



Trunk-based Development



“Las ramas crean distancias entre los desarrolladores y nosotros no queremos eso”

Frank Compagner, Guerrilla Games



Miguel Angel Martínez Espichan

Arquitecto Experto en DevOps



Apoyo a las empresas a mejorar su competitividad a través de la implementación de soluciones en DevOps

 mmartinezepichan@gmail.com

 www.linkedin.com/in/mmartinezepichan

Agenda

Revisemos los siguientes puntos importantes

- 1 **Introducción y Hallazgos**
- 2 **¿Qué es trunk-based development?**
- 3 **¿Como se implementa?**
- 4 **Demo**



Introducción y Hallazgos

Contexto, propósito y porque hacer Trunk-based development

- 1 Introducción y Hallazgos
- 2 ¿Qué es trunk-based development?
- 3 ¿Como se implementa?
- 4 Demo



¿Que es el control código fuente?

“.. es la práctica de **seguimiento** y **administración** de los **cambios** en el código..”



¿Que es una rama y por que crear ramas?

Git permite crear una **copia de su repositorio**, conocida como **rama**. Con esta **rama**, puede **trabajar** en su código con **independencia** de la **versión estable** de la base de código

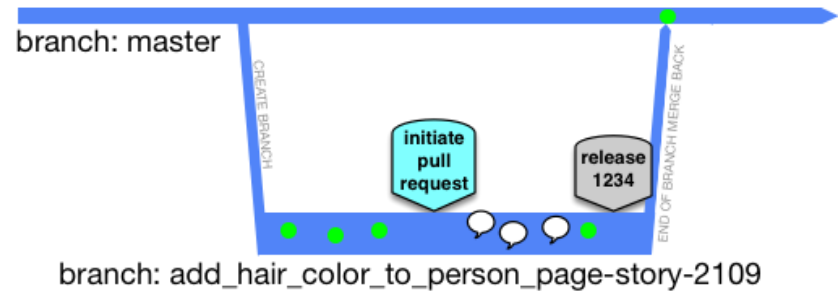
¿Que es Git?

Git es un **sistema** de **administración de código** fuente **distribuido** de código abierto.

Modelos de ramificaciones

Github flow

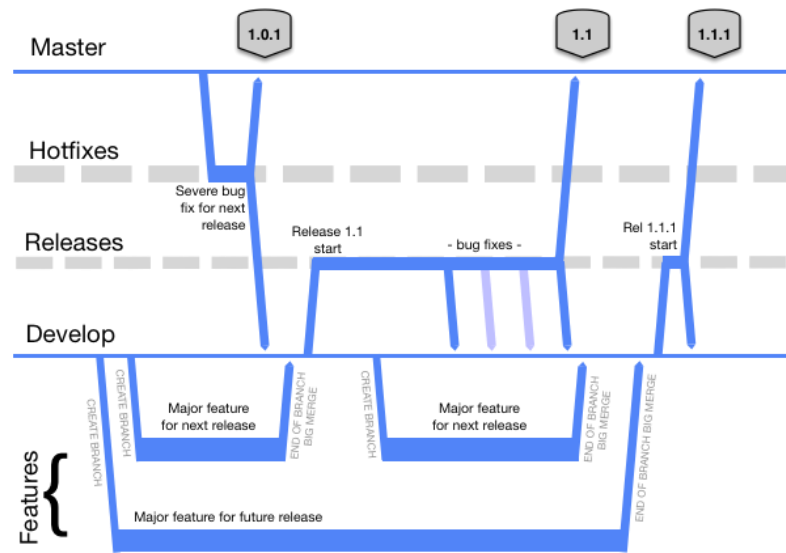
".. Es un **sistema basado en Pull Request**(peticiones de cambios) y ramas de corta duración.."



Problema: Puede que no se reintegre el código a la rama master

Gitflow

“.. define un **modelo estricto de ramificación** diseñado alrededor de la publicación del proyecto..”



Problema: Retrasa la integración continua, ramas que nunca se integran y funcionalidades aisladas

Contexto



En el Estado de DevOps en 2019, DORA **considera clave el uso de trunkbase para mejorar el rendimiento de entrega y operación de software**



Investigation y recolección datos de 31,000 profesionales alrededor del mundo.

Objetivos POR REVISAR



¿Qué modelo utilizan las grandes compañías?

