



Mindset Devops

Mayo 2023

Quién soy



manu.fosela@kairosds.com



@manufosela



in/manufosela

Índice


- Mindset Devops
- Desarrollo resiliente (XP)
- Integración Continua
- Virtualización
- Control de versiones
- Arquitectura Hexagonal
- **Todo Es Código**

Anexos:

- DASA
- Métricas DORA
- SRE



Mindset Devops



El Devops en muchos sitios se entiende como un rol, un trabajo,
una persona...

Devops es un tipo de Mindset agile



Toda persona que esté en un equipo de desarrollo:
Ingenieros, SM, PO, Diseñadores, etc... **debe tener mindset Devops.**



¿Y qué significa tener un mindset DevOps?

- Aumentar el TTM (Time To Market)
- Aumentar la resiliencia en el desarrollo
- Aumentar la usabilidad, vistosidad y accesibilidad
- Reducir las pruebas manuales y los errores
- Reducir fricciones entre roles/departamentos

Desarrollo Resiliente: XP

XP

Es una **metodología de desarrollo de software** orientada a la imprevisibilidad de los requisitos del cliente poniendo énfasis en iteraciones muy cortas que aporten valor rápido al cliente y favorezcan su feedback

Con XP conseguimos:

- Reducir el riesgo del proyecto
- Mejorar la respuesta ante cambios en el negocio
- Mejorar la productividad a lo largo de toda la vida del software
- Producir software de calidad a un ritmo sostenible en el tiempo

Se caracteriza por:

- Funcionar bien con equipos de cualquier tamaño
- Adaptarse a requisitos vagos altamente cambiantes
- Provocar feedback del cliente lo antes posible para ir adaptando sus cambios
- Al tener una fuerte disciplina de tests e integración continua, los fallos se encuentran lo más pronto posible.

Valores XP

- **Comunicación:** constante a todos los niveles: con el cliente y entre los miembros del equipo. El éxito del proyecto está más en riesgo por una mala comunicación que por un mal conocimiento técnico.
- **Simplicidad:** siempre tener en la cabeza KISS, todo lo que se haga debe ser lo más simple e inmediato posible sin pensar en cosas futuras que puedan o no pasar (YAGNI).
- **Feedback:** recibir feedback rápido del cliente es la clave del éxito. Los cambios son inevitables y debemos adaptar la arquitectura para poder soportarlos.

