

# Programación Segura

Presentación

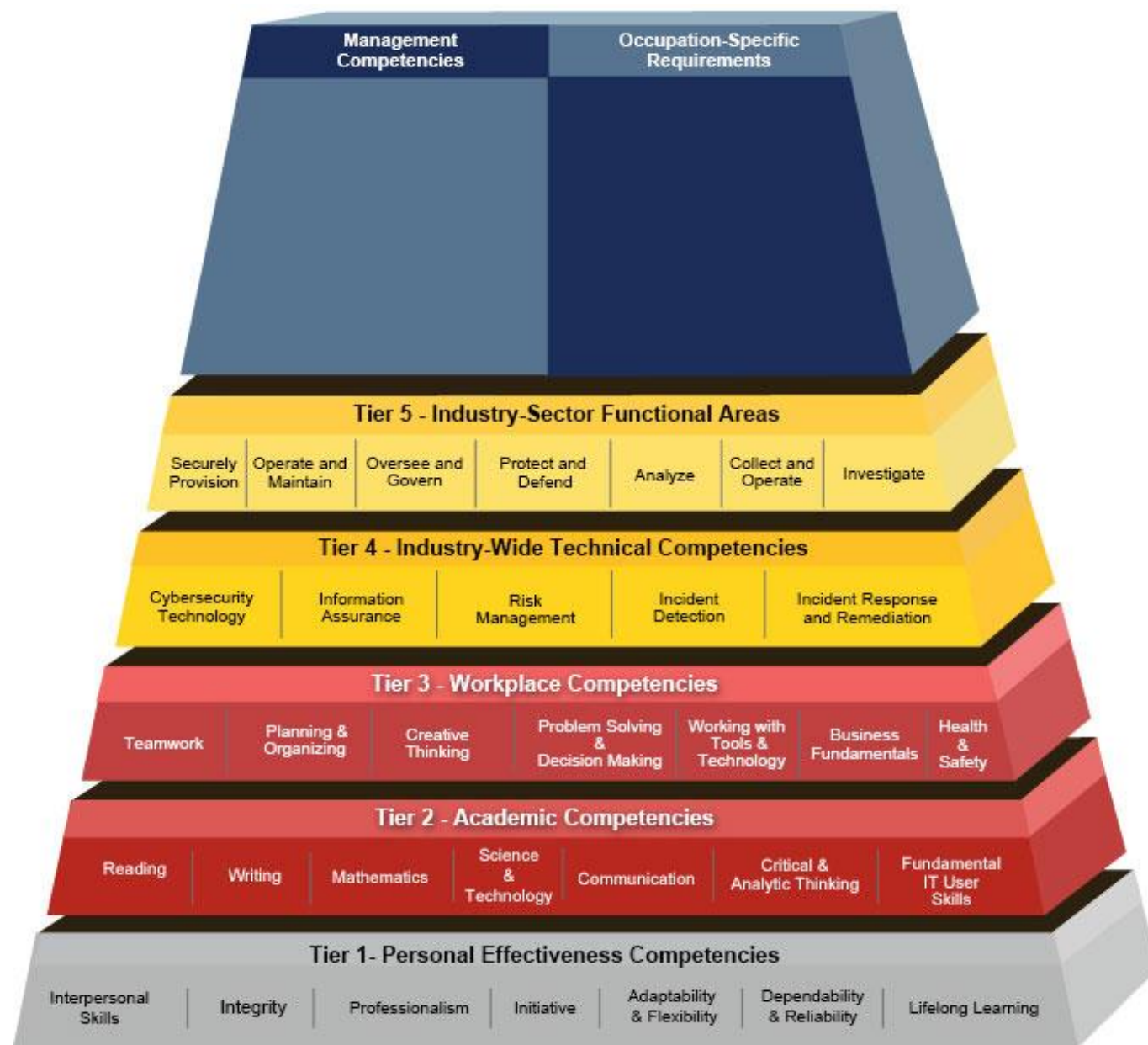
## 2. Seguridad en el desarrollo

Competencias, roles y equipos

# Competencias

- Las competencias didácticas son las capacidades humanas que constan de diferentes conocimientos, habilidades, pensamientos, carácter y valores de manera integral en las distintas interacciones que tienen las personas para la vida en los ámbitos personal, social y laboral<sup>(1)</sup>
- Se adquieren como un proceso
  - Sensibilizar
  - Presentar
  - Asentar
  - Mantener
- Algunos de ellos son intrínsecos, pero en otros casos es necesario su adquisición.