Modelo de ramas en las empresa de más representativas de Internet

facebook

Trunk-based



Trunk-based

Google

Trunk-based

amazon

Trunk-based

Otras compañías relevantes



Trunk-based



Github flow

¿Qué es trunk-based development?

Definición, objetivos, diferencias entre modelos existentes

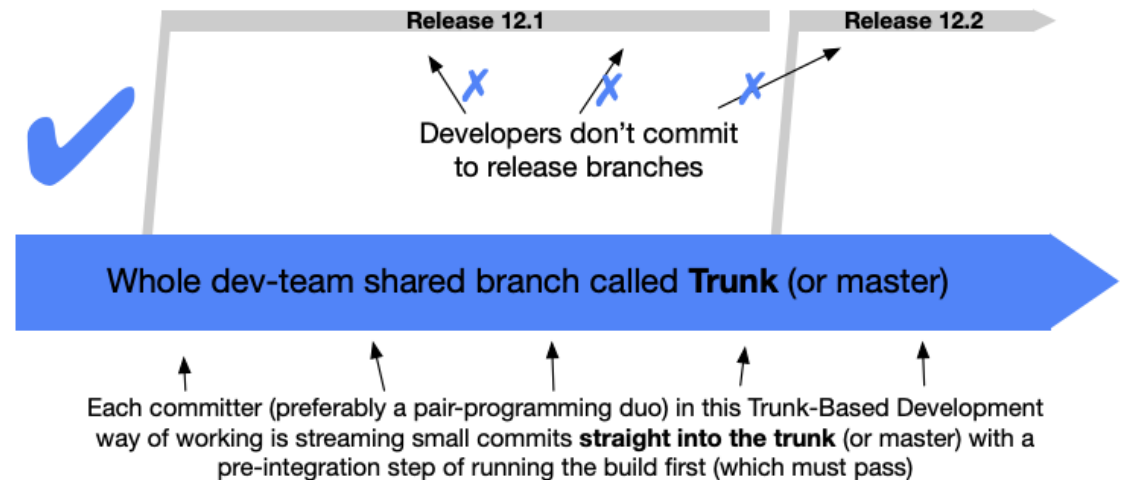
- 1 Introducción y Hallazgos
- 2 ¿Qué es trunk-based development?
- 3 ¿Como se implementa?
- 4 **Demo**



Trunk-based development

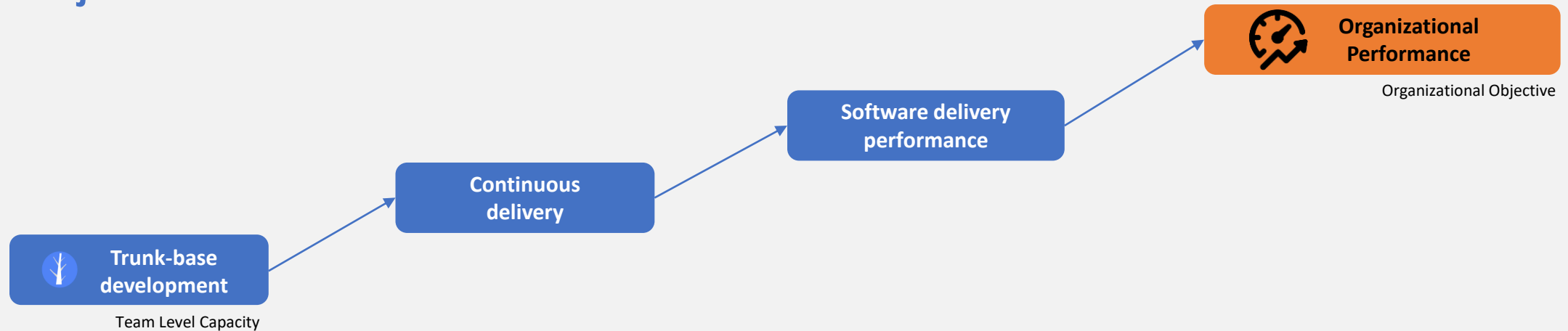
Es un modelo de ramificación para el desarrollo de software. Los desarrolladores colaboran en el código en una única rama llamada Trunk.