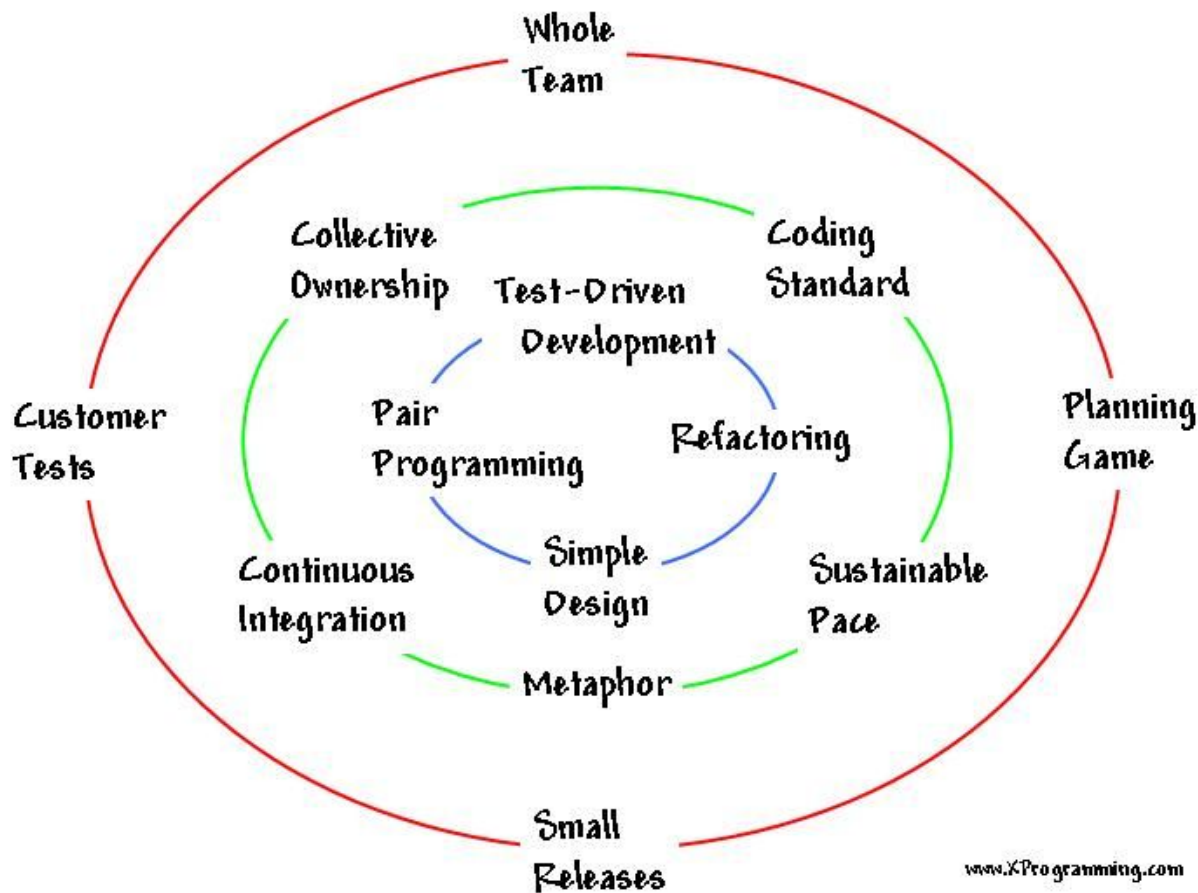
Valores XP

- **Coraje:** entendido como el no miedo a decir la verdad sobre el progreso y las estimaciones, así como expresar los problemas en la daily, todos formamos un equipo.
- **Respeto:** el respeto entre los miembros del equipo personal y profesionalmente, es la clave para trabajar en equipo.

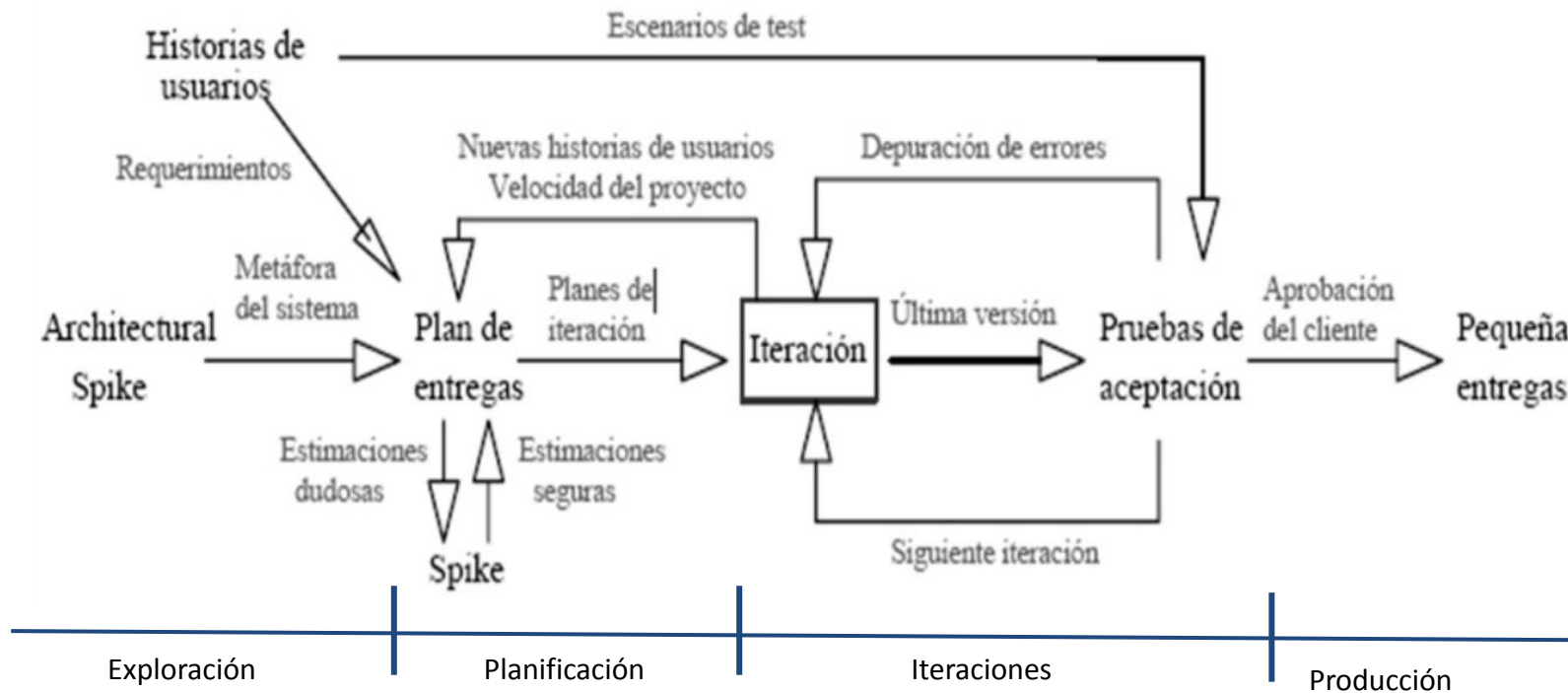
Valores XP

- **Coraje:** entendido como el no miedo a decir la verdad sobre el progreso y las estimaciones, así como expresar los problemas en la daily, todos formamos un equipo.
- **Respeto:** el respeto entre los miembros del equipo personal y profesionalmente, es la clave para trabajar en equipo.

