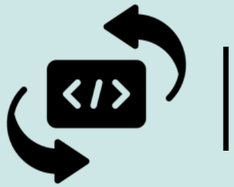
Entornos DevSecOps

PRESENTADO POR **LUIS FELIPE OSPINA** Y **TOMAS ENRIQUE VIVARES**

¿Qué es DevOps?

DevOps es una filosofía cultural, metodológica y técnica que fusiona el desarrollo de software (Dev) con las operaciones de TI (Ops). Su meta es acortar el ciclo de vida del desarrollo, aumentar la calidad del software y permitir la entrega continua de valor.

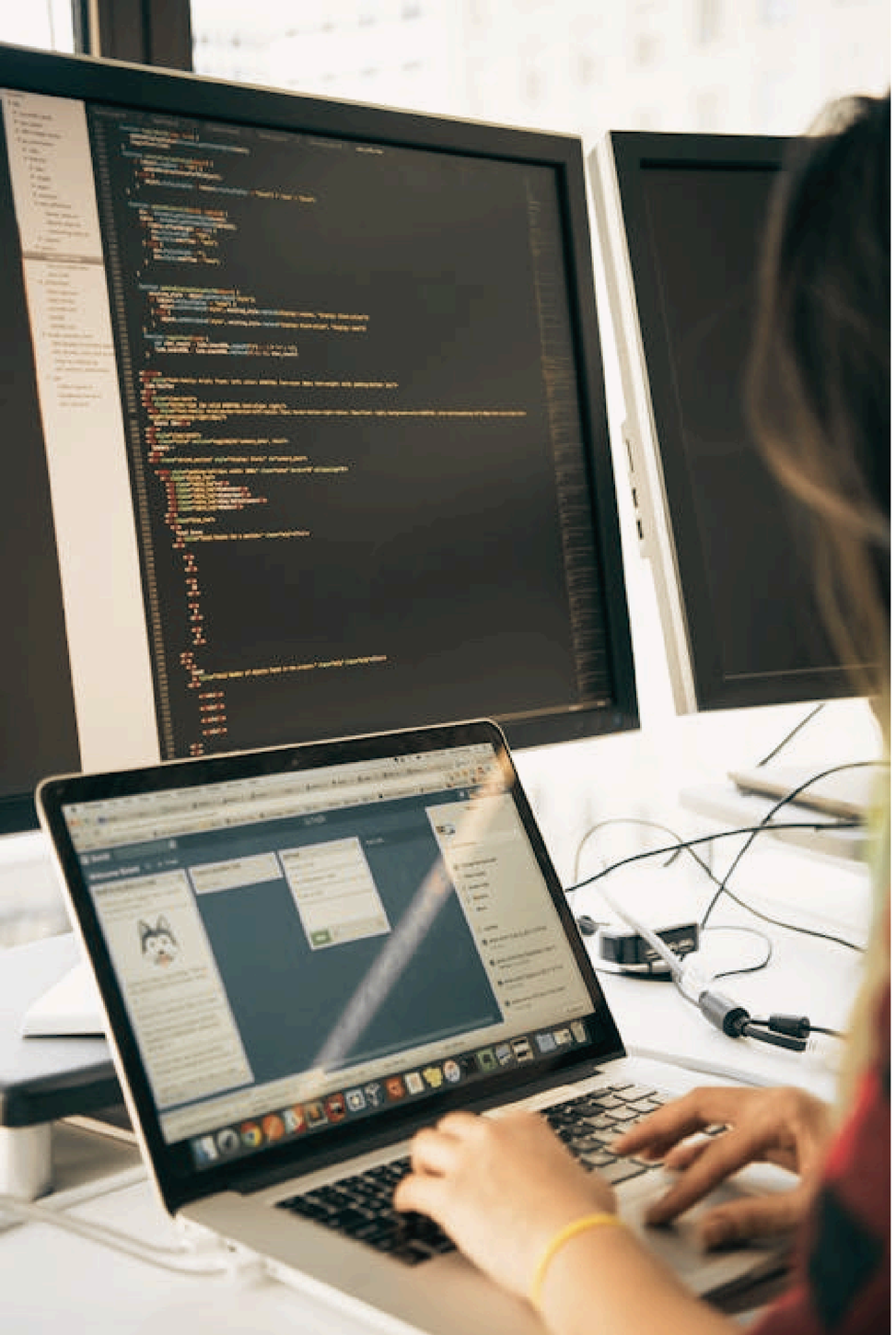
Principales conceptos



Integración Continua (CI)
Los desarrolladores suben cambios frecuentes al repositorio. Cada cambio dispara procesos automáticos de build y pruebas. Permite detectar errores rápidamente y evitar “sorpresas” al final.

