INTEGRANTES:

- Samuel Alzate
- Juan Jose Rivera
- Maria Camila Rojas

Fase 1 -Requisitos de Seguridad (Teoría + Redacción)

Control de acceso:
Restricción del acceso a la información y sistemas basados en roles y permisos, para evitar la exposición a usuarios no autorizados.

Seguridad de red:
Implementación de firewalls, sistemas de detección de intrusiones, y VPNs para proteger la red contra ataques externos e internos.

Protección de terminales:
Uso de antivirus, antimalware, y la aplicación de parches de seguridad para proteger los dispositivos de los usuarios.

Seguridad de la nube:
Implementación de medidas de seguridad específicas para los servicios en la nube, como cifrado de datos, control de acceso y monitorización.

Copias de seguridad:
Realización periódica de copias de seguridad de los datos para poder recuperarlos en caso de pérdida o corrupción.

Gestión de contraseñas:
Uso de contraseñas fuertes y complejas, así como la implementación de políticas de gestión de contraseñas.

Seguridad física:
Protección de los servidores, equipos de red y otros activos de tecnología en instalaciones seguras y restringidas.

Formación y conciencia:
Capacitación a los empleados sobre ciberseguridad, para que reconozcan y eviten amenazas, así como para fomentar una cultura de seguridad.

Cumplimiento normativo:

Cumplimiento de las regulaciones y estándares de ciberseguridad, como ISO 27001 y GDPR.
Gestión de riesgos:

Identificación, evaluación y mitigación de los riesgos de ciberseguridad.

Monitorización y respuesta a incidentes:

Implementación de sistemas de monitorización para detectar amenazas y respuesta a incidentes de ciberseguridad.

Beneficios de implementar requisitos de seguridad:

Protección de datos confidenciales: Evita la pérdida, robo o divulgación no autorizada de información sensible.

Evitación de interrupciones en el negocio: Reduce el riesgo de cortes de servicio o inactividad de los sistemas.

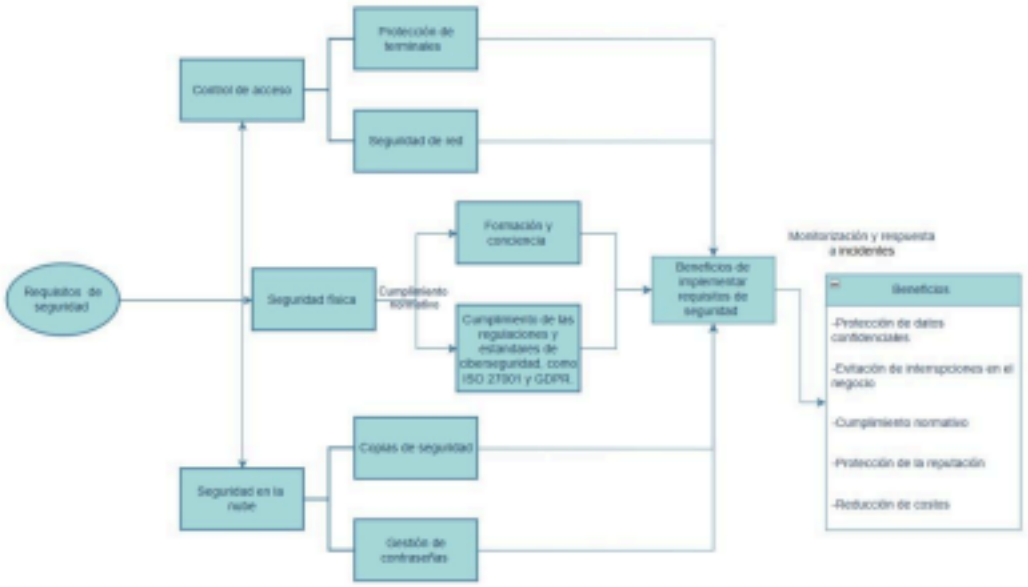
Cumplimiento normativo: Garantiza el cumplimiento de las regulaciones y estándares de ciberseguridad.

Protección de la reputación: Evita la pérdida de confianza de clientes y socios comerciales.

Reducción de costos: Mitiga los costos asociados con incidentes de ciberseguridad, como la recuperación de datos y las multas.

Fase 2 – Diseño Seguro (Análisis + Diagrama)

Diagrama de arquitectura



Archivo config.py

```python
app = Flask(__name__)
app.config['SECRET_KEY'] = 'tu_clave_secreta'  # Cambia esto por una clave más segura
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///database.db'  # Base de datos SQL
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

db = SQLAlchemy(app)

login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view = 'login'
```

Fase 3 – Implementación Segura (Código)

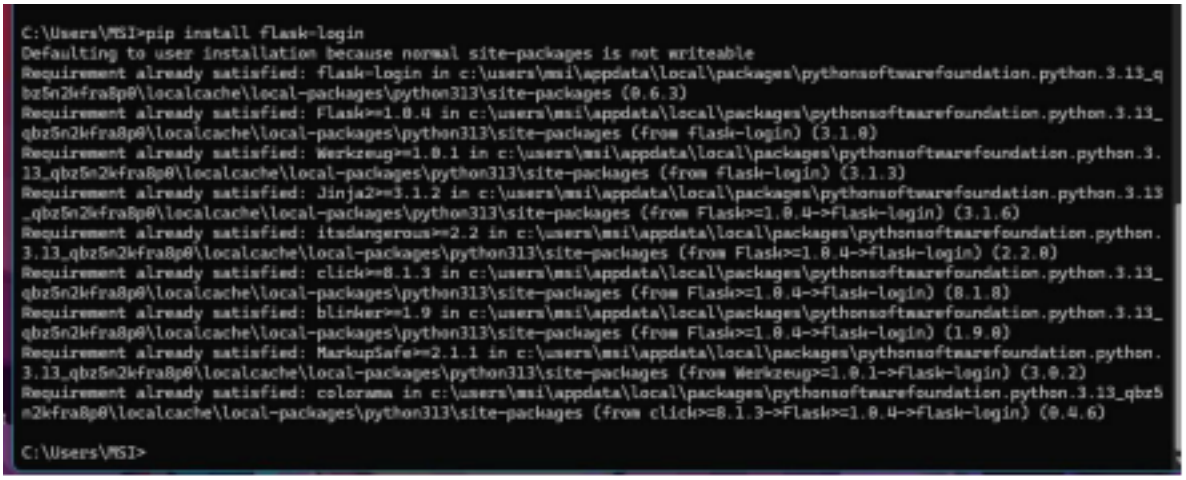
Flask

```
C:\Users\MSI>pip install flask
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: flask in c:\users\msi\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5n2k
fra8p0\localcache\local-packages\python313\site-packages (3.1.0)
Requirement already satisfied: Werkzeug>=3.1 in c:\users\msi\appdata\local\packages\pythonsoftwarefoundation.python.3.13
_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from flask) (3.1.3)
Requirement already satisfied: Jinja2>=3.1.2 in c:\users\msi\appdata\local\packages\pythonsoftwarefoundation.python.3.13
_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from flask) (3.1.6)
Requirement already satisfied: itsdangerous>=2.2 in c:\users\msi\appdata\local\packages\pythonsoftwarefoundation.python.
3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from flask) (2.2.0)
Requirement already satisfied: click>=8.1.3 in c:\users\msi\appdata\local\packages\pythonsoftwarefoundation.python.3.13_
qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from flask) (8.1.8)
Requirement already satisfied: blinker>=1.9 in c:\users\msi\appdata\local\packages\pythonsoftwarefoundation.python.3.13_
qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from flask) (1.9.0)
Requirement already satisfied: colorama in c:\users\msi\appdata\local\packages\pythonsoftwarefoundation.python.3.13_qbz5
n2kfra8p0\localcache\local-packages\python313\site-packages (from click>=8.1.3->flask) (0.4.6)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\msi\appdata\local\packages\pythonsoftwarefoundation.python.3.
13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from Jinja2>=3.1.2->flask) (3.0.2)
```