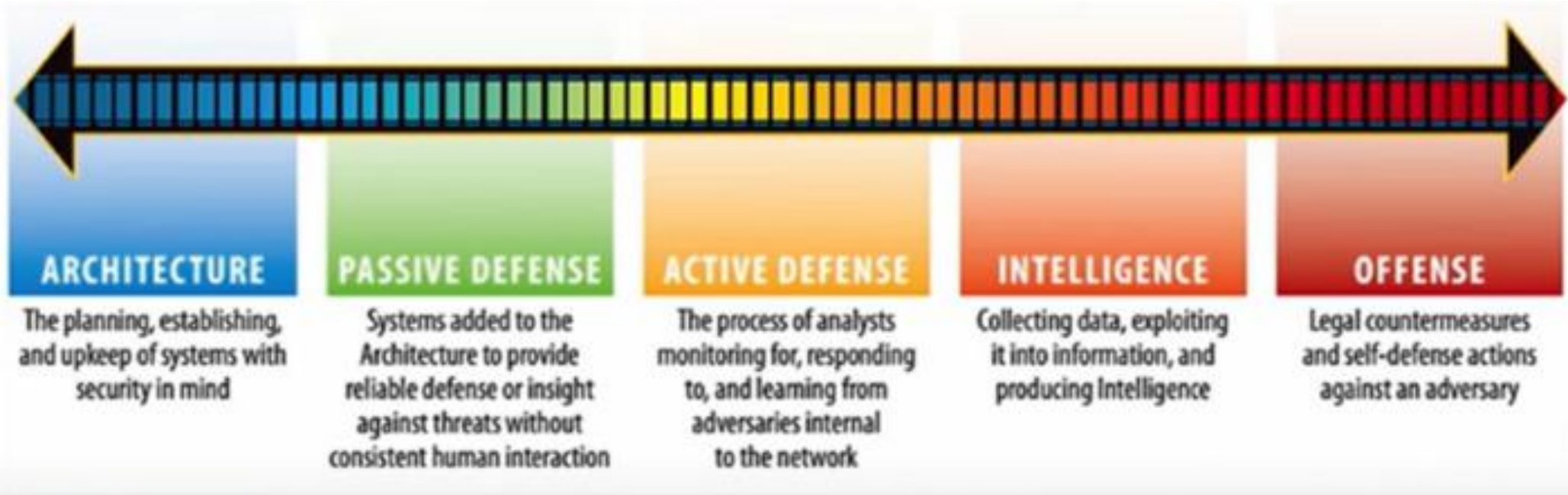
(1) Tigelaar, D. E. H.; Dolmans, D. H. J. M.; Wolfhagen, I. H. A. P.; van der Vleuten, C. P. M. (2004). «The Development and Validation of a Framework for Teaching Competencies in Higher Education» [Desarrollo y validación de un marco para las competencias docentes en la educación superior]. Higher Education (en inglés) 48 (2): 253-268. ISSN 1573-174X



# Roles

1. Chief Information Security Officer (CISO)
  2. Security Manager
  3. Security Engineer
  4. Security Analyst
- En el documento original Workforce Framework for Cybersecurity (NICE Framework) nos ofrece una idea clara de las áreas, roles y tareas, aunque esté algo desfasado  
<http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-181.pdf?trackDocs=NIST.SP.800-181.pdf>
  - Exploramos los dos siguientes enlaces para hacernos una idea más actual:
    - <https://cybersn.com/cybersecurity-career-center/>
    - <https://www.cyberseek.org/pathway.html>
- Pero hay muchos otros son posibles y en constante evolución:
- <https://www.sans.org/cybersecurity-careers/20-coolest-cyber-security-careers/>