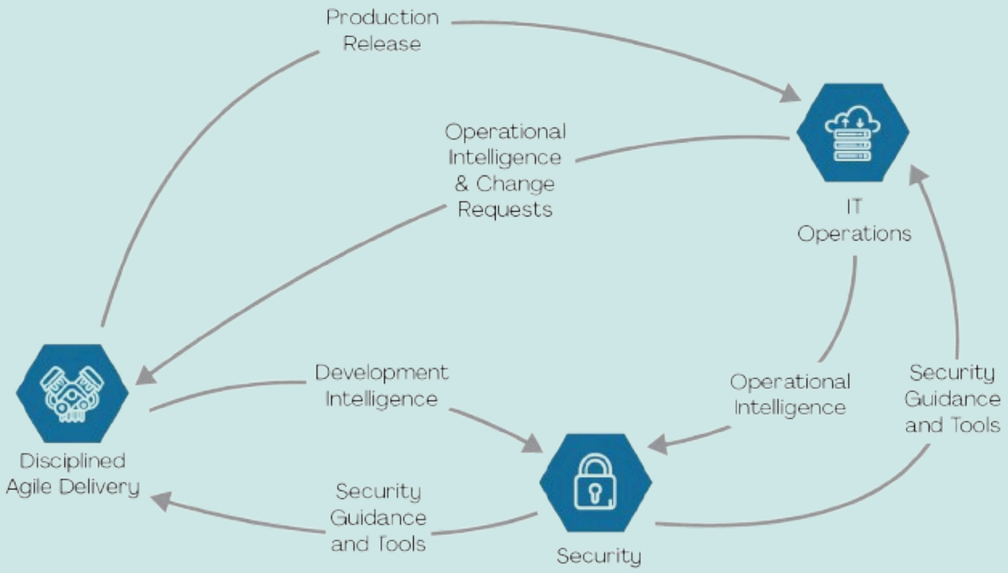
Entrega/Despliegue Continuo (CD)
Automatización de la puesta en producción. Reduce tiempos de liberación y errores humanos.
Ejemplo: pipelines con Jenkins, GitHub Actions, GitLab CI/CD.

Infraestructura como Código (IaC)
Configuración de servidores, redes y entornos mediante código. Asegura consistencia y reproducibilidad.
Ejemplo: Terraform, Ansible, AWS CloudFormation.



¿Qué es DevSecOps?

DevSecOps es la evolución de DevOps que integra la seguridad como un principio esencial desde el inicio. Su lema es: **“Seguridad como código, responsabilidad compartida en todo el ciclo de vida del software”.**



Conceptos y terminología

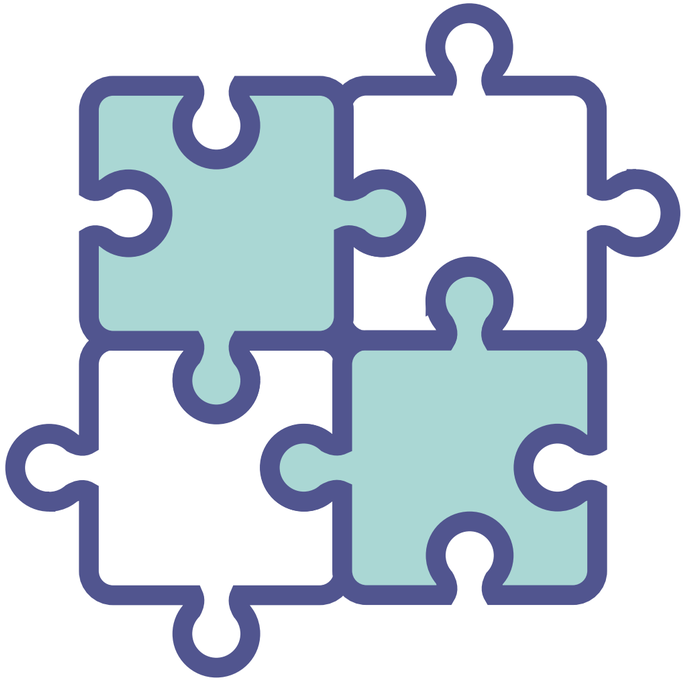
Shift-left security

Llevar las pruebas de seguridad a las fases tempranas del desarrollo.

Responsabilidad compartida

Todos los equipos participan en asegurar el software:

- Dev → escriben código seguro.
- Sec → proveen reglas, políticas y herramientas.
- Ops → aseguran infraestructura, despliegue y monitoreo.



Pipeline seguro

Integrar escáneres automáticos en CI/CD:

- SAST (Static Application Security Testing).
- DAST (Dynamic Application Security Testing).
- SCA (Software Composition Analysis).

Seguridad como código

Definir políticas de seguridad (ej. control de acceso, cifrado) en IaC y en pipelines.

Modelo conceptual de DevSecOps

1. **Planificación:** Identificación de riesgos, requisitos de seguridad, modelado de amenazas.
2. **Codificación:** Uso de repositorios seguros, escaneo de dependencias, pre-commit hooks.
3. **Build & Test:** Escaneo estático (SAST), dinámico (DAST) y análisis de composición de software (SCA).
4. **Despliegue:** Escaneo de contenedores, validación de configuraciones, gestión de secretos.
5. **Operación:** Monitoreo de seguridad, detección de anomalías, gestión de incidentes.
6. **Feedback y mejora continua:** Integrar lecciones aprendidas para fortalecer el ciclo.

Relevancia en el contexto de seguridad en los datos

- Confianza del cliente y reputación: Entregar software seguro aumenta la confianza en los servicios y productos de la organización.
- Cumplimiento regulatorio: Apoya certificaciones y normativas como ISO 27001, GDPR, PCI DSS, HIPAA.
- Resiliencia y reducción de riesgos: Acelera la respuesta ante incidentes y minimiza el tiempo de exposición.
- Protección de datos sensibles: DevSecOps asegura que los datos personales, financieros o confidenciales estén protegidos desde el diseño hasta la operación.
- Prevención proactiva: Se detectan vulnerabilidades antes de que sean explotadas.
- Escalabilidad y automatización: La seguridad se aplica de forma uniforme en múltiples proyectos y entornos cloud.

Thank you very much!

PRESENTADO POR **LUIS FELIPE OSPINA** Y **TOMAS ENRIQUE VIVARES**