00-00.sql.gz
```

### *escaneaYmanda.bash*

```
#hemos de tener nmap en la raspberry pi
#!/bin/bash
# Configuración
SUBNET="192.168.1.0/24"
MYSQL_USER="root"
MYSQL_PASS="root"
DB="wishield"
```



```

CLAVE="TuClaveAES123"
# Escaneo con Nmap
echo "🔍 Escaneando red..."
RESULTS=$(nmap -sn $SUBNET | awk '/Nmap scan report/{ip=$NF}/MAC
Address:/{print ip,$3}' | sed 's/[()]/g')
# Insertar resultados
for line in $RESULTS; do
    IP=$(echo $line | awk '{print $1}')
    MAC=$(echo $line | awk '{print $2}')
    echo "📁 Registrando IP: $IP | MAC: $MAC"
    mysql -u$MYSQL_USER -p$MYSQL_PASS $DB -e "
        INSERT INTO Dispositivos (usuario_id, mac_address, ip_address,
        tipo_dispositivo)
        VALUES (1, AES_ENCRYPT('$MAC', '$CLAVE'), AES_ENCRYPT('$IP', '$CLAVE'),
        'Detectado por Pi');
    "
done
echo "✅ Escaneo completado e insertado."
#para crontab: crontab -e
# 0 * * * * /home/pi/scripts/scan_and_send.sh >> /var/log/wishield_scan.log 2>&1

```

## Python

### *Envio\_pi.py*

```

# Envía datos de conexión de dispositivos desde la Raspberry Pi a la base de datos
de WiShield
import mysql.connector
from datetime import datetime
import time
import random
# Configuración de conexión
config = {
    'host': 'localhost',      # Cambiar si la BD está en otra máquina
    'user': 'pi_user',       # Usuario con permisos LIMITADOS
    'password': 'clave_pi_segura',
    'database': 'wishield'
}
# Simular dispositivos conectados
dispositivos = [801, 802, 803, 804]
while True:
    try:
        conn = mysql.connector.connect(**config)
        cursor = conn.cursor()

        dispositivo_id = random.choice(dispositivos)
        inicio = datetime.now()
        fin = inicio # Para la simulación, sesión instantánea

```

```

        red_id = random.randint(1, 3) # Simulamos que hay 3 redes configuradas
        sql = """
        INSERT INTO sesiones_conexion (dispositivo_id, timestamp_inicio,
timestamp_fin, red_id)
        VALUES (%s, %s, %s, %s)
        """

        cursor.execute(sql, (dispositivo_id, inicio, fin, red_id))
        conn.commit()
        print(f"✅ Sesión insertada: Dispositivo {dispositivo_id}, Red {red_id}, {inicio}")

        cursor.close()
        conn.close()
    except mysql.connector.Error as e:
        print(f"❌ Error al insertar sesión: {e}")
    time.sleep(10) # Esperar 10 segundos antes de enviar otro dato

```

## SQL

*Wishield.sql*

```
CREATE SCHEMA wishield;
```

-- Creamos las tablas

```
CREATE TABLE `usuarios` (
  `usuario_id` int(11) NOT NULL,
  `nombre` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `rol` enum('estudiante','invitado','admin') NOT NULL,
  `contraseña` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `dispositivos` (
  `dispositivo_id` int(11) NOT NULL,
  `usuario_id` int(11) NOT NULL,
  `mac_address` varchar(17) NOT NULL,
  `ip_address` varchar(15) DEFAULT NULL,
  `tipo_dispositivo` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `eventos_monitorizacion` (
  `evento_id` int(11) NOT NULL,
  `mac_address` varchar(17) NOT NULL,
  `timestamp` datetime NOT NULL,
  `rssi` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;
```

```
CREATE TABLE `vulnerabilidades` (
```

```

`vulnerabilidad_id` int(11) NOT NULL,
`dispositivo_id` int(11) NOT NULL,
`tipo_vulnerabilidad` varchar(255) NOT NULL,
`severidad` enum('baja','media','alta','crítica') NOT NULL,
`fecha_deteccion` date NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE `logs_acceso` (
  `id` int(11) NOT NULL,
  `usuario_id` int(11) NOT NULL,
  `fecha_hora` datetime NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE `recuperacion_tokens` (
  `id` int(11) NOT NULL,
  `usuario_id` int(11) NOT NULL,
  `token` varchar(255) NOT NULL,
  `fecha_expiracion` datetime NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE `redes` (
  `red_id` int(11) NOT NULL,
  `nombre` varchar(50) NOT NULL,
  `tipo` enum('segura','pública','administrativa') NOT NULL,
  `configuracion` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE `sesiones_conexion` (
  `sesion_id` int(11) NOT NULL,
  `dispositivo_id` int(11) NOT NULL,
  `timestamp_inicio` datetime NOT NULL,
  `timestamp_fin` datetime DEFAULT NULL,
  `red` enum('administrativa','estudiantes','invitados') NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

```

CREATE TABLE `auditoria` (
  `id` int(11) NOT NULL,
  `usuario_id` int(11) DEFAULT NULL,
  `accion` varchar(255) DEFAULT NULL,
  `fecha` timestamp NOT NULL DEFAULT current_timestamp()
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

```

-- Disparadores o Triggers

DELIMITER \$\$

```

CREATE TRIGGER `tr_prevent_admin_delete` BEFORE DELETE ON `usuarios`
FOR EACH ROW BEGIN

```

```

    IF OLD.rol = 'admin' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = '🚫 No se puede eliminar un usuario con rol de
administrador.';
    END IF;
END
$$
DELIMITER ;

DELIMITER $$
CREATE TRIGGER `tr_token_reset_cleanup` BEFORE INSERT ON
`recuperacion_tokens` FOR EACH ROW BEGIN
    DELETE FROM recuperacion_tokens
    WHERE usuario_id = NEW.usuario_id;
END
$$
DELIMITER ;

DELIMITER $$
CREATE TRIGGER `tr_auto_revisar_vuln` AFTER INSERT ON `dispositivos` FOR
EACH ROW BEGIN
    IF NEW.tipo_dispositivo = 'Smart TV' THEN
        INSERT INTO Vulnerabilidades (dispositivo_id, tipo_vulnerabilidad, severidad,
fecha_deteccion)
        VALUES (NEW.dispositivo_id, 'Fuga de datos detectada', 'crítica', CURDATE());
    ELSEIF NEW.tipo_dispositivo = 'Smartphone' THEN
        INSERT INTO Vulnerabilidades (dispositivo_id, tipo_vulnerabilidad, severidad,
fecha_deteccion)
        VALUES (NEW.dispositivo_id, 'Intento de acceso no autorizado', 'alta',
CURDATE());
    END IF;
END
$$
DELIMITER ;

-- Procedimientos
DELIMITER $$
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_actividad_por_fecha` ()
BEGIN
    SELECT DATE(timestamp_inicio) AS fecha, COUNT(*) AS sesiones
    FROM Sesiones_Conexion
    WHERE timestamp_inicio >= CURDATE() - INTERVAL 7 DAY
    GROUP BY DATE(timestamp_inicio)
    ORDER BY fecha ASC;
END$$

```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_dispositivos_por_tipo`
() BEGIN
    SELECT tipo_dispositivo, COUNT(*) AS total
    FROM Dispositivos
    GROUP BY tipo_dispositivo
    ORDER BY total DESC;
END$$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_insertar_usuario` (IN
`p_nombre` VARCHAR(100), IN `p_email` VARCHAR(100), IN `p_contraseña`
VARCHAR(255), IN `p_rol` VARCHAR(50), IN `p_mac_address` VARCHAR(100), IN
`p_ip_address` VARCHAR(100), IN `p_tipo_dispositivo` VARCHAR(50), IN
`p_clave_encryptacion` VARCHAR(255)) BEGIN
    DECLARE uid INT;
    INSERT INTO usuarios (nombre, email, contraseña, rol)
    VALUES (p_nombre, p_email, p_contraseña, p_rol);
    SET uid = LAST_INSERT_ID();
    INSERT INTO Dispositivos (usuario_id, mac_address, ip_address,
tipo_dispositivo)
    VALUES (
        uid,
        AES_ENCRYPT(p_mac_address, p_clave_encryptacion),
        AES_ENCRYPT(p_ip_address, p_clave_encryptacion),
        p_tipo_dispositivo
    );
    SELECT uid AS nuevo_usuario_id;
END$$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_logs_por_usuario` ()
BEGIN
    SELECT u.usuario_id, u.nombre, u.email, MAX(l.fecha_hora) AS ultima_conexion
    FROM usuarios u
    LEFT JOIN logs_acceso l ON u.usuario_id = l.usuario_id
    GROUP BY u.usuario_id, u.nombre, u.email
    ORDER BY ultima_conexion DESC;
END$$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_sesiones_activas` ()
BEGIN
    SELECT COUNT(*) AS total_activas
    FROM Sesiones_Conexion
    WHERE timestamp_fin IS NULL;
END$$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_total_por_rol` () BEGIN
    SELECT rol, COUNT(*) AS total
    FROM usuarios
```

```
GROUP BY rol;
END$$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_total_vulnerabilidades`
() BEGIN
    SELECT tipo_vulnerabilidad, COUNT(*) AS total
    FROM Vulnerabilidades
    GROUP BY tipo_vulnerabilidad;
END$$
```

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `sp_usuarios_por_rol` (IN
`p_rol` VARCHAR(50)) BEGIN
    SELECT usuario_id, nombre, email
    FROM usuarios
    WHERE rol = p_rol;
END$$
```

```
DELIMITER ;
```

```
-- Volcado de datos para la tabla `redes`
INSERT INTO `redes` (`red_id`, `nombre`, `tipo`, `configuracion`) VALUES
(1, 'Red Estudiantes', 'segura', 'WPA2, filtrado MAC, segmentación VLAN'),
(2, 'Red Invitados', 'pública', 'Portal cautivo, autenticación temporal'),
(3, 'Red Administrativa', 'segura', 'VPN, control de acceso, segmentación'),
(4, 'Red IoT', '', 'Aislada para dispositivos IoT');
```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
```