<https://trunkbaseddevelopment.com/>



Trunk-based development

Objetivos



Accelerate: State of DevOps 2019

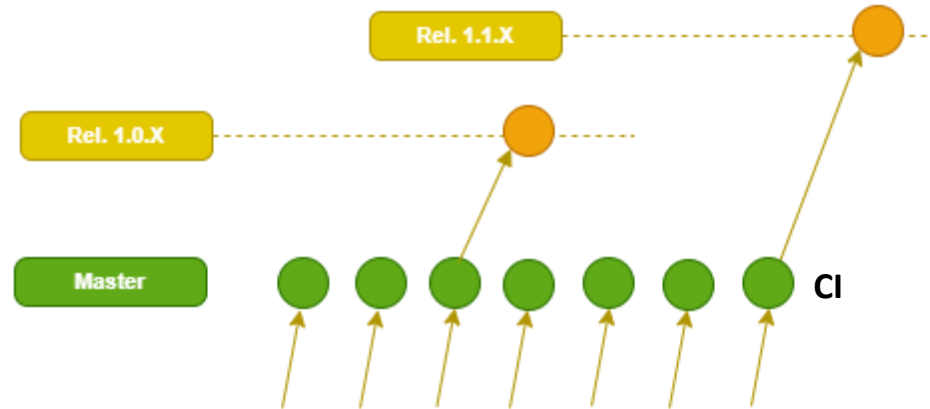


Beneficios

- Cualquier **desarrollo** nuevo está **inmediatamente disponible** para todos
- Lo que se encuentra en **Trunk** está **disponible para ser desplegado**
- **Evita tener ramas aisladas** que nunca se fusionan con la principal
- **Apoya la integración continua** de código

Trunk-based Development o Gitflow

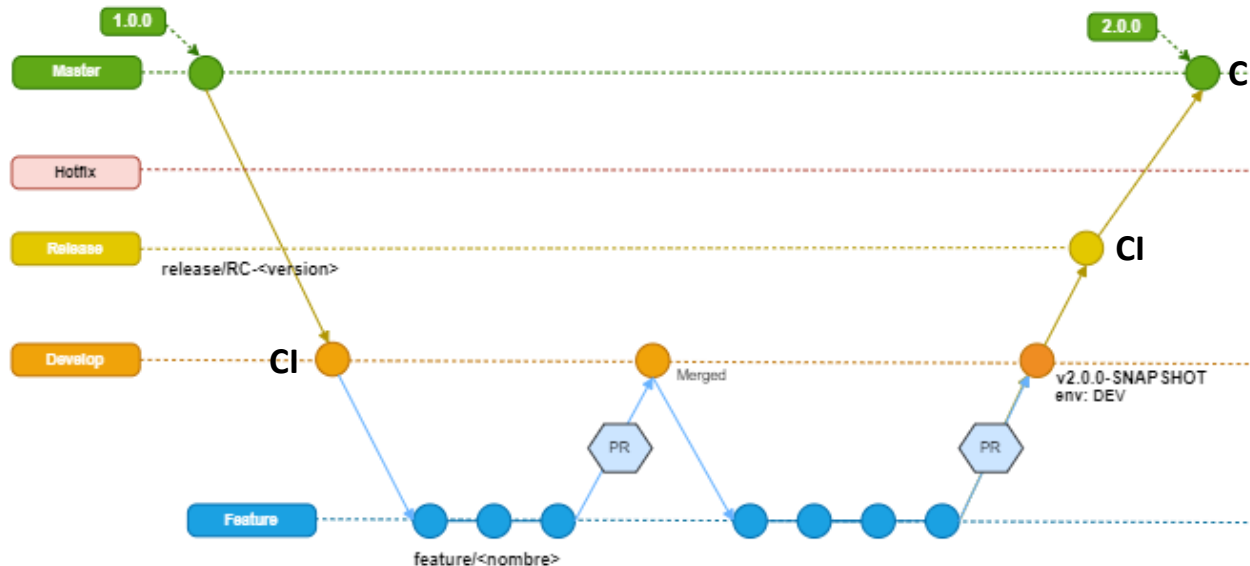
Trunk-based Development



¿Cuándo utilizar TBD?

