



# DevOps



CERTIFICACIÓN

Universitaria

modo **online**



**mundosE**  
PEOPLE & BUSINESS SCHOOL

**USL** Universidad  
San Ignacio de Loyola

# Encuentro Jenkins- Parte 2

Daniel Ojcius

# Acuerdos

## NORMAS DE COMPORTAMIENTO

Si quieres aportar , no lo dudes!

Para evitar ruidos, silenciar los mic.

Sesión grabada

# Agenda

## TEMAS A DESARROLLAR

- Seguridad en Jenkins
- Jenkins + Docker
- Pipelines
- Branching Model
- Gitflow
- CI/CD
- Practica.



# Seguridad en Jenkins

Algunas recomendaciones abarcan:

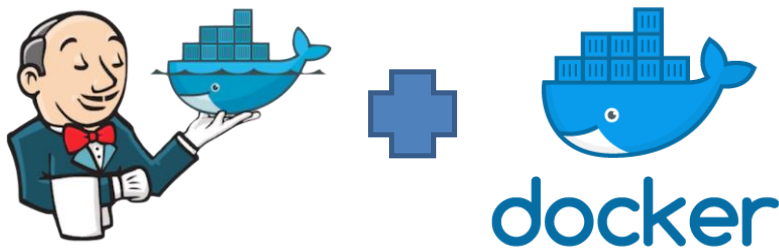
- Proteger Jenkins de internet
- Establecer reglas de firewall
- Poner Jenkins detras de una vpn
- Bloquear puertos para solo acceder por vpn de la empresa.
- Agregar whitelist (github, gitlab)
- Establecer autenticaciones y autorizaciones.
- Mantener Jenkins y plugins siempre actualizados



**Jenkins**



# Jenkins + Docker



- Instalacion
- Como conviven?
- Que podemos hacer?
- DEMO

# Pipelines

Un conjunto de pasos a través de los cuales el código va desde las fuentes a ser un ejecutable en algún ambiente de desarrollo