```
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS
*/;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION
*/;
/*!40101 SET NAMES utf8mb4 */;
```

```
-- Volcado de datos para la tabla `eventos_monitorizacion`
INSERT INTO `eventos_monitorizacion` (`evento_id`, `mac_address`,
`timestamp`, `rssi`) VALUES
(1, 'AA:BB:CC:DD:EE:01', '2024-03-18 08:15:00', -45),
(2, 'AA:BB:CC:DD:EE:01', '2024-03-19 08:45:00', -50),
(3, 'AA:BB:CC:DD:EE:02', '2024-03-18 09:45:00', -60),
(4, 'AA:BB:CC:DD:EE:03', '2024-03-18 07:50:00', -50),
```

```
-- Volcado de datos para la tabla `usuarios`
```

```

INSERT INTO `usuarios` (`usuario_id`, `nombre`, `email`, `rol`, `contraseña`)
VALUES
(101, 'Kristin Martínez', 'kristin.martínez0@test.com', 'estudiante',
'$2y$10$3eCQLPd5NH4NzBN2.OlbNuJYQO0vYvmo64oPKh1bQljzdoZJkr/nS'),
(102, 'Irantzu Romeu', 'irantzu.romeu1@test.com', 'invitado',
'$2y$10$rtwGFbmdoicJqYulej.frORScdFRSioPGDGaUXWAWWh0OOtauJ0YqW'),
(103, 'Ricard Jaumandreu', 'ricard.jaumandreu2@test.com', 'estudiante',
'$2y$10$7oiJ0Hi/HBkto2FADkY1ZOx7uYBVlvKsrE1eXTeK2LrU.TO7QZjmK'),
(104, 'Sara Romeu', 'sara.romeu3@test.com', 'estudiante',
'$2y$10$K4/PPE1worXfvYKevOJg6uVrC4jf3SJiAm9RjVtnfsTDVwrJtGrW2');

```

-- Índices para tablas volcadas

```

ALTER TABLE `auditoria`
ADD PRIMARY KEY (`id`);

```

```

ALTER TABLE `dispositivos`
ADD PRIMARY KEY (`dispositivo_id`),
ADD UNIQUE KEY `mac_address` (`mac_address`),
ADD KEY `usuario_id` (`usuario_id`);

```

```

ALTER TABLE `eventos_monitorizacion`
ADD PRIMARY KEY (`evento_id`);

```

```

ALTER TABLE `logs_acceso`
ADD PRIMARY KEY (`id`),
ADD KEY `usuario_id` (`usuario_id`);

```

```

ALTER TABLE `recuperacion_tokens`
ADD PRIMARY KEY (`id`),
ADD KEY `usuario_id` (`usuario_id`);

```

```

ALTER TABLE `redes`
ADD PRIMARY KEY (`red_id`);

```

```

ALTER TABLE `sesiones_conexion`
ADD PRIMARY KEY (`sesion_id`),
ADD KEY `dispositivo_id` (`dispositivo_id`);

```

```

ALTER TABLE `usuarios`
ADD PRIMARY KEY (`usuario_id`),
ADD UNIQUE KEY `email` (`email`);

```

```

ALTER TABLE `vulnerabilidades`
ADD PRIMARY KEY (`vulnerabilidad_id`),
ADD KEY `dispositivo_id` (`dispositivo_id`);

```



```

-- AUTO_INCREMENT de las tablas volcadas
ALTER TABLE `auditoria`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;

ALTER TABLE `dispositivos`
  MODIFY `dispositivo_id` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=801;
--
ALTER TABLE `eventos_monitorizacion`
  MODIFY `evento_id` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=12;

ALTER TABLE `logs_acceso`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;

ALTER TABLE `recuperacion_tokens`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;

ALTER TABLE `redes`
  MODIFY `red_id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=5;

ALTER TABLE `sesiones_conexion`
  MODIFY `sesion_id` int(11) NOT NULL AUTO_INCREMENT,
  AUTO_INCREMENT=801;

-- Verificamos la estructura de las tablas
DESC Usuarios;
DESC Dispositivos;
DESC Sesiones_Conexion;
DESC Eventos_Monitorizacion;
DESC Vulnerabilidades;
DESC Redes;

-- Verificamos claves primarias y foraneas
SELECT
  TABLE_NAME, COLUMN_NAME, CONSTRAINT_NAME,
  REFERENCED_TABLE_NAME, REFERENCED_COLUMN_NAME
FROM
  information_schema.KEY_COLUMN_USAGE
WHERE
  TABLE_SCHEMA = 'proyecto'
  AND REFERENCED_TABLE_NAME IS NOT NULL;

-- Validamos relaciones con consultas JOIN
SELECT u.nombre, d.mac_address, d.ip_address, d.tipo_dispositivo
FROM Usuarios u
JOIN Dispositivos d ON u.usuario_id = d.usuario_id;

```

```

SELECT s.sesion_id, u.nombre, d.mac_address, s.timestamp_inicio,
s.timestamp_fin, s.red
FROM Sesiones_Conexion s
JOIN Dispositivos d ON s.dispositivo_id = d.dispositivo_id
JOIN Usuarios u ON d.usuario_id = u.usuario_id;

```

```

SELECT v.vulnerabilidad_id, u.nombre, d.mac_address, v.tipo_vulnerabilidad,
v.severidad, v.fecha_deteccion
FROM Vulnerabilidades v
JOIN Dispositivos d ON v.dispositivo_id = d.dispositivo_id
JOIN Usuarios u ON d.usuario_id = u.usuario_id;

```

```

-- Buscamos posibles errores en las relaciones
SHOW ENGINE INNODB STATUS;

```

```

-- Buscamos datos huérfanos
SELECT * FROM Dispositivos d
LEFT JOIN Sesiones_Conexion s ON d.dispositivo_id = s.dispositivo_id
WHERE s.sesion_id IS NULL;

```

```

SELECT * FROM Vulnerabilidades v
LEFT JOIN Sesiones_Conexion s ON v.dispositivo_id = s.dispositivo_id
WHERE s.sesion_id IS NULL;

```

```

-- Creamos roles
CREATE ROLE admin;
CREATE ROLE usuario;
CREATE ROLE invitado;

```

```

-- Asignamos permisos
GRANT ALL PRIVILEGES ON wishield.* TO 'admin'@'localhost';
GRANT SELECT ON wishield.usuarios TO 'usuario'@'localhost';

```

```

-- Creamos una vista restringida para invitados
CREATE VIEW usuarios_invitados AS
SELECT usuario_id, nombre, email FROM usuarios WHERE rol = 'invitado';
GRANT SELECT ON usuarios_invitados TO 'invitado'@'localhost';

```

#### *Triggers.sql:*

```

DELIMITER //
-- Trigger 1: Logs de acceso
CREATE TRIGGER tr_log_acceso_usuario
AFTER INSERT ON logs_acceso
FOR EACH ROW
BEGIN
    -- Puede extenderse para registrar estado o auditoría

```

```

END //
-- Trigger 2: Evitar eliminar admins
CREATE TRIGGER tr_prevent_admin_delete
BEFORE DELETE ON usuarios
FOR EACH ROW
BEGIN
    IF OLD.rol = 'admin' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = '🚫 No se puede eliminar un usuario con rol de
administrador.';
    END IF;
END //
-- Trigger 3: Añadir vulnerabilidad automática tras insertar dispositivo
CREATE TRIGGER tr_auto_revisar_vuln
AFTER INSERT ON Dispositivos
FOR EACH ROW
BEGIN
    IF NEW.tipo_dispositivo = 'Smart TV' THEN
        INSERT INTO Vulnerabilidades (dispositivo_id, tipo_vulnerabilidad, severidad,
fecha_deteccion)
        VALUES (NEW.dispositivo_id, 'Fuga de datos detectada', 'crítica', CURDATE());
    ELSEIF NEW.tipo_dispositivo = 'Smartphone' THEN
        INSERT INTO Vulnerabilidades (dispositivo_id, tipo_vulnerabilidad, severidad,
fecha_deteccion)
        VALUES (NEW.dispositivo_id, 'Intento de acceso no autorizado', 'alta',
CURDATE());
    END IF;
END //
-- Trigger 4: Eliminar tokens anteriores antes de insertar uno nuevo
CREATE TRIGGER tr_token_reset_cleanup
BEFORE INSERT ON recuperacion_tokens
FOR EACH ROW
BEGIN
    DELETE FROM recuperacion_tokens
    WHERE usuario_id = NEW.usuario_id;
END //
DELIMITER ;

```

#### *Procedimientos\_wishield.sql*

```

DELIMITER //
CREATE PROCEDURE sp_insertar_usuario (
    IN p_nombre VARCHAR(100),
    IN p_email VARCHAR(100),
    IN p_contraseña VARCHAR(255),
    IN p_rol VARCHAR(50),
    IN p_mac_address VARCHAR(100),
    IN p_ip_address VARCHAR(100),

```

```

    IN p_tipo_dispositivo VARCHAR(50),
    IN p_clave_encryptacion VARCHAR(255)
)
BEGIN
    DECLARE uid INT;
    INSERT INTO usuarios (nombre, email, contraseña, rol)
    VALUES (p_nombre, p_email, p_contraseña, p_rol);
    SET uid = LAST_INSERT_ID();
    INSERT INTO Dispositivos (usuario_id, mac_address, ip_address,
tipo_dispositivo)
    VALUES (
        uid,
        AES_ENCRYPT(p_mac_address, p_clave_encryptacion),
        AES_ENCRYPT(p_ip_address, p_clave_encryptacion),
        p_tipo_dispositivo
    );
    SELECT uid AS nuevo_usuario_id;
END //