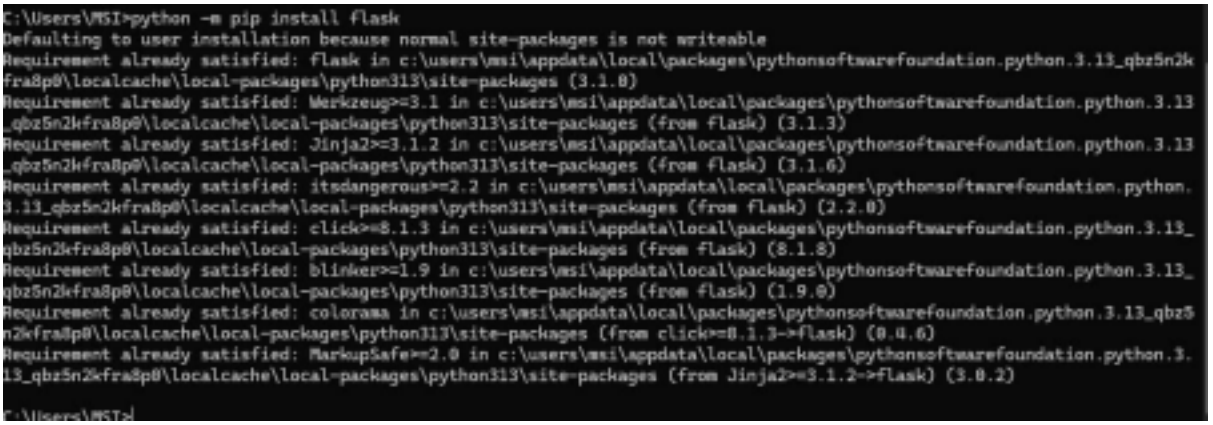
Sqlalchemy

```
C:\Users\MSI>pip install sqlalchemy
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: sqlalchemy in c:\users\msi\appdata\local\packages\pythonsoftwarefoundation.python.3.13_
z5n2kfra8p0\localcache\local-packages\python313\site-packages (2.0.40)
Requirement already satisfied: greenlet>=1 in c:\users\msi\appdata\local\packages\pythonsoftwarefoundation.python.3.13_
qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from sqlalchemy) (3.2.1)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\msi\appdata\local\packages\pythonsoftwarefoundation
python.3.13_qbz5n2kfra8p0\localcache\local-packages\python313\site-packages (from sqlalchemy) (4.13.2)
```

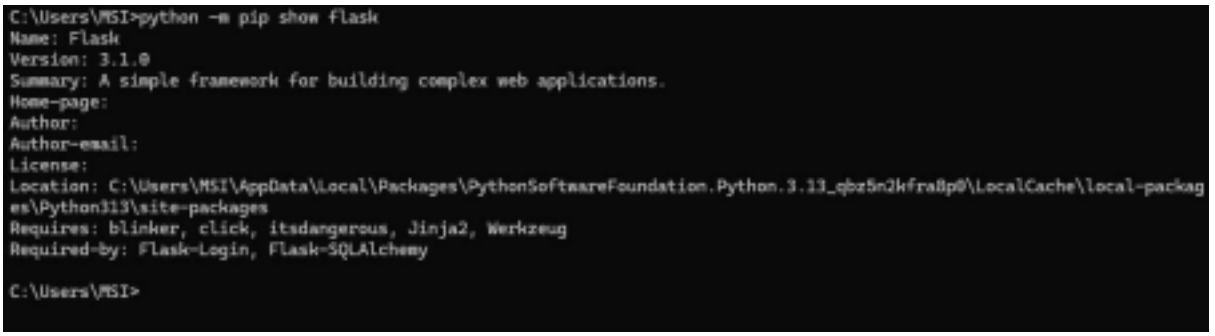
Flask-login



Instalación de flask en python



Verificar que flask este en Python



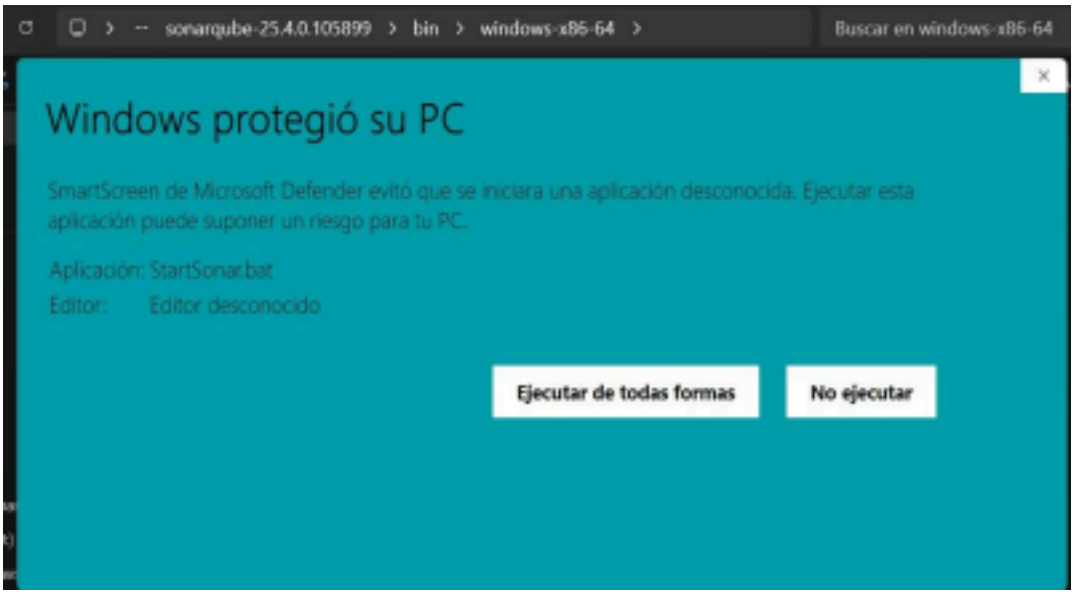
Uso del ORM

```python
# --- Inicialización del ORM (SQLAlchemy) ---
db = SQLAlchemy(app)

# --- Definición de modelos con el ORM ---
class User(db.Model):  # ORM: Representación de la tabla 'User'
    id = db.Column(db.Integer, primary_key=True)  # Llave primaria
    username = db.Column(db.String(80), unique=True, nullable=False)  # Nombre de usuario único
    email = db.Column(db.String(120), unique=True, nullable=False)  # Email único

    def __repr__(self):
        return f"<User {self.username}>"

# --- Rutas de la aplicación ---
@app.route('/')
def index():
    users = User.query.all()  # ORM: Consulta de todos los usuarios
    return render_template('index.html', users=users)

@app.route('/add', methods=['POST'])
def add_user():
    if request.method == 'POST':
        username = request.form['username']
        email = request.form['email']
        # ORM: Creación de un nuevo registro en la tabla 'User'
        new_user = User(username=username, email=email)
        db.session.add(new_user)  # ORM: Agregar el usuario a la sesión
        db.session.commit()  # ORM: Confirmar cambios en la base de datos
        return redirect(url_for('index'))

@app.route('/delete/<int:id>')
def delete_user(id):
    user_to_delete = User.query.get_or_404(id)  # ORM: Búsqueda de usuario por ID
    db.session.delete(user_to_delete)  # ORM: Eliminar el registro
    db.session.commit()  # ORM: Confirmar cambios
    return redirect(url_for('index'))
```

```python
# --- Inicialización de la base de datos ---
with app.app_context():
    db.create_all()  # ORM: Crear las tablas en la base de datos

# --- Ejecutar la aplicación ---
if __name__ == '__main__':
    app.run(debug=True)
```