# Pipeline - Completo

Descargar Repositorio - DEV

Ejecutar pruebas de análisis de código estático - DEV

Verificar compilación - DEV

Ejecutar pruebas unitarias – DEV

Ejecutar pruebas de humo - DEV

Empaquetar - DEV

Instalar en ambiente QA

Ejecutar pruebas de humo - QA

Ejecutar pruebas de regresión - QA

Ejecutar pruebas de integración - QA

Ejecutar pruebas de seguridad - QA

Ejecutar pruebas de rendimiento/carga - QA

Instalar en ambiente STAGING

Ejecutar pruebas de aceptación - STAGING

Ejecutar pruebas de seguridad - STAGING

Ejecutar pruebas de rendimiento/carga – STAGING

Instalar en ambiente de producción: PROD

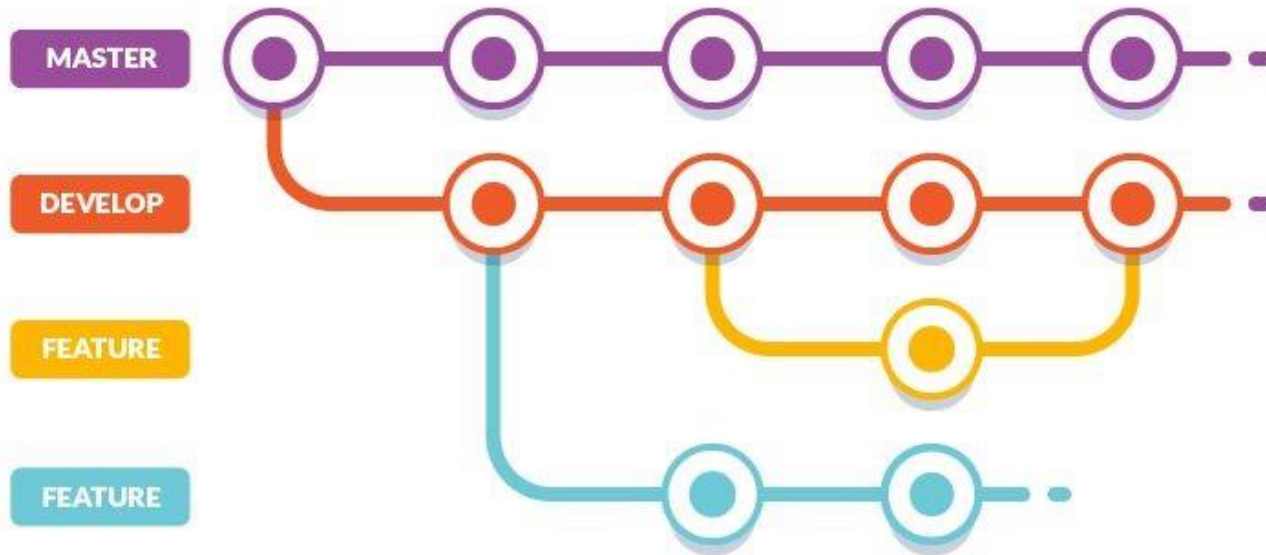
Ejecutar pruebas de humo – PROD

Ejecutar pruebas de seguridad - PROD





# Branching model



# Branching Model

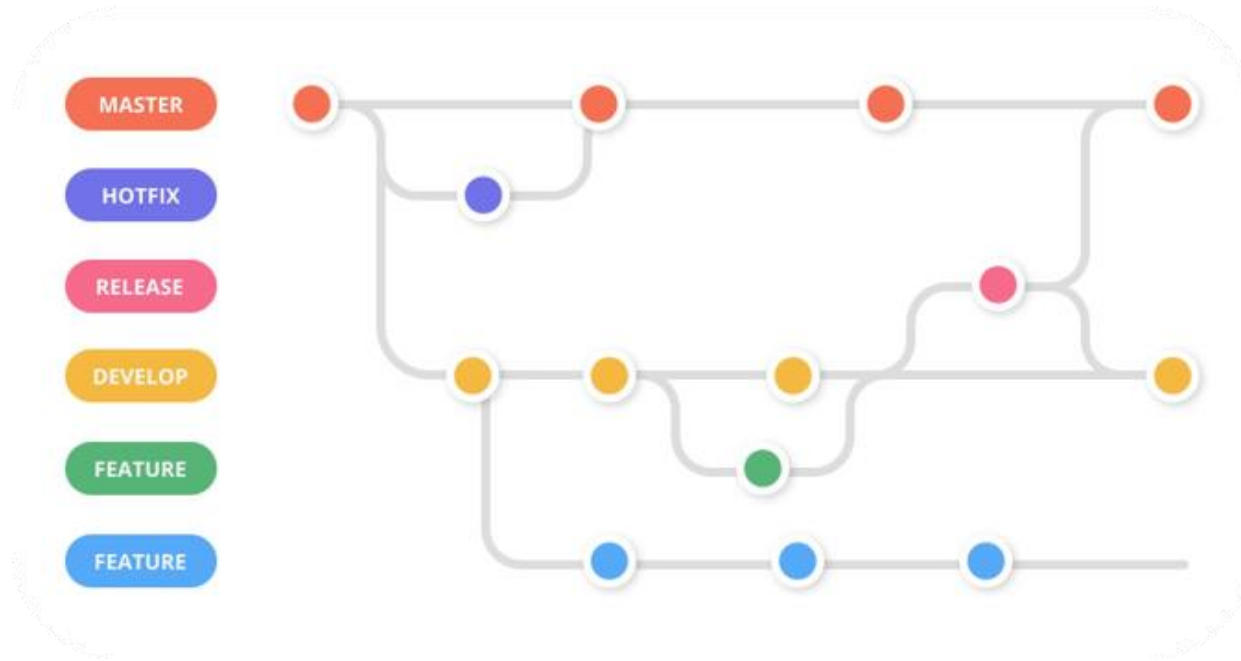
Es un acuerdo en cómo se va a manejar en el código