CREATE PROCEDURE sp_total_vulnerabilidades ()
BEGIN
    SELECT tipo_vulnerabilidad, COUNT(*) AS total
    FROM Vulnerabilidades
    GROUP BY tipo_vulnerabilidad;
END //

CREATE PROCEDURE sp_usuarios_por_rol(IN p_rol VARCHAR(50))
BEGIN
    SELECT usuario_id, nombre, email
    FROM usuarios
    WHERE rol = p_rol;
END //

CREATE PROCEDURE sp_total_por_rol()
BEGIN
    SELECT rol, COUNT(*) AS total
    FROM usuarios
    GROUP BY rol;
END //
DELIMITER ;
DELIMITER //
-- Mostrar los logs de acceso por usuario (nombre, email, última conexión)
CREATE PROCEDURE sp_logs_por_usuario ()
BEGIN
    SELECT u.usuario_id, u.nombre, u.email, MAX(l.fecha_hora) AS ultima_conexion
    FROM usuarios u
    LEFT JOIN logs_acceso l ON u.usuario_id = l.usuario_id

```

```

        GROUP BY u.usuario_id, u.nombre, u.email
        ORDER BY ultima_conexion DESC;
END //
-- Total de sesiones activas por día (últimos 7 días)
CREATE PROCEDURE sp_actividad_por_fecha ()
BEGIN
    SELECT DATE(timestamp_inicio) AS fecha, COUNT(*) AS sesiones
    FROM Sesiones_Conexion
    WHERE timestamp_inicio >= CURDATE() - INTERVAL 7 DAY
    GROUP BY DATE(timestamp_inicio)
    ORDER BY fecha ASC;
END //
-- Total de dispositivos por tipo (para gráfico de pastel o barras)
CREATE PROCEDURE sp_dispositivos_por_tipo ()
BEGIN
    SELECT tipo_dispositivo, COUNT(*) AS total
    FROM Dispositivos
    GROUP BY tipo_dispositivo
    ORDER BY total DESC;
END //
DELIMITER ;

```

## PHP y HTML

*Wishield.php*

```

<?php
session_start();
$_SESSION["usuario_id"] = 1;
$_SESSION["nombre"] = "Admin Temporal";
$_SESSION["rol"] = "admin";

Control de IPs autorizadas
$ips_autorizadas = [$ip_actual]; // Añade otras IPs si lo ves necesario
$ip_actual = $_SERVER['REMOTE_ADDR'];
if (!in_array($ip_actual, $ips_autorizadas)) {
    header("HTTP/1.1 403 Forbidden");
    echo "<h1 style='color: red;'>🚫 Acceso denegado</h1>";
    echo "<p>IP bloqueada: <strong>$ip_actual</strong></p>";
    /exit;
}
require_once 'config.php'; // Clave secreta para AES
// Proteger acceso solo para admins
if (!isset($_SESSION["usuario_id"])) {
    header("Location: login.php");
    exit;
}
if ($_SESSION["rol"] !== "admin") {

```

```

    echo "<h2 style='color: red; text-align: center;'>🚫 Acceso denegado. Solo para
administradores.</h2>";
    exit;
}

```

```
// Activar errores
```

```
ini_set('display_errors', 1);
```

```
error_reporting(E_ALL);
```

```
mysqli_report(MYSQLI_REPORT_ERROR | MYSQLI_REPORT_STRICT);
```

```
// Conexión
```

```
$conexion = new mysqli("localhost", "root", "", "wishield");
```

```
if ($conexion->connect_error) {
```

```
    die("Error de conexión: " . $conexion->connect_error);
```

```
}
```

```
// Paginación
```

```
$registros_por_pagina = 20;
```

```
$pagina_actual = isset($_GET['pagina']) ? (int)$_GET['pagina'] : 1;
```

```
$offset = ($pagina_actual - 1) * $registros_por_pagina;
```

```
// Procesar formulario
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
```

```
    $nombre = trim($_POST['nombre']);
```

```
    $email = trim($_POST['email']);
```

```
    $password = password_hash($_POST['password'], PASSWORD_BCRYPT);
```

```
    $rol = trim($_POST['rol']);
```

```
    $tipo_dispositivo = trim($_POST['tipo_dispositivo']);
```

```
    $mac_address = trim($_POST['mac_address']);
```

```
    $ip_address = trim($_POST['ip_address']);
```

```
    // Insertar usuario
```

```
    $sql = "INSERT INTO usuarios (nombre, email, rol, contraseña) VALUES (?, ?, ?,
?)";
```

```
    $stmt = $conexion->prepare($sql);
```

```
    $stmt->bind_param("ssss", $nombre, $email, $rol, $password);
```

```
    if ($stmt->execute()) {
```

```
        $usuario_id = $conexion->insert_id;
```

```
        // Insertar dispositivo con cifrado AES
```

```
        $sql_dispositivo = "INSERT INTO Dispositivos (usuario_id, mac_address,
ip_address, tipo_dispositivo)
```

```
        VALUES (?, AES_ENCRYPT(?, ?), AES_ENCRYPT(?, ?), ?)";
```

```
        $stmt2 = $conexion->prepare($sql_dispositivo);
```

```
        $stmt2->bind_param("issss", $usuario_id, $mac_address, CLAVE_SECRETA,
$ip_address, CLAVE_SECRETA, $tipo_dispositivo);
```

```
        $stmt2->execute();
```

```
        // Comprobar vulnerabilidades
```

```
        $sql_vuln = "SELECT * FROM Vulnerabilidades WHERE dispositivo_id IN (
        SELECT id FROM Dispositivos WHERE usuario_id = ?)";
```

```
        $stmt3 = $conexion->prepare($sql_vuln);
```

```

$stmt3->bind_param("i", $usuario_id);
$stmt3->execute();
$result = $stmt3->get_result();
echo "<div style='color:green; margin: 10px;'>Usuario y dispositivo registrados
correctamente.</div>";
if ($result->num_rows > 0) {
    echo "<div style='color:red; margin: 10px;'>¡Este usuario tiene dispositivos
vulnerables!</div>";
}
$stmt2->close();
$stmt3->close();
} else {
    echo "Error al agregar usuario: " . $stmt->error;
}
$stmt->close();
}
// Filtro de búsqueda
$condicion = "";
if (isset($_GET['buscar']) && $_GET['buscar'] !== "") {
    $buscar = $conexion->real_escape_string($_GET['buscar']);
    $condicion = "WHERE u.nombre LIKE '%$buscar%' OR u.rol LIKE '%$buscar%'";
}
// Total para paginación
$sql_total = "SELECT COUNT(*) as total FROM usuarios u
JOIN Dispositivos d ON u.usuario_id = d.usuario_id
$condicion";
$res_total = $conexion->query($sql_total);
$total_filas = $res_total->fetch_assoc()['total'];
$total_paginas = ceil($total_filas / $registros_por_pagina);
$sql = "SELECT u.nombre, u.email, u.rol,
d.tipo_dispositivo,
d.mac_address,
d.ip_address
FROM usuarios u
JOIN Dispositivos d ON u.usuario_id = d.usuario_id
$condicion
ORDER BY u.usuario_id DESC
LIMIT $registros_por_pagina OFFSET $offset";
$resultado = $conexion->query($sql);
if (!$resultado) {
    die("✗ Error en la consulta de usuarios: " . $conexion->error);
}

?>
<!DOCTYPE html>
<html lang="es">
<head>

```

```

<meta charset="UTF-8">
<title>WiShield · Registro de Usuarios</title>
<style>
  body {
    font-family: 'Segoe UI', sans-serif;
    background-color: #f2f4f8;
    margin: 0;
    padding: 0;
  }
  nav {
    background: #2c3e50;
    padding: 12px 20px;
    display: flex;
    gap: 20px;
  }
  nav a {
    color: #ecf0f1;
    text-decoration: none;
    font-weight: bold;
  }

  h2, h3 {
    color: #2c3e50;
  }
  .container {
    max-width: 1000px;
    margin: 30px auto;
    padding: 0 20px;
  }
  .card {
    background-color: white;
    padding: 20px;
    border-radius: 12px;
    box-shadow: 0 2px 10px rgba(0,0,0,0.1);
    margin-bottom: 30px;
  }
  form label {
    display: block;
    margin-bottom: 12px;
  }
  input[type="text"],
  input[type="email"],
  select {
    width: 100%;
    padding: 10px;
    border-radius: 6px;
    border: 1px solid #ccc;
  }

```



```

        margin-top: 5px;
    }
    input[type="submit"] {
        background-color: #3498db;
        color: white;
        border: none;
        padding: 10px 16px;
        border-radius: 6px;
        font-weight: bold;
        cursor: pointer;
        margin-top: 10px;
    }
    input[type="submit"]:hover {
        background-color: #2980b9;
    }
    table {
        width: 100%;
        border-collapse: collapse;
        background-color: white;
    }
    th, td {
        padding: 10px;
        border: 1px solid #ddd;
        text-align: left;
    }
    th {
        background-color: #2980b9;
        color: white;
    }
    }

    .search-box {
        display: flex;
        gap: 10px;
        margin-bottom: 20px;
    }
    .search-box input[type="text"] {
        flex: 1;
    }
    }
</style>
</head>
<body>
<nav>
    <a href="wishield.php"><img alt="house icon" data-bbox="368 818 388 838"/> Registro</a>
    <a href="dashboard.php"><img alt="bar chart icon" data-bbox="388 838 408 858"/> Dashboard</a>
    <a href="http://localhost/phpmyadmin" target="_blank"><img alt="key icon" data-bbox="648 858 668 878"/> phpMyAdmin</a>
    <span style="flex-grow: 1;"></span>

```