Prácticas XP



Prácticas XP



Integración Continua

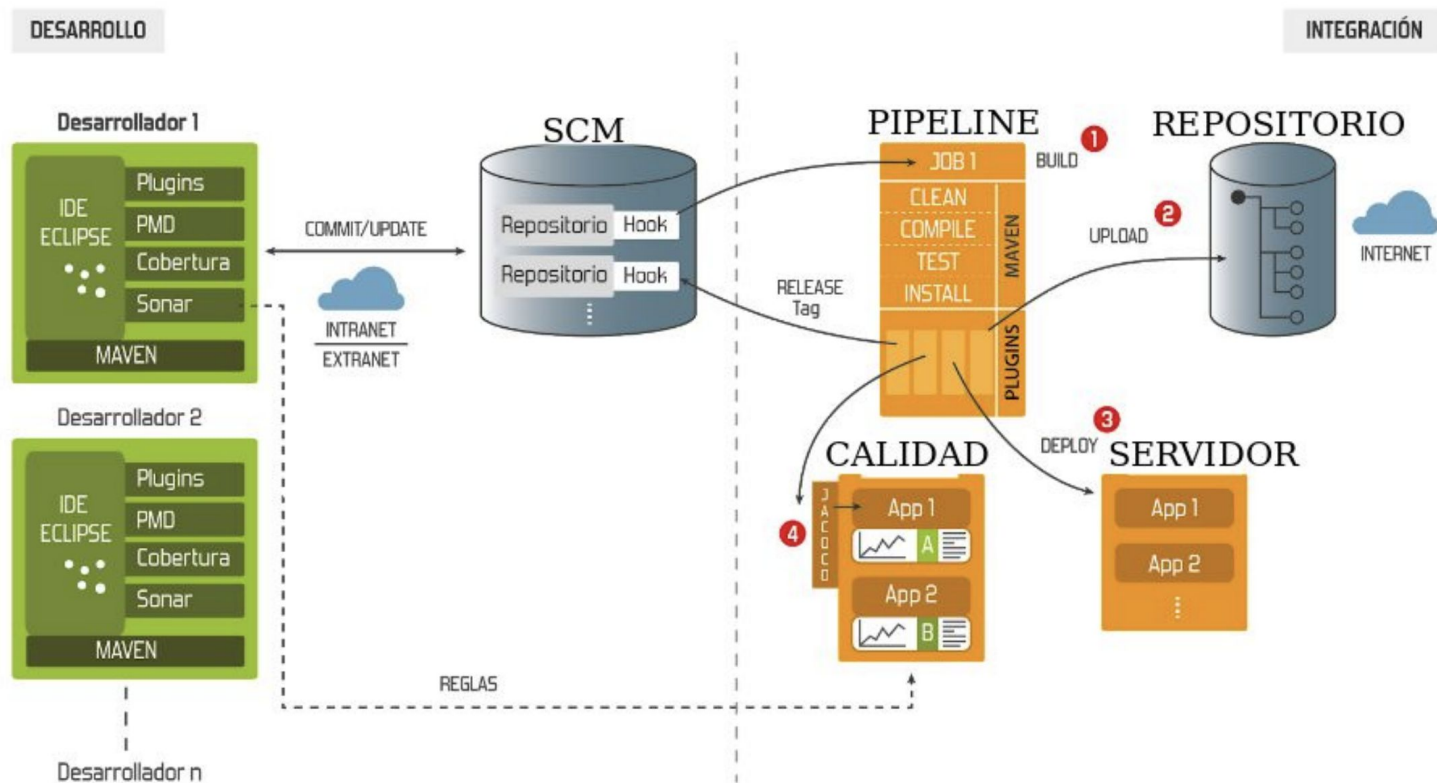
Integración Continua

- Es una parte fundamental del Mindset DevOps que permite **reducir el tiempo de entrega al mínimo** combinado con el despliegue continuo varias veces al día si fuera necesario.
- Reduce el tamaño de las entregas pudiendo ser más fácilmente validadas.
- Consiste en hacer integraciones automáticas de un proyecto lo más a menudo posible para así **poder detectar fallos cuanto antes**.
- Cada integración es verificada por un **build automático** que incluye la fase de ejecución de tests, para detectar los errores de compilación y de deficiencias en las historias de usuario.