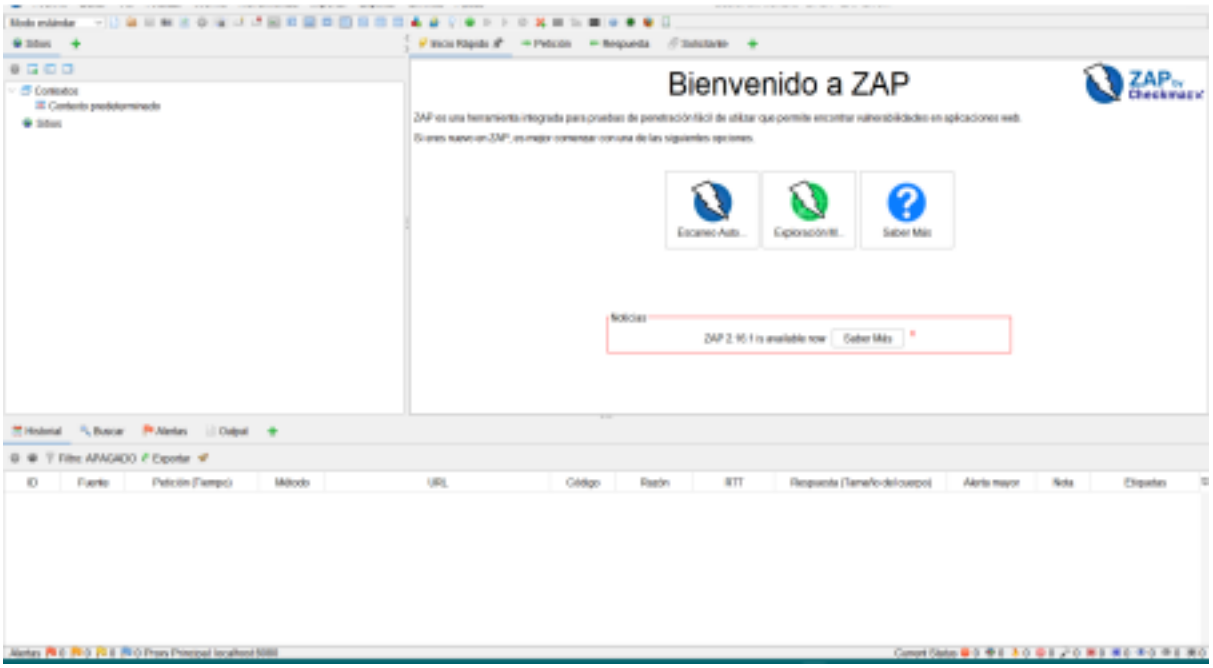
Fase 4 – Pruebas de Seguridad (SAST y DAST)

