

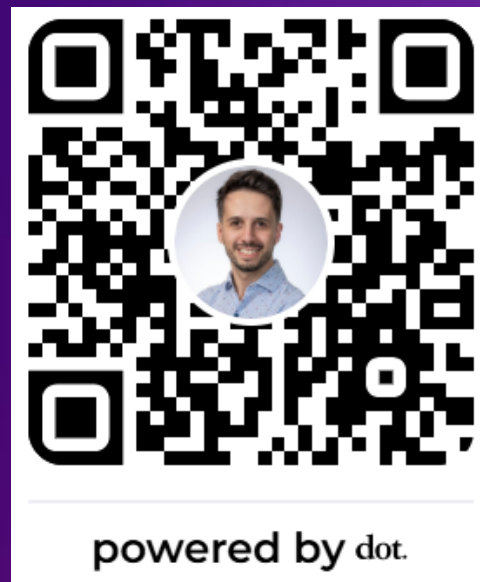


DevOps Multiverse

Y sus variantes

Axel Labruna

Business Solution Architect @Nubiral
AWS Community Builder
DevOps Institute Ambassador





El recorrido de hoy



1  Definiciones clave

3  Herramientas por metodología

5  Tendencias y demanda laboral

7  Recursos y certificaciones

2  Conceptos fundamentales

4  Comparaciones

6  Cómo iniciar tu carrera

8  Conclusión



Érase una vez en un reino desarrollo tradicional





¿Qué es DevOps?

“ DevOps es una metodología que integra el desarrollo de software (Dev) y las operaciones de TI (Ops) para mejorar la colaboración, automatización y entrega continua de valor.

Características principales:



Integración y entrega continua

Automatización del ciclo de vida del desarrollo



Cultura de colaboración

Equipos de desarrollo y operaciones trabajan juntos



Automatización

Reducción de tareas manuales y procesos repetitivos



Medición continua

Monitoreo y optimización del rendimiento

Beneficios:



Mayor velocidad en la entrega de software



Mejora de la calidad y estabilidad del producto



Reducción de fallos en producción y tiempo de recuperación



Herramientas clave en DevOps



CI/CD

Integración y despliegue continuo: automatización del ciclo de vida de desarrollo



Jenkins

Servidor de automatización open-source



GitLab CI

CI/CD integrado con GitLab



GitHub Actions

Flujos de trabajo en repositorio



CircleCI

Plataforma cloud de CI/CD



Contenedores

Empaquetado, distribución y ejecución de aplicaciones aisladas



Docker

Plataforma de contenedores



Kubernetes

Orquestación de contenedores



Containerd

Runtime de contenedores



Harbor

Registro de imágenes



Infraestructura como código

Gestión y aprovisionamiento de infraestructura mediante definiciones en código



Terraform

Infraestructura declarativa



Ansible

Automatización de configs



CloudFormation

IaC para AWS



Pulumi

IaC programática



Monitoreo

Observabilidad, análisis y alertas sobre infraestructura y servicios



Prometheus

Recolección de métricas



Grafana

Visualización de datos



Datadog

Monitoreo cloud



ELK Stack

Análisis de logs



¿Qué es DevSecOps?

DevSecOps integra la seguridad en todo el ciclo de vida de desarrollo y operaciones. Esta metodología incorpora el enfoque "Security as Code", donde las prácticas de seguridad son responsabilidad de todos, no solo del equipo especializado.

Filosofía "Security by Design":



Seguridad desde el inicio

No como una capa añadida al final



Responsabilidad compartida

Todos contribuyen a la seguridad del código



Automatización de seguridad

Integrada en pipelines de CI/CD



Detección temprana

Vulnerabilidades identificadas en fases iniciales

Herramientas destacadas:



Análisis de código

SonarQube, Checkmarx, Fortify



Escaneo de dependencias

Snyk, OWASP Dependency-Check

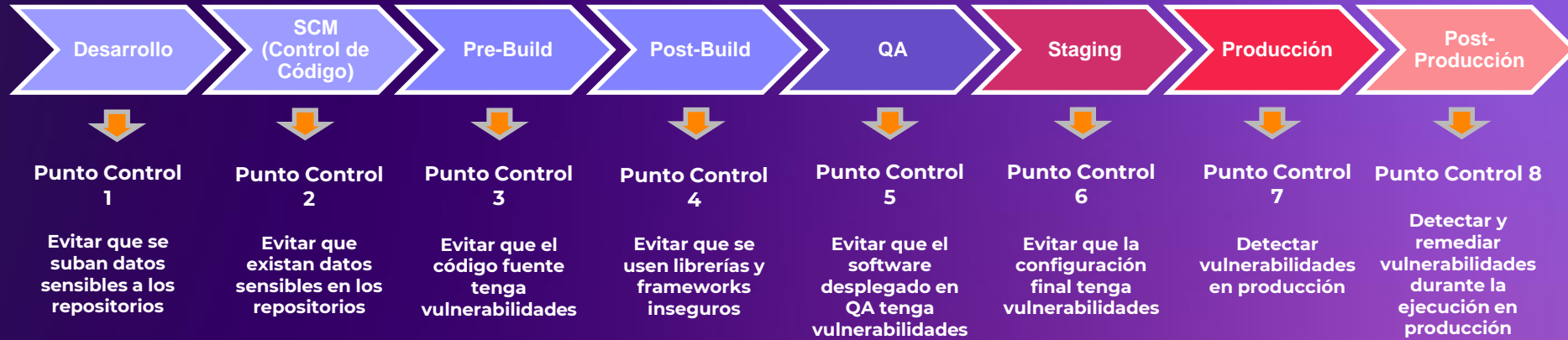


Seguridad en contenedores

Aqua, Trivy, Clair



El súper flujo DevSecOps



Cómo lo hacemos							
<ul style="list-style-type: none">- IDE Plugin- Pre Commit Hooks- Detección de Secretos	<ul style="list-style-type: none">- Post Commit Hooks- Detección de Secretos- Escaneo Repositorio	<ul style="list-style-type: none">- SAST- Testing Continuo- Unit Testing	<ul style="list-style-type: none">- Escaneo SCA- Container & IaC Scan	<ul style="list-style-type: none">- DAST- IAST- Integration testing	<ul style="list-style-type: none">- Escaneo de vulnerabilidades- Validación Compliance- Chaos Engineering	<ul style="list-style-type: none">- PenTesting- Chaos Engineering	<ul style="list-style-type: none">- Monitoreo- Alertas



Herramientas clave en DevSecOps

DevSecOps integra seguridad en todo el ciclo de vida del desarrollo. Estas herramientas permiten detectar, prevenir y mitigar vulnerabilidades desde el inicio del desarrollo hasta la implementación.

Análisis de Seguridad



SonarQube

Análisis estático de código para detectar bugs, vulnerabilidades y "code smells"



Checkmarx

Análisis de seguridad de aplicaciones (SAST) para identificar vulnerabilidades en el código

Escaneo de Dependencias



OWASP Dependency-Check

Identifica componentes vulnerables en las dependencias del proyecto



Snyk

Detecta y corrige vulnerabilidades en dependencias y contenedores en tiempo real

Seguridad en Contenedores



Aqua Security

Seguridad para contenedores y aplicaciones cloud native desde el desarrollo hasta producción



Trivy

Escáner simple y completo para vulnerabilidades en contenedores e imágenes Docker



Consejo: Integra estas herramientas en tu pipeline de CI/CD para automatizar la detección temprana de vulnerabilidades.



¿Qué es SRE (Site Reliability Engineering)?

SRE es una disciplina creada por Google que aplica principios de ingeniería de software a las operaciones de TI, con énfasis en fiabilidad, automatización y escalabilidad.

Principios clave:



Error Budgets

Balance entre innovación y fiabilidad



Monitorización

Basada en métricas y alertas efectivas



Automatización

Reducción de tareas manuales (toil)



Ingeniería de sistemas

Enfoque sistémico para operaciones de TI

SLI/SLO/SLA:



SLI

Service Level Indicators: métricas que miden el rendimiento real (latencia, disponibilidad, tasa de errores).



SLO

Service Level Objectives: objetivos internos de rendimiento, como "99.9% de disponibilidad en 30 días".



SLA

Service Level Agreements: contratos con consecuencias financieras o legales por incumplimiento.



Herramientas clave en SRE

Observabilidad



Prometheus

Sistema de monitoreo y alerta de código abierto con modelo de datos dimensional



Grafana

Plataforma para visualizar y analizar métricas a través de dashboards personalizables



Datadog

Solución de monitoreo para cloud, servidores, bases de datos y servicios



Gestión de Incidentes



PagerDuty

Plataforma de gestión de incidentes con alertas, escalamiento y guardias



Opsgenie

Sistema de alertas y notificaciones con integración a múltiples herramientas



Incident.io

Herramienta para coordinar y documentar la respuesta a incidentes



Automatización



Terraform

Herramienta de infraestructura como código para crear y modificar recursos



Kubernetes

Orquestador de contenedores para despliegue y escalado automatizado



Scripts personalizados

Bash, Python o Go para automatizar tareas comunes de SRE





Consejo SRE: Las herramientas por sí solas no son suficientes. SRE utiliza el concepto de "presupuesto de error" (error budget) para balancear la fiabilidad y la velocidad de innovación.





