



**First OWASP DAY**  
Chile 2010



**The OWASP Foundation**  
<http://www.owasp.org>



## Temario

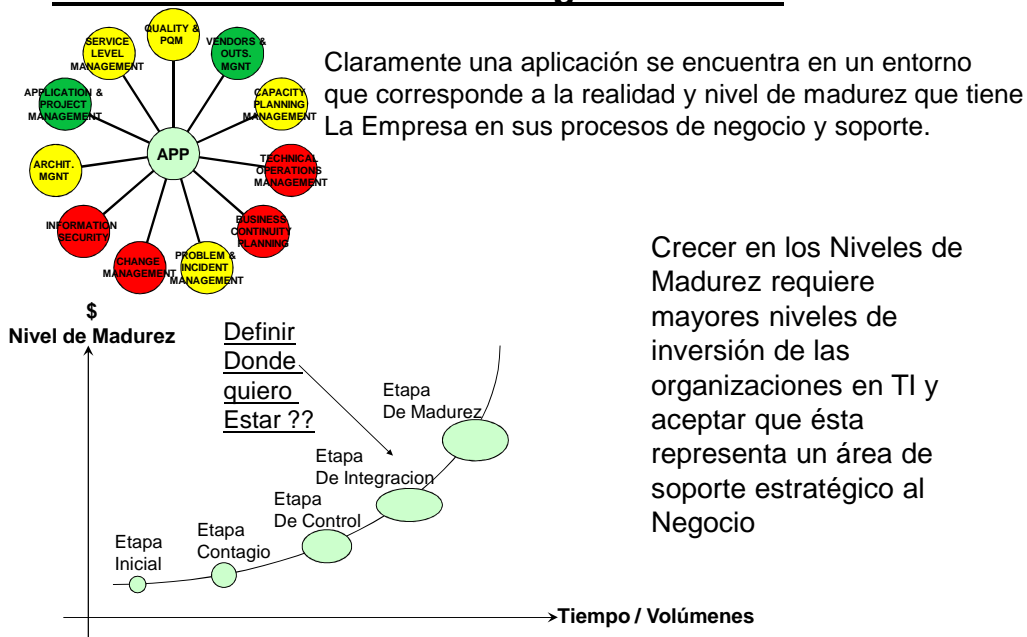
- OWASP Guide
- Evolución TI en las Organizaciones
- Capítulo: Principios de Codificación Segura en SDLC
- Contexto y referencias OWASP
- Modelo de Protección de la Información
- Principios para Aplicaciones Seguras desde el Diseño
- Flujo Modelo Amenazas
- Ejemplos

2

# OWASP Guide

- Guía para construir aplicaciones y servicios WEB seguros
- Varios editores, autores, directores de proyecto (40 aproximadamente.)
- Publicación gratuita y abierta a interesados en mejorar la seguridad de aplicaciones
- Conjunto de definiciones, alertas, vulnerabilidades y buenas prácticas. No impone, pero sugiere
- **Copyright © 2002-2005. The Open Web Application Security Project (OWASP). Todos los derechos reservados. Se concede permiso para copiar, distribuir y / o modificar este documento siempre que se incluya este aviso de derechos de autor y se mantenga la atribución a OWASP.**

