

# Gestión Avanzada de la Configuración de Sistemas Software

Pablo Sánchez

Dpto. Ingeniería Informática y Electrónica  
Universidad de Cantabria, Santander (Spain)  
[p.sanchez@unican.es](mailto:p.sanchez@unican.es)



# Advertencia

Todo el material contenido en este documento no constituye en modo alguno una obra de referencia o apuntes oficiales mediante los cuales se puedan preparar de manera autónoma las pruebas necesarias para superar la asignatura.

Este documento contiene exclusivamente una serie de diapositivas cuyo objetivo es servir de complemento visual a las actividades realizadas en el aula.

Dicho de forma más clara, **estas transparencias no son apuntes y su objetivo no es en modo alguno servir para que el alumno pueda preparar la asignatura.**

# Objetivos del Tema

- 1 Recordar y comprender la importancia y objetivos de la gestión de la configuración de sistemas sw.
- 2 Comprender los conceptos de *integración y entrega continua*.
- 3 Comprender el concepto de *DevOps*.
- 4 Ser capaz de crear y fusionar ramas utilizando **Git**.
- 5 Ser capaz de entender el concepto de flujo de trabajo en gestión de la configuración.

# Objetivos del Tema

- 1 Recordar y comprender la importancia y objetivos de la gestión de la configuración de sistemas sw.
- 2 Comprender los conceptos de *integración y entrega continua*.
- 3 Comprender el concepto de *DevOps*.
- 4 Ser capaz de crear y fusionar ramas utilizando **Git**.
- 5 Ser capaz de entender el concepto de flujo de trabajo en gestión de la configuración.

# Objetivos del Tema

- 1 Recordar y comprender la importancia y objetivos de la gestión de la configuración de sistemas sw.
- 2 Comprender los conceptos de *integración y entrega continua*.
- 3 Comprender el concepto de *DevOps*.
- 4 Ser capaz de crear y fusionar ramas utilizando **Git**.
- 5 Ser capaz de entender el concepto de flujo de trabajo en gestión de la configuración.

# Objetivos del Tema

- 1 Recordar y comprender la importancia y objetivos de la gestión de la configuración de sistemas sw.
- 2 Comprender los conceptos de *integración y entrega continua*.
- 3 Comprender el concepto de *DevOps*.
- 4 Ser capaz de crear y fusionar ramas utilizando **Git**.
- 5 Ser capaz de entender el concepto de flujo de trabajo en gestión de la configuración.

# Objetivos del Tema

- 1 Recordar y comprender la importancia y objetivos de la gestión de la configuración de sistemas sw.
- 2 Comprender los conceptos de *integración y entrega continua*.
- 3 Comprender el concepto de *DevOps*.
- 4 Ser capaz de crear y fusionar ramas utilizando **Git**.
- 5 Ser capaz de entender el concepto de flujo de trabajo en gestión de la configuración.

# Bibliografía



Ian Sommerville.

*Software Engineering.*

Addison Wesley, 9 edition, April 2010.



Scott Chacon.

*Pro Git.*

Apress, 2 edition 2014.



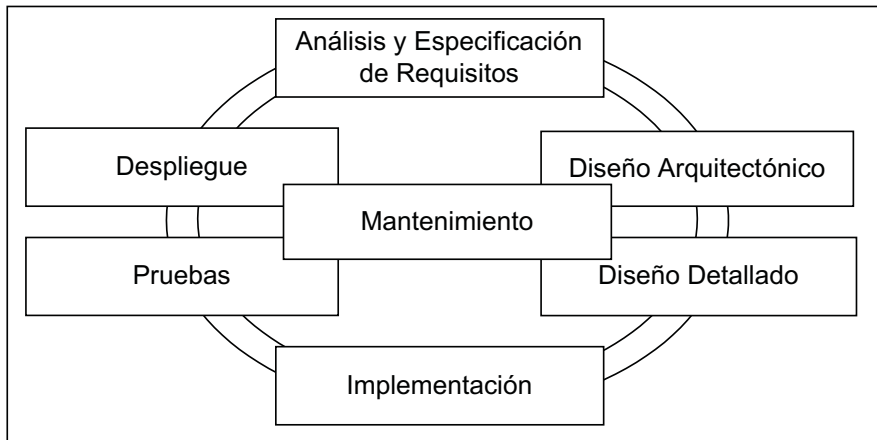
# Índice

- 1 **Introducción**
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
- 4 Control de Versiones con Git
- 5 Sumario

