



Software Configuration Manager

Modelo Operativo Propuesto

Christian Camilo Gil

Arquitecto

christiangi@lagash.com



Agenda

1. Conceptos
2. Estrategias de Control de Versiones
3. Estrategia Claro
4. Modelo Operativo
5. Preguntas y Respuestas

Conceptos



**Check-
in**

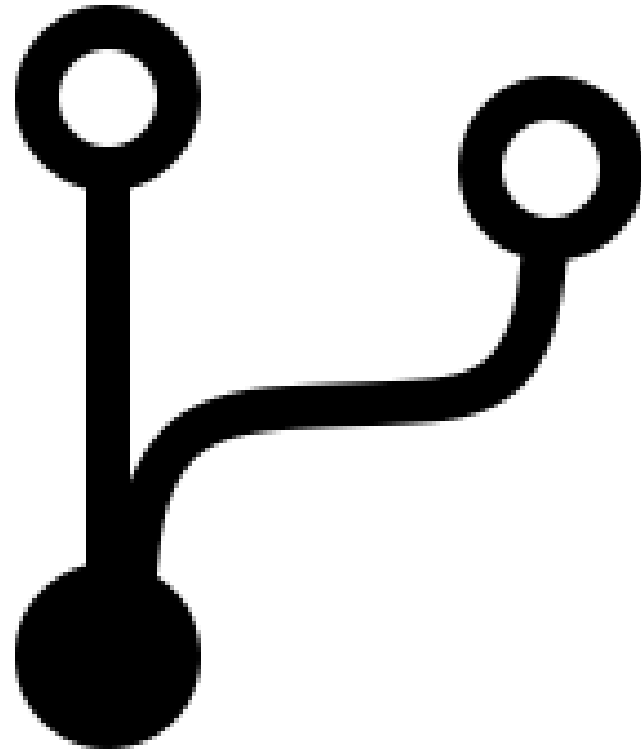
Qué es?

- Operación Central de la Gestión de Versiones de Software.
- Almacena una nueva versión de uno o más archivos dentro del repositorio.
- La suma de todos los check-ins constituye la historia de todo el contenido dentro del repositorio.
- Reflejan la integridad de un entregable.

Buenas Prácticas

- Un check-in es una operación atómica.
- Un check-in debería tener un único propósito.
- Mantener un alcance estrecho del check-in.
- La descripción debe comunicar la descripción del cambio.
- Cada check-in debe ser consistente.
- Cada check-in debe estar completo.
- Usar validaciones previas.
- Asegurarse que un check-in es trazable.
- Un check-in deseable es reversible.

Conceptos



Ramas

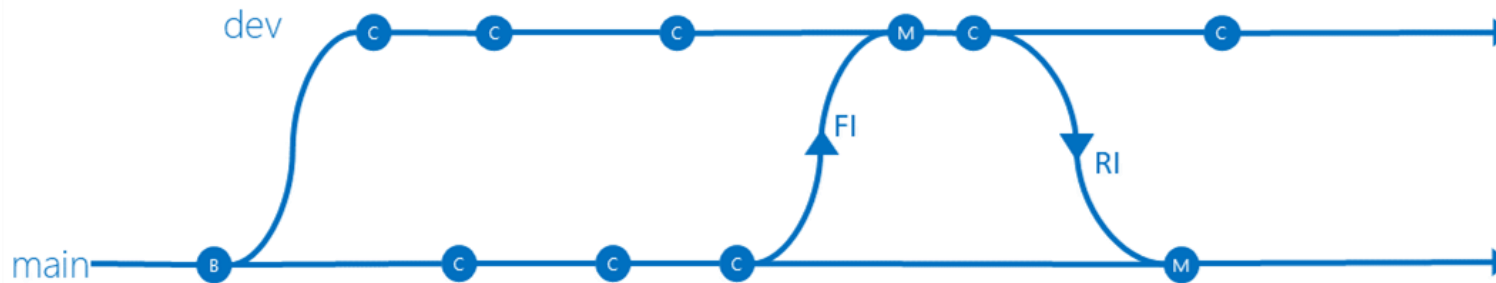
Qué es?

- Permite dirigir cada incidente o requerimiento dentro del desarrollo de software.
- Permite distinguir variadas versiones del software, las que están en construcción, pruebas y operativas.