## Evolución de las TI en las Organizaciones

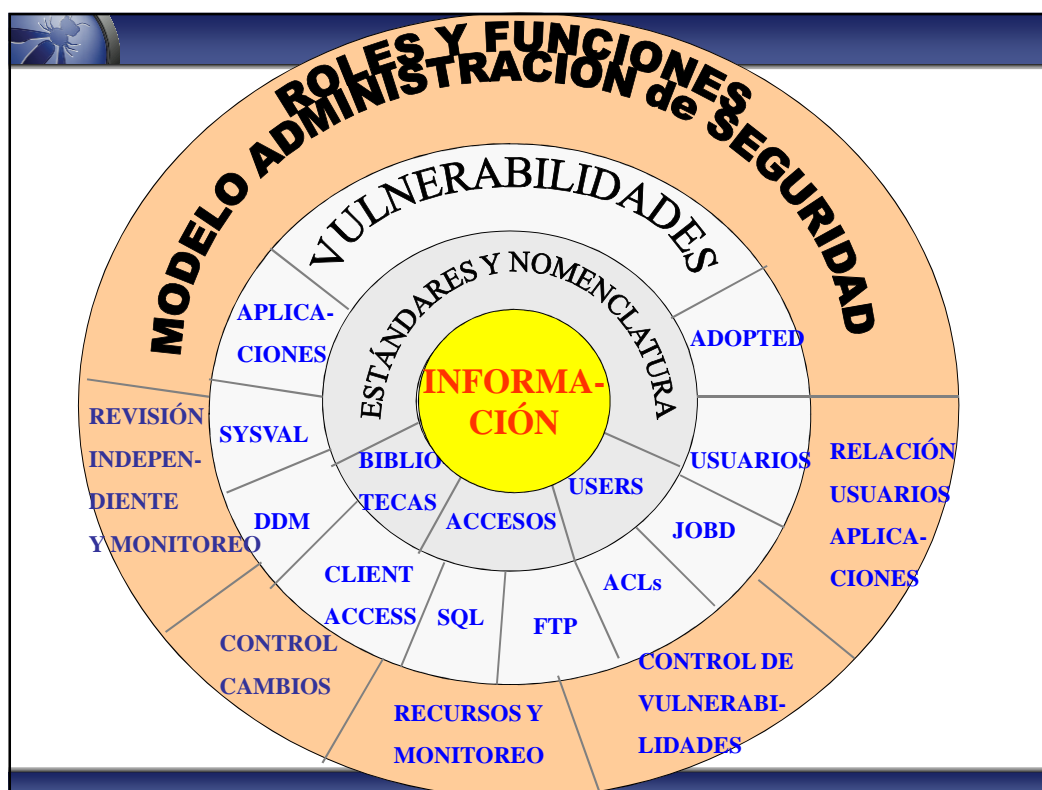


## Capítulo: Principios de Codificación Segura en SDLC

- **La elección de una metodología de desarrollo no es tan importante como el simple hecho de poseer una.**
- **El desarrollo Ad hoc no es lo suficiente estructurado para producir aplicaciones seguras.**
- **Elegir la metodología adecuada. Considerar:**
  - Complejidad: puede sobreburocratizar identificando demasiados roles diferentes
  - Fuerte aceptación de diseño, testeo y documentación
  - Espacios donde se puedan insertar controles de seguridad (tales como análisis de riesgo de amenazas, revisiones por parte de pares, revisiones de código, etc.)
  - Que funcione para el tamaño y nivel de madurez de la organización
  - Tenga potencial de reducir tasa actual de errores y de mejorar la productividad de los desarrolladores.

## Contexto y Referencias OWASP

- Los principios de seguridad tales como **confidencialidad, integridad, y disponibilidad** – aunque son importantes, amplios y vagos – no cambian
- OWASP propicia:
  - Adoptar un Modelo de Riesgo de Amenazas: Microsoft / Trike / CVSS / AS4360 / y Octave
  - Mejores prácticas de mercado: CobIT / ISO 27001 / ISO 17799 / PCI / SOX
  - Una gestión organizacional que abogue por la seguridad
  - Políticas de seguridad documentadas y apropiadamente basadas en estándares nacionales-internacionales
  - Una metodología de desarrollo con adecuados puntos de control y actividades de seguridad
  - Gestión segura de versiones y configuración
  - Lograr un “SDL”: Security Development Lifecycle