# Equipos



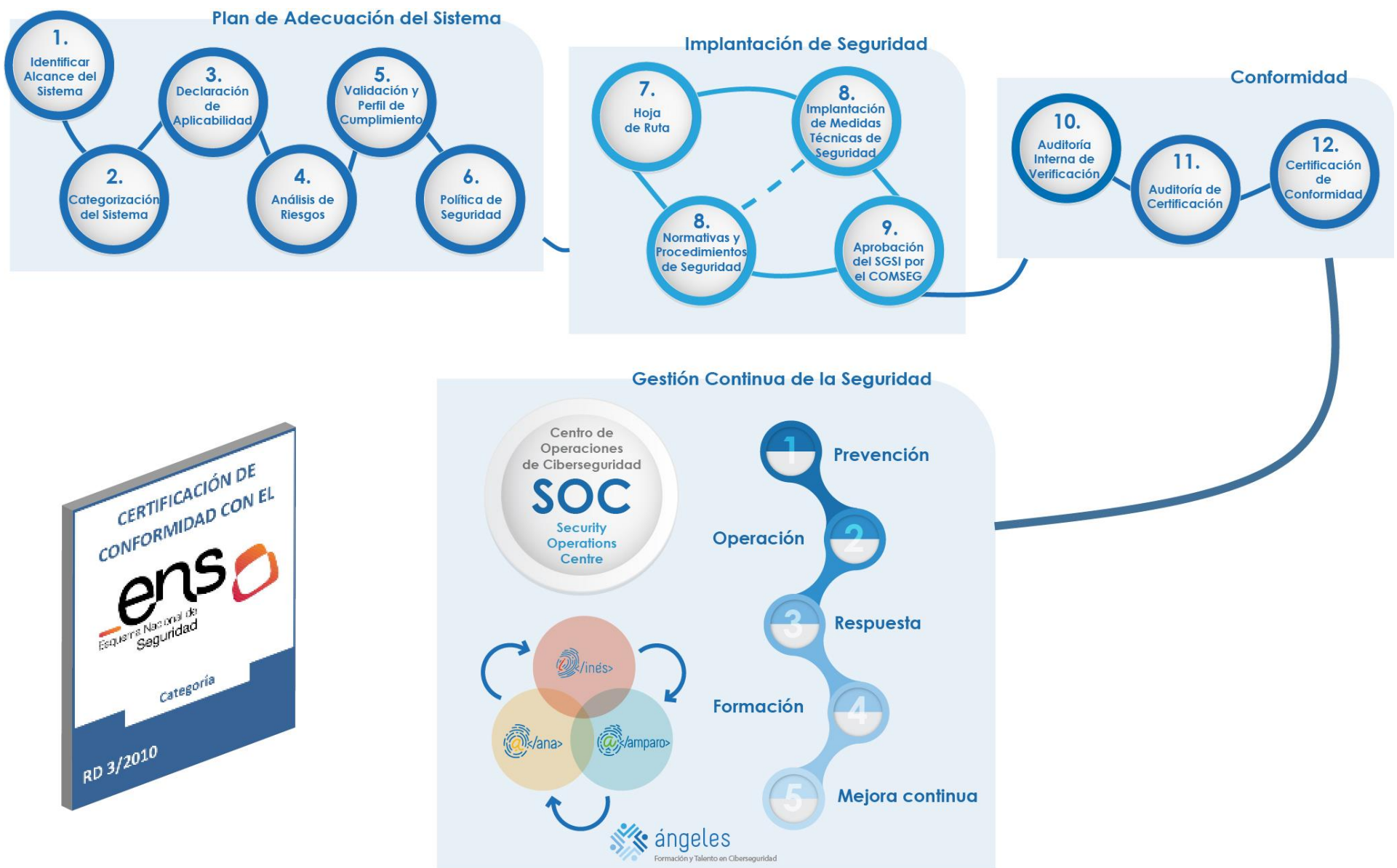
# Compartiendo información de seguridad

- National Initiative for Cybersecurity Education (NICE)  
<https://www.nist.gov/itl/applied-cybersecurity/nice>
- Instituto Nacional de Ciberseguridad  
<https://www.incibe.es/>
- Centro Criptológico Nacional  
<https://www.ccn-cert.cni.es/>
- European Union Agency for Cybersecurity  
<https://www.enisa.europa.eu/>