```

<span style="color: #ecf0f1;"><img alt="user icon" data-bbox="422 88 442 108"/> <?php echo $_SESSION["nombre"]; ?> (<?php
echo $_SESSION["rol"]; ?>)</span>
<a href="logout.php" style="margin-left: 20px;"><img alt="logout icon" data-bbox="575 125 595 145"/> Cerrar sesión</a>
</nav>
<div class="container">
<div class="card">
<form method="GET" class="search-box">
<input type="text" name="buscar" placeholder="Buscar por nombre o rol..."
value="<?php echo isset($_GET['buscar']) ?
htmlspecialchars($_GET['buscar']) : ''; ?>">
<input type="submit" value="Buscar">
</form>
</div>
<div class="card">
<h2>Agregar nuevo usuario + dispositivo</h2>
<form method="POST">
<label>Nombre:
<input type="text" name="nombre" required>
</label>
<label>Email:
<input type="email" name="email" required>
</label>
<label>Contraseña:
<input type="password" name="password" required>
</label>
<label>Rol:
<select name="rol">
<option value="invitado">Invitado</option>
<option value="estudiante">Estudiante</option>
<option value="admin">Administrador</option>
</select>
</label>
<label>Tipo de dispositivo:
<select name="tipo_dispositivo">
<option value="Laptop">Laptop</option>
<option value="Smartphone">Smartphone</option>
<option value="Tablet">Tablet</option>
<option value="Smart TV">Smart TV</option>
<option value="Smartwatch">Smartwatch</option>
<option value="Consola de videojuegos">Consola de
videojuegos</option>
</select>
</label>
<label>MAC Address:
<input type="text" name="mac_address" required>
</label>
<label>IP Address:

```

```

        <input type="text" name="ip_address" required>
    </label>
    <input type="submit" value="Agregar Usuario">
</form>
</div>
<div class="card">
<h3>📋 Usuarios registrados y sus dispositivos</h3>
<table>
    <tr>
        <th>Nombre</th>
        <th>Email</th>
        <th>Rol</th>
        <th>Tipo de Dispositivo</th>
        <th>MAC Address</th>
        <th>IP Address</th>
    </tr>
    <?php while ($fila = $resultado->fetch_assoc()): ?>
    <tr>
        <td><?php echo htmlspecialchars($fila['nombre']); ?></td>
        <td><?php echo htmlspecialchars($fila['email']); ?></td>
        <td><?php echo htmlspecialchars($fila['rol']); ?></td>
        <td><?php echo htmlspecialchars($fila['tipo_dispositivo']); ?></td>
        <td><?php echo htmlspecialchars($fila['mac_address']); ?></td>
        <td><?php echo htmlspecialchars($fila['ip_address']); ?></td>
    </tr>
    <?php endwhile; ?>
</table>
</div>
    <div style="text-align: center; margin-top: 20px;">
    <?php if ($pagina_actual > 1): ?>
        <a href="?pagina=<?php echo $pagina_actual - 1; ?>&buscar=<?php echo
urlencode($_GET['buscar'] ?? ''); ?>">« Anterior</a>
    <?php endif; ?>

    <?php for ($i = 1; $i <= $total_paginas; $i++): ?>
        <a href="?pagina=<?php echo $i; ?>&buscar=<?php echo
urlencode($_GET['buscar'] ?? ''); ?>"
            style="<?php echo ($i == $pagina_actual) ? 'font-weight: bold; text-
decoration: underline;' : ''; ?>">
            <?php echo $i; ?>
        </a>
    <?php endfor; ?>

    <?php if ($pagina_actual < $total_paginas): ?>
        <a href="?pagina=<?php echo $pagina_actual + 1; ?>&buscar=<?php echo
urlencode($_GET['buscar'] ?? ''); ?>">Siguiente »</a>
    <?php endif; ?>

```

```

</div>
</div>
</div>
</body>
</html>

```

### *Dashboard.php*

```

<?php
ob_start();
session_start();
$_SESSION["usuario_id"] = 1;
$_SESSION["nombre"] = "Admin Temporal";
$_SESSION["rol"] = "admin";

// Control de IPs autorizadas
// $ips_autorizadas = ['127.0.0.1', '::1']; // Añade otras IPs si lo ves necesario
// $ip_actual = $_SERVER['REMOTE_ADDR'];
// if (in_array($ip_actual, $ips_autorizadas)) {
//   header("HTTP/1.1 403 Forbidden");
//   echo "<h1 style='color: red;'>⊘ Acceso denegado</h1>";
//   echo "<p>IP bloqueada: <strong>$ip_actual</strong></p>";
//   exit;
// }

if (!isset($_SESSION["usuario_id"])) {
    header("Location: login.php");
    exit;
}
require_once 'config.php';
$conexion = new mysqli("localhost", "root", "", "wishield");

if ($conexion->connect_error) {
    die("Error de conexión: " . $conexion->connect_error);
}
// Usuarios por rol
$res_usuarios = $conexion->query("CALL sp_total_por_rol()");
$usuarios_data = [];
while ($row = $res_usuarios->fetch_assoc()) {
    $usuarios_data[] = $row;
}
$res_usuarios->close();
$conexion->next_result();
// Sesiones activas por red
$res_sesiones = $conexion->query("CALL sp_sesiones_activas()");
$sesiones_data = [];
while ($row = $res_sesiones->fetch_assoc()) {
    $sesiones_data[] = $row;
}

```

```

}
$res_sesiones->close();
$conexion->next_result();
// Vulnerabilidades por severidad
$res_vuln = $conexion->query("CALL sp_total_vulnerabilidades()");
$vuln_data = [];
while ($row = $res_vuln->fetch_assoc()) {
    $vuln_data[] = $row;
}
$res_vuln->close();
$conexion->next_result();
// Logs por usuario (NO cerrar aún, lo haces en el HTML)
$res_logs = $conexion->query("CALL sp_logs_por_usuario()");
$conexion->next_result(); // Dejar esto para liberar el siguiente CALL
// Actividad por fecha (igual)
$res_actividad = $conexion->query("CALL sp_actividad_por_fecha()");
$conexion->next_result();
// Dispositivos conectados por tipo
$res_disp = $conexion->query("CALL sp_dispositivos_por_tipo()");
$tipos = [];
$valores = [];
while ($fila = $res_disp->fetch_assoc()) {
    $tipos[] = $fila['tipo_dispositivo'];
    $valores[] = $fila['total'];
}
$res_disp->close();
$conexion->next_result();
$conexion->close();
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Dashboard WiShield</title>
    <script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
    <style>
        body {
            font-family: Arial, sans-serif;
            background: #f0f4f8;
            padding: 20px;
        }
        nav {
            background: #2c3e50;
            padding: 12px;
            display: flex;
            gap: 20px;
        }
    </style>

```

```

nav a {
    color: #ecf0f1;
    text-decoration: none;
    font-size: 16px;
}
.chart-row {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
    justify-content: center;
}
.chart-container {
    background: white;
    padding: 20px;
    border-radius: 12px;
    box-shadow: 0 4px 10px rgba(0,0,0,0.1);
    width: 350px;
}
canvas {
    max-width: 100%;
}

h1 {
    text-align: center;
    color: #2c3e50;
}
</style>
</head>
<body>
<nav>
    <a href="wishield.php">🏠 Registro</a>
    <a href="dashboard.php">📊 Dashboard</a>
    <a href="https://localhost/phpmyadmin" target="_blank">🔗 phpMyAdmin</a>
    <span style="flex-grow: 1;"></span>
    <span style="color: #ecf0f1;">👤 <?php echo $_SESSION["nombre"] . " (" .
$_SESSION["rol"] . ")"; ?></span>
    <a href="logout.php" style="margin-left: 20px;">🚪 Cerrar sesión</a>
</nav>
<h1>📊 Dashboard WiShield</h1>
<div class="chart-row">
    <div class="chart-container">
        <h3>👥 Usuarios por rol</h3>
        <canvas id="usuariosChart"></canvas>
    </div>
    <div class="chart-container">

```

```

        <h3>📶 Sesiones activas por red</h3>
        <canvas id="sesionesChart"></canvas>
    </div>
    <div class="chart-container">
        <h3>🔒 Vulnerabilidades por severidad</h3>
        <canvas id="vulnChart"></canvas>
    </div>
    <div class="chart-container">
        <h3>📱 Dispositivos por tipo</h3>
        <canvas id="graficaDispositivos"></canvas>
    </div>
</div>
<h3>📄 Últimos accesos de usuarios</h3>
<table border='1' cellpadding='6' style='border-collapse: collapse; background:
white;'>
    <tr style='background: #34495e; color: white;'>
        <th>Usuario</th><th>Email</th><th>Último acceso</th>
    </tr>
    <?php while ($fila = $res_logs->fetch_assoc()): ?>
    <tr>
        <td><?php echo htmlspecialchars($fila['nombre']); ?></td>
        <td><?php echo htmlspecialchars($fila['email']); ?></td>
        <td><?php echo $fila['ultima_conexion'] ?? '—'; ?></td>
    </tr>
    <?php endwhile; $res_logs->close(); ?>
</table>
<h3>📅 Actividad (últimos 7 días)</h3>
<table border='1' cellpadding='6' style='border-collapse: collapse; background:
white;'>
    <tr style='background: #2ecc71; color: white;'>
        <th>Fecha</th><th>Sesiones iniciadas</th>
    </tr>
    <?php while ($fila = $res_actividad->fetch_assoc()): ?>
    <tr>
        <td><?php echo $fila['fecha']; ?></td>
        <td><?php echo $fila['sesiones']; ?></td>
    </tr>
    <?php endwhile; $res_actividad->close(); ?>
</table>
<script>
const usuariosData = <?php echo json_encode($usuarios_data); ?>;
const sesionesData = <?php echo json_encode($sesiones_data); ?>;
const vulnData = <?php echo json_encode($vuln_data); ?>;
const tiposDisp = <?php echo json_encode($tipos); ?>;
const valoresDisp = <?php echo json_encode($valores); ?>;

```