Modelos de Ramificación

- Línea de Código
- Ramas por Release/Hito
- Ramas por Estado
- Aislamiento para Desarrollo
- Modelo de Línea Principal
- Flujos

Aislamiento para Desarrollo



* Tomado de: Microsoft, Learn about branching strategies for Team Foundation Version Control (TFVC) and how to select an effective strategy, en: <https://docs.microsoft.com/en-us/vsts/tfvc/branching-strategies-with-tfvc>

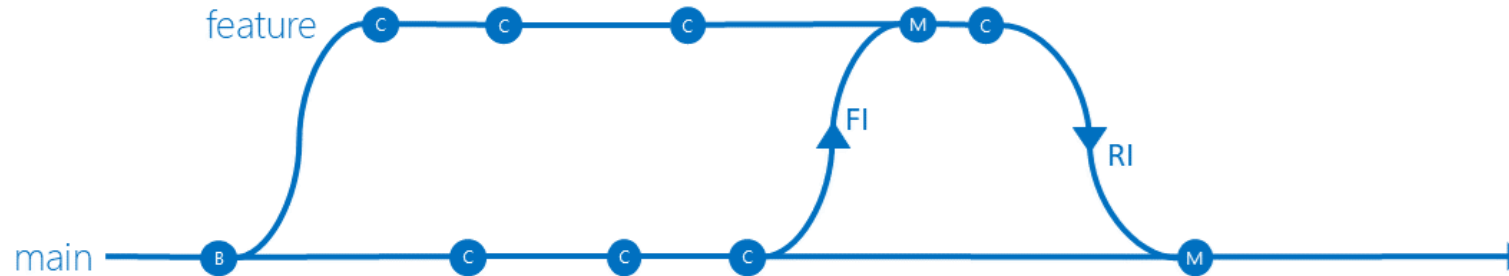
Características

- Cada uno de los miembros del equipo ven los cambios inmediatamente, mejorado la comunicación y evita las sorpresas de liberación de versión.
- Es propicio para proyectos complejos o de experimentación.
- Brinda una rama para que los desarrolladores puedan liberar cambios incrementales.

Buenas Prácticas

- Construir una rama sobre políticas incompatibles.
- Evitar desarrollos extensos en una rama separada.
- Establecer una nueva base de desarrollo regularmente.

Aislamiento por Característica



* Tomado de: Microsoft, Learn about branching strategies for Team Foundation Version Control (TFVC) and how to select an effective strategy, en: <https://docs.microsoft.com/en-us/vsts/tfvc/branching-strategies-with-tfvc>

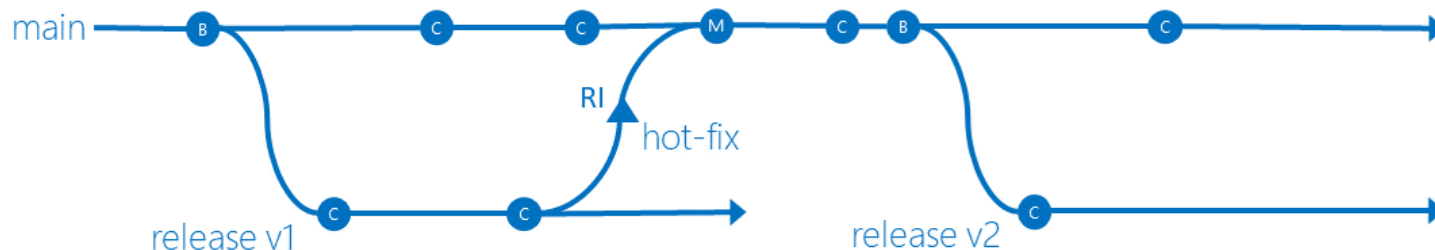
Características

- Se crea una nueva rama específicamente para adicionar una característica al producto.
- El Rollback sobre la línea principal puede ser costosa

Buenas Prácticas

- Construir una rama sobre nuevas características que pueda tener el producto.
- Reintegrar a la rama principal debe estar acompañada por una DoD.