- **En Pilotos, en startups**
Se necesita obtener un producto lo más rápido posible
- **Cuando se necesita velocidad/cambios rápidos**
Cuando se tienen que dar cambios muy rápidos, cuando se está explorando
- **Equipo con mucha experiencia**
Hay necesidad de cambios rápidos, que se cuente con desarrolladores autónomos, que se trabaje con alto nivel de calidad, que el equipo esté comprometido en corregir problemas centralizados lo más pronto posible

Gitflow



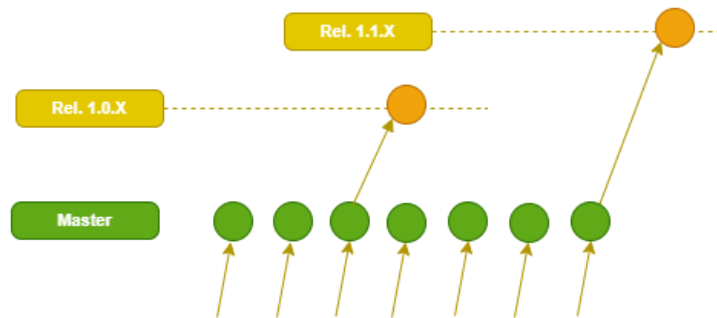
¿Cuándo utilizar Gitflow?

- **Proyectos Open Source**
Múltiples fuentes, múltiples colaboradores y existe necesidad de revisión exhaustiva)
- **Equipo está conformado por muchos desarrolladores junior**
Se necesita un control exhaustivo del desarrollo y cumplimiento de calidad
- **En productos establecidos críticos y controlados**
Donde el foco es la mejora de rendimiento y aumento de capacidades de carga

Tipos de Trunk-based Development

Trunk-Based Development

Para equipos pequeños, se trabaja en un único branch

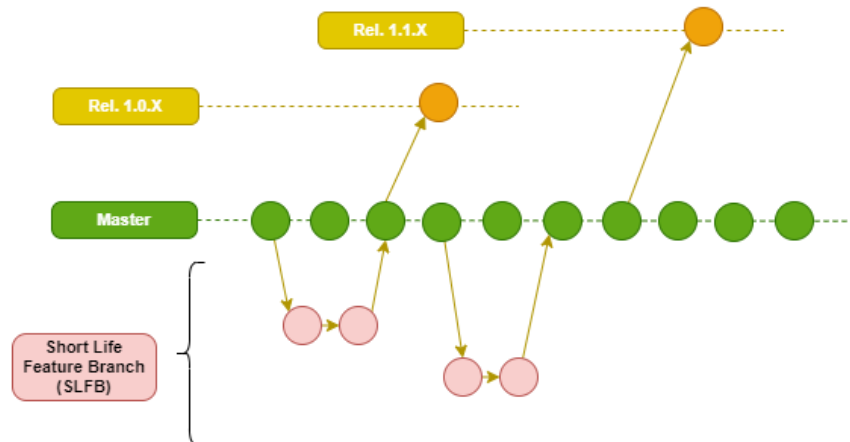


¿Cuándo utilizar TBD?

- En equipos pequeños
- En cambios pequeños
- Preferiblemente aplicar desarrollo de programación de pares
- Exige probar el código antes de desplegarlo

Scaled Trunk-Based Development

Para equipos grandes, se trabaja en short-live feature branch



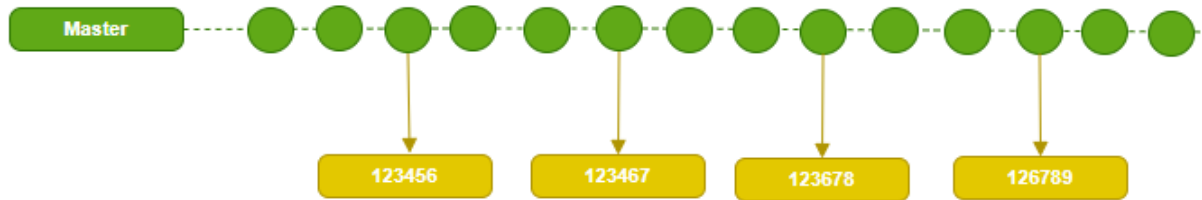
¿Cuándo utilizar STBD?

