

Visor de redes

Demo web estática para visualizar grafos (JSON) en 3D y simular un ataque básico por eliminación aleatoria de nodos.

Idunn Freya

Mar 2026

ESTADO PUBLICABLE (DEMO) **INICIO** Dec 2025 **VERSIÓN** v1.0.0
REPO [idunnmaria/visor-resiliencia-redes](#)

RESUMEN

Visor de redes es una demo técnica (sin backend) para cargar un grafo en JSON, renderizarlo en 3D y simular un ataque simple (eliminación aleatoria de nodos) para observar cambios de conectividad y fragmentación. Incluye ejemplos listos para probar en menos de 1 minuto y scripts opcionales para generar/atacar redes.

Software y sistemas

Web app (estática) + scripts auxiliares

HTML/CSS/JS + Three.js · Python + NetworkX

STACK TECNOLÓGICO

| | |
|------------------------|--|
| BACKEND | N/A (app estática) |
| FRONTEND | HTML · CSS · JavaScript (vanilla) · Three.js (CDN) |
| INFRAESTRUCTURA | Hosting estático / GitHub Pages (opcional) |
| TESTING | Smoke tests manuales (navegador + scripts) |

OBJETIVOS

- Cargar JSON por archivo (sin copy/paste)
- Visualizar red en 3D (nodos + enlaces)
- Simular ataque básico (random node removal) y observar cambios

- Mantener UI unificada (theme + header)
- Validación de JSON con errores legibles (mejora futura)
- Drag & drop + "Cargar ejemplo" (mejora futura)

ARQUITECTURA

Proyecto dividido en: - **App estática** (HTML) con UI unificada mediante `assets/theme.css` . - **Scripts opcionales** (Python) para generar redes y producir versiones atacadas. - **Ejemplos incluidos** en `examples/` para probar sin instalar nada.

| MÓDULO | DESCRIPCIÓN | ESTADO |
|--|---|--------|
| <code>index.html</code> | Home/intro + enlaces a visor y simulador | ✓ |
| <code>visor_viewer.html</code> | Visor 3D + carga de JSON por archivo | ✓ |
| <code>simulador_resiliencia.html</code> | Simulador (inicializar/similar/reset) + carga de JSON | ✓ |
| <code>assets/theme.css</code> | Theme común + componentes UI | ✓ |
| <code>scripts/data_generator.py</code> | Generador de datos de red (ejemplos) | ✓ |
| <code>scripts/attack_simulator.py</code> | Ataque por eliminación aleatoria + métricas básicas | ✓ |
| <code>examples/</code> | JSON de ejemplo listos para probar | ✓ |

HITOS

- Mar 2026
 UI unificada (theme + header) y carga de JSON por archivo.

- Mar 2026
 Simulador reconstruido con una sola UI (sin overlays legacy).

- Mar 2026
 Ejemplos incluidos en `examples/` para prueba inmediata.

DECISIONES TÉCNICAS Y NOTAS

DECISIÓN

App estática (sin backend). Máxima portabilidad y mantenimiento mínimo; ideal para demo de portfolio. .

DECISIÓN

Modelo de ataque simple. MRandom node removal para observar conectividad/fragmentación sin prometer simulación avanzada.

NOTA

Formatos JSON. Soporte tolerante a formato "nuevo" (`nodes/edges`) y "legacy" (`data/nodes/connections`) para no romper compatibilidad.

ATENCIÓN

Dependencias CDN. Correcto para demo; si se quisiera "producción seria", conviene fijar versiones y añadir SRI o empaquetar.

INSTALACIÓN Y DESPLIEGUE

Probar la demo en local

```
``bash python3 -m http.server 8000 # abrir: # http://localhost:8000/index.html
```