Ventajas de la Integración Continua

- Eliminar dependencias de entornos de desarrollo locales.
- Reducir el tiempo de integración de distintas piezas.
- Se detectan errores lo más pronto posible.
- Testeos inmediatos tras un cambio en el código.
- Disponibilidad continua de la última versión del código para tests, demos...
- Aumenta la visibilidad del proyecto.
- Genera confianza del equipo sobre el producto.
- Limita las promociones entre entornos con defectos.
- Integra la calidad aumentando la mantenibilidad del código.

Ciclo de la Integración Continua

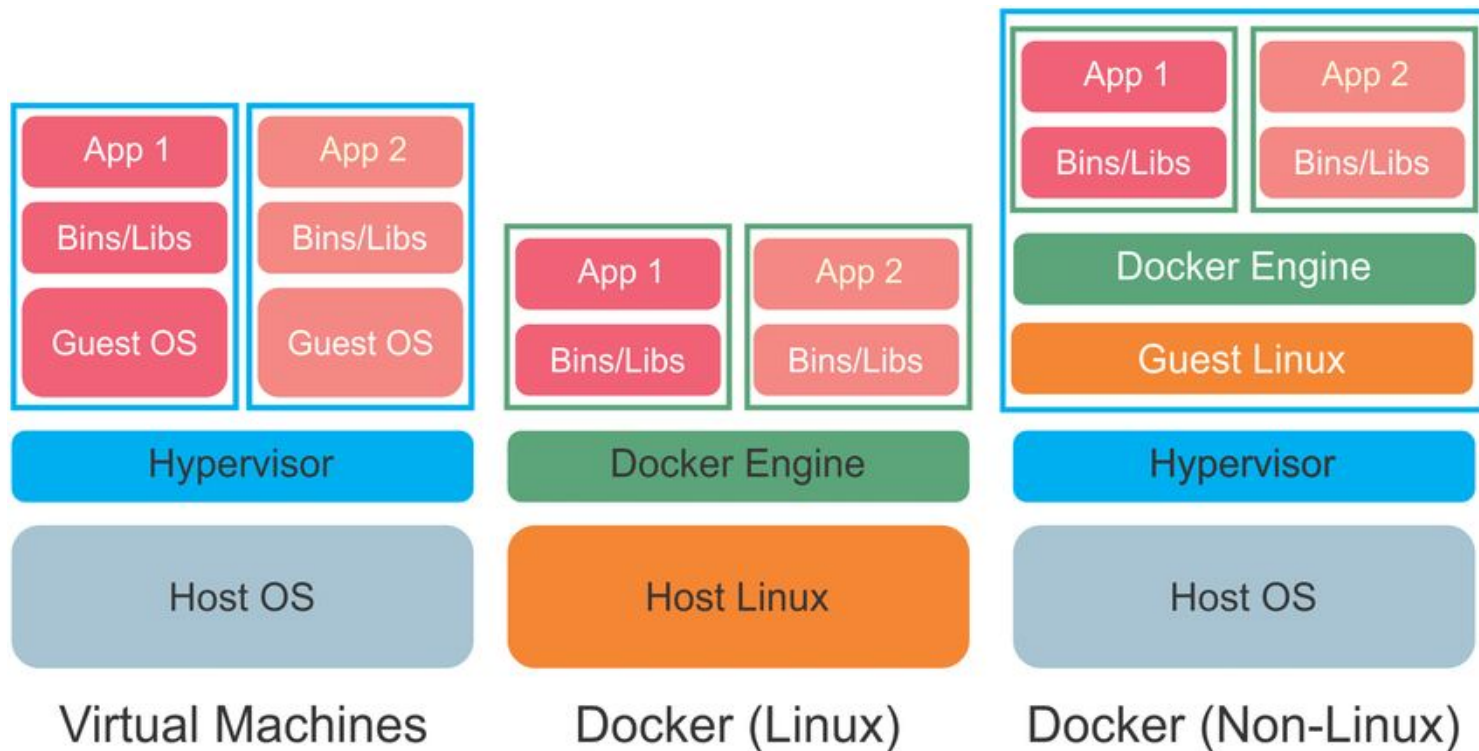


Virtualización

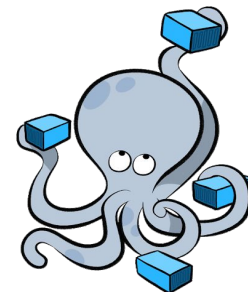
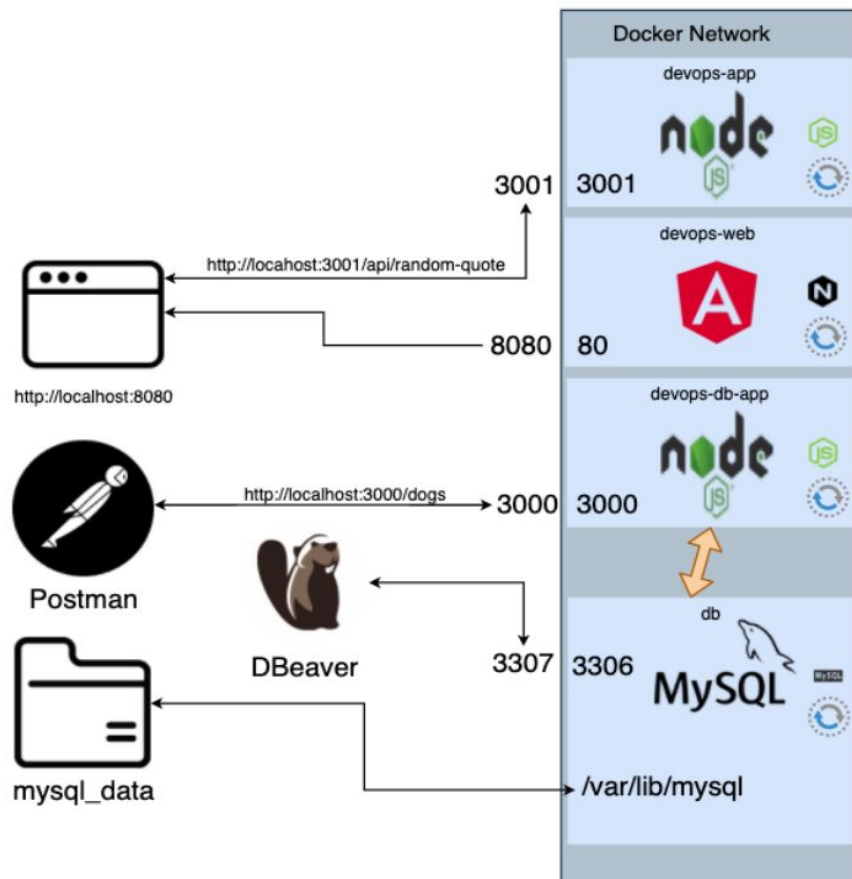
Virtualización