- Equipo de medio tamaño
- Se requiere que los cambios sean validados antes de llegar al Trunk

Releases usando Trunk-based Development

Release from Trunk

Recomendado para TBD

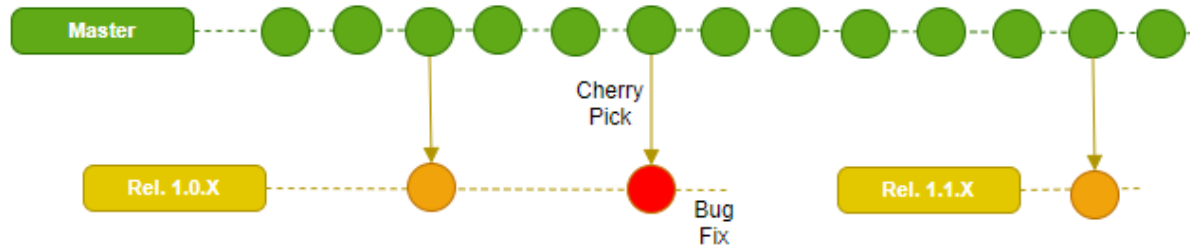


Consideraciones

- Para equipos con alta cadencia de lanzamientos
- Los bugs se corrigen en el siguiente lanzamiento

Branch for release

Recomendado para Scaled TBD



Consideraciones

- Se genera una rama release pocos días antes del lanzamiento a producción
- No se hace push a la rama release
- Las ramas release no se fusionan con master
- Si se detecta un bug entre lanzamientos, a través de cherry-pick se llevan los commits a la rama release y se lanza una nueva versión

Introducción y Hallazgos

Contexto, propósito y porque hacer Trunk-based development

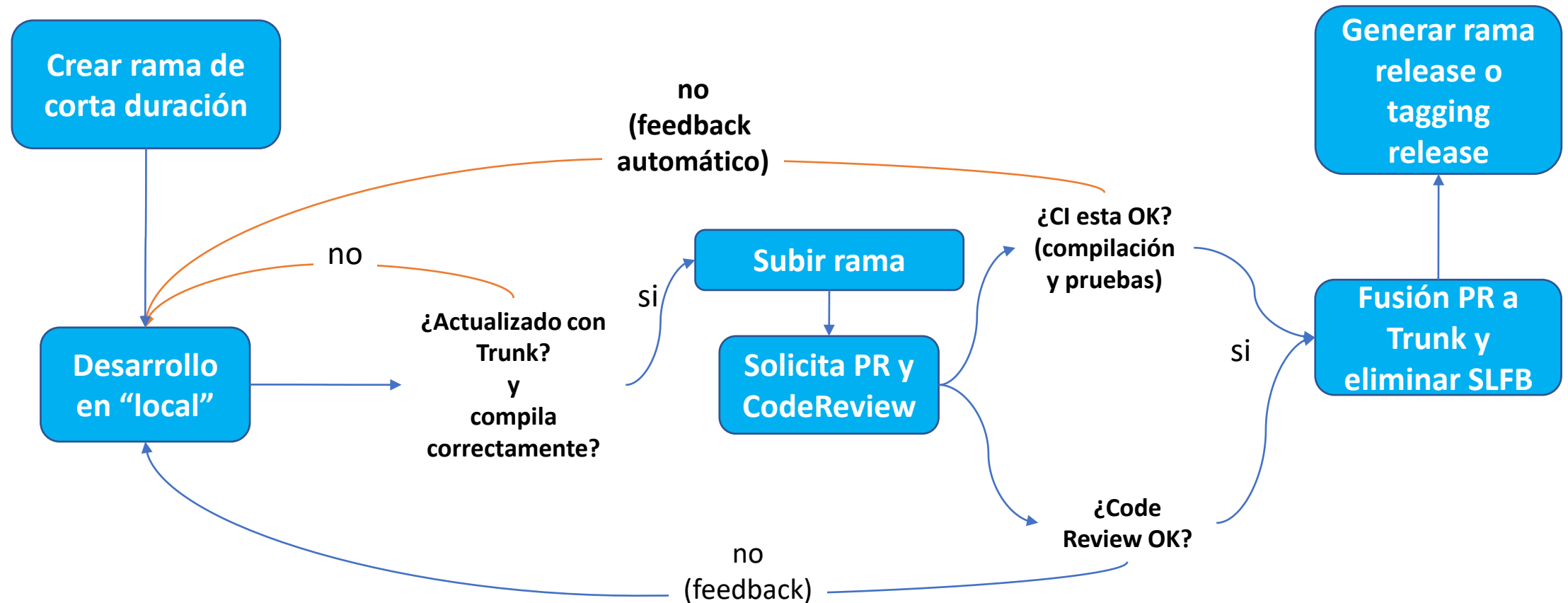
- 1 Introducción y Hallazgos
- 2 ¿Qué es trunk-based development?
- 3 ¿Como se implementa?
- 4 Demo



