

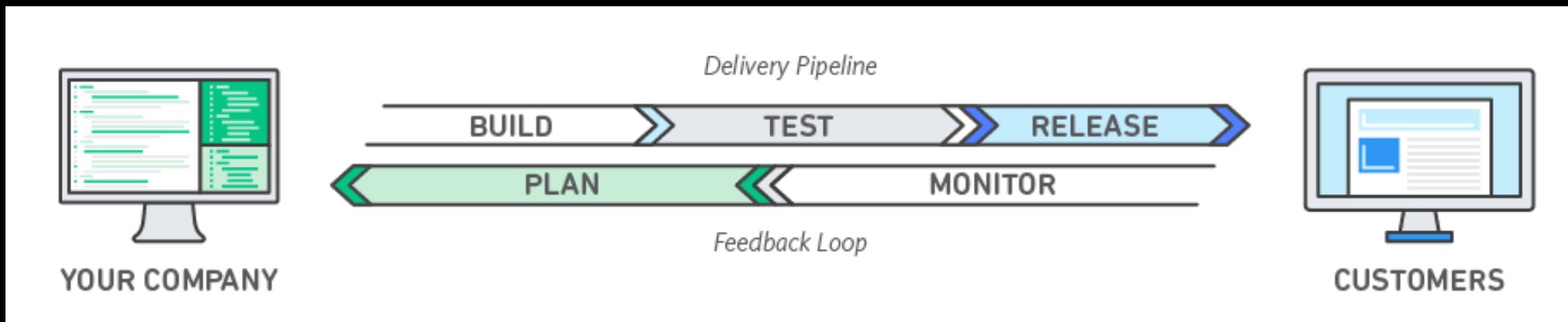
# TÉCNICAS DE INTEGRACIÓN DE CÓDIGO

DevOps / Introducción Flujos de Trabajo

# Desarrollo en el mundo real



Con los roles de "Empresa" y "Cliente, podemos plantear una estructura como esta:



# ¿Cómo es el "Delivery Pipeline"?

Un Delivery Pipeline cambia entre distintas empresas, y no tiene que apegarse a una estructura definida.

Sin embargo, podemos decir que todas necesitan, de una forma u otra, dos principales grupos:




*Development*

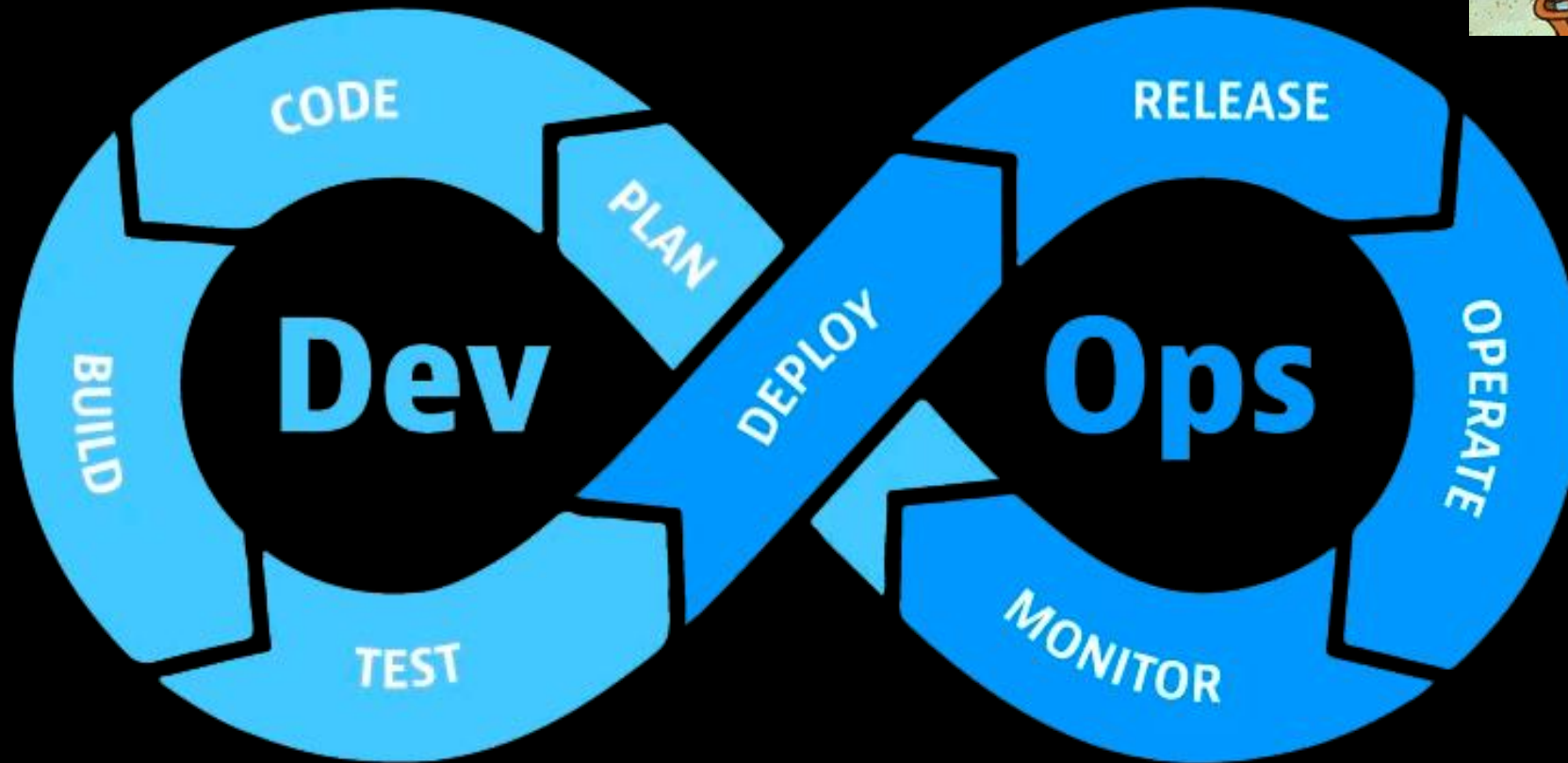


*Operations*

# Con esto en cuenta... DevOps... ¿qué es y qué no?

DevOps es...	DevOps NO es...
Una unión fancy de las palabras "Development" y "Operations."	Una tecnología o una herramienta de trabajo específica.
Una metodología o modelo de trabajo donde los equipos de operaciones y desarrollo se unen.	Un lenguaje de programación.
Un conjunto de prácticas y filosofías que buscan optimizar tiempos, reducir errores y tener en cuenta la mejora continua.	Un rol de trabajo o un título. No existe un perfil "DevOps".
Un modelo de trabajo que hace uso de distintas prácticas y tecnologías para hacer el desarrollo más colaborativo y eficiente.	Una marca de chocolate. 

# ¿Cómo funciona?



# Lo más importante: Automatización



Con DevOps, se busca hacer lo más posible de forma automática.

¿Por qué?

- Eficiencia
- Consistencia
- Escalabilidad
- Estandarización
- CI (Continuous Improvement).

# Herramientas de DevOps

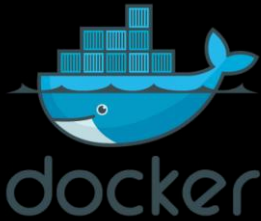
¿En qué etapa pondrías qué?