```

new Chart(document.getElementById('usuariosChart'), {
  type: 'pie',
  data: {
    labels: usuariosData.map(x => x.rol),
    datasets: [{
      data: usuariosData.map(x => x.total),
      backgroundColor: ['#3498db', '#2ecc71', '#f1c40f']
    }]
  }
});

```

```

new Chart(document.getElementById('sesionesChart'), {
  type: 'bar',
  data: {
    labels: sesionesData.map(x => x.red),
    datasets: [{
      label: 'Sesiones activas',
      data: sesionesData.map(x => x.total),
      backgroundColor: '#9b59b6'
    }]
  }
});

```

```

new Chart(document.getElementById('vulnChart'), {
  type: 'doughnut',
  data: {
    labels: vulnData.map(x => x.severidad),
    datasets: [{
      data: vulnData.map(x => x.total),
      backgroundColor: ['#e74c3c', '#f39c12', '#27ae60', '#34495e']
    }]
  }
});

```

```

new Chart(document.getElementById('graficaDispositivos'), {
  type: 'bar',
  data: {
    labels: tiposDisp,
    datasets: [{
      label: 'Cantidad',
      data: valoresDisp,
      backgroundColor: ['#3498db', '#9b59b6', '#f1c40f', '#2ecc71', '#e74c3c'],
      borderColor: '#2c3e50',
      borderWidth: 1
    }]
  },
  options: {

```



```

        responsive: true,
        plugins: {
            legend: { display: false }
        },
        scales: {
            y: { beginAtZero: true }
        }
    }
});
</script>
</body>
</html>
<?php ob_end_flush(); ?>

```

### *Config.php*

```

<?php
define("CLAVE_SECRETA", "WiShieldClaveSegura2025!");
?>
Enviar_token.php <?php
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;
require 'src/PHPMailer.php';
require 'src/SMTP.php';
require 'src/Exception.php';
require_once 'config.php';
$conexion = new mysqli("localhost", "root", "", "wishield");
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = trim($_POST['email']);
    // Buscar usuario por email
    $stmt = $conexion->prepare("SELECT usuario_id, nombre FROM usuarios
WHERE email = ?");
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $resultado = $stmt->get_result();
    if ($resultado->num_rows === 1) {
        $usuario = $resultado->fetch_assoc();
        $usuario_id = $usuario["usuario_id"];
        $nombre = $usuario["nombre"];
        // Generar token y caducidad (ej. 1 hora)
        $token = bin2hex(random_bytes(32));
        $expiracion = date("Y-m-d H:i:s", strtotime("+1 hour"));
        // Guardar en base de datos
        $insert = $conexion->prepare("INSERT INTO tokens_recuperacion (usuario_id,
token, expiracion) VALUES (?, ?, ?)");
        $insert->bind_param("iss", $usuario_id, $token, $expiracion);
        $insert->execute();
    }
}

```

```

// Crear enlace
$enlace = "http://localhost/wishield/reset_password.php?token=$token";
// Configurar PHPMailer
$mail = new PHPMailer(true);
try {
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'astoreth@gmail.com'; // 📧 Pon tu correo
    $mail->Password = 'xfad nfmr gqnn mjqb'; // 📧 Aquí tu contraseña de
aplicación
    $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS;
    $mail->Port = 587;
    $mail->setFrom('Tastoreth@gmail.com', 'WiShield');
    $mail->addAddress($email, $nombre);
    $mail->isHTML(true);
    $mail->Subject = '🔒 Recuperación de contraseña WiShield';
    $mail->Body = "Hola <strong>$nombre</strong>,<br><br>
        Has solicitado recuperar tu contraseña.<br>
        <a href='$enlace'>Haz clic aquí para crear una nueva
contraseña</a><br><br>
        Este enlace caduca en 1 hora.";
    $mail->send();
    echo "<h3 style='color:green;'>✅ Se ha enviado un correo con
instrucciones de recuperación.</h3>";
} catch (Exception $e) {
    echo "<h3 style='color:red;'>❌ Error al enviar el correo: {$mail-
>ErrorInfo}</h3>";
}
} else {
    echo "<h3 style='color:red;'>❌ No se encontró ninguna cuenta con ese
email.</h3>";
}
}
?>

```

#### *Generador\_datos.php*

```

<?php
ini_set('max_execution_time', 700); // aumenta tiempo de ejecución si metes
muchos sino da error
$conexion = new mysqli("localhost", "root", "", "wishield");

if ($conexion->connect_error) {
    die("Error de conexión: " . $conexion->connect_error);
}

```

```
$TOTAL = 100; // Cambia este número a 100, 500, 1000...
```

```
$roles = ['invitado', 'estudiante', 'admin'];
$dispositivos = ['Laptop', 'Smartphone', 'Tablet', 'Smart TV', 'Smartwatch', 'Consola
de videojuegos'];
$redes = ['Red Estudiantes', 'Red Invitados', 'Red Administrativa'];
$severidades = ['baja', 'media', 'alta', 'crítica'];
function generarMAC() {
    $mac = [];
    for ($i = 0; $i < 6; $i++) {
        $mac[] = strtoupper(str_pad(dechex(rand(0, 255)), 2, '0', STR_PAD_LEFT));
    }
    return implode(':', $mac);
}
function generarIP() {
    return '192.168.' . rand(1, 3) . '.' . rand(10, 250);
}
function nombreFalso() {
    $nombres = ["Ana", "Carlos", "Lucía", "Jorge", "Valentina", "Raúl", "Marina",
"David", "Paula", "Sergio", "Raquel", "Abraham", "Luisa", "Marivega", "Antonio",
"Mercé", "Vanesa", "Javier", "Marina", "Abril", "MariCarmen", "Sara", "Diana",
"Andrea", "Felix", "Silvia", "Irantzu", "Arturo", "Kristin", "Nacho", "Ricard", "Elena",
"Ben", "Aritz", "John", "Rosa", "Rúben", "Isabel", "Jezabella", "Carmen", "Armando",
"Blanca", "Lidia", "Andrés", "Covadonga", "Renee", "Bojan", "Sonia", "Alba", "Jerson",
"Edurne", "Diego", "Duncan", "Sandra", "Alexandra", "Alejandro", "Sandro", "Xavier",
"Samuel", "Nick", "Miles", "Louis", "Thais", "Eire", "Isaac", "Iria", "Mikel", "Nicolas",
"Gemma", "Patxi", "Pascu", "Alyona", "Leiona", "Leo", "Marc", "Marcos",
"Mark", "Elvira", "Fermin", "Dolores", "Pere", "Pedro", "Peter", "Altagracia", "Amadora",
"Apolinario", "Arnulfo", "Arsenio", "Bonifacio", "Burgundófora", "Cipriniano",
"Cojoncio", "Digna", "Diosnelio", "Dombina", "Escolástico", "Estanislada",
"Expiración", "Froilana", "Froilán", "Fulgencio", "Fulgencia", "Ruperta",
"Gumersindo", "Diogenes", "Hermógenes", "Montse", "Hierónides", "Hercules",
"Iluminado", "Ladislao", "Elsa", "Elso", "Elba", "Luzdivino", "Marciana", "Marcial",
"Oristila", "Pantaleona", "Pantaleón", "Yorinda", "Yoringel", "Piedrasantas",
"Protasio", "Segismundo", "Tesifonte", "Penitencia", "Paz", "Renata", "Aitor", "Iker",
"Markel", "Eneko", "Oier", "Xaiba", "Argi", "Abba", "Brais", "Drac", "Eilán", "Elm",
"Jofre", "Guifré", "Enzo", "Eros", "Jano, Elían", "Eros", "Milos", "Anne", "Serge", "Uriel",
"Otto", "Zigor", "Salomón", "Ezequiel", "Aaron", "Georgina", "Laia", "Obdulia",
"William", "James", "John", "Robert", "Michael", "Thomas", "David", "George", "Jane",
"Sarah", "April", "Emily", "Rachel", "Amber", "Charlotte", "Madison", "Brooke",
"Amy", "Hunter", "Martin", "Bernard", "Thomas", "Petit", "Robert", "Richard", "Durand",
"Dubois", "Moreau", "Laurent", "Lambert", "Leroy", "Dupont", "Gabriel", "Colin",
"Lemaire", "Fontaine", "Blanchard", "Faure", "Chevalier", "Mathieu", "Morin",
"Legrand", "Robin", "Nicolas", "Blanc", "Masson", "Marchand", "Etsuko", "Hoshiko",
"Izumi", "Kagumi", "Kagome", "Kaoru", "Hana", "Sakura", "Himari", "Rin", "Kaguya",
"Yuna", "Kenshin", "Aki", "Akihito", "Hiro", "Akihiro", "Daiki", "Ryu", "Ryota", "Masaru",
```

"Hiroshi", "Shinosuke", "Hina", "Seitaro", "Kanakano", "Nobunaga", "Hideyoshi",  
 "Shingen", "Yoshimoto", "Masamune", "Ieyasu"];

```
$apellidos = ["García", "López", "Sánchez", "Martínez", "Ruiz", "Gómez", "Díaz",  

  "Pérez", "Torres", "de la Vega",  

  "Valentino", "Abril", "del Carmen", "Artemisa", "Botelli", "de la Marina", "Silvo",  