Flujo de Scaled Trunk-Based Development - Resumen

Scaled Trunk base

Para equipos grandes, se trabaja en short-live feature branch



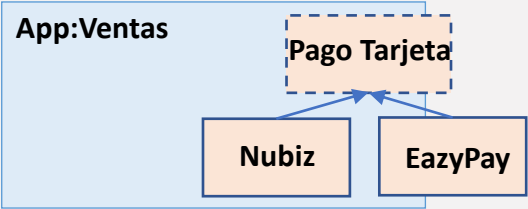
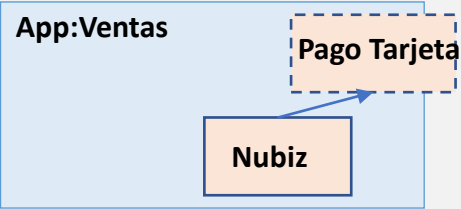
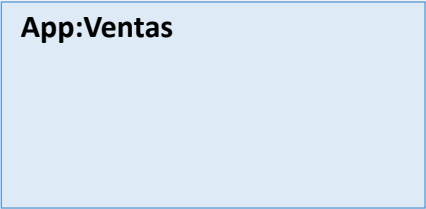
Prácticas de ayudan a Trunkbase development

Feature Flags

```
if args.contains("--withPagoTarjeta") {  
    feature = new EazyPayPagoTarjeta();  
}
```

```
bootContainer.addComponent(  
    classFromName(  
        config.get("PagoTarjetaClass")  
    )  
);
```

Branch by Abstraction

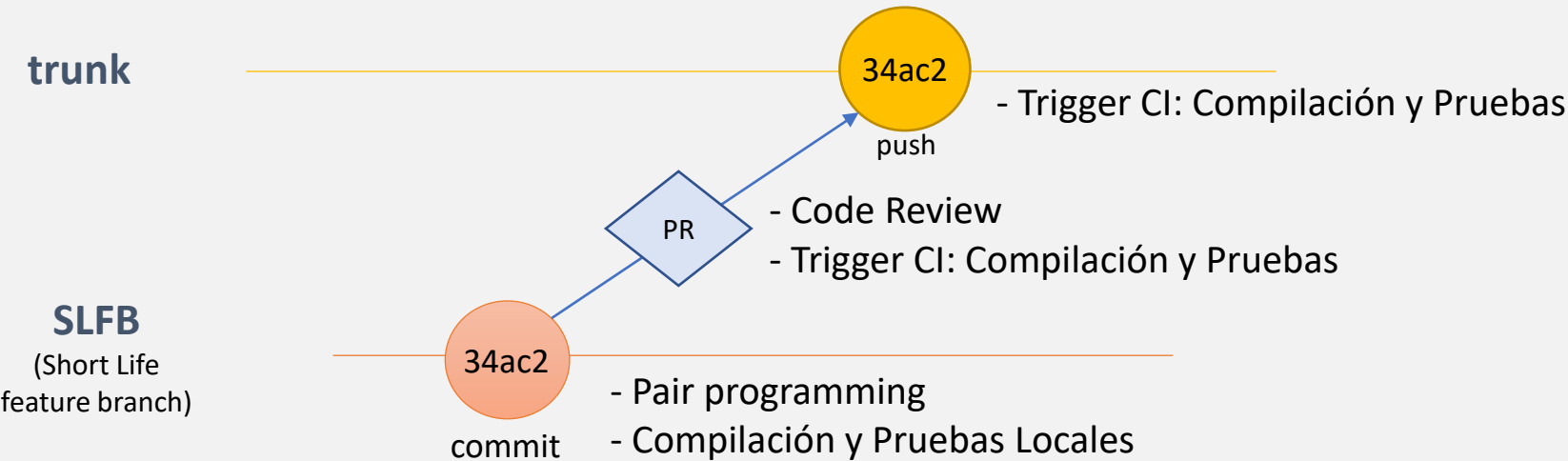


Continous Integration

Pull Request / Code Review

(Scaled Trunkbase)

Pair Programming



Demo Time!