La virtualización es una tecnología que permite crear servicios de TI útiles, con recursos que están tradicionalmente limitados al hardware. Gracias a que distribuye las funciones de una máquina física entre varios usuarios o entornos, posibilita el uso de toda la capacidad de la máquina.

Virtualización



Virtualización con Docker



—

Control de versiones

Control de versiones Git

Es un sistema de control de versiones distribuido.

Permite centralizar toda la información en un repositorio, evitando la copia desactualizada de distintas fuentes (copias en local con el día)

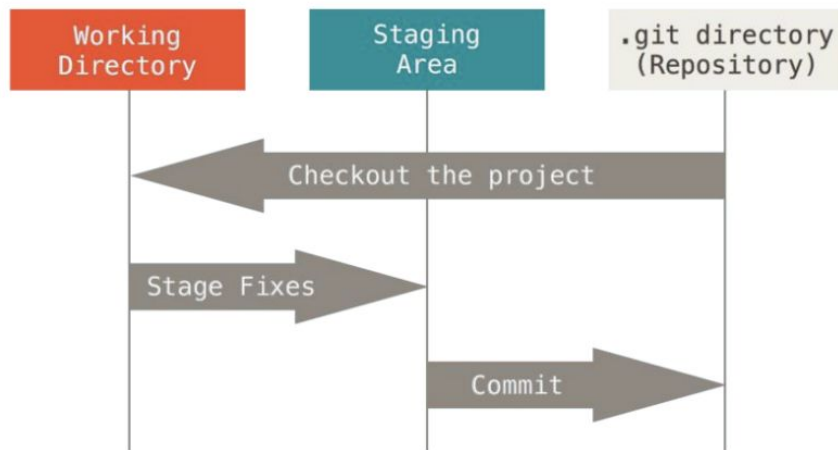
Cada desarrollador tiene una copia en local del repositorio, lo que permite trabajar en modo offline.

Cuando se hace commit de los cambios se actualiza el repositorio local y cuando y cuando se hace push se comparten con el resto de desarrolladores a través del repositorio remoto.