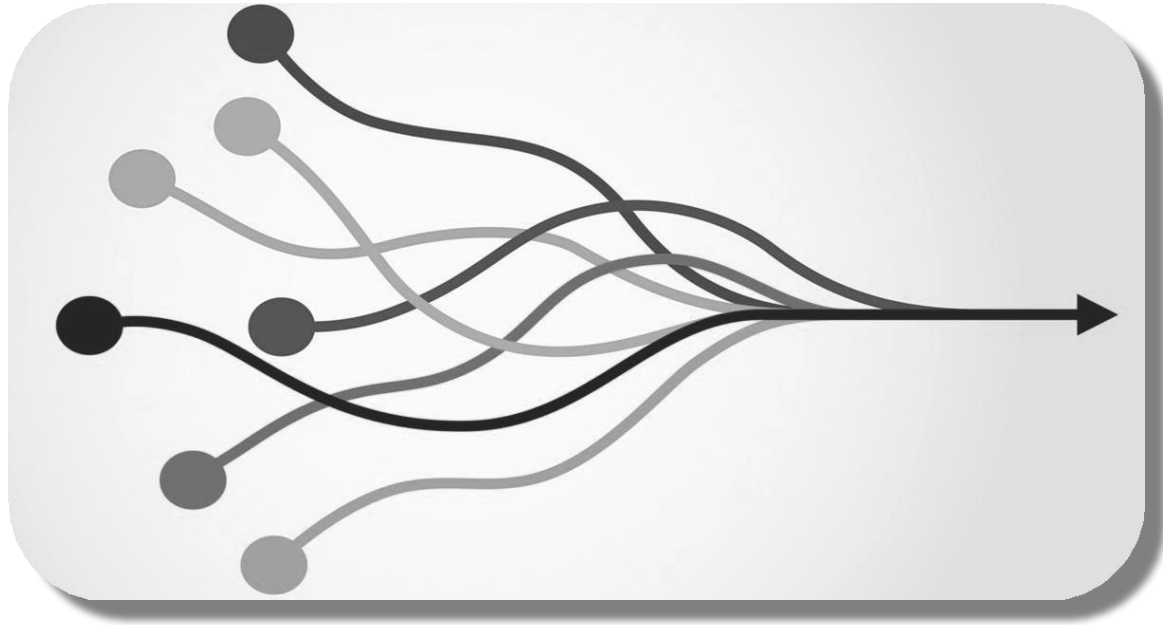
- El proceso de modificaciones
- Versionado

Algunos de los acuerdos (o estándares) mas usados son

- Git Flow
- Trunk Based Development

# Gitflow





# 3 Continuous Integration

# Que es ?

Un conjunto de pasos dentro del pipeline cuyo es objetivo es garantizar que el código sea funcional u operativo con respecto a unas determinadas pautas.

- El Código compila
- El Código pasa todas las pruebas unitarias
- El Código pasa las pruebas de sintaxis o estructura del texto
- El Código pasa las pruebas de humo
- El Código pasa las pruebas de regresión
- El Código pasa las pruebas de seguridad