- **Usar SonarQube: incluir captura del informe.**

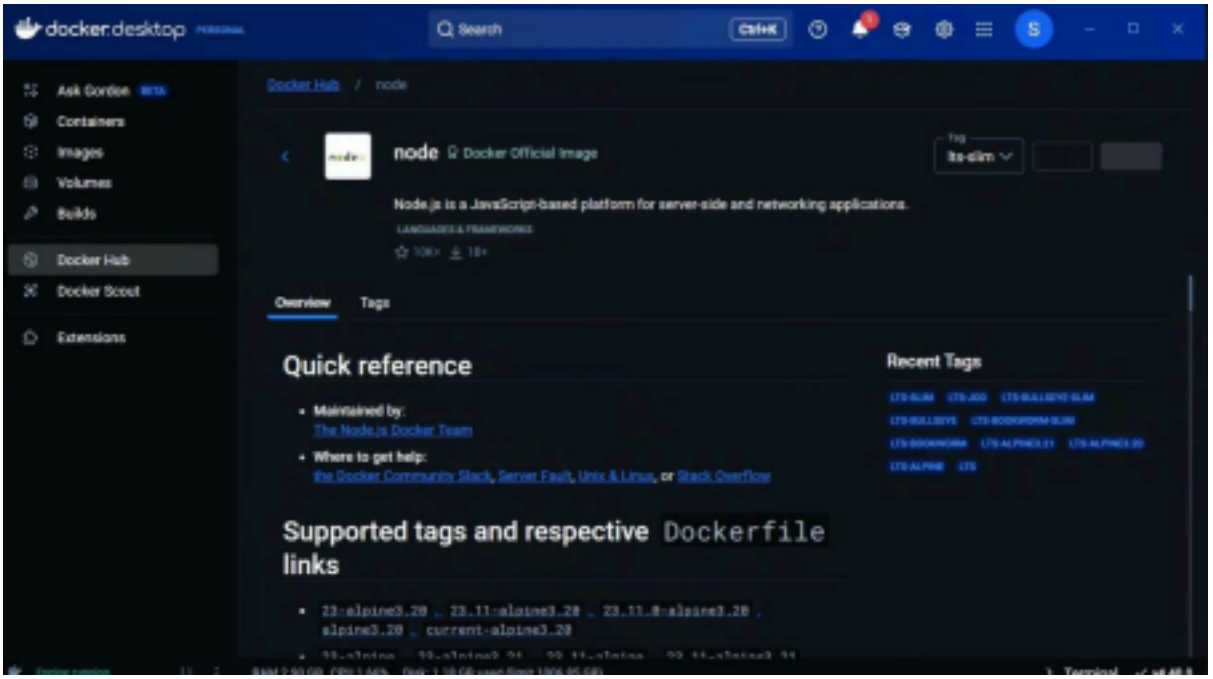


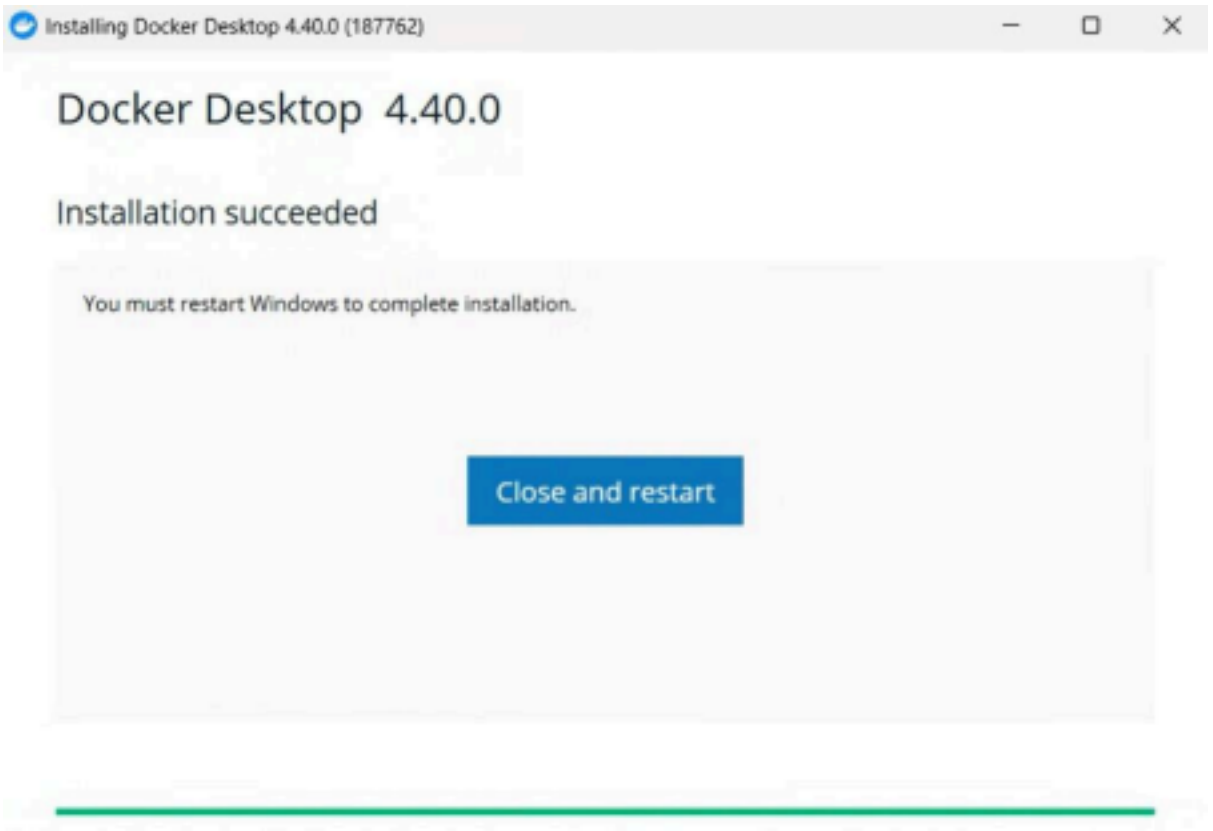
Windows Defender SmartScreen está protegiendo tu computadora, por ende cuando se intenta ejecutar el proyecto nos envía esta advertencia diciéndonos que windows no reconoce el programa como seguro y a la hora de desactivar el defenders abre un cmd por 0.3 segundos y vuelve a cerrarlo, esto pasa porque directamente el s.o que lo cierra ya que lo detecta como una amenaza para todo el ordenador.