Ciclo de vida de Git

El workflow básico sería:

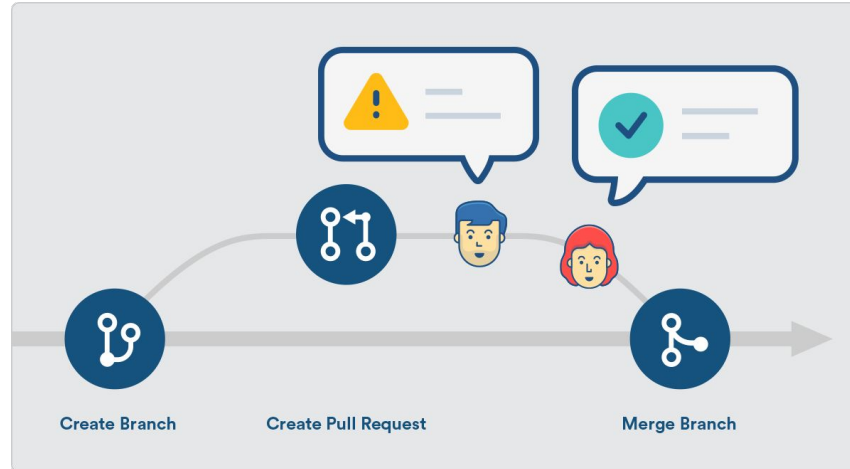
- Se modifica un fichero en tu directorio de trabajo.
- Marcas los ficheros modificados para añadirse al área de Staged (`git add --all`)
- Subes todos los ficheros del área Staged al repositorio local de Git. (`git commit -m "cambio" && git push origin master`)



Git: Pull Request / Merge Request

Es una buena práctica bien extendida que tiene los siguientes beneficios:

- Mejora la calidad del producto.
- Mejora las capacidades de los recursos menos experimentados.
- Hace fluir el conocimiento del producto.
- Mal utilizado puede ser un cuello de botella (único revisor experto)



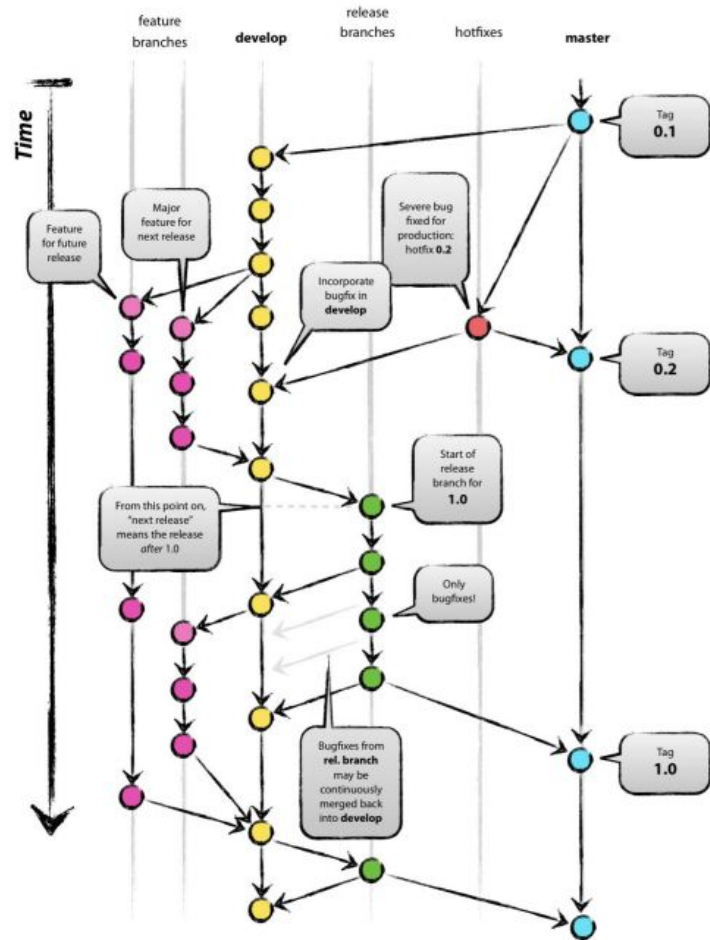
Marco de trabajo: GitFlow

Ramas principales:

- Son ramas permanentes
- No se trabaja directamente sobre ellas. Se protegen master y develop.

