

infraestructura como código

IaaS

ocultar/mostrar

- adaptar metodologías de desarrollo al mundo de sistemas
- control de versiones
- lenguajes de programación
- gestión de la configuración
- despliegues automatizados
- basado en pruebas: integración continua, entrega continua y despliegue continuo
- AUTOMATIZACIÓN

DevOps

ocultar/mostrar

desarrollo del software

- análisis
- diseño
- desarrollo
- pruebas
- despliegue

Metodologías ágiles

- Manifiesto
- XP
- Lean
- Scrum
- Test Driven Development

Integración continua (CI)

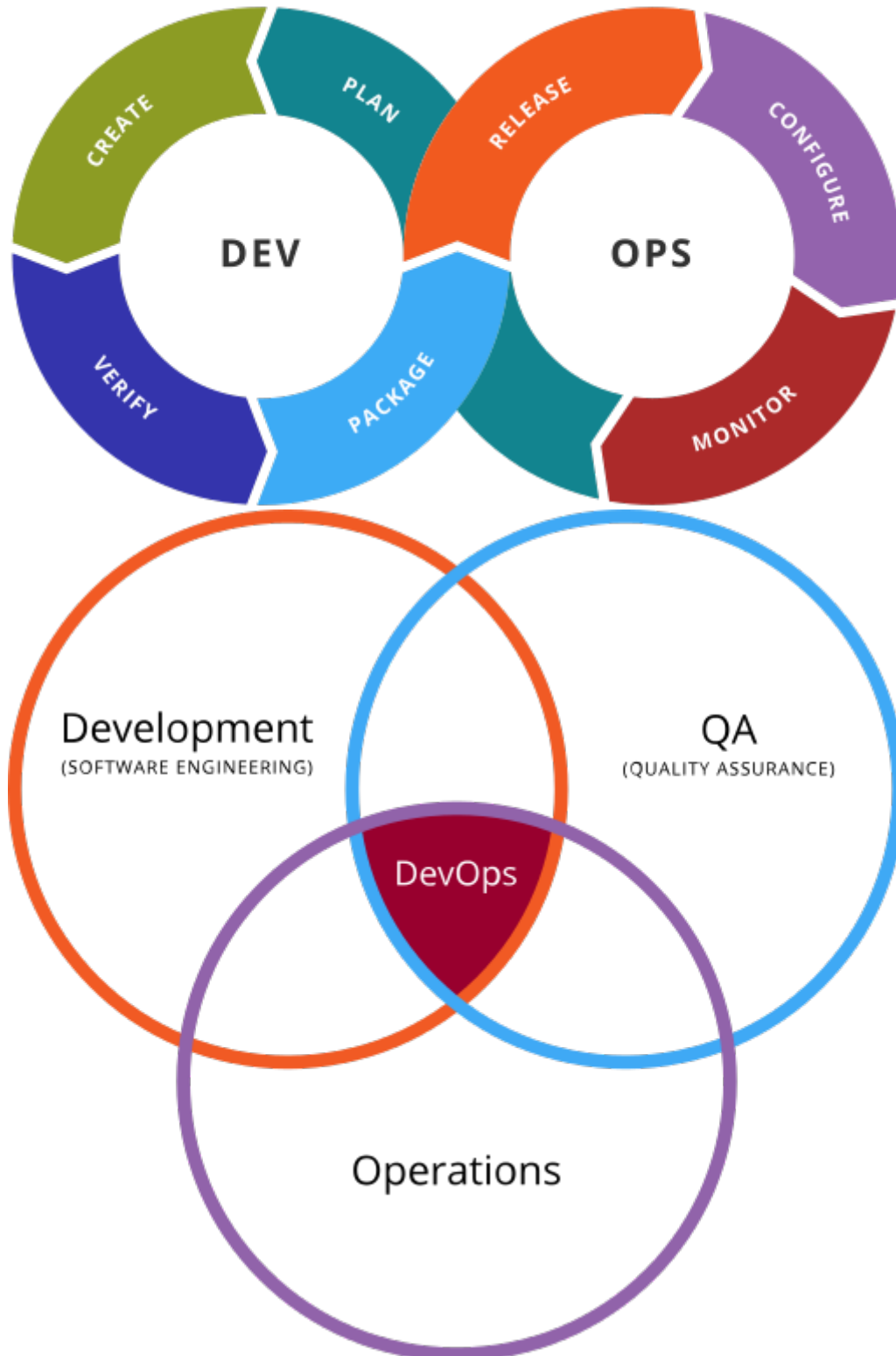
- práctica habitual en desarrollo ágil
- se aplican cambios de forma continua
- se automatiza integración: compilación y pruebas entre componentes
 - no solo en el componente, si no en la interacción entre ellos
- objetivo: detectar pronto los errores
- aplicaciones utilizadas:
 - control de versiones: GIT, SVN, ...
 - Software CI: jenkins, bamboo, travis, ...