# El Esquema Nacional de Seguridad

- ¿Por qué? Ya que nos lo van a pedir
    - “Tiene por objeto establecer la política de seguridad en la utilización de medios electrónicos y está constituido por principios básicos y requisitos mínimos que permitan una protección adecuada de la información”
- [https://es.wikipedia.org/wiki/Esquema\\_Nacional\\_de\\_Seguridad](https://es.wikipedia.org/wiki/Esquema_Nacional_de_Seguridad)
- RD 03/2010 “Esquema Nacional de Seguridad” (ENS)
  - RD 951/2015 que lo modifica
  - Dependiente del Centro Criptológico Nacional
    - <https://ens.ccn.cni.es/es>
  - Guías
    - <https://www.ccn-cert.cni.es/guias/guias-series-ccn-stic/800-guia-esquema-nacional-de-seguridad.html>





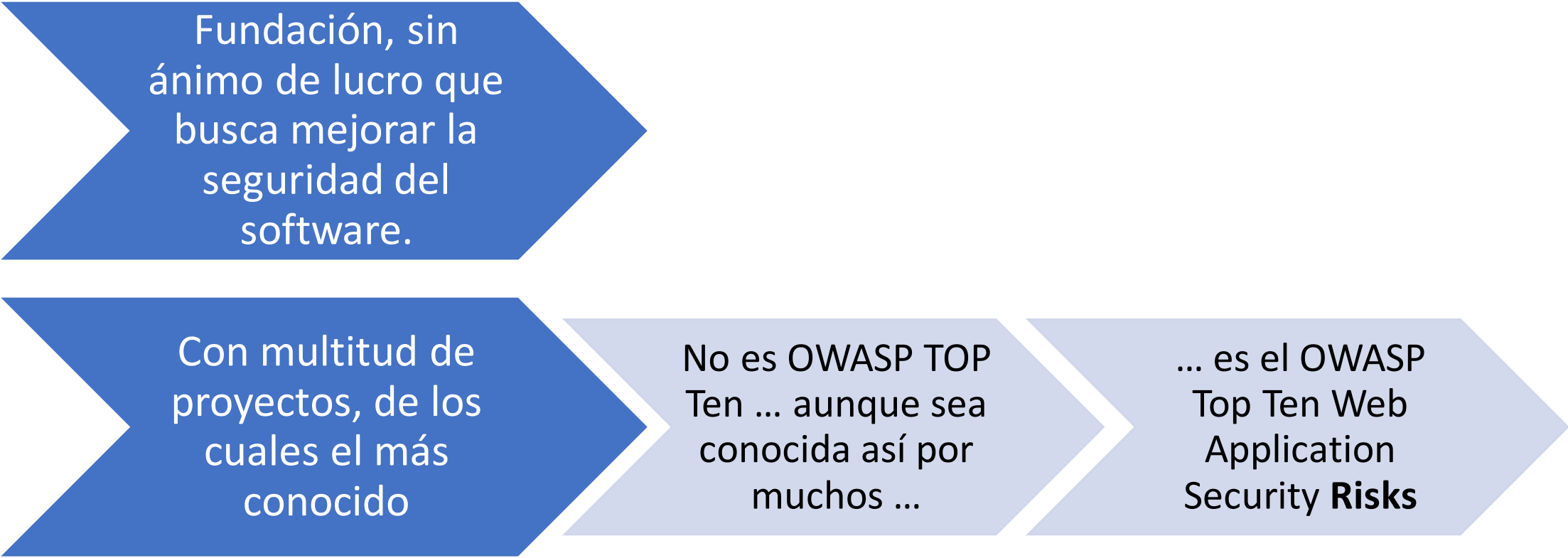


# 3. OWASP Top 10

¿Sí, sí, pero cuál de ellos?