  "Irantzu", "Aritz",  

  "Armandez", "Nieve", "de Covadonga", "Kermit", "Bojan", "Albar", "Jetson", "Dhu",  

  "Sandro", "Miles", "Lioncourt",  

  "Eire", "Patel", "Alyona", "Leona", "Fermin", "Altagracia", "Amadora", "Apolinario",  

  "Arnulfo", "Arsenio", "Bonifacio", "Burgundófora", "Cipriniano", "Cojoncio", "Digna",  

  "Diosnelio", "Dombina", "Escolástico", "Estanislada", "Expiración", "Froilana",  

  "Gumersindo", "Diogenes", "Hermógenes", "Hierónides", "Hercules",  

  "Iluminado", "Ladislao", "Elsa", "Elso", "Elba", "Luzdivino", "Marciana", "Marcial",  

  "Oristila", "Pantaleona", "Pantaleón", "Yorinda", "Yoringel", "Piedrasantas",  

  "Protasio", "Segismundo", "Tesifonte", "de la Penitencia", "de la Paz", "de Cabeza",  

  "de Barriga", "Bronca", "Segura", "Fina", "Delano", "Gil", "de Dios", "Surero",  

  "Cremento", "Montada", "Trozado", "Tresado", "Mento", "Mingo", "Busado",  

  "Fermizo", "Japón", "Masdeu", "Cuesta", "Mogollón", "Amor", "Jurado", "Arrimadas",  

  "Seisdedos", "Pieplano", "Gol", "Gordo", "Nito", "del Bosque", "del Pozo", "Salido",  

  "Campofrío", "Ladrón", "Honesto", "Diezhandino", "Honrado", "Calavera", "Cortada",  

  "del Rosal", "Alegre", "Pieldelobo", "Bonachera", "Zas", "Perroverde", "Alcoholado",  

  "Gandula", "Chinchurreta", "de la Repolla", "Parahoy", "Paramí", "Verdugo",  

  "Pichilengue", "Karamoko", "Moto", "Vergassola", "Esario", "Osario", "Flores",  

  "Golon", "Smith", "Jones", "Williams", "Brown", "Hunter", "Martin", "Bernard",  

  "Thomas", "Petit", "Robert", "Richard", "Durand", "Dubois", "Moreau", "Laurent",  

  "Lambert", "Leroy", "Dupont", "Gabriel", "Colin", "Lemaire", "Fontaine", "Blanchard",  

  "Faure", "Chevalier", "Mathieu", "Morin", "Legrand", "Robin", "Nicolas", "Blanc",  

  "Masson", "Marchand", "Tanaka", "Yamada", "Nakamura", "Ishikawa", "Yamamoto",  

  "Yamagawa", "Yoshida", "Suzuki", "Kimura", "Nishimura", "Madarama",  

  "Matsudaira", "Mihura", "Minagawa", "Minami", "Miyake", "Mizoguchi", "Mori",  

  "Murakami", "Date", "Oda", "Toyotoma", "Ueda", "Tokugawa", "Takeda", "Imagawa"];  

  return $nombres[array_rand($nombres)] . ' ' . $apellidos[array_rand($apellidos)];  

}
```

```
// Generador masivo  

for ($i = 0; $i < $TOTAL; $i++) {  

  $nombre = nombreFalso();  

  $email = strtolower(str_replace(' ', '.', $nombre)) . $i . '@test.com';  

  $rol = $roles[array_rand($roles)];  

  $stmt = $conexion->prepare("INSERT INTO usuarios (nombre, email, rol) VALUES  

  (?, ?, ?)");  

  $stmt->bind_param("sss", $nombre, $email, $rol);  

  $stmt->execute();  

  $usuario_id = $conexion->insert_id;  

  $stmt->close();  

}
```

```

// Crear dispositivo
$mac = generarMAC();
$ip = generarIP();
$tipo = $dispositivos[array_rand($dispositivos)];
$stmt = $conexion->prepare("INSERT INTO Dispositivos (usuario_id,
mac_address, ip_address, tipo_dispositivo) VALUES (?, ?, ?, ?)");
$stmt->bind_param("isss", $usuario_id, $mac, $ip, $tipo);
$stmt->execute();
$dispositivo_id = $conexion->insert_id;
$stmt->close();
// Crear sesión de conexión aleatoria
$inicio = date("Y-m-d H:i:s", strtotime("-" . rand(0, 10) . " days " . rand(0, 23) . "
hours"));
$fin = rand(0, 1) ? date("Y-m-d H:i:s", strtotime($inicio . " + " . rand(1, 3) . "
hours")) : null;
$red = $redes[array_rand($redes)];
$stmt = $conexion->prepare("INSERT INTO Sesiones_Conexion (dispositivo_id,
timestamp_inicio, timestamp_fin, red) VALUES (?, ?, ?, ?)");
$stmt->bind_param("isss", $dispositivo_id, $inicio, $fin, $red);
$stmt->execute();
$stmt->close();
// Posiblemente añadir una vulnerabilidad
if (rand(0, 3) === 0) { // ~25% de los dispositivos
    $tipo_vuln = "Simulada: " . ['Puerto abierto', 'Fuga de datos', 'Malware', 'Acceso
no autorizado'][rand(0, 3)];
    $severidad = $severidades[array_rand($severidades)];
    $fecha = date("Y-m-d", strtotime("-" . rand(1, 7) . " days"));
    $stmt = $conexion->prepare("INSERT INTO Vulnerabilidades (dispositivo_id,
tipo_vulnerabilidad, severidad, fecha_deteccion) VALUES (?, ?, ?, ?)");
    $stmt->bind_param("isss", $dispositivo_id, $tipo_vuln, $severidad, $fecha);
    $stmt->execute();
    $stmt->close();
}
}
$csv = fopen("usuarios_generados.csv", "w");
fputcsv($csv, ["nombre", "email", "rol", "contraseña_plana"]);

for ($i = 0; $i < $TOTAL; $i++) {
    $nombre = nombreFalso();
    $email = strtolower(str_replace(' ', '', $nombre)) . $i . '@test.com';
    $rol = $roles[array_rand($roles)];
    // ⚡ Contraseña generada (simple para pruebas, puedes mejorarla)
    $pass_plana =
substr(str_shuffle('abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789'), 0, 8);
    $pass_hash = password_hash($pass_plana, PASSWORD_BCRYPT);

```

```

// Guardar en BD
$stmt = $conexion->prepare("INSERT INTO usuarios (nombre, email, rol,
contraseña) VALUES (?, ?, ?, ?)");
$stmt->bind_param("ssss", $nombre, $email, $rol, $pass_hash);
$stmt->execute();
$usuario_id = $conexion->insert_id;
$stmt->close();
// Guardar en CSV
fputcsv($csv, [$nombre, $email, $rol, $pass_plana]);
// El resto del script (dispositivos, sesiones, etc.) igual...
}
fclose($csv);

echo "<h2>✅ $TOTAL usuarios insertados con sus dispositivos, sesiones y
vulnerabilidades aleatorias.</h2>";
$conexion->close();
?>

```

### *Index.php*

```

<?php
session_start();
?>

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>WiShield · Inicio</title>
  <style>
    body {
      font-family: 'Segoe UI', sans-serif;
      background: #ecf0f1;
      margin: 0;
      padding: 0;
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100vh;
    }

    .card {
      background: white;
      padding: 30px 40px;
      border-radius: 12px;
      box-shadow: 0 4px 15px rgba(0,0,0,0.1);
      text-align: center;
    }

```

```

}
h1 {
    margin-bottom: 20px;
    color: #2c3e50;
}

a {
    display: block;
    margin: 10px 0;
    padding: 12px 18px;
    background: #3498db;
    color: white;
    text-decoration: none;
    border-radius: 6px;
    font-weight: bold;
    transition: background 0.2s ease;
}
a:hover {
    background: #2980b9;
}

.info {
    margin-top: 20px;
    color: #555;
}

</style>
</head>
<body>
<div class="card">
    <h1>🔒 WiShield - Panel Principal</h1>
    <?php if (!isset($_SESSION["usuario_id"])): ?>
        <a href="login.php">Iniciar sesión</a>
    <?php else: ?>
        <a href="dashboard.php">📊 Dashboard</a>
        <?php if ($_SESSION["rol"] === "admin"): ?>
            <a href="wishield.php">📝 Registro de usuarios</a>
            <a href="logs.php">🕒 Logs de acceso</a>
        <?php endif; ?>
        <a href="logout.php">🚪 Cerrar sesión</a>
        <div class="info">Sesión iniciada como <strong><?php echo
$_SESSION["nombre"]; ?></strong> (<?php echo $_SESSION["rol"]; ?></div>
        <?php endif; ?>
    </div>
</body>
</html>

```

### *Insertar\_proc.php*

```
<?php
require_once 'config.php';
$conexion = new mysqli("localhost", "root", "", "wishield");

// Datos simulados para test
$nombre = "Procedimiento Test";
$email = "proc_test@example.com";
$password = password_hash("segura123", PASSWORD_BCRYPT);
$rol = "invitado";
$mac = "00:11:22:33:44:55";
$ip = "192.168.0.250";
$tipo = "Smartphone";

$stmt = $conexion->prepare("CALL sp_insertar_usuario(?, ?, ?, ?, ?, ?, ?, ?)");
$stmt->bind_param("sssssss",
    $nombre,
    $email,
    $password,
    $rol,
    $mac,
    $ip,
    $tipo,
    CLAVE_SECRETA
);
$stmt->execute();
$resultado = $stmt->get_result();
if ($resultado && $fila = $resultado->fetch_assoc()) {
    echo "✅ Usuario insertado con ID: " . $fila['nuevo_usuario_id'];
} else {
    echo "❌ Algo salió mal al insertar.";
}
?>
```

### *Login.php*