Llevémoslo a cabo

- 1 Introducción y Hallazgos
- 2 ¿Qué es trunk-based development?
- 3 ¿Como se implementa?
- 4 Demo

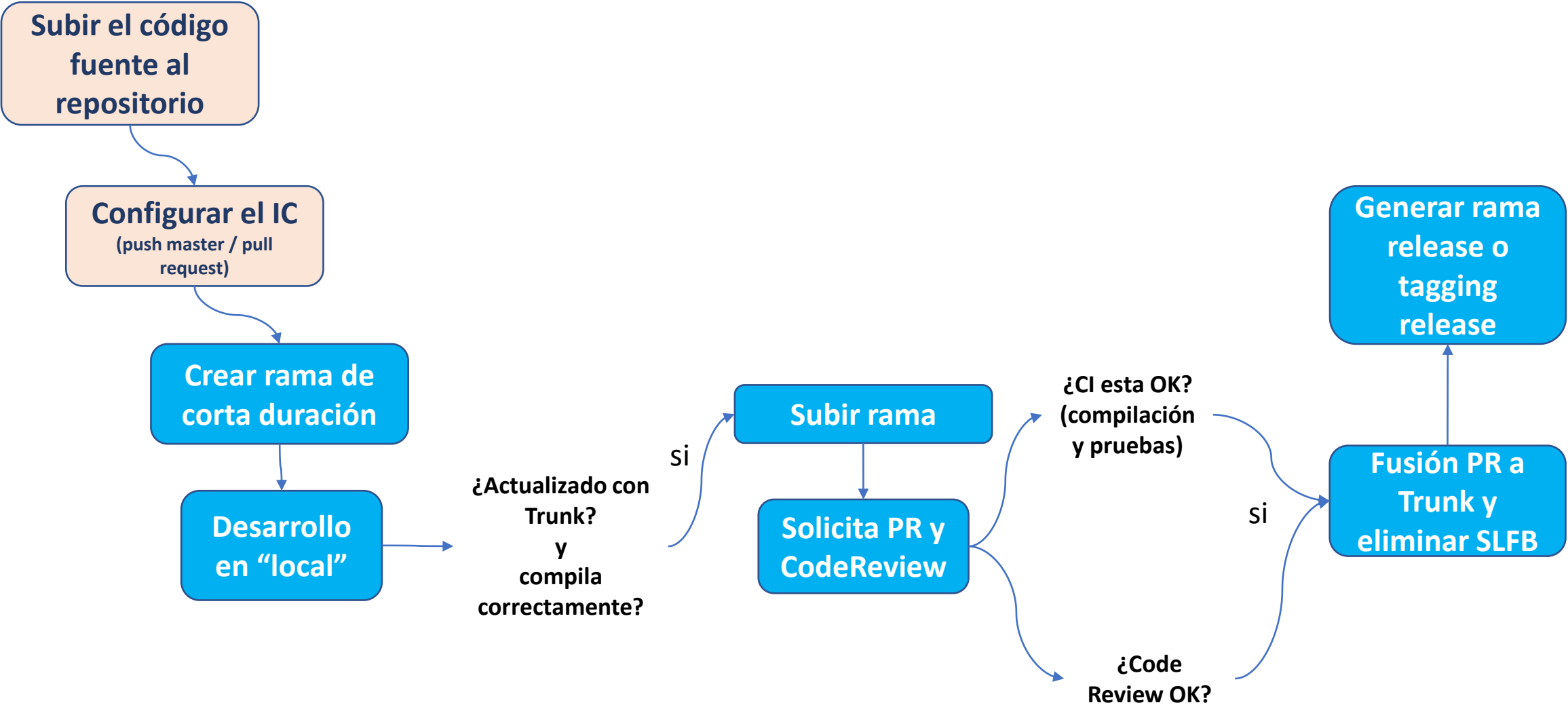


Demo Scaled Trunk-base - Happy Flow



<https://github.com/mmartinezepichan/demo-java-project-base.git>

- Pre-requisitos
- Pasos





Gracias !



Miguel Martínez Espichan
Arquitecto Experto en DevOps



mmartinezepichan@gmail.com



www.linkedin.com/in/mmartinezepichan