



# Seguridad en Aplicaciones Web Modernas


## Introducción

- Las aplicaciones web modernas usan APIs, microservicios y nube.
- Esto mejora el desarrollo, pero también incrementa los riesgos.
- Hoy abordaremos:
  - Seguridad en APIs RESTful
  - Seguridad en la nube
  - OWASP Top 10 y cómo mitigar riesgos



**OPENAPI**  
INITIATIVE

## Seguridad en APIs RESTful

- Las APIs son esenciales pero vulnerables.
- Riesgos comunes:
  - **SQL Injection**
  - **Cross-Site Scripting (XSS)**
  - **Cross-Site Request Forgery (CSRF)**
-  **Soluciones:**
  - Consultas preparadas / ORMs
  - Sanitización de entradas
  - Tokens CSRF y cookies con SameSite



## Autenticación y Autorización

- Usa estándares modernos como:
    - **JWT** (JSON Web Tokens)
    - **OAuth 2.0**
  - Aplica control de acceso por roles (**RBAC**)
- 🔑 Ejemplo: solo usuarios “admin” pueden borrar registros.





## Validación y Sanitización en APIs

- Nunca confíes en los datos del cliente.
- Valida todos los datos entrantes y salientes.
- Protege tu API del mal uso.



## Seguridad en la Nube

Plataformas más usadas:

- **AWS, Azure, Google Cloud Platform**

Modelo de responsabilidad compartida:

- **Proveedor:** Infraestructura
- **Tú:** Datos, configuración y accesos



Google Cloud

## Buenas Prácticas Cloud

- Rotación de claves y tokens
- Cifrado en tránsito y en reposo
- Monitoreo con SIEM
- Pruebas de seguridad periódicas
- MFA para accesos críticos



## OWASP Top 10 (2021)

1. Broken Access Control
2. Cryptographic Failures
3. Injection (SQL, LDAP, OS...)
4. Insecure Design
5. Security Misconfiguration
6. Vulnerable Components
7. Auth Failures
8. Integrity Failures
9. Logging Failures
10. Server-Side Request Forgery (SSRF)



# OWASP

Open Web Application  
Security Project



## ¿Por qué importa?

Las vulnerabilidades OWASP pueden provocar:

- Fugas de datos
- Robo de identidad
- Pérdidas económicas
- Daño a la reputación



## Mitigación según OWASP

- Control de acceso fuerte
- Cifrado robusto
- Parches de seguridad frecuentes
- DevSecOps: pruebas + revisión continua
- Registro y monitoreo activo



## Conclusión

- La seguridad moderna no es opcional.
- Debe aplicarse en:
  - Diseño
  - Desarrollo
  - Producción
- ✓ Cultura de prevención
- ✓ Equipos comprometidos
- ✓ Aplicaciones confiables y seguras