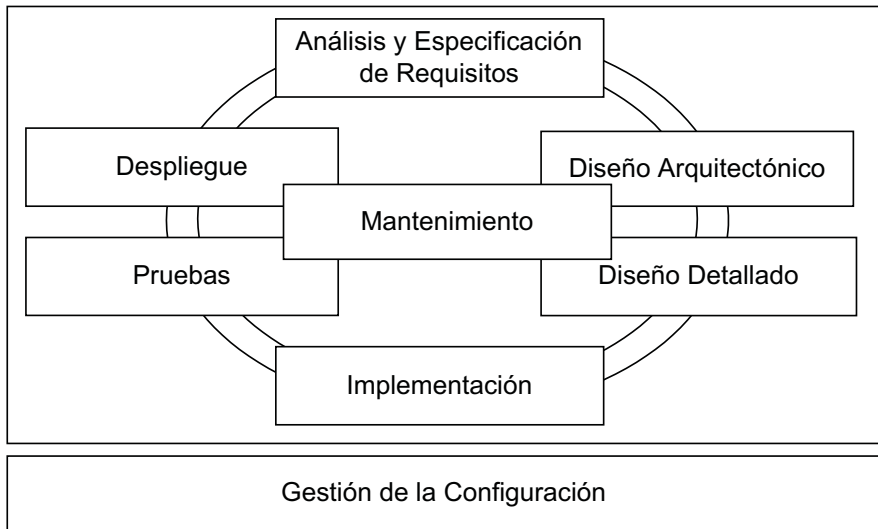
# Índice

- 1 Introducción
  - Ciclo de Vida Software
  - Motivación y Objetivos
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
- 4 Control de Versiones con Git
- 5 Sumario

# Gestión de la Configuración de Sistemas Software



# Gestión de la Configuración de Sistemas Software



# Índice

- 1 Introducción
  - Ciclo de Vida Software
  - **Motivación y Objetivos**
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
- 4 Control de Versiones con Git
- 5 Sumario

# ¿Por Qué Necesitamos la Gestión de la Configuración?

- 1 ¿Qué he cambiado? ¿Cómo hago si no hubiese pasado nada?
- 2 Problema de la copia correcta.
- 3 Desarrollo distribuido de software.
- 4 Problema de que *Google* encuentra los archivos mejor que yo.

# ¿Por Qué Necesitamos la Gestión de la Configuración?

- 1 ¿Qué he cambiado? ¿Cómo hago si no hubiese pasado nada?
- 2 Problema de la copia correcta.
- 3 Desarrollo distribuido de software.
- 4 Problema de que *Google* encuentra los archivos mejor que yo.

# ¿Por Qué Necesitamos la Gestión de la Configuración?

- 1 ¿Qué he cambiado? ¿Cómo hago si no hubiese pasado nada?
- 2 Problema de la copia correcta.
- 3 Desarrollo distribuido de software.
- 4 Problema de que *Google* encuentra los archivos mejor que yo.



# ¿Por Qué Necesitamos la Gestión de la Configuración?

- 1 ¿Qué he cambiado? ¿Cómo hago si no hubiese pasado nada?
- 2 Problema de la copia correcta.
- 3 Desarrollo distribuido de software.
- 4 Problema de que *Google* encuentra los archivos mejor que yo.