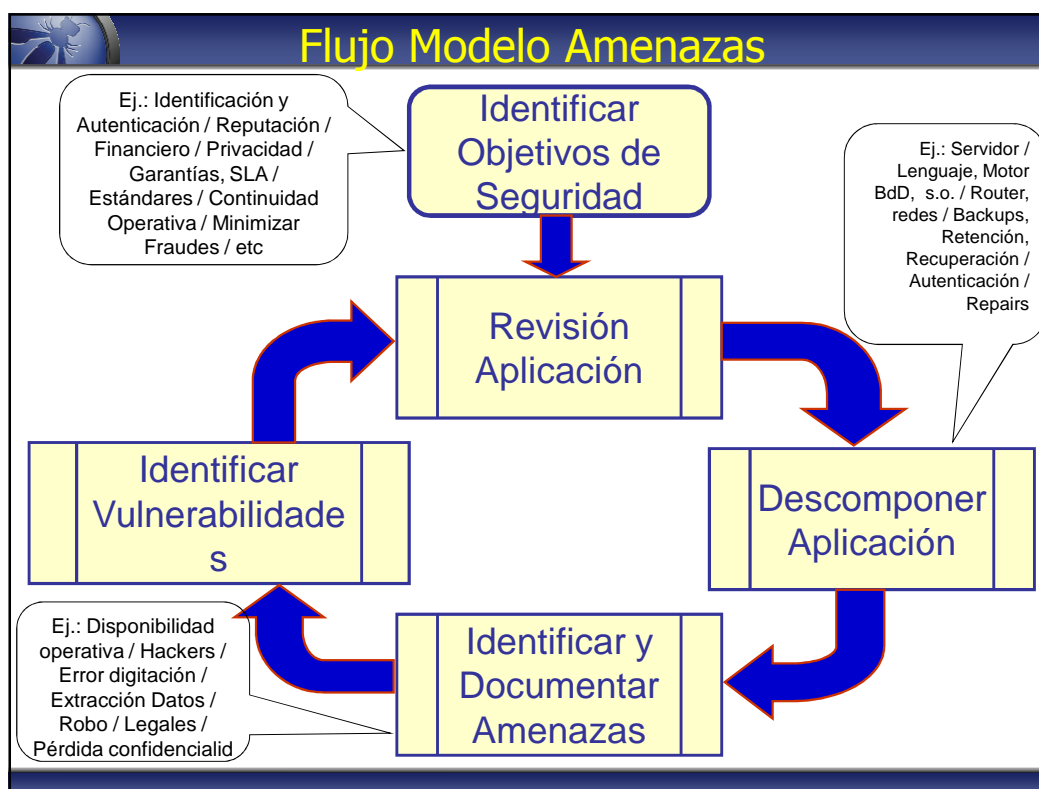


## Principios para Aplicaciones Seguras desde el Diseño

- Análisis de Riesgos
- Identificación de amenazas, revisión de pares, revisiones de código, Fuerte aceptación de diseño, testeo, Ethical Hacking, documentación, etc...
- Clasificación de Activos: La selección de controles sólo es posible después de clasificar los datos a proteger tales como Datos, Código SW, Configuración FW
- Orientación arquitectura (p.ej. "la capa Web no debe llamar a la base de datos directamente", Aislar Producción de Desarrollo con Firewall)
- Niveles mínimos de documentación requerida
- Requerimientos de testeo mandatorios (inspecciones, Ethical Hacking, comprobar en el tiempo)
- Niveles mínimos de comentarios entre código y estilo de comentarios preferidos
- Manejo de excepciones
- Control de integridad del código fuente

## Principios para Aplicaciones Seguras desde el Diseño

- La arquitectura de seguridad empieza el día en que se modelan los requisitos del negocio, y
- no termina nunca hasta que la última copia de su aplicación es retirada
- Seguridad por defecto (p.ej. estructura passwords, valores sistemas, continuidad operativa,..)
- Principio del mínimo privilegio
- Fallos de manera segura
- Los sistemas externos son inseguros: la confianza implícita de ejecutar sistemas externos, no está garantizada
- Segregación de Funciones: Identificar funciones sensibles
- No confiar la seguridad en la oscuridad, sino hacer lo contrario
- Incorporar bitácoras: Logs, Journals, prender los Audit journals, y disponer de un mecanismo de su revisión. Principio de NO-Repudiación





## Ejemplo, Análisis de Protección : Accesos

### VULNERABILIDADES

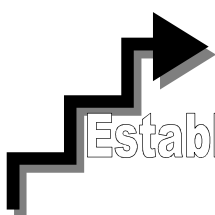
- Procedimientos y metodología (nomenclatura, clasificación, control de cambios, privilegio usuarios, revisión journals)
- Datos de Producción no encriptados usados en Areas de Desarrollo, Testing u otros
- Cxs remotas, filtros, ftp, User-ids privilegiados sin control, configuración Routers
- Tareas programadas desprotegidas. Etc...

ACCESOS

CONTROLES DE PROTECCIÓN

ACTIVOS  
DE  
INFORMA-  
CIÓN

## Se inicia y logra con el compromiso y responsabilidad del Top Management



Estableciendo objetivos y planes

Trabajando en Equipo



Escogiendo herramientas adecuadas