```
<?php
session_start();
$conexion = new mysqli("localhost", "root", "", "wishield");

if ($conexion->connect_error) {
    die("Error de conexión: " . $conexion->connect_error);
}
$mensaje = "";
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $email = trim($_POST["email"]);
```



```

$password = $_POST["password"];
$stmt = $conexion->prepare("SELECT usuario_id, nombre, contraseña, rol FROM
usuarios WHERE email = ?");
$stmt->bind_param("s", $email);
$stmt->execute();
$resultado = $stmt->get_result();
if ($resultado->num_rows === 1) {
    $usuario = $resultado->fetch_assoc();
    if (password_verify($password, $usuario["contraseña"])) {
        $_SESSION["usuario_id"] = $usuario["usuario_id"];
        $_SESSION["nombre"] = $usuario["nombre"];
        $_SESSION["rol"] = $usuario["rol"];
        $stmt_log = $conexion->prepare("INSERT INTO logs_acceso (usuario_id)
VALUES (?)");
        $stmt_log->bind_param("i", $usuario["usuario_id"]);
        $stmt_log->execute();
        $stmt_log->close();
        header("Location: dashboard.php");
        exit;
    } else {
        $mensaje = "✗ Contraseña incorrecta.";
    }
} else {
    $mensaje = "✗ No se encontró un usuario con ese email.";
}
$stmt->close();
}
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Login WiShield</title>
    <style>
        body {
            font-family: Arial;
            background: #f1f2f6;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }
        .login-box {
            background: white;
            padding: 30px;
            border-radius: 12px;
            box-shadow: 0 4px 12px rgba(0,0,0,0.15);

```

```

        width: 300px;
    }
    input[type="email"], input[type="password"] {
        width: 100%;
        padding: 10px;
        margin: 12px 0;
        border-radius: 6px;
        border: 1px solid #ccc;
    }
    input[type="submit"] {
        background-color: #3498db;
        color: white;
        padding: 10px;
        border: none;
        width: 100%;
        border-radius: 6px;
        cursor: pointer;
    }
    .mensaje {
        margin-top: 10px;
        color: red;
        font-weight: bold;
    }
}
</style>
</head>
<body>
    <div class="login-box">
        <h2>🔒 Login WiShield</h2>
        <form method="POST">
            <input type="email" name="email" placeholder="Correo electrónico"
required>
            <input type="password" name="password" placeholder="Contraseña"
required>
            <input type="submit" value="Iniciar sesión">
        </form>
        <?php if ($mensaje): ?>
            <div class="mensaje"><?php echo $mensaje; ?></div>
        <?php endif; ?>
    </div>
</body>
</html>

```

### *Logout.php*

```
<?php
session_start();
session_unset(); // Borra todas las variables de sesión
session_destroy(); // Destruye la sesión
header("Location: login.php");
exit;
```

### *logs.php*

```
<?php
session_start();

if (!isset($_SESSION["usuario_id"]) || $_SESSION["rol"] !== "admin") {
    echo "<h2 style='color: red; text-align: center;'>🚫 Acceso denegado. Solo para administradores.</h2>";
    exit;
}

$conexion = new mysqli("localhost", "root", "", "wishield");
if ($conexion->connect_error) {
    die("Error de conexión: " . $conexion->connect_error);
}

$sql = "SELECT l.fecha_hora, u.nombre, u.email
        FROM logs_acceso l
        JOIN usuarios u ON l.usuario_id = u.usuario_id
        ORDER BY l.fecha_hora DESC";
$resultado = $conexion->query($sql);
?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Historial de Accesos · WiShield</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f6f9;
            margin: 0;
            padding: 0;
        }
        nav {
            background-color: #2c3e50;
            padding: 12px 20px;
            display: flex;
            gap: 20px;
```

```

}
nav a {
    color: #ecf0f1;
    text-decoration: none;
    font-weight: bold;
}
.container {
    max-width: 900px;
    margin: 30px auto;
    padding: 20px;
    background: white;
    border-radius: 12px;
    box-shadow: 0 3px 10px rgba(0,0,0,0.1);
}
table {
    width: 100%;
    border-collapse: collapse;
    margin-top: 15px;
}
th, td {
    border: 1px solid #ddd;
    padding: 10px;
}
th {
    background-color: #3498db;
    color: white;
}
</style>
</head>
<body>
<nav>
    <a href="wishield.php">🏠 Registro</a>
    <a href="dashboard.php">📊 Dashboard</a>
    <a href="logs.php">🕒 Logs de acceso</a>
    <span style="flex-grow: 1;"></span>
    <span style="color: #ecf0f1;">👤 <?php echo $_SESSION["nombre"]; ?> (<?php
echo $_SESSION["rol"]; ?>)</span>
    <a href="logout.php">🚪 Cerrar sesión</a>
</nav>
<div class="container">
    <h2>🕒 Historial de accesos</h2>
    <table>
        <tr>
            <th>Nombre</th>
            <th>Email</th>
            <th>Fecha y hora de acceso</th>

```

```

</tr>
<?php while($fila = $resultado->fetch_assoc()): ?>
<tr>
    <td><?php echo htmlspecialchars($fila['nombre']); ?></td>
    <td><?php echo htmlspecialchars($fila['email']); ?></td>
    <td><?php echo htmlspecialchars($fila['fecha_hora']); ?></td>
</tr>
<?php endwhile; ?>
</table>
</div>
</body>
</html>

```

### *Recuperar.php*

```

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Recuperar contraseña</title>
</head>
<body>
    <h2>🔒 ¿Olvidaste tu contraseña?</h2>
    <form method="POST" action="enviar_token.php">
        <label>Introduce tu email:
            <input type="email" name="email" required>
        </label>
        <br><br>
        <input type="submit" value="Enviar enlace de recuperación">
    </form>
</body>
</html>

```

### *Reporte\_semanal.php*

```

<?php
require 'src/PHPMailer.php';
require 'src/SMTP.php';
require 'src/Exception.php';
require 'language/phpmailer.lang-es.php';
use PHPMailer\PHPMailer\PHPMailer;
$conexion = new mysqli("localhost", "root", "", "wishield");
if ($conexion->connect_error) {
    die("Error de conexión: " . $conexion->connect_error);
}
// Consultas
$nuevos_usuarios = $conexion->query("SELECT COUNT(*) AS total FROM usuarios
WHERE fecha_creacion >= CURDATE() - INTERVAL 7 DAY")->fetch_assoc()['total'];

```

```

$nuevas_sesiones = $conexion->query("SELECT COUNT(*) AS total FROM
Sesiones_Conexion WHERE timestamp_inicio >= CURDATE() - INTERVAL 7 DAY")-
>fetch_assoc()['total'];
$nuevas_vulnerabilidades = $conexion->query("SELECT COUNT(*) AS total FROM
Vulnerabilidades WHERE fecha_deteccion >= CURDATE() - INTERVAL 7 DAY")-
>fetch_assoc()['total'];
// Obtener correos de administradores
$correos = [];
$result = $conexion->query("SELECT email FROM usuarios WHERE rol = 'admin'");
while ($row = $result->fetch_assoc()) {
    $correos[] = $row['email'];
}
$conexion->close();
// HTML del correo
$fecha = date('Y-m-d');
$reporteHTML = "
    <h2><img alt="WiShield logo" data-bbox="205 365 225 380"/> Reporte Semanal WiShield - $fecha</h2>
    <ul>
        <li><img alt="User icon" data-bbox="205 405 225 420"/> Nuevos usuarios registrados: <strong>$nuevos_usuarios</strong></li>
        <li><img alt="Key icon" data-bbox="205 425 225 440"/> Sesiones iniciadas: <strong>$nuevas_sesiones</strong></li>
        <li><img alt="Shield icon" data-bbox="205 445 225 460"/> Vulnerabilidades detectadas:
        <strong>$nuevas_vulnerabilidades</strong></li>
    </ul>
    <p style='color: #888;'>Enviado automáticamente por el sistema WiShield</p>
";
// Envío del correo
$mail = new PHPMailer(true);
try {
    $mail->isSMTP();
    $mail->Host    = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'tuemail@gmail.com';    // <- tu email real
    $mail->Password = 'contraseña_de_aplicacion'; // <- tu contraseña de
aplicación
    $mail->SMTPSecure = 'tls';
    $mail->Port    = 587;
    $mail->CharSet  = 'UTF-8';

    $mail->setFrom('tuemail@gmail.com', 'Sistema WiShield');

    foreach ($correos as $correo) {
        $mail->addAddress($correo);
    }
    $mail->isHTML(true);
    $mail->Subject = "<img alt="Document icon" data-bbox="315 860 335 875"/> Reporte semanal WiShield ($fecha)";
    $mail->Body    = $reporteHTML;

```

```

$mail->send();
echo "✅ Reporte enviado correctamente a administradores.";
} catch (Exception $e) {
    echo "❌ Error al enviar el reporte: {$mail->ErrorInfo}";
}
?>

```

### *Reset\_password.php*