# Situación a evitar

Nombre	Tamaño	Tipo	Fecha de modificación
Images		Carpeta	22/02/2010 12:34
imagesSource		Carpeta	22/02/2010 12:34
presentations		Carpeta	22/02/2010 12:34
~\4080908-AMPLE-WP2-D2-3-2.1(Carlos).doc	1 KB	Documento de Micr...	10/09/2008 11:28
~\WRL2913.tmp	1.909 KB	Archivo TMP	10/09/2008 9:54
20080328-AMPLE-WP2-T2-4-M2.3(finalChrista).doc	1.522 KB	Documento de Micr...	30/04/2008 18:52
20080328-AMPLE-WP2-T2-4-M2.3_final_.pdf	510 KB	Foxit PDF Document	24/09/2008 9:00
20080430-AMPLE-WP2-T2-4(Christa).doc	1.559 KB	Documento de Micr...	05/05/2008 17:38
20080430-AMPLE-WP2-T2-4.doc	1.542 KB	Documento de Micr...	05/05/2008 10:16
20080430-AMPLE-WP2-T2-4.pdf	730 KB	Foxit PDF Document	30/04/2008 18:53
20080430-AMPLE-WP2-T2-4(Aless_comments).doc	1.582 KB	Documento de Micr...	24/07/2008 13:27
20080530-AMPLE-WP2-T2-4.doc	1.579 KB	Documento de Micr...	30/05/2008 13:41
20080708-AMPLE-WP2-D2-3-1-1.doc	110 KB	Documento de Micr...	09/07/2008 14:05
20080712-AMPLE-WP2-D2-3-(Ludger).doc	1.632 KB	Documento de Micr...	05/09/2008 10:32
20080722-AMPLE-WP2-D2-3-1.4.doc	964 KB	Documento de Micr...	23/07/2008 14:00
20080722-AMPLE-WP2-D2-3-1.5.doc	980 KB	Documento de Micr...	06/08/2008 13:11
20080831-AMPLE-WP2-D2-3-2.0.doc	1.883 KB	Documento de Micr...	03/09/2008 12:04
20080903-AMPLE-WP2-D2-3-2.1.doc			
20080908-AMPLE-WP2-D2-3-2.1(Carlos).doc			
20080910-AMPLE-WP2-D2-3-3.0.doc			
20080910-AMPLE-WP2-D2-3-3.0-LF.doc	1.702 KB	Documento de Micr...	11/09/2008 9:44
20080911-AMPLE-WP2-D2-3(firstDraft).doc	1.685 KB	Documento de Micr...	11/09/2008 10:09
20080911-AMPLE-WP2-D2-3(firstDraft).pdf	1.380 KB	Foxit PDF Document	11/09/2008 10:10
20080911-AMPLE-WP2-D2-3(firstDraft)_aless.doc	1.709 KB	Documento de Micr...	17/09/2008 9:59
20080911-AMPLE-WP2-D2-3(firstDraft)_aless+LF.doc	1.712 KB	Documento de Micr...	18/09/2008 10:24
20080915-AMPLE-WP2-D2-3.doc	1.792 KB	Documento de Micr...	18/09/2008 10:41
20080918-AMPLE-WP2-D2-3.doc	1.766 KB	Documento de Micr...	18/09/2008 10:44
20080918-AMPLE-WP2-D2-3.pdf	1.388 KB	Foxit PDF Document	18/09/2008 11:13

Tipo: Documento de Microsoft Office Word 97-2003  
 Autor: PC  
 Fecha de modificación: 06/08/2008 13:11  
 Tamaño: 980 KB

# Índice

- 1 Introducción
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
- 4 Control de Versiones con Git
- 5 Sumario

# Definiciones

## Configuration Item

Artefacto o conjunto de artefactos susceptible de poseer varias versiones.

## Version

Instancia de un ítem de configuración que difiere de alguna manera de otras instancias del mismo artefacto.

## Release

Versión de un ítem de configuración (puede ser un sistema entero) que se distribuye a los clientes.

# Definiciones

## Configuration Item

Artefacto o conjunto de artefactos susceptible de poseer varias versiones.

## Version

Instancia de un ítem de configuración que difiere de alguna manera de otras instancias del mismo artefacto.

## Release

Versión de un ítem de configuración (puede ser un sistema entero) que se distribuye a los clientes.

# Definiciones

## Configuration Item

Artefacto o conjunto de artefactos susceptible de poseer varias versiones.

## Version

Instancia de un ítem de configuración que difiere de alguna manera de otras instancias del mismo artefacto.

## Release

Versión de un ítem de configuración (puede ser un sistema entero) que se distribuye a los clientes.

# Definiciones

## Baseline

Conjunto de versiones concretas de los diferentes items de configuración, que constituyen un estado significativo, claramente identificado, controlado y aprobado en la evolución de un producto.

# Índice

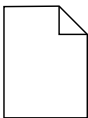
- 1 Introducción
- 2 Terminología Básica
- 3 **Procesos de Gestión de la Configuración Sw**
- 4 Control de Versiones con Git
- 5 Sumario



# Índice

- 1 Introducción
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
  - Gestión de la Configuración Sw
  - Integración y Entrega Continua
  - DevOps
- 4 Control de Versiones con Git
- 5 Sumario

# Áreas de la Gestión de la Configuración Sw



Petición de  
Cambio

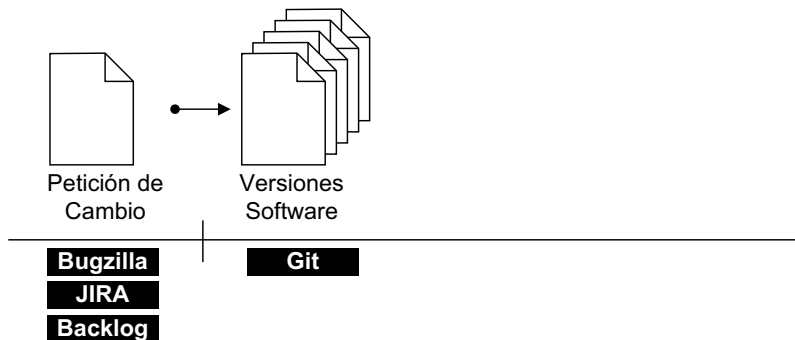
---

**Bugzilla**