# OWASP

## (Open Web Application Security Project)



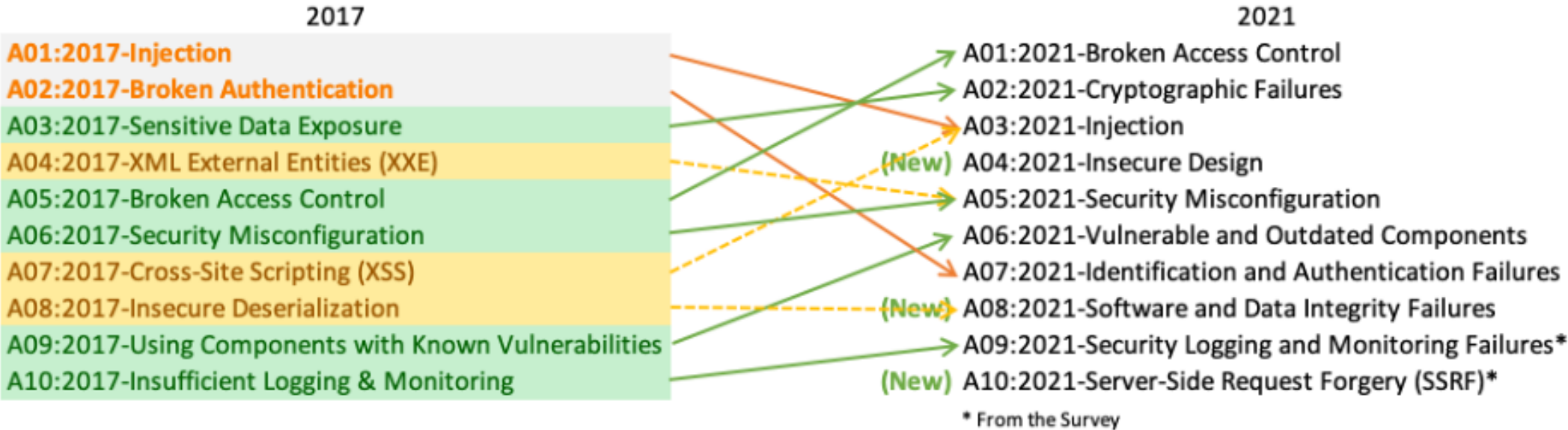
# Diferentes proyectos

- Hemos resaltado:
  - OWASP ASVS (Application Security Verification Standard)
    - Requisitos y criterios de verificación
  - OWASP TG (Testing Guide)
    - Una referencia general de como acceder a sistemas
- Pero en realidad está dividido en multitud de proyectos que se relacionan entre sí (se complementan... y suplementan en ocasiones)
- Veamos en su web...

# Otros proyectos de interés

- OWASP Proactive Controls (OPC)
- OWASP Cheat Sheet Series (OCSS)
- Software Assurance Maturity Model