¿Qué es Platform Engineering?

“ Platform Engineering es la disciplina que se enfoca en la creación y mantenimiento de plataformas internas para que los equipos de desarrollo puedan construir, desplegar y operar software de manera más eficiente, segura y autónoma.

Objetivos principales:

-  Mejorar la experiencia de desarrollo
Reducir la fricción para los desarrolladores
-  Implementar controles de seguridad
De forma transparente y aplicación centralizada

-  Estandarizar infraestructuras
Crear patrones y servicios reutilizables
-  Acelerar el tiempo de entrega
Reducir el trabajo manual repetitivo

Beneficios para desarrolladores:

- ✓ Autoservicio: acceso a recursos sin depender de otros equipos
- ✓ Enfoque en código de negocio en lugar de configuraciones
- ✓ Aceleración de onboarding y adopción de mejores prácticas





Herramientas clave en Platform Engineering

Las herramientas de Platform Engineering permiten crear, mantener y mejorar plataformas internas para equipos de desarrollo, facilitando la entrega de software de manera más eficiente y segura.

Plataformas internas

Backstage

Plataforma open source de Spotify para crear portales de desarrolladores

Port

Portal de autoservicio para recursos de ingeniería y catálogos de software

Humanitec

Plataforma para crear PaaS personalizados para equipos de desarrollo

Infraestructura

Terraform

Herramienta de infraestructura como código para provisionar recursos

Kubernetes

Orquestador de contenedores para gestionar microservicios

Crossplane

Framework para construir abstracciones de infraestructura cloud

Automatización y CI/CD

Jenkins

Servidor de automatización open source para CI/CD

GitHub Actions

Automatización integrada con GitHub para workflows CI/CD




Tekton

Framework cloud-native para construir pipelines CI/CD

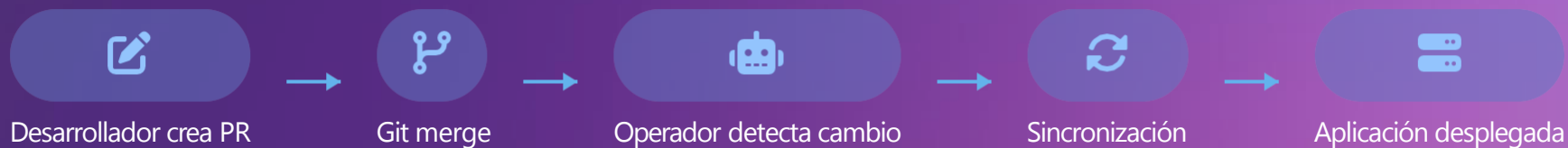


¿Qué es GitOps?

GitOps es un paradigma operativo que utiliza Git como única fuente de verdad para definir y gestionar infraestructuras y aplicaciones, automatizando despliegues utilizando flujos de trabajo basados en Git.

-  Sistema declarativo
Todo el sistema se describe como código en repositorios Git
-  Cambios automatizados
Agentes detectan diferencias y sincronizan automáticamente
-  Estado deseado versionado
Git es la única fuente de verdad para el estado del sistema
-  Reconciliación continua
Sistema auto-reparable que mantiene el estado deseado

Flujo de trabajo:



Beneficios:

-  Mayor transparencia y trazabilidad con historial de cambios en Git
-  Recuperación ante desastres simplificada mediante reconciliación automática
-  Mejor colaboración entre equipos de desarrollo y operaciones
-  Simplificación del proceso de despliegue y reducción de errores



Herramientas clave en GitOps

“ GitOps utiliza Git como fuente de verdad para definir y gestionar infraestructura y aplicaciones. Las herramientas GitOps automatizan el proceso de despliegue basándose en los cambios realizados en repositorios Git.