Ramas de soporte:

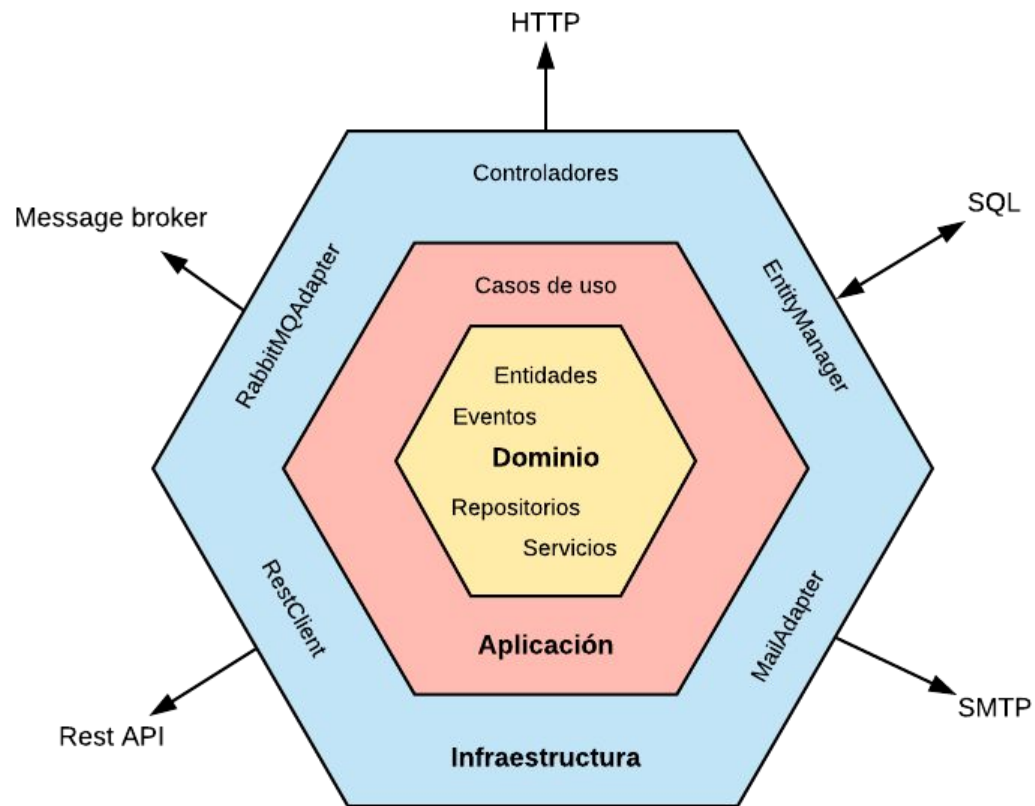
- Feature (parte de develop)
- Release (previo a master)
- Hotfix (parte de master)



—

Arquitectura Hexagonal

Arquitectura Hexagonal



Todo Es Código

Todo Es Código

Todo Es Código

Todo Es Cultura

Todo Es Cloud

Todo Es CX(Customer eXperience)