Aislamiento por Release



* Tomado de: Microsoft, Learn about branching strategies for Team Foundation Version Control (TFVC) and how to select an effective strategy, en: <https://docs.microsoft.com/en-us/vsts/tfvc/branching-strategies-with-tfvc>

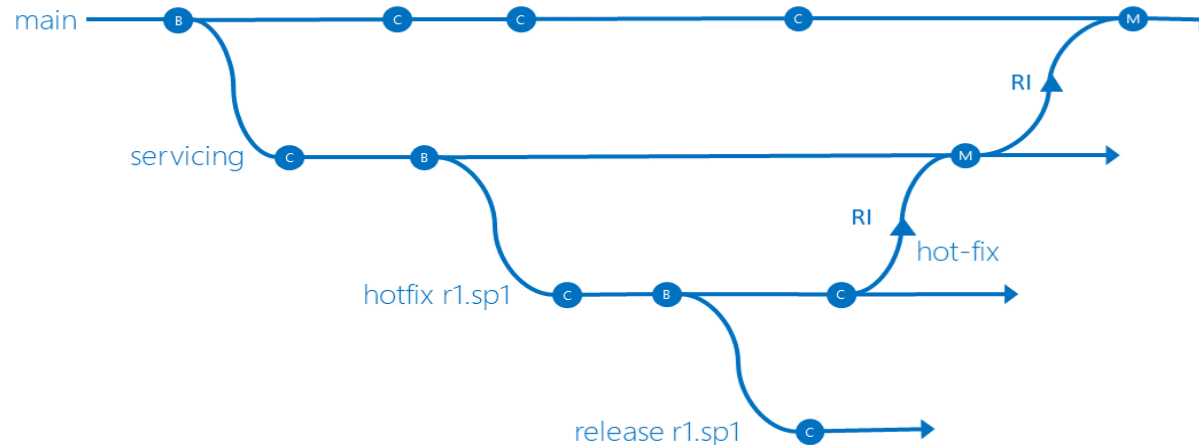
Características

- Consiste en congelar el código y crear nuevas ramas para evitar que las nuevas características choquen con los esfuerzos de estabilización.
- Buscan no detener los desarrollos futuros y evitar cambiar las políticas sobre la línea principal, ramificando una nueva línea por reléase desde la línea principal.
- Permite crear varias ramas al mismo tiempo
- Es útil para crear demos del producto o una versión de pre-lanzamiento como hitos.

Buenas Prácticas

- Crear un reléase por cada rama.
- Crear una rama en lugar de congelar el código.
- Definir una política separada para una nueva rama de reléase
- No reusar ramas de reléase para múltiples releases.

Aislamiento por Servicio, Release y Hotfix



* Tomado de: Microsoft, Learn about branching strategies for Team Foundation Version Control (TFVC) and how to select an effective strategy, en: <https://docs.microsoft.com/en-us/vsts/tfvc/branching-strategies-with-tfvc>