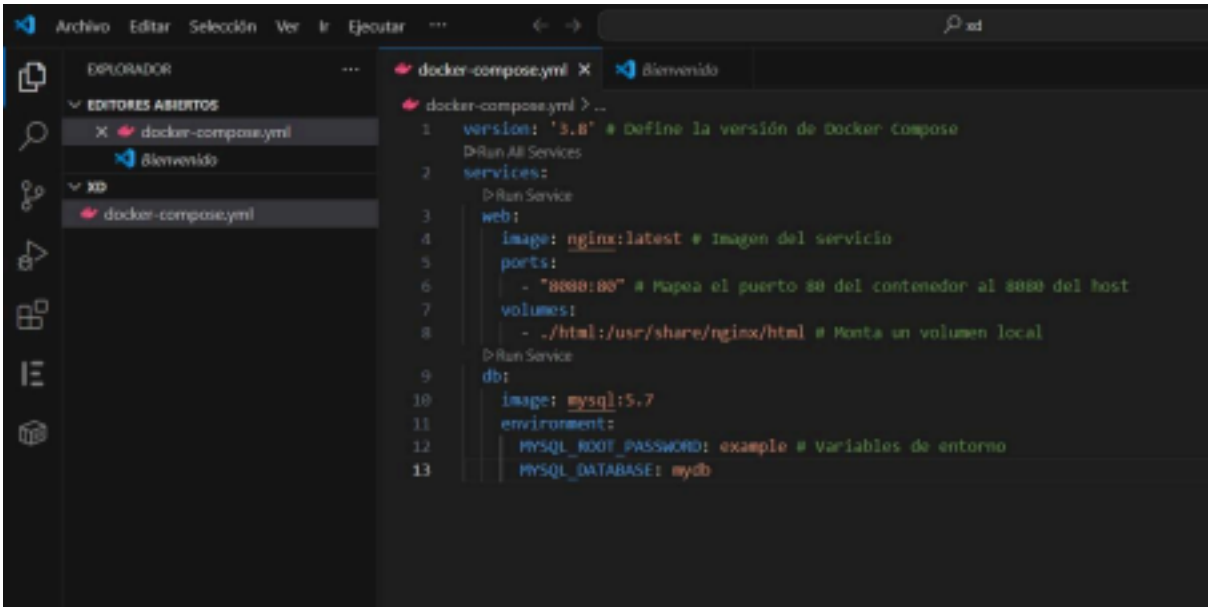
OWASP ZAP



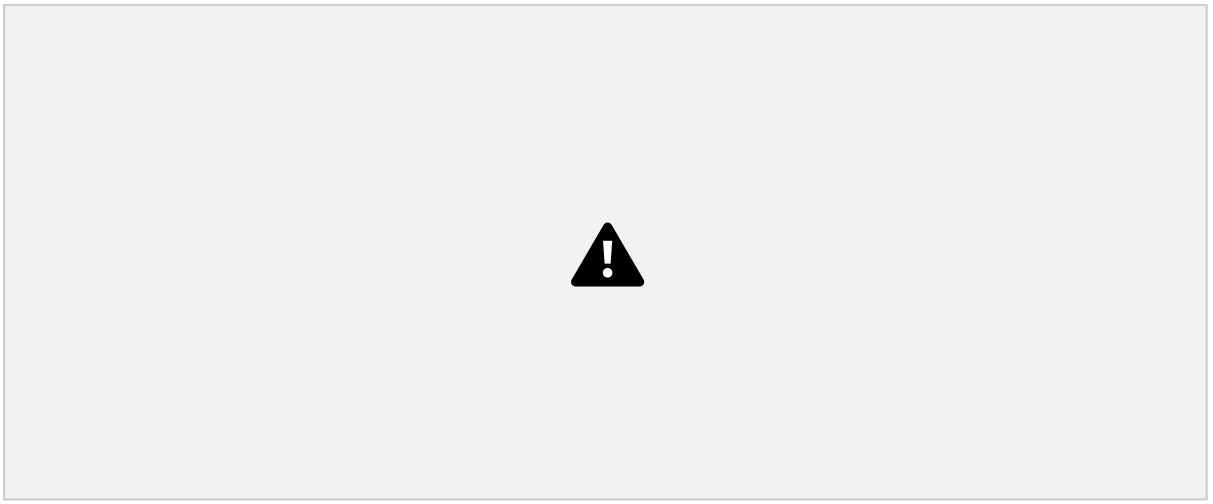
Fase 5 – Despliegue Seguro (Docker)


docker desktop
Search
Ctrl+K
Ask Gordon
Containers
Images
Volumes
Builds
Docker Hub
Docker Scout
Extensions
Docker Hub / node
node
Docker Official Image
Node.js is a JavaScript-based platform for server-side and networking applications.
Overview
Tags
Quick reference
Maintained by:
The Node.js Docker Team
Where to get help:
the Docker Community Slack, Server Fault, Unix & Linux, or Stack Overflow
Recent Tags
Supported tags and respective Dockerfile links

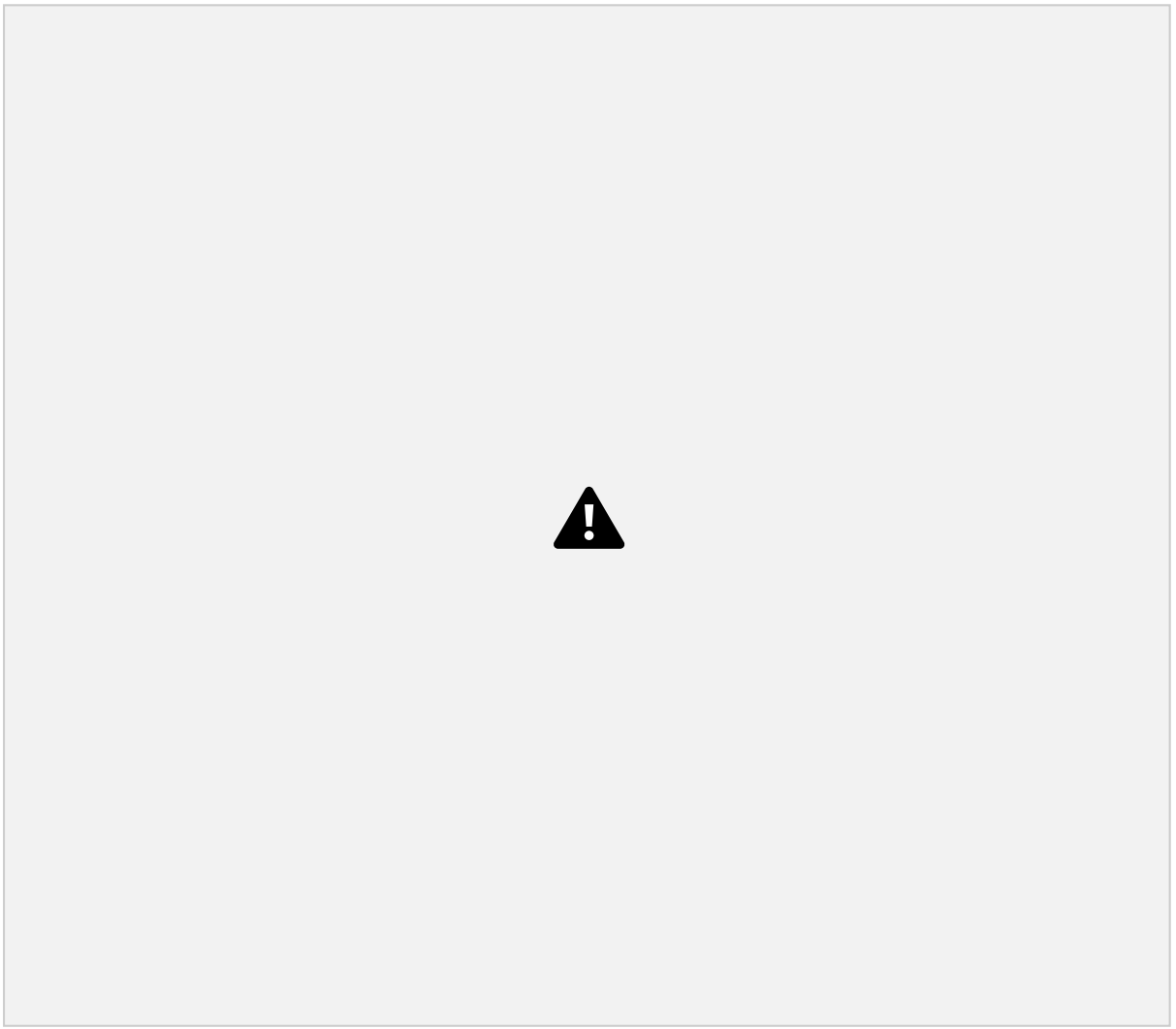

Installing Docker Desktop 4.40.0 (187762)
Docker Desktop 4.40.0
Installation succeeded
You must restart Windows to complete installation.
Close and restart




```
 Directorio de C:\Users\Samue\xd

28/04/2025  06:16 p. m.    <DIR>          .
28/04/2025  12:40 p. m.    <DIR>          ..
28/04/2025  06:13 p. m.               414 docker-compose.yml
               1 archivos            414 bytes
               2 dirs  678.940.536.832 bytes libres

C:\Users\Samue\xd>docker-compose updocker-compose up
Usage:  docker compose [OPTIONS] COMMAND

Define and run multi-container applications with Docker

Options:
      --all-resources              Include all resources, even those not
                                   used by services
      --ansi string                Control when to print ANSI control
                                   characters ("never"|"always"|"auto")
                                   (default "auto")
      --compatibility              Run compose in backward compatibility mode
      --dry-run                    Execute command in dry run mode
      --env-file stringArray       Specify an alternate environment file
  -f, --file stringArray           Compose configuration files
      --parallel int               Control max parallelism, -1 for
                                   unlimited (default -1)
      --profile stringArray        Specify a profile to enable
      --progress string            Set type of progress output (auto,
                                   tty, plain, json, quiet) (default "auto")
      --project-directory string   Specify an alternate working directory
                                   (default: the path of the, first
                                   specified, Compose file)
  -p, --project-name string        Project name

Commands:
  attach      Attach local standard input, output, and error streams to a service's running container
  build       Build or rebuild services
  commit      Create a new image from a service container's changes
  config      Parse, resolve and render compose file in canonical format
  cp          Copy files/folders between a service container and the local filesystem
  create      Creates containers for a service
  down        Stop and remove containers, networks
  events      Receive real time events from containers
  exec        Execute a command in a running container
  export      Export a service container's filesystem as a tar archive
  images      List images used by the created containers
  kill        Force stop service containers
  logs        View output from containers
```