Objetivos y responsabilidades

- desarrolladores (devs) que quieren pasar a producción todas las modificaciones
- administradores (ops) que quieren mantener el sistema funcionando sin errores

DevOps

- Desarrolladores
- Testeadores (Q&A)
- Operadores



Entrega y despliegue continuo

- Más allá de CI
 - entrega continua : Integración continua, despliegue manual

- entrega continua + despliegue continuo = todo automatizado
- Se automatiza la generación de una versión publicable (entrega continua)
- se automatiza su paso a producción (despliegue continuo)
- incluye en el ciclo a Q&A y Ops

Aplicaciones utilizadas

- Control de versiones
- herramientas de empaquetado
- orquestación de escenarios
 - los escenarios de desarrollo y producción han de ser iguales (para la correcta comprobación de la aplicación)
- automatización de la configuración
 - herramientas de configuración de la aplicación: crea/replica el escenario necesario para la app

¿devops sin devs?

- ¿podemos tratar la configuración de sistemas como código?
 - si, ya no hacemos el proceso en la máquina, desarrollamos el código que generará esa máquina con sus requisitos
 - desarrollo el escenario que despliega las configuraciones de nuestro sistema
- ¿un administrador que sólo programa y monitoriza?
- Administramos sin tocar en producción
 - administrador de sistemas: ejemplo: aplicación de parche de seguridad → aplico los cambios en mi entorno de desarrollo, veo que eso no rompe la aplicación → se aplica en producción
- ¿desarrolladores de sistemas?
 - metodologías ágiles
 - trabajo en equipo
 - desarrollo
 - cambio de paradigma

Site Reliability Engineer (SRE)

- no necesariamente integrado con devs
- centrado en el entorno de producción
- gestión automática de infraestructura
- actualizaciones
- respuesta a eventos
- <https://devops.com/sre-vs-devops-false-distinction/>

Lenguaje de marcas

ocultar/mostrar

¿no es mejor usar word?

- los formatos binarios con «informática de los 90s», su objetivo no declarado en la incompatibilidad
- formatos en texto plano
- legibles tanto por aplicaciones como humanos
- fácilmente modificables
- su éxito radica en la adaptabilidad de aplicaciones y son esenciales hoy en día

XML

- eXtended Markup Language

- estandar de la W3C
- muchos lenguajes que cumplen las reglas
- formato estricto
- documento bien formado
- validación
- esquemas XML (DTD, XML Schema)
 - XML Schema: definición de como ha de ser el XML y puedo validar que un XML cumple con las reglas establecidas
- XML ha quedado restringido a documentos complejos
 - usado en el principio del API RESTFUL

JSON

- JavaScript Object Notation
- Más sencillo que XML
- Sin esquema y no validable
- Muy extendido en APIs REST

YAML

- YAML Ain't Markup Language
- Más sencillo aún que JSON
- Relacionado con definición de aplicaciones sencillas
- Actualmente es el formato «de moda»

HSL

Ejemplos

```
<?xml version="1.0" encoding="ID0-8859-1"?>
<bookstore>
  <book category="COOKING">
    <title lang="en">Everyday Italian</title>
    <author>Giada de Laurentis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
</bookstore>
```

```
{ books: [
  {
    "title": "Everyday Italian",
    "author": "Giada de Laurentis",
    "year": "2005",
    "price": "30.00"
  }
]
```

books:

-

```
title: "Everyday Italian"
author: "Giada de Laurentis"
year: 2005
price: "30.00"
```

Lenguaje de programación más usados

ocultar/mostrar

Compilados VS interpretados

- en sistemas, típicamente interpretados
- más sencillos
- fácilmente modificables
- pero menos eficientes
- sobre gustos... colores

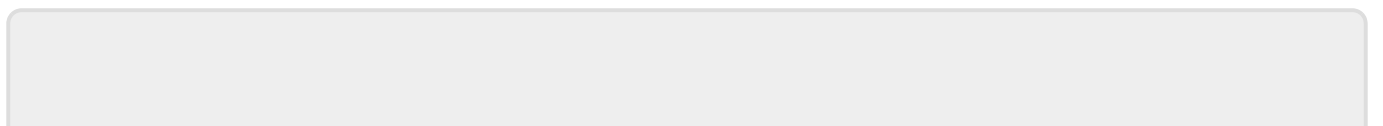
Algunos lenguajes de programación

- ¿bash?
 - pobre en estructuras
- Perl
 - cayendo en desuso
 - sintaxis compleja, difícil captar usuarios
- Python
 - simpleza en sintaxis
 - powerful
 - fácil reutilización de código (o encontrar biblioteca)
- Ruby
 - estilo python
- Nodejs
 - Google
- Go
 - compilado
 - en discusión si es el más adecuado para sistemas
 - la tendencia en las últimas herramientas en este sentido es con este lenguaje
- Rust
 - en desarrollo

Sistema de Control de Versiones

ocultar/mostrar

X



From: <https://miguelangel.torresegea.es/wiki/> - **miguel angel torres egea**

Permanent link: <https://miguelangel.torresegea.es/wiki/info:cursos:openwebinars:intro-cloud-computing:iaac?rev=1530025627>

Last update: **26/06/2018 08:07**