Por lo general a esto le decimos que el código sea integrable

# Pipeline - DEV

Descargar Repositorio - DEV

Ejecutar pruebas de análisis de código estático - DEV

Verificar compilación - DEV

Ejecutar pruebas unitarias – DEV

Ejecutar pruebas de humo - DEV

Empaquetar - DEV

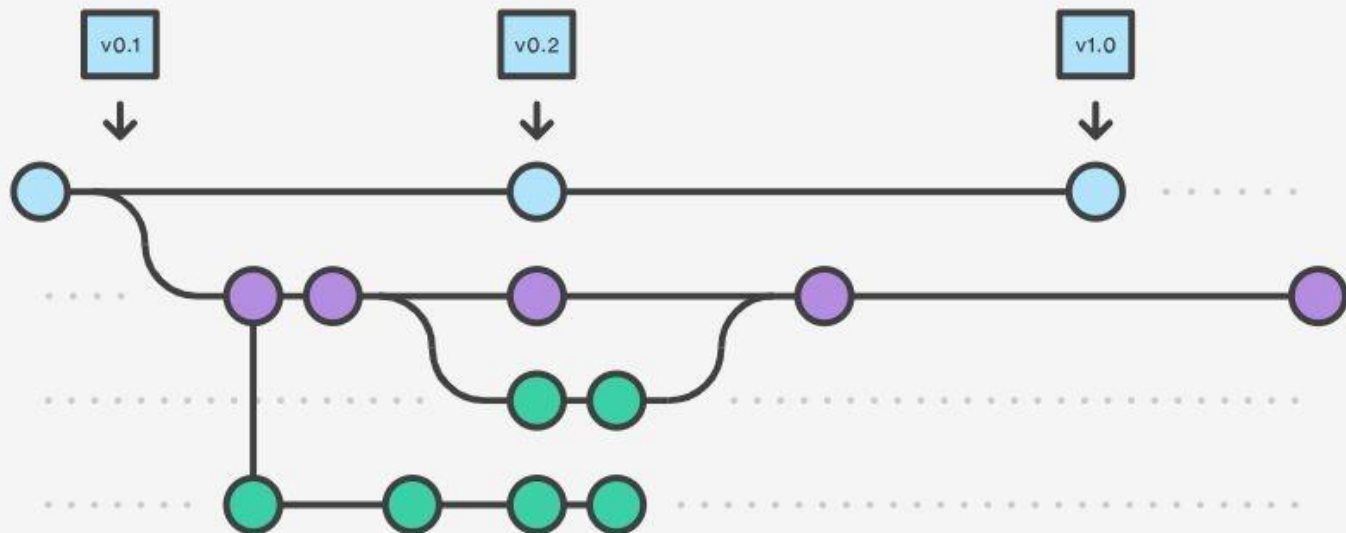


Master

Develop

Feature

Feature



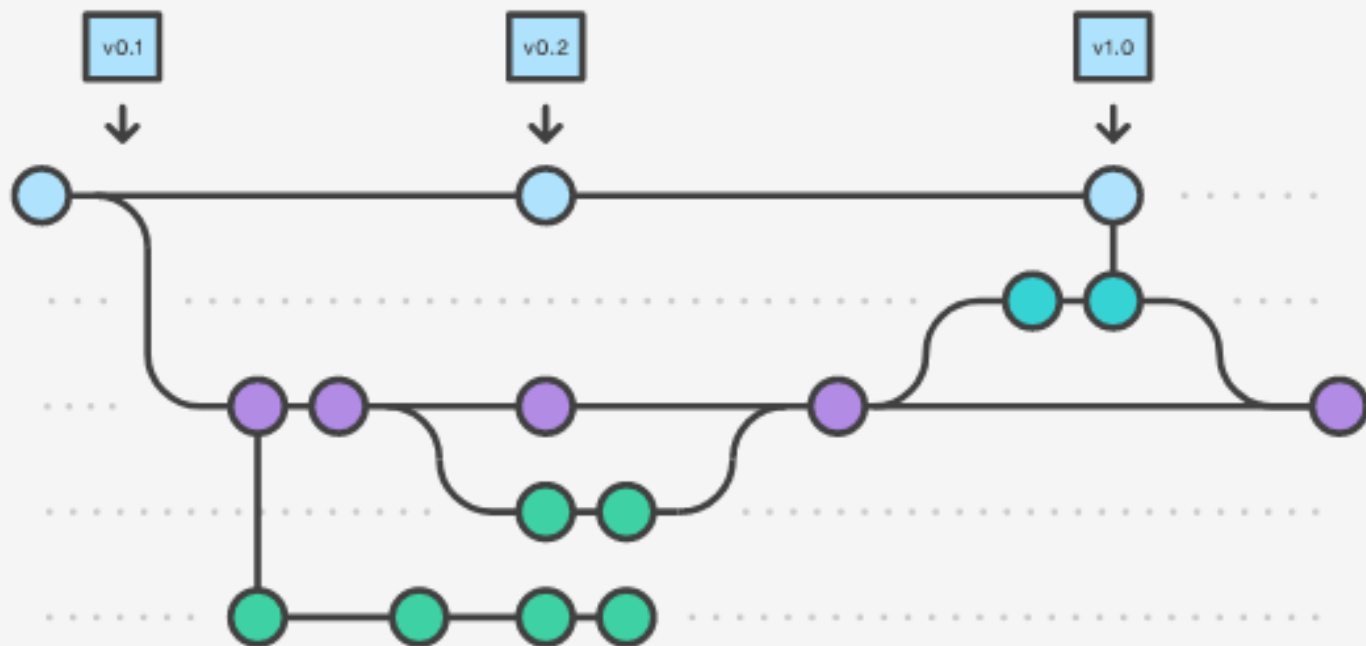
Main

Release

Develop

Feature

Feature





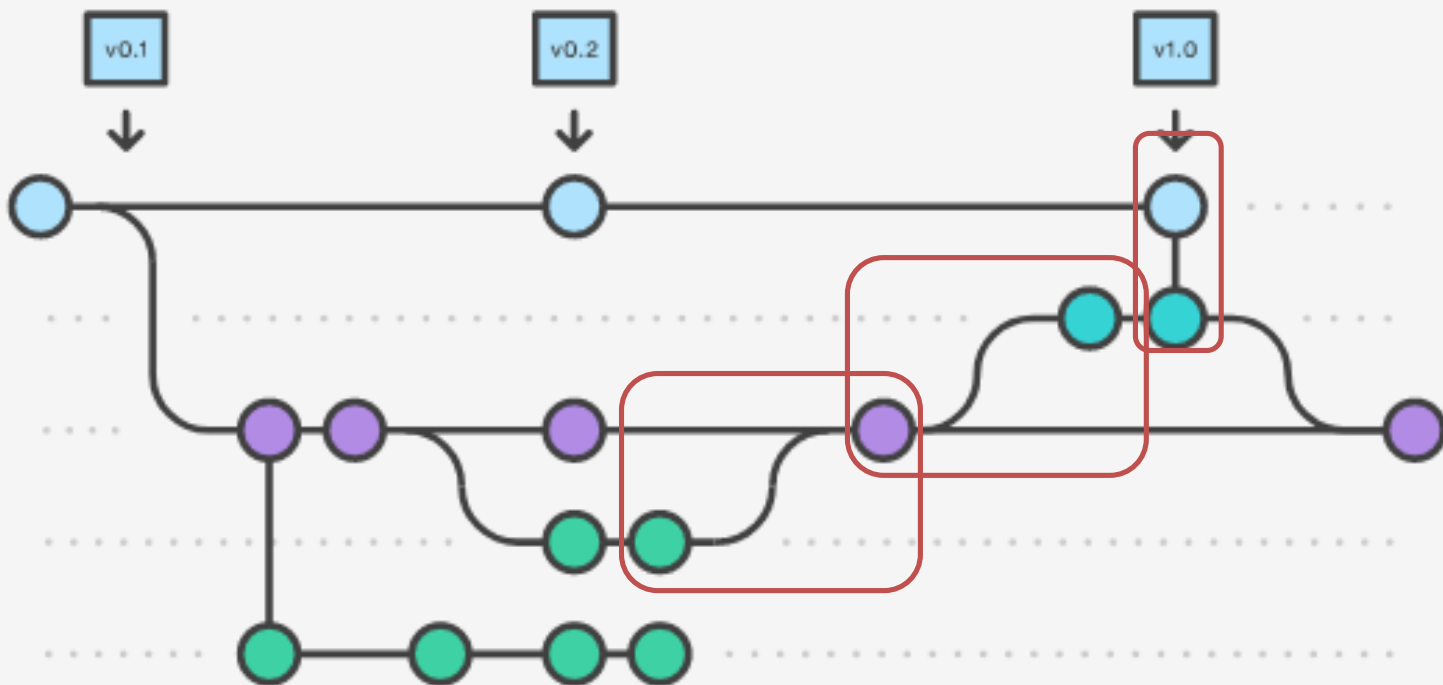
Main

Release

Develop

Feature

Feature





# 4 Continuous Delivery

# Que es ?

Un conjunto de pasos dentro del pipeline cuyo es objetivo es garantizar que el código sea entregable (instalable o deployable)

- El Código pasa las pruebas de integración
- El Código pasa las pruebas de rendimiento/carga
- El Código pasa las pruebas de seguridad
- El Código se empaqueta y tiene una versión específica

# Pipeline QA

Instalar en ambiente QA

Ejecutar pruebas de humo - QA

Ejecutar pruebas de regresión - QA

Ejecutar pruebas de integración - QA

Ejecutar pruebas de seguridad - QA

Ejecutar pruebas de rendimiento/carga - QA