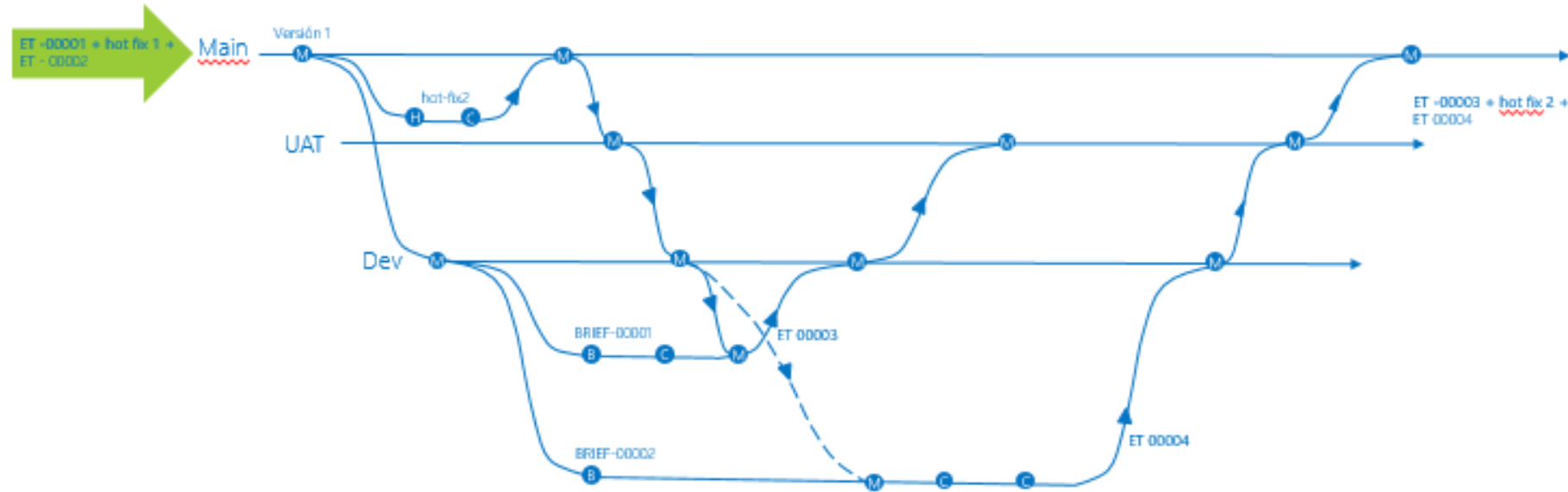
Características

- Es posible añadir ramas considerando los hotfix.
- Buscan no detener los desarrollos futuros y evitar cambiar las políticas sobre la línea principal, ramificando una nueva línea por release desde la línea principal.
- Permite crear varias ramas al mismo tiempo
- Es útil para crear demos del producto o una versión de pre-lanzamiento como hitos.

Buenas Prácticas

- Bloquear el release para evitar modificaciones, las modificaciones futuras se realizan sobre la del servicio..
- Crear nuevas ramas de servicio y release para releases subsiguientes si se requiere dicho nivel de aislamiento.

Estrategia para Claro



Características

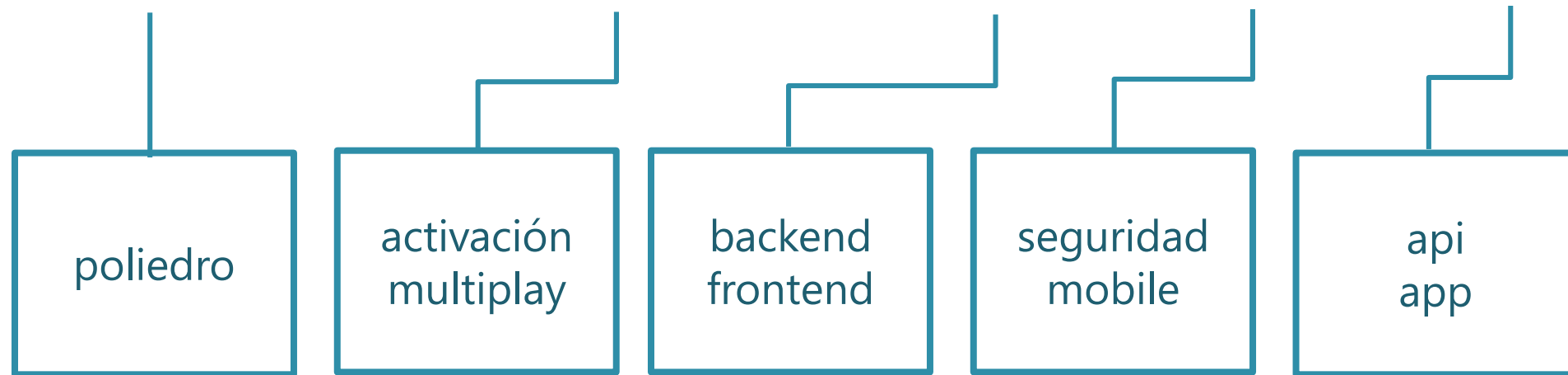
- Se combina el Aislamiento para Desarrollo, con Aislamiento por Release.
- Se crean sub-ramas de la rama Dev por Brief y cada commit se asocia con la Especificación Técnica.
- Las sub-ramas creadas son eliminadas en el proceso de merge