```

<?php
require_once 'config.php';
$conexion = new mysqli("localhost", "root", "", "wishield");
$token = $_GET['token'] ?? "";
$valido = false;
// Verificar token válido y no caducado
$stmt = $conexion->prepare("SELECT usuario_id FROM tokens_recuperacion
    WHERE token = ? AND expiracion > NOW()");
$stmt->bind_param("s", $token);
$stmt->execute();
$resultado = $stmt->get_result();
if ($resultado->num_rows === 1) {
    $valido = true;
    $usuario = $resultado->fetch_assoc();
    $usuario_id = $usuario['usuario_id'];
}
if ($_SERVER["REQUEST_METHOD"] === "POST" &&
    isset($_POST["nueva_contraseña"])) {
    $nueva_contraseña = password_hash($_POST["nueva_contraseña"],
    PASSWORD_BCRYPT);
    // Actualizar contraseña
    $update = $conexion->prepare("UPDATE usuarios SET contraseña = ? WHERE
    usuario_id = ?");
    $update->bind_param("si", $nueva_contraseña, $_POST['usuario_id']);
    $update->execute();
    // Eliminar token
    $delete = $conexion->prepare("DELETE FROM tokens_recuperacion WHERE
    usuario_id = ?");
    $delete->bind_param("i", $_POST['usuario_id']);
    $delete->execute();
    echo "<h3 style='color:green;'>✅ Tu contraseña se ha actualizado
    correctamente. Ya puedes iniciar sesión.</h3>";
    exit;
}
?>
<!DOCTYPE html>
<html lang="es">

```

```

<head>
  <meta charset="UTF-8">
  <title>Restablecer contraseña</title>
</head>
<body>
  <h2>🔒 Restablecer contraseña</h2>
  <?php if ($valido): ?>
    <form method="POST">
      <input type="hidden" name="usuario_id" value="<?php echo $usuario_id;
?>">
      <label>Nueva contraseña:
        <input type="password" name="nueva_contraseña" required>
      </label><br><br>
      <input type="submit" value="Actualizar contraseña">
    </form>
  <?php else: ?>
    <p style="color:red;">🚫 Este enlace no es válido o ha expirado.</p>
  <?php endif; ?>
</body>
</html>

```



### Glosario:

- **AES (Advanced Encryption Standard):** Algoritmo de cifrado simétrico utilizado para proteger datos sensibles, como direcciones MAC o IPs.
- **Backup:** Copia de seguridad de la base de datos que permite restaurar el sistema en caso de fallo.
- **Base de Datos Relacional:** Modelo de almacenamiento estructurado en tablas conectadas por relaciones, como MySQL.
- **Bash:** Lenguaje de scripting en sistemas Unix/Linux. Se ha utilizado para automatizar tareas como escaneo de red y backups.
- **Chart.js:** Librería de JavaScript usada para generar gráficas dinámicas en el dashboard web.
- **Crontab:** Programador de tareas de Unix/Linux que permite ejecutar scripts automáticamente en horarios definidos.
- **CSV (Comma-Separated Values):** Formato de archivo de texto plano para intercambio de datos tabulados, compatible con Power BI.
- **Dashboard:** Panel de control visual que muestra indicadores clave (KPI) como vulnerabilidades, sesiones o usuarios activos.
- **Dispositivo IoT: Objeto** conectado a internet (como un smartphone, TV o consola) que puede enviar o recibir datos.
- **Encriptación:** Proceso de transformar información en un formato seguro e ilegible sin una clave.
- **Fetch\_assoc():** Función de PHP usada para recuperar resultados de una consulta SQL como array asociativo.
- **FOREIGN KEY:** Clave foránea que conecta dos tablas entre sí para mantener la integridad referencial.
- **Grafana / Power BI:** Herramientas de Business Intelligence para visualización y análisis de datos.
- **MySQL:** Sistema de gestión de bases de datos usado en el backend de WiShield.
- **nmap:** Herramienta de escaneo de redes utilizada para detectar dispositivos conectados.
- **PHP:** Lenguaje de programación web utilizado en la creación de la interfaz WiShield.
- **phpMyAdmin:** Aplicación web para gestionar bases de datos MySQL de forma visual.
- **Raspberry Pi:** Minicomputadora de bajo coste usada como nodo de análisis, recolección de datos y monitorización.
- **Rol (de usuario):** Permisos asignados a cada usuario, como admin, estudiante o invitado, que limitan su acceso al sistema.

- **Script:** Archivo de instrucciones automatizadas que ejecutan acciones específicas en el sistema.
- **SQL (Structured Query Language):** Lenguaje usado para consultar y manipular datos en bases de datos relacionales.
- **Stored Procedure (Procedimiento Almacenado):** Conjunto de instrucciones SQL predefinidas que se ejecutan en el servidor para optimizar tareas.
- **Trigger (Disparador):** Fragmento de código SQL que se ejecuta automáticamente cuando ocurre un evento en la base de datos (como insertar un dispositivo).
- **Usuario Admin:** Usuario con privilegios completos dentro del sistema, incluyendo el acceso a los logs, registros y configuración.
- **Vulnerabilidad:** Debilidad de seguridad detectada en un dispositivo o sistema, clasificada por su severidad (alta, media, baja...).

## Referencias y Recursos:

- <https://www.w3schools.com/sql/default.asp>
- <https://dspace.uazuay.edu.ec/bitstream/datos/2326/1/06827.pdf>
- <https://dev.mysql.com/doc/refman/8.0/en/scalar-subqueries.html>
- <https://medium.com/@manutej/mastering-sql-subqueries-comprehensive-guide-633dc50ac294>
- <https://mysqlia.com.ar/tutoriales/>
- <https://www.arteco-consulting.com/articulos/tutorial-sql/>
- <https://www.datacamp.com/tutorial/views-in-sql>
- <https://mapamental.com.es/diagrama-de-bases-de-datos/>
- <https://medium.com/enredando-con-bases-de-datos/bbdd-diagrama-b%C3%A1sico-5591f9f919b5>
- [https://itacademy.barcelonactiva.cat/pluginfile.php/29116/mod\\_page/content/13/Editor%20Power%20Query.pdf](https://itacademy.barcelonactiva.cat/pluginfile.php/29116/mod_page/content/13/Editor%20Power%20Query.pdf)
- [https://itacademy.barcelonactiva.cat/pluginfile.php/29116/mod\\_page/content/13/Or%C3%ADgens%20de%20dades%20i%20connectors.pdf](https://itacademy.barcelonactiva.cat/pluginfile.php/29116/mod_page/content/13/Or%C3%ADgens%20de%20dades%20i%20connectors.pdf)
- <https://mysqlia.com.ar/general/connect-power-bi-with-mysql/>
- <https://keepcoding.io/blog/que-es-la-granularidad-de-los-datos/>
- [https://itacademy.barcelonactiva.cat/pluginfile.php/29116/mod\\_page/content/13/Oriegen%20C%20transformaci%C3%B3n%20i%20c%C3%A0rrega%20de%20dades.pdf](https://itacademy.barcelonactiva.cat/pluginfile.php/29116/mod_page/content/13/Oriegen%20C%20transformaci%C3%B3n%20i%20c%C3%A0rrega%20de%20dades.pdf)
- <https://www.akkio.com/post/kpis-for-data-analytics>
- <https://blog.bismart.com/10-mejores-power-bi-dashboards-2021>
- [https://itacademy.barcelonactiva.cat/pluginfile.php/29122/mod\\_page/content/7/DAX.pdf](https://itacademy.barcelonactiva.cat/pluginfile.php/29122/mod_page/content/7/DAX.pdf)
- <https://www.pontia.tech/10-consejos-buenas-practicas-para-mejorar-la-visualizacion-de-datos-en-power-bi/>
- <https://biuwer.com/es/blog/como-elegir-el-grafico-adecuado-para-tus-datos/>
- <https://www.datacamp.com/tutorial/power-bi-dashboards-vs-reports-a-comprehensive-guide>
- <https://www.datacamp.com/es/tutorial/sql-triggers>
- [https://codigofacilito.com/articulos/triggers\\_mysql](https://codigofacilito.com/articulos/triggers_mysql)
- [https://es.wikipedia.org/wiki/Procedimiento\\_almacenado](https://es.wikipedia.org/wiki/Procedimiento_almacenado)
- [https://docs.oracle.com/cd/E12151\\_01/doc.150/e12155/triggers\\_proc\\_mysql.htm#g1049668](https://docs.oracle.com/cd/E12151_01/doc.150/e12155/triggers_proc_mysql.htm#g1049668)
- <https://github.com/suresh-pokharel/forgot-password>
- [https://www.youtube.com/watch?v=m52ljs78S24&list=PL0eyrZgxdwhwwQQZA79OzYwI5ewA7HQih&ab\\_channel=DaniKrossing](https://www.youtube.com/watch?v=m52ljs78S24&list=PL0eyrZgxdwhwwQQZA79OzYwI5ewA7HQih&ab_channel=DaniKrossing)
- [https://www.youtube.com/watch?v=LC9GaXkdxF8&list=PL0eyrZgxdwhyfSPF6sHd7Ibm3R0THoOJd&ab\\_channel=DaniKrossing](https://www.youtube.com/watch?v=LC9GaXkdxF8&list=PL0eyrZgxdwhyfSPF6sHd7Ibm3R0THoOJd&ab_channel=DaniKrossing)
- <https://medium.com/%40soulaimaneyh/secure-your-php-application-with-encryption-functions-fc8b8ebd019d>
- <https://stackoverflow.com/questions/1289061/best-way-to-use-php-to-encrypt-and-decrypt-passwords>
- <https://stackify.com/how-to-load-test-your-php-website/>
- <https://www.inmotionhosting.com/support/server/server-usage/how-to-stress-test-your-website/>
- [https://reintech.io/blog/creating-interactive-charts-graphs-php-pchart-library?utm\\_source=chatgpt.com](https://reintech.io/blog/creating-interactive-charts-graphs-php-pchart-library?utm_source=chatgpt.com)
- <https://canvasjs.com/php-charts/>
- [https://www.browserstack.com/guide/php-web-development?utm\\_source=chatgpt.com](https://www.browserstack.com/guide/php-web-development?utm_source=chatgpt.com)
- [https://www.w3schools.com/php/?utm\\_source=chatgpt.com](https://www.w3schools.com/php/?utm_source=chatgpt.com)