git



Kubernetes



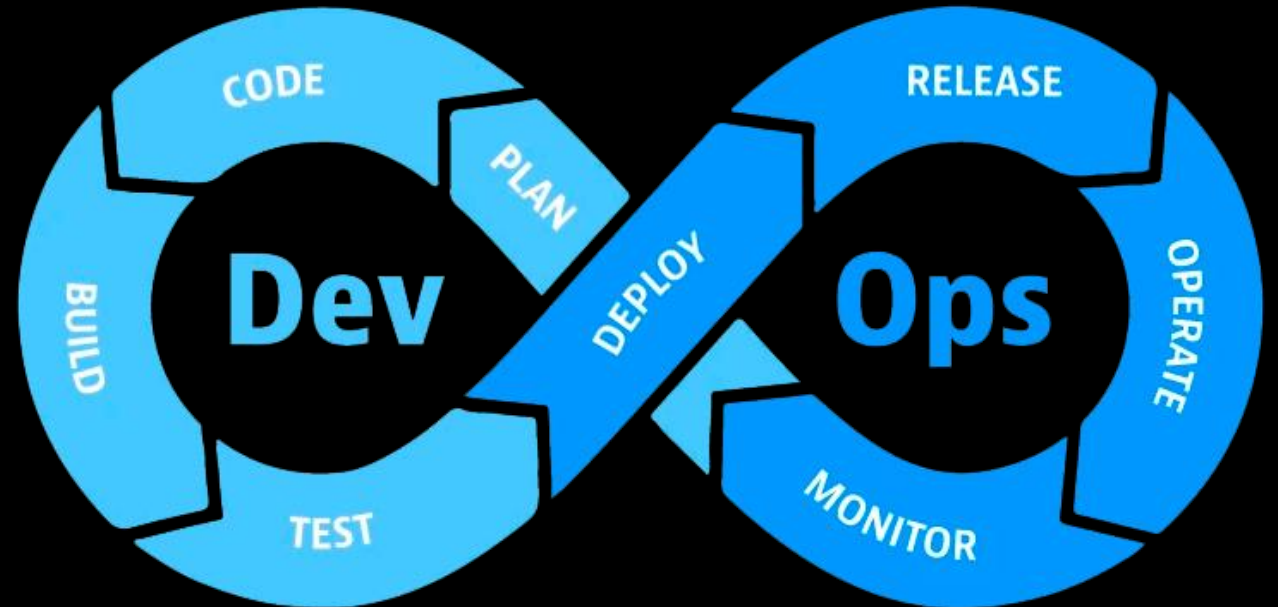
Rainforest QA



Jenkins



Sublime Text



# En equipos de 3 integrantes, determinen en qué parte del flujo propuesto de DevOps irían lo siguiente:

1. Los desarrolladores se reúnen con el equipo de soporte para agendar la fecha de solución a los tickets que se tienen.
  2. Tu equipo de QA hace un sistema con Selenium para probar el Frontend de tu WebApp.
3. Juan, desarrollador de Backend, modifica las líneas 15 y 16 de su archivo ClientsController.php para arreglar un typo.
4. El equipo de operaciones verifica en DataDog el tráfico de su aplicación WEB para ver si requieren un servidor más robusto.
5. Mónica tiene un script de Linux que se ejecuta para compilar el código desde Jenkins cuando se sube a producción, y lo edita para que incluya una nueva dependencia que le indicaron los desarrolladores que necesitará en la compilación.
  6. El Product Owner de tu aplicación prioriza los features que se lanzarán al mercado la siguiente semana.
7. Se sube al servidor de pruebas el contenedor de Docker para verificar la funcionalidad de un feature nuevo.
  8. Se publica la nueva versión de la aplicación de TickTick.
9. El equipo de desarrollo se reúne para revisar en conjunto porqué la compilación de su aplicación de Sencha no está funcionando.
  10. Pancho Tablas hace Pull de los cambios que hizo Rosa Meltrozo para retomar el desarrollo donde ella se quedó.
11. Se hace un análisis del uso de CPU en el servidor de producción para ver las horas pico de uso.
12. Se recibe una llamada de un cliente molesto por un feature que no funciona y se crea un ticket de soporte a partir de ella.
13. Se edita el script que se tiene en Jenkins que corre las pruebas de Backend al subir cambios al servidor de producción.
  14. Se detecta una brecha de seguridad en la aplicación WEB que es canalizada al equipo de Network.



# Flujos de Trabajo en Git



Un flujo de trabajo es una estructura recomendada para usar Git.

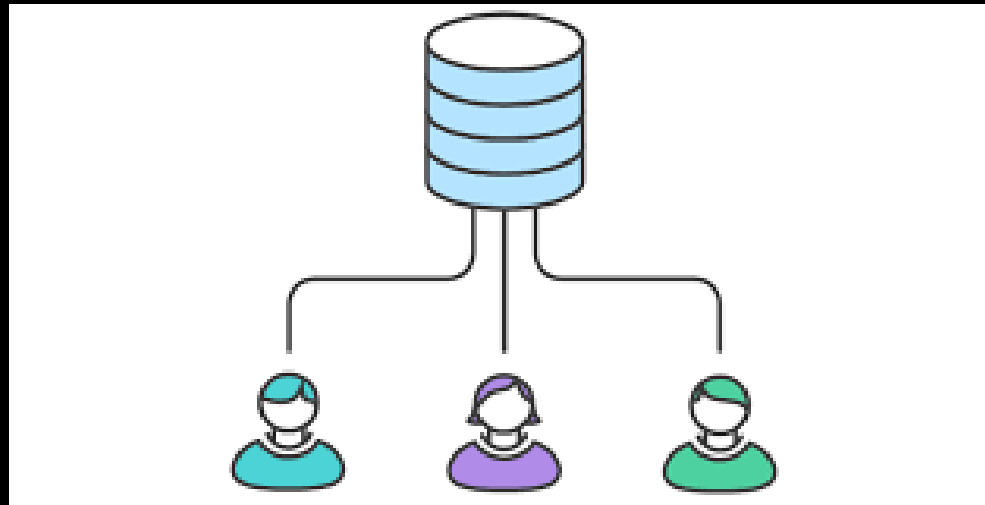
Un flujo de trabajo que se adapte a tu equipo debe ser:

- De escala correcta (¿funciona para un equipo de 2, 3, 10, 100 personas?)
- Permite modificaciones al código de forma fácil y rápida.
  - Debe ser fácil de adaptarse a él.

# Flujo de trabajo centralizado

Los desarrolladores clonan el "repositorio central" en su computadora de forma local. Cada quien tiene una copia del proyecto.

Todos trabajan sobre la rama "main".



¿Ventajas? ¿Desventajas?

# Tarea:

1. En un par de párrafos, imagina un software (nuevo o existente), y escribe de la forma más técnica que puedas cómo se desarrollaría o se le daría mantenimiento a ese software utilizado DevOps (Ej. En una app WEB, se planea cuándo se hará el "Home", y el "Login". Luego, se desarrolla por los developers, que lo empaquetan para que se haga el build...).
2. Describe, en un par de líneas, un proyecto de software, o las condiciones de dicho proyecto (tecnologías usadas, cantidad de miembros del equipo, tamaño del proyecto) en las que te convendría usar un flujo de trabajo centralizado. También explica un proyecto en el que NO sería conveniente usar este flujo de trabajo.