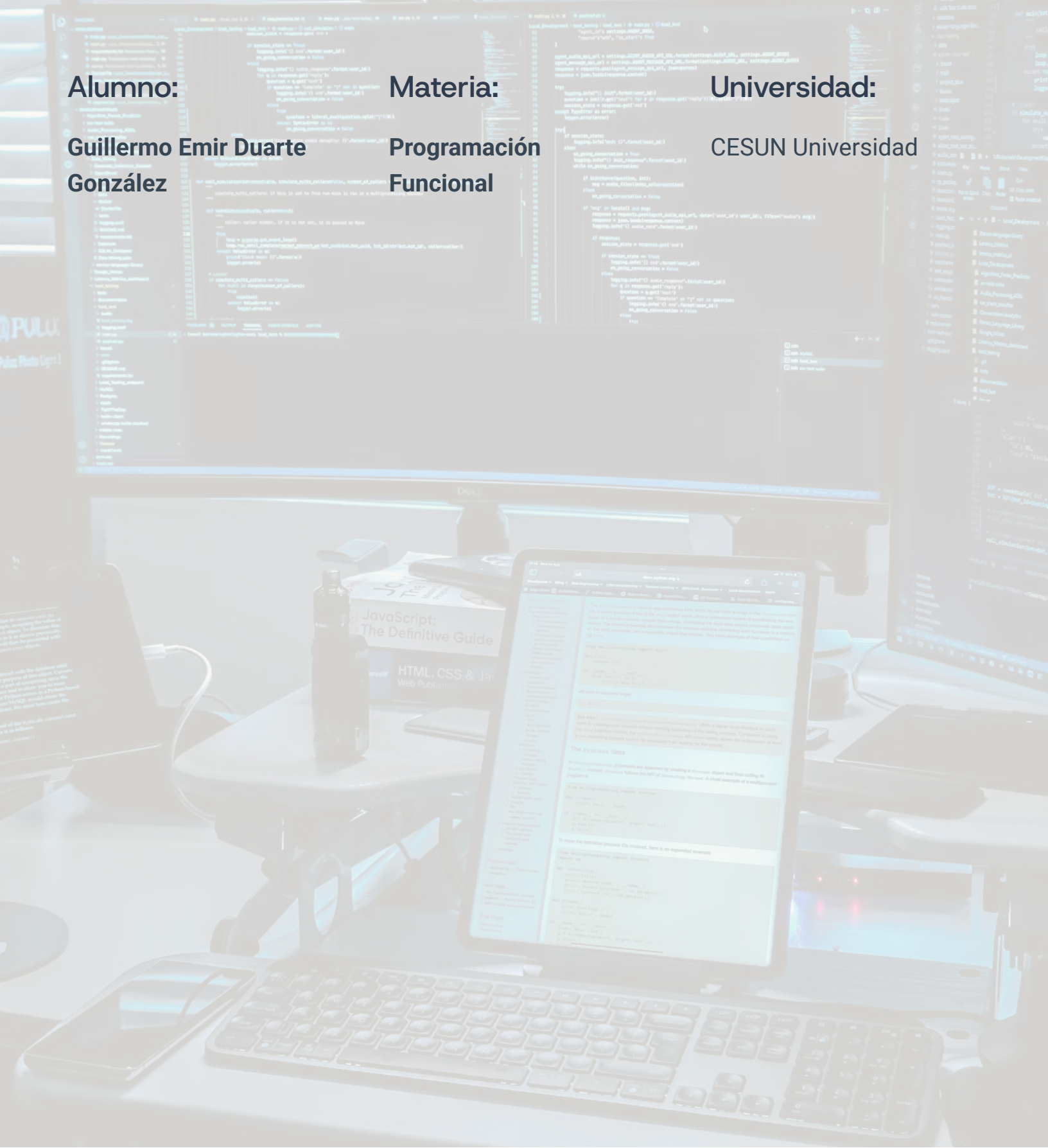
Análisis de Vulnerabilidades con OWASP ZAP



Alumno:	Materia:	Universidad:
Guillermo Emir Duarte González	**Programación Funcional**	CESUN Universidad

🔍 1. Descripción del Proyecto

Este proyecto consiste en aplicar pruebas de seguridad web utilizando la herramienta **OWASP ZAP (Zed Attack Proxy)** sobre el sistema **Tienda de Autos**, alojado en GitHub Pages. El objetivo principal fue **identificar vulnerabilidades comunes** relacionadas con cabeceras de seguridad HTTP, configuraciones CORS y políticas de transporte seguro.

⚙️ 2. Metodología

Herramienta utilizada:

OWASP ZAP 2.16.1

Tipo de análisis:

Escaneo pasivo y activo

Proxy configurado:

localhost:8080

URL analizada:

https://emirdg1.github.io/backend/

Navegador utilizado:

Chrome controlado por ZAP

Durante las pruebas se exploraron todas las secciones de la aplicación, incluyendo el index, imágenes, y archivos JavaScript del proyecto.

🚨 3. Principales Vulnerabilidades Detectadas

Vulnerabilidad	Nivel de Riesgo	Descripción	Recomendación
Content Security Policy (CSP) Header Not Set	Medio	Falta de la cabecera CSP, lo que puede permitir ataques XSS.	Configurar la cabecera Content-Security-Policy.
Strict-Transport-Security Header Not Set	Medio	No hay cabecera HSTS, lo que reduce la protección HTTPS.	Añadir Strict-Transport-Security: max-age=31536000; includeSubDomains.
Cross-Domain Misconfiguration	Bajo	Permite solicitudes desde orígenes externos.	Limitar los orígenes en Access-Control-Allow-Origin.
X-Content-Type-Options Header Missing	Bajo	Archivos podrían ejecutarse como scripts.	Agregar X-Content-Type-Options: nosniff.

📸 4. Evidencias del Escaneo

Capturas obtenidas directamente desde OWASP ZAP:

Panel de Sitios:

Muestra todas las rutas escaneadas (/backend, /img, /script.js).

Alertas:

Detalle de cada vulnerabilidad detectada.

Resumen General:

Vista global del reporte con niveles de riesgo.

🧩 5. Conclusiones

El análisis de seguridad reveló que la aplicación, aunque funcional, carece de algunas **cabeceras de seguridad esenciales** que protegen contra ataques como XSS o inyección de contenido. Se recomienda aplicar las mejoras sugeridas en la configuración del servidor o mediante un servicio intermedio (por ejemplo, Cloudflare o un servidor Nginx).

> Estas pruebas demuestran la importancia de la **auditoría continua de seguridad** en aplicaciones web antes de su despliegue en entornos de producción.

📂 6. Estructura del Repositorio

```
SeguridadZAP/
├── seguridad/
│   ├── reporte_zap_backend.pdf
│   ├── resumen_reporte_zap.pdf
│   └── capturas/
│   ├── alerta_CSP.png
│   ├── alerta_StrictTransport.png
│   ├── alerta_CrossDomain.png
│   ├── panel_sites.png
└── README.md
```