Dockerfile



Fase 6 – Mantenimiento y Monitorización

1. Estrategia de actualización y parches

a. Planificación de actualizaciones

- **Frecuencia de revisiones:** Implementa una política de revisión mensual para dependencias y paquetes (como Flask, SQLAlchemy y Werkzeug) utilizando herramientas como pip list --outdated.
- **Actualizaciones críticas:** Prioriza actualizaciones de seguridad. Configura alertas automáticas con GitHub Dependabot o PyPI para identificar vulnerabilidades conocidas.

b. Proceso de despliegue de parches

- **Entorno de prueba:** Siempre prueba las actualizaciones y parches en un entorno de preproducción antes de implementarlos en producción.

- **Automatización:** Usa herramientas como Docker para crear contenedores consistentes, facilitando el despliegue de nuevas versiones con cambios de configuración o parches aplicados.

- **Historial de cambios:** Lleva un registro en un archivo CHANGELOG.md para documentar cada actualización o parche aplicado.

c. Automatización de tareas

- **Scripts de actualización:** Crea scripts de mantenimiento, por ejemplo, para realizar migraciones de bases de datos con Flask-Migrate:flask db upgrade

2. Configuración de monitoreo básico

El monitoreo es crucial para garantizar el rendimiento y la disponibilidad de tu

aplicación. **a.** Usar ELK Stack (Elasticsearch, Logstash, Kibana)

ELK Stack es ideal para centralizar y visualizar logs. Aquí tienes una configuración básica: - Configurar Flask-Logging: Agrega logs en tu aplicación Flask:import logging from logging.handlers import RotatingFileHandler

```
if not app.debug:
 handler = RotatingFileHandler('app.log', maxBytes=10000, backupCount=1)
handler.setLevel(logging.INFO)
 app.logger.addHandler(handler)
```

- Instalar ELK Stack:- Descarga e instala Elasticsearch, Logstash y Kibana desde Elastic.co.
- Configura un archivo de Logstash (logstash.conf) para leer los logs de tu aplicación:input {

```
file {
 path => "/ruta/a/app.log"
 start_position => "beginning"
 }
}
 output {
 elasticsearch {
 hosts => ["http://localhost:9200"]
 }
}
```

- Iniciar Kibana:- Accede a Kibana en http://localhost:5601 para visualizar los logs procesados.

b. Usar Prometheus para métricas
Prometheus se usa para monitorear métricas en tiempo real y generar alertas.
- Exponer métricas en Flask con Prometheus-Client: Instala la librería:pip install prometheus-client
Integra métricas en tu aplicación:from prometheus_client import Counter, generate_latest from flask import Response

```
REQUEST_COUNT = Counter('request_count', 'Total de solicitudes', ['method', 'endpoint'])

@app.before_request
def before_request():
  REQUEST_COUNT.labels(method=request.method, endpoint=request.path).inc()

@app.route('/metrics')
def metrics():
 return Response(generate_latest(), mimetype='text/plain')
```

- Configurar Prometheus:- Agrega la URL de tus métricas (http://localhost:5000/metrics) en el archivo de configuración de Prometheus:scrape_configs:

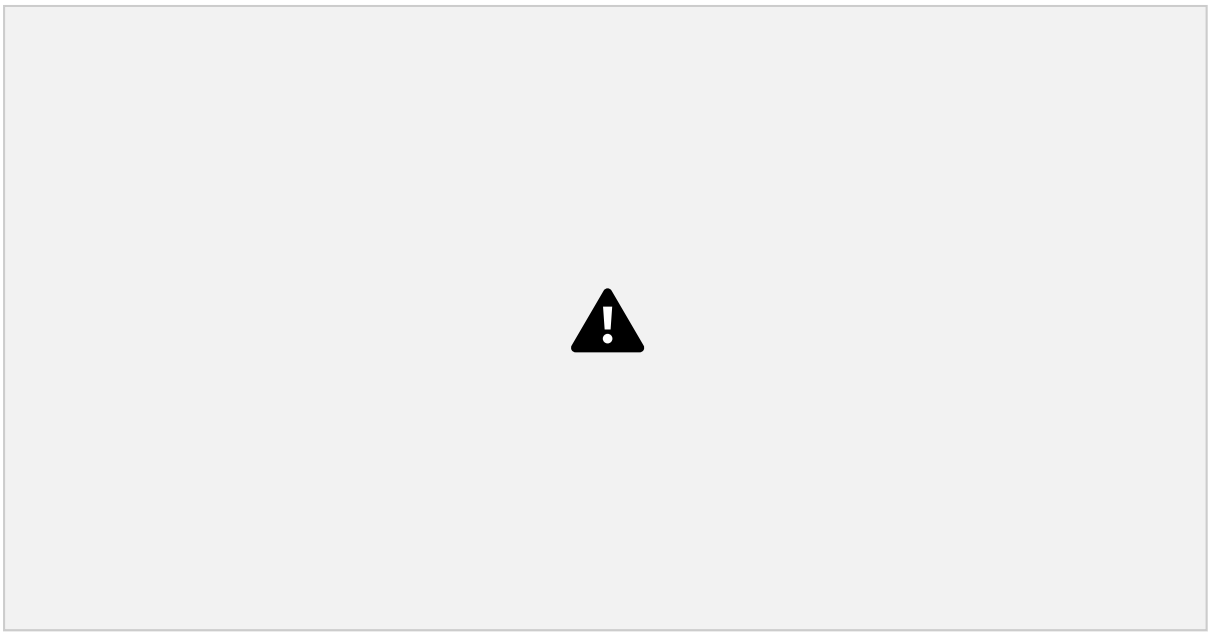
```
 - job_name: 'flask_app'
 static_configs:
 - targets: ['localhost:5000']
```

- Visualizar métricas:- Inicia Prometheus y accede a http://localhost:9090 para consultar métricas como solicitudes por endpoint.

PoC 2: Evaluación de riesgos y plan de tratamiento

Parte 1 – Escaneo de Vulnerabilidades:

Ya adjuntamos las fotos de descarga del Owasp Zap ahora agregaremos las de nessus y como vamos a crear el entorno controlado puede ser en una Vb o en lo que nos pueda servir o DVWA



VULNERABILIDADES:

- **Cabecera Content Security Policy (CSP) No configurada**

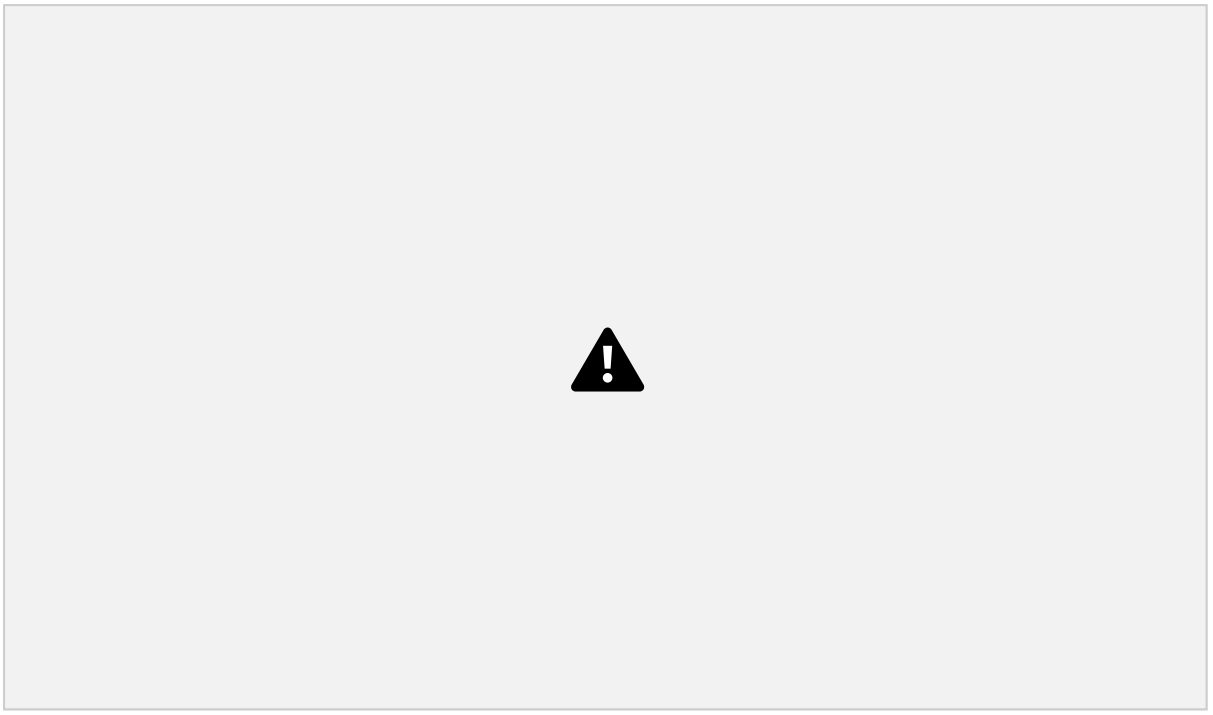
  La aplicación no implementa una política de seguridad de contenidos (CSP), debemos recordar que (CSP) ayuda a prevenir ataques como Cross (XSS) SIte Scripting al restringir de donde puede cargar el contenido la página, esto puede tener un riesgo alto.

- **Divulgación de información mediante la cabecera HTTP “Server”**

  El servidor está exponiendo su tecnología backend (por ejemplo, ASP.NET) mediante la cabecera X-Powered-By. Esto le facilita a los atacantes identificar tecnologías y versiones específicas para explotar vulnerabilidades conocidas, lo cual puede tener un riesgo de medio.

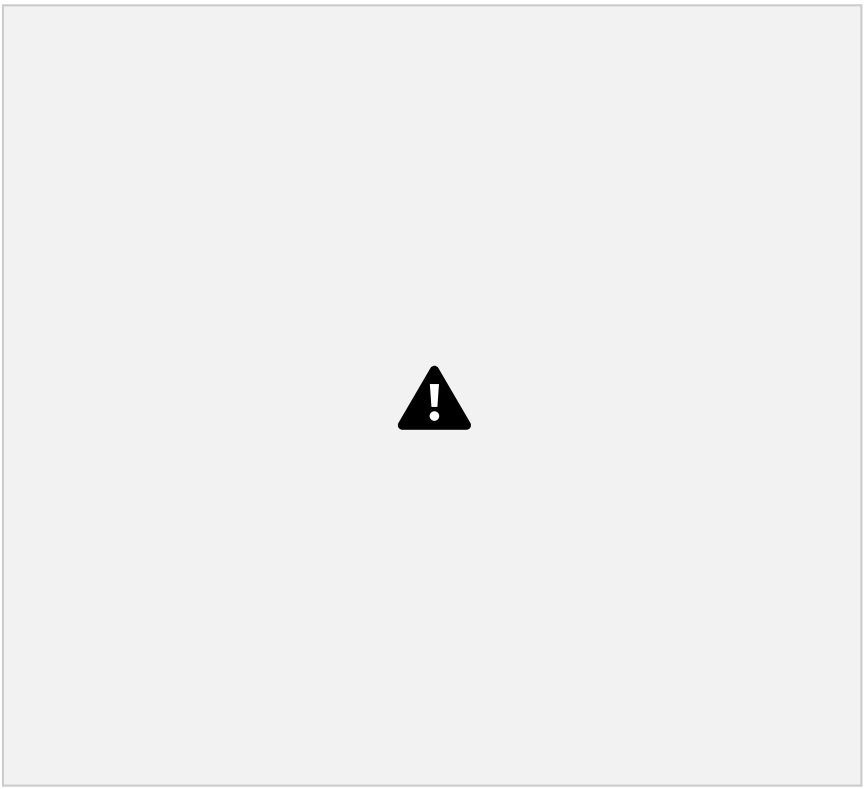
- **Falta de cabecera Anti-Clickjacking**

  No se detectó la cabecera `X-Frame-Options` o equivalente que proteja contra ataques de clickjacking. Esto permite que la página pueda ser incrustada en un iframe malicioso,esto puede tener un riesgo alto.