Buenas Prácticas

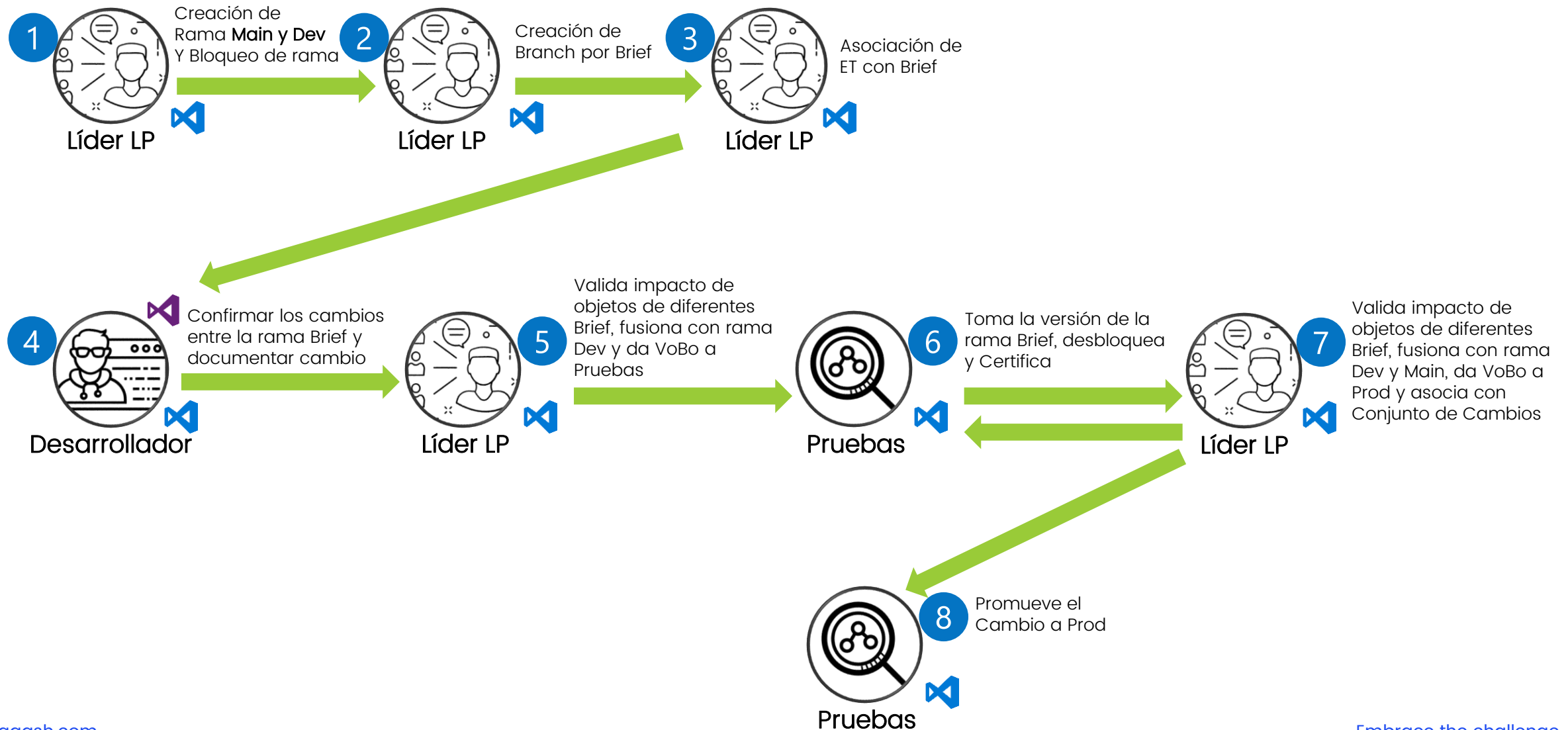
- La estrategia se aplica tanto para TFSVC como Git
- Se requiere la asociación de los Work Items con el control de versiones
- Es el inicio para realizar la integración continua (CI)

Nombramiento Repositorios

<Proyecto Equipo> <Producto/Equipo> - <Capa> - <Servicio> - <Tecnología>



Modelo Operativo de Control de Versiones



Preguntas y Respuestas

Por qué las ramas deben tener vida corta?

R/ Debido a que se evitan problemas de conciliación o merge

Por qué ramificar únicamente si es necesario?

R/ Hacia DevOps es necesario considerar la automatización del build, pruebas y desarrollo. El Cambio es continuo, frecuente y las operaciones de merge pueden requerir de intervención manual. Por lo tanto se recomienda evitar las ramas y confiar en otras estrategias como la función de alternar.

Puede un código base enlazado una rama entre Team Projects?

R/ Aunque no es recomendado a menos que se deban compartir código y no tengan procesos común.

Por qué eliminar las ramas?

R/ El objetivo es regresar los cambios a la rama principal tan pronto como es posible y mitigar los problemas de merge en el futuro. Las ramas no usadas y usadas temporalmente causan confusión y gastos generales para el equipo.

Son similares TFVC & Git?

R/ Las estrategias de aislamiento son similares



Let's keep in touch

Argentina | Buenos Aires
+54 (11) 4982 4185
info@lagash.com

Chile | Santiago de Chile
+56 (2) 2231 9428
info_chile@lagash.com

Colombia | Bogotá
+57 (1) 750 5276
info_colombia@lagash.com

México | México DF
+52 (55) 6394 0617
info_mexico@lagash.com

USA | Seattle - US
+1 844 4 Lagash
infousa@lagash.com

Uruguay | Montevideo
+598 2623 2546
info_uy@lagash.com