Git & Plataformas Git

GitHub, GitLab, Bitbucket

- ✓ Almacenamiento de manifiestos declarativos
- ✓ Control de versiones y colaboración
- ✓ Trazabilidad y auditoría de cambios



Argo CD

Controlador de despliegue continuo declarativo

- ✓ UI Web avanzada para visualizar despliegues
- ✓ Sincronización automática con Git
- ✓ Soporte para múltiples clusters



Flux CD

Controlador GitOps de CNCF

- ✓ Enfoque minimalista y sin servidor
- ✓ Integración nativa con notificaciones
- ✓ Potente sistema de reconciliación



Kubernetes

Plataforma de orquestación de contenedores

- ✓ API declarativa para definir estados objetivo
- ✓ Control de versiones por objeto
- ✓ Integración con operadores de GitOps





**“Quien conoce al
enemigo y se conoce a sí
mismo, disputa 100
combates sin peligro”**

~ Sun Tzu ~



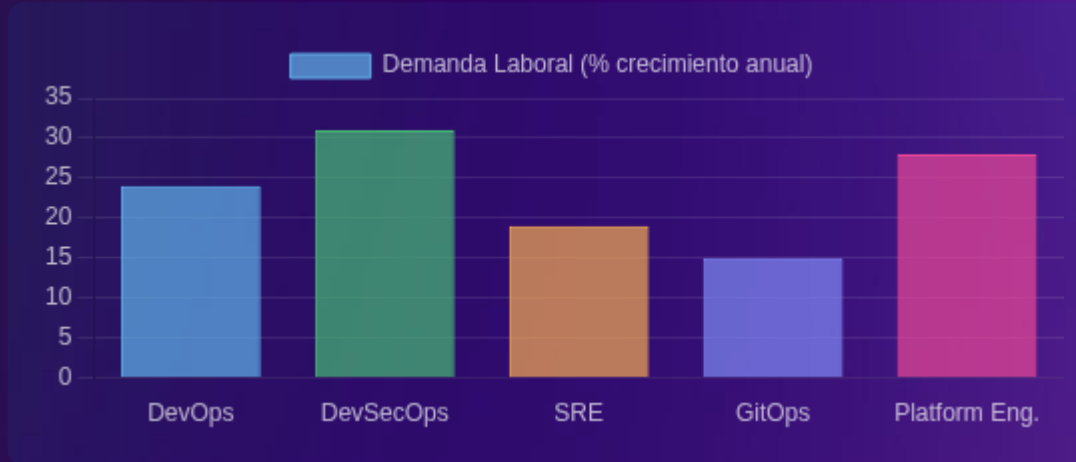
**“Quien conoce su
aplicación y se conoce a sí
mismo, ejecuta 100
deploys sin peligro”**

~ El hermano DevSecOps de Sun Tzu ~



Resumen clave

Tendencias y demanda laboral 2025



Crecimiento de salarios
25-40% superior a roles tradicionales de desarrollo

Adopción empresarial
El 83% de empresas están implementando prácticas DevOps

DevSecOps en auge
Crecimiento del 47% en requisitos de seguridad integrada

💡 Recomendaciones para arrancar

- 1 Comienza con lo fundamental
 - ✓ Domina Linux y scripting básico (Bash, Python)
 - ✓ Aprende conceptos de redes y virtualización
- 2 Construye un laboratorio práctico
 - ✓ Despliega entornos con VMs o servicios cloud gratuitos
 - ✓ Experimenta con Docker, Git y CI/CD pipelines sencillos
- 3 Participa en la comunidad
 - ✓ Contribuye a proyectos open source desde tu primer año
 - ✓ Construye un portfolio personal en GitHub



Algunos tips

Cursos Gratuitos y de Pago

Cloud
AWS Skill Builder, MS Learn

Udemy
DevOps Bootcamp y Prácticas

KodeKloud
Kubernetes y Docker práctico

A Cloud Guru
SRE y Platform Engineering

Certificaciones Recomendadas

AWS: Arma tu propio path

Kubernetes: CKA, CKAD, CKS

GitOps: GitOps Fundamentals (CNCF)

DevSecOps: DevSecOps Foundation (DevOps Institute)

Azure: Arma tu propio path

GitHub: Fundamentals, Copilot, Advanced Security

Comunidades y Foros

Roxs Ross / Tech With Nana
Jamie Ynoñan / Óscar Lobaton

Animate a explorar, descubrir
y compartir!

Matias Armándola / Edu Spotti
Andrea Griffiths / Mabel Gerónimo

Meetup & Eventos
KubeCon, DevOpsDays

¿Cómo empezar tu carrera?

- Aprende lo básico primero: Linux, redes, bash/scripting
- Crea un entorno de laboratorio personal con VMs o cloud gratuito
- Contribuye a proyectos open source para ganar experiencia
- Artículos en Medium, Dev.to, LinkedIn. Labos hands-on de comunidades, referentes, programas de providers (GitHub, Azure, AWS)



“Los planes no son
nada, la planificación
lo es todo”

~ Dwight Eisenhower ~



Muchas gracias!