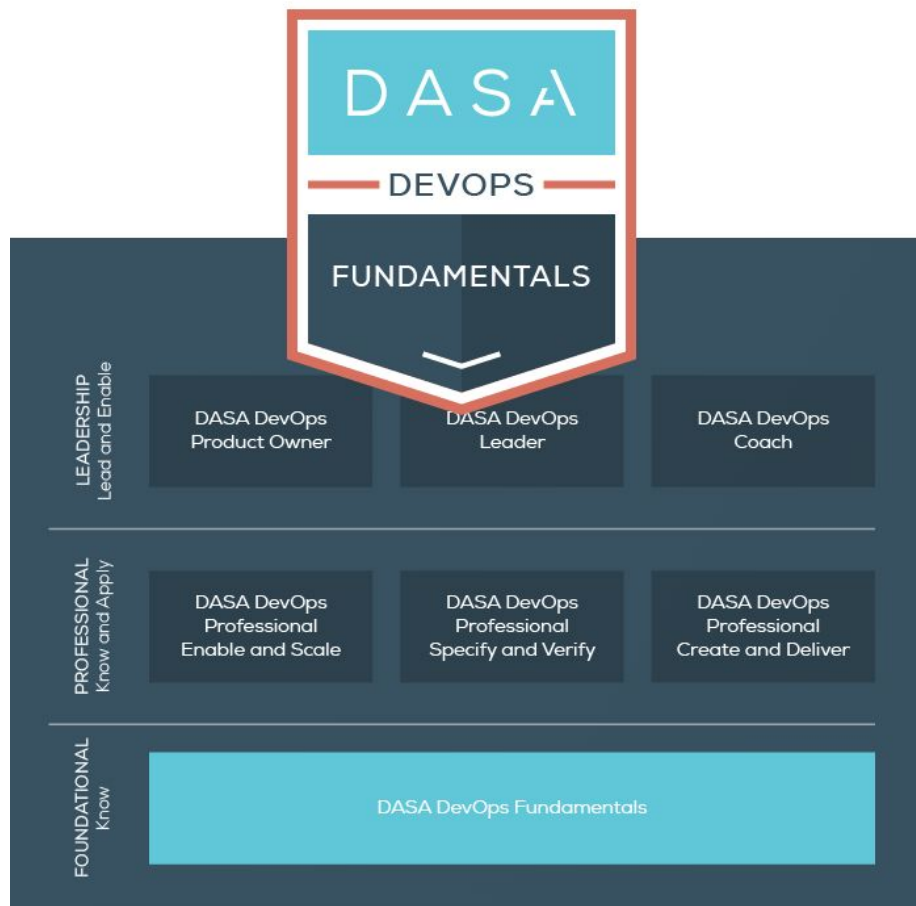
Todo Es Componente

—

Para complementar

D.A.S.A.

Devops Agile Skills Association



Métricas DORA

D.O.R.A es el acrónimo de **DevOps Research and Assessment** un grupo de investigación de Google.

Dicho grupo ha definido unas métricas que permiten a los equipos medir y analizar su rendimiento, con ellas en la mano los equipos serán capaces de tomar una serie de medidas tratando de mejorar su rendimiento.

Métricas DORA

La existencia de estas métricas es muy importante, ya que permite valorar de forma estandarizada y objetiva cuál es el estado de los equipos en lo que respecta a su fase de desarrollo. Previamente esta valoración, al no tener este tipo de métricas, era muy subjetiva, lo que hacía que algunos equipos pensarán que todo estaba bien dentro de sus equipos de ingeniería cuando la realidad era bastante distinta.

Lograr unas buenas métricas será un gran indicador para los equipos en lo que se refiere a una entrega de valor más frecuente, con mayor calidad y en el menor tiempo posible. Algo que sin duda es un indicador súper positivo para cualquier equipo de ingeniería.

Métricas DORA

Frecuencia de implementación (DF)

¿Cuántas veces despliega tu equipo a producción? ¿Hablamos del rango de semanas? ¿Tal vez días? ¿Horas?. Este intervalo de tiempo es una de las métricas que permite conocer cuán rápido es capaz un equipo de llevar nuevos cambios a producción.

Tiempo de espera para cambios (LT)

¿Trabajas en un equipo donde cada integrante del mismo trabaja de forma asíncrona creando posteriormente un Pull-Request con los cambios para que sean revisados? ¿O por el contrario trabajas en un equipo donde se aplican técnicas como el pair programming donde no existen las Pull-Request y los cambios se llevan directamente a la rama principal?

Métricas DORA

Cambiar la tasa de fallas (CFR)

Las dos primeras métricas buscan reflejar cómo de rápido puede entregar valor un equipo, está en cambio nos va a permitir conocer cómo de fiables son los cambios que el equipo lleva a producción.

Tiempo medio de recuperación (MTTR)

Seguro que alguna vez has vivido un despliegue a producción en el que algo sale mal y es necesario volver a la versión anterior. Este proceso conocido como rollback y dependiendo del equipo el tiempo requerido para restaurar una versión anterior puede variar desde el rango de segundos a incluso horas.

S.R.E. (Ingeniero de Confiabilidad del Sitio)

Los **SRE** son sucesores de los equipos de operación típicos y tradicionales. Se puede decir que **SRE** aporta más disciplina al campo de la ingeniería de software.

La ingeniería de confiabilidad del sitio es donde los ingenieros crean sistemas que pueden operar aplicaciones de manera confiable. El inicio del concepto **SRE** comenzó en Google, y esto ha sido muy bien documentado en Google SRE Book .

S.R.E. (Ingeniero de Confiabilidad del Sitio)

Con la ayuda de **SRE**, los ingenieros pueden definir objetivos a nivel de **servicio**. Una vez que se definen los objetivos, se desarrollan los sistemas para lograr estos objetivos. Finalmente, estos sistemas se expanden aún más en plataformas, que abordan áreas como la gestión de incidentes, la mitigación de fallas e incluso la eliminación de los puntos únicos de falla.

Un aspecto muy crítico del enfoque SRE es que **cada falla** se considera una falla y **se realiza un análisis detallado de la causa raíz** para encontrar la razón de la falla. Además, se introducen medidas correctivas a nivel de sistema para el progreso continuo en el aspecto de confiabilidad.

AHORA SÍ

GRANCIAS

Que es como **gracias** pero a lo **GRANDE**

[@manufosela](#)

[in/manufosela](#)



—

—

—

—

—

—

—