# Nos basamos en la 2017 ... pero ya está la 2021

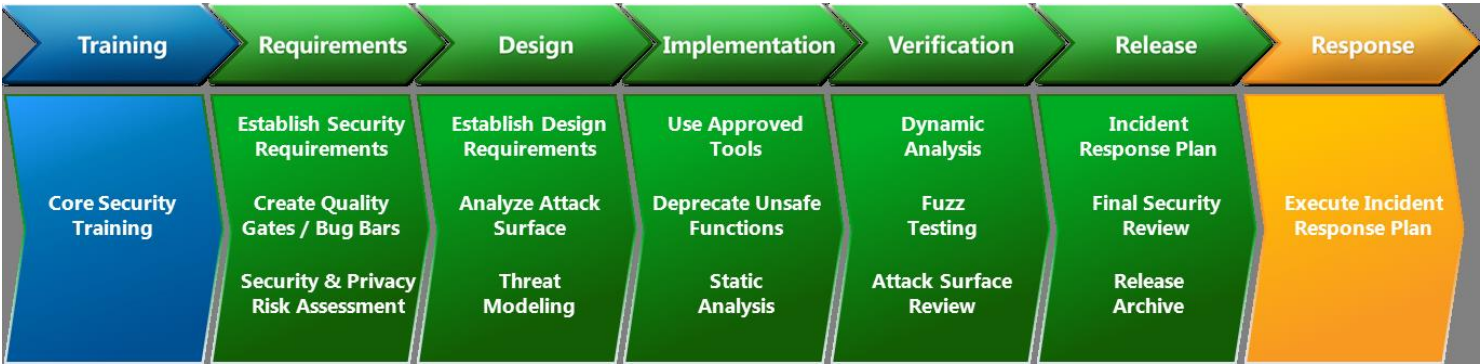


# 4. Ciclo de Vida de Desarrollo SW seguro

Tradicional y ágil



# En el SDL de MicroSoft

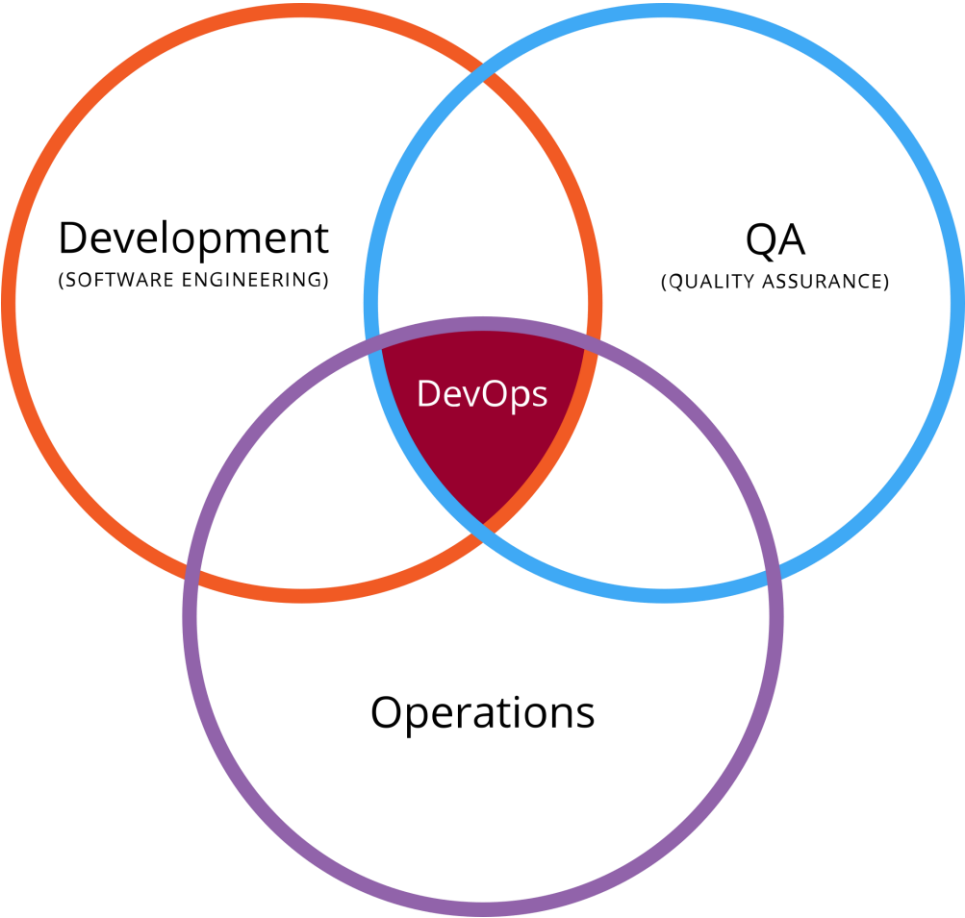


The SDL Optimization Model

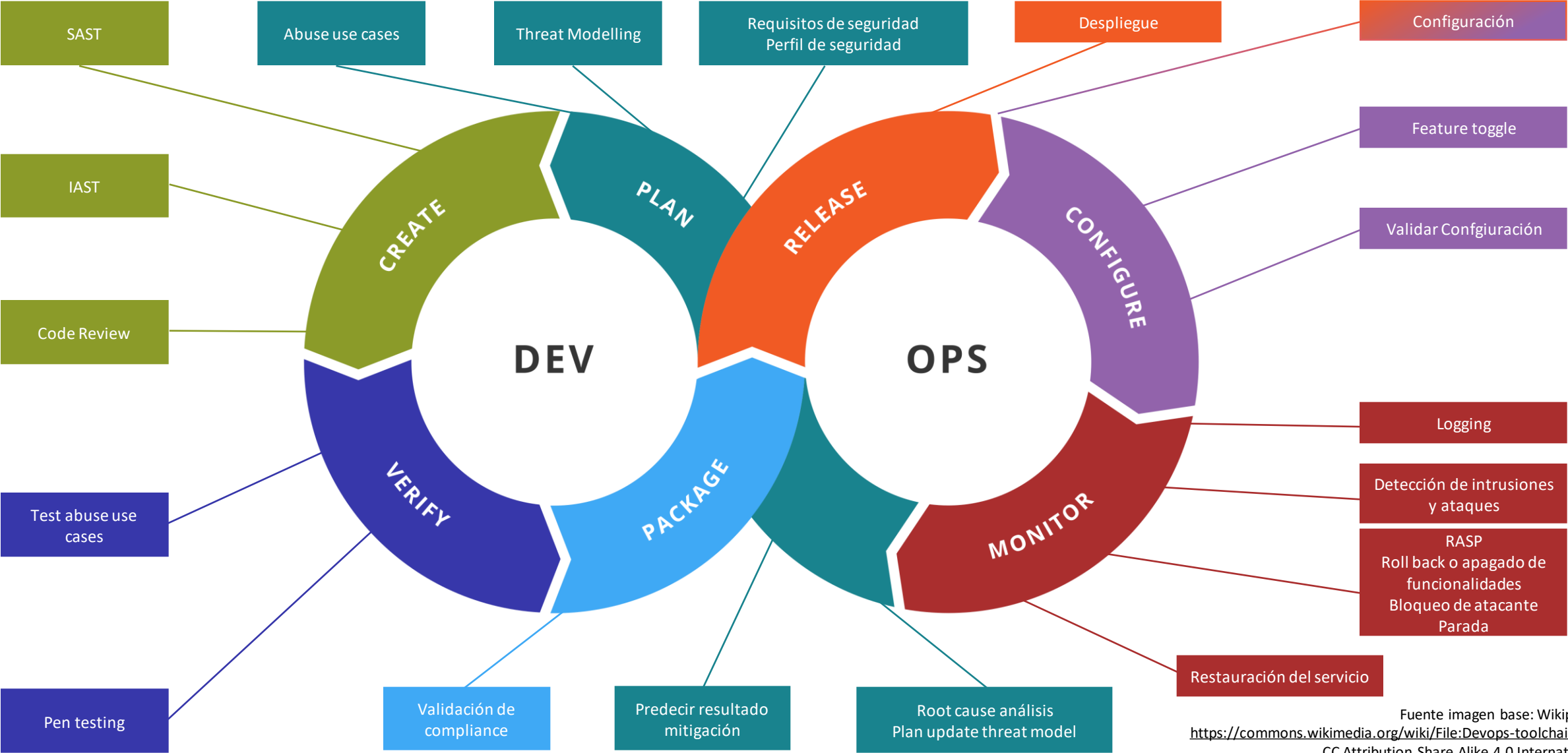


Simplified Implementation of the SDL.doc  
<http://www.microsoft.com/sdl>  
CC Attribution-ShareAlike 3.0 Unported

# DevOps



# DevSecOps



# 5. Herramientas para el desarrollo SW seguro

# Tenemos para distintos usos

- SAST: Static Application Security Testing
- DAST: Dynamic Application Security Testing
- IAST: Interactive Application Security Testing
- WAF: Web Application Firewall
- RASP: Run-time Application Security Protection

# 6. Taller de pentesting

Manos a la obra

# 6.a Una aplicación insegura

...o dos

# OWASP Juice Shop

<https://owasp.org/www-project-juice-shop/>



# OWASP WebGoat

<https://owasp.org/www-project-webgoat/>

# 6.b Herramientas de pen-testing

... más algunas de base

# Aislamiento del entorno de trabajo

Buscamos

- Un entorno controlado y aislado
- Optamos por ejecutar máquinas virtuales:
  - Vagrant para el control
  - VirtualBox como software de virtualización
  - Dentro del mismo instalaremos Docker (script)
- Alternativas durante el curso
  - Utilizar otro proveedor de virtualización
  - Crear manualmente una imagen propia, Ubuntu 20.04 based (Ubuntu, mint ...)
  - Kali linux

# Herramientas generales

Instalamos en la máquina virtual:

- git
- Chrome, Firefox
- Docker
- node/npm
- VSCode
- ZAP

# Pentesting – Atendiendo al conocimiento del sistema

Puede ser realizado por equipos internos o externos.

- Black Box: Desconocemos detalles internos del sistema
- White Box: Podemos llegar a tener acceso a la documentación, fuentes...
- Grey Box: Escenario intermedio

# Pen-testing

- OWASP ZAP (DEMO)
  - Proxy
  - Spider
  - Atacante
  - Automatización
- Burp Suite (alternativa)
- Metasploit (demasiado agresiva)
- ... más muchas otras que vamos a encontrar en

[https://owasp.org/www-project-web-security-testing-guide/v41/6-Appendix/A-Testing\\_Tools\\_Resource](https://owasp.org/www-project-web-security-testing-guide/v41/6-Appendix/A-Testing_Tools_Resource)

# 6.c Ejecución de diferentes ataques

Comienza el taller