# 5 Continuous Deployment

# Que es ?

Un conjunto de pasos dentro del pipeline cuyo es objetivo es garantizar que el código sea deployado (instalado correctamente)

- El paquete con el código nuevo es instalado
- La nueva funcionalidad pasa las pruebas de aceptación
- La nueva funcionalidad pasa las pruebas de seguridad
- La nueva funcionalidad pasa las pruebas de rendimiento/carga

# Pipeline STAGING

Instalar en ambiente STAGING

Ejecutar pruebas de aceptación -  
STAGING

Ejecutar pruebas de seguridad - STAGING

Ejecutar pruebas de rendimiento/carga – STAGING



# Repaso

## Objetivos:

Que puedan instalar localmente Jenkins, puedan visualizar la aplicabilidad de cada una de las tareas que ofrece y como esto se relaciona con CI/CD



# Recomendaciones

**Paciencia:** Tanto si tienen un nivel inicial o avanzado, aprender a lidiar con los errores y no frustrarse

**Estudio:** Complementar los contenidos del curso buscando información extra pero sin diversificar

**Practica:** Practicar, practicar, practicar...

# CONTACTO

[mundose@ojcius.com](mailto:mundose@ojcius.com)

*Muchas gracias!*

**mundosE**  
PEOPLE & BUSINESS SCHOOL