VULNERABILIDADES:

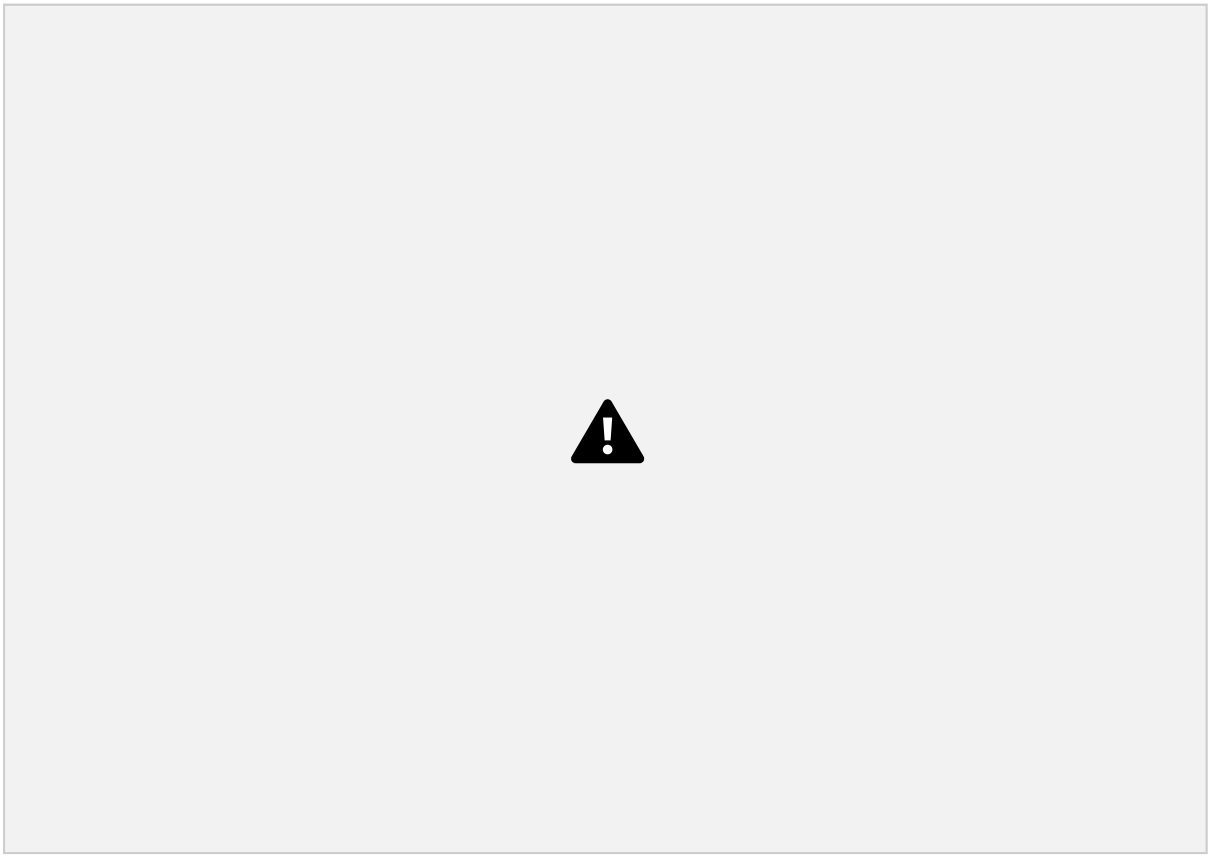
- No se encontraron vulnerabilidades, ya que solo permite que se le registren IP



Por medio de este código en la terminal averiguamos IP, la cual debería ser dirección LAN, luego colocamos la dirección de nuestro sitio con su respectivo puerto ***(https://192.168.56.1:44371/Default.aspx)***

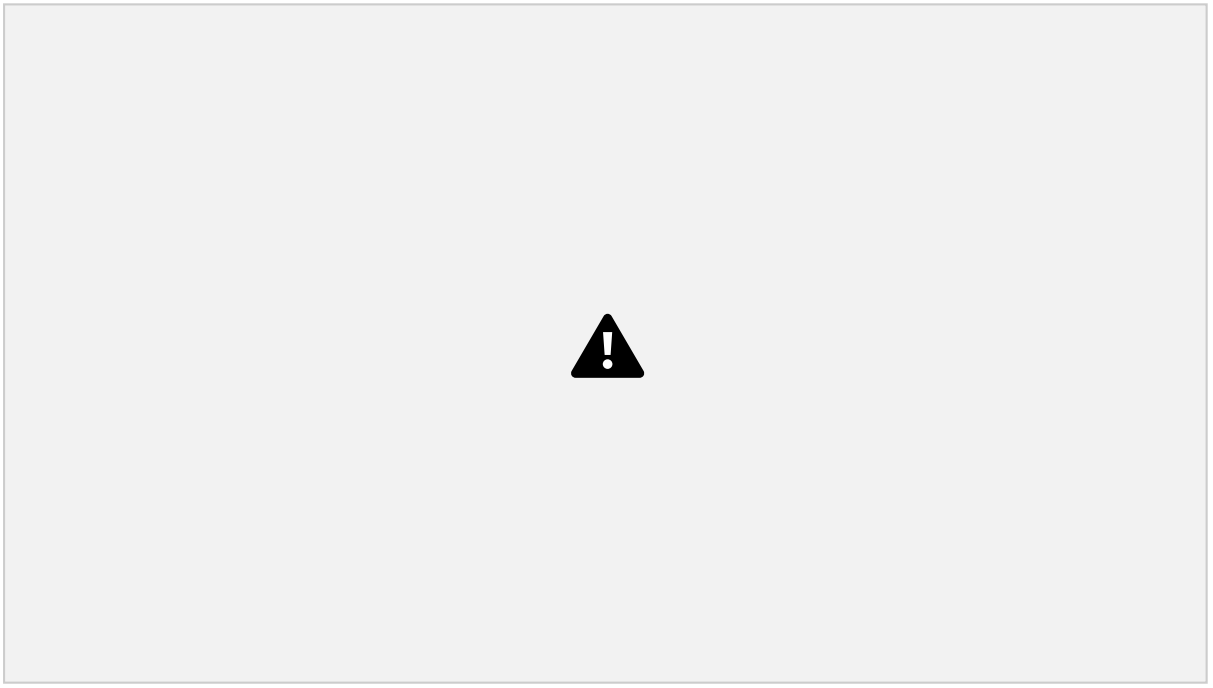
Parte 3 – Matriz de Riesgos

- ***Generar matriz de impacto/probabilidad.***



Parte 4 – Medidas de Mitigación

- Asociar contramedidas específicas a cada riesgo (código + explicación).



explicación de las mejoras:

- **Protege contra ataques de Clickjacking**

El encabezado `X-Frame-Options` evita que tu aplicación sea incrustada en un marco o iframe en otro sitio web, lo que protege contra ataques de clickjacking.

- **Evita ataques MIME**

El encabezado `X-Content-Type-Options` asegura que el navegador no interprete tipos

MIME incorrectos, lo que podría ser explotado por un atacante.

● **Refuerza la seguridad de transporte con HSTS**

El encabezado `Strict-Transport-Security` obliga al navegador a usar HTTPS en todas las comunicaciones con tu sitio, reduciendo el riesgo de ataques como man-in-the-middle**.**

● **Elimina encabezados innecesarios para ocultar información**

La eliminación del encabezado `X-Powered-By` evita que los atacantes obtengan información sobre el software de tu servidor, como el uso de ASP.NET.

● **Configura políticas de caché**

El encabezado `Cache-Control` asegura que datos sensibles no se almacenen innecesariamente en cachés, protegiendo información confidencial.

Parte 5 – Implementación

● Mostrar cambios en código para prevenir al menos 3 vulnerabilidades. ● Comprobamos que al aplicar el código logramos eliminar 5 vulnerabilidades de las 9 que salieron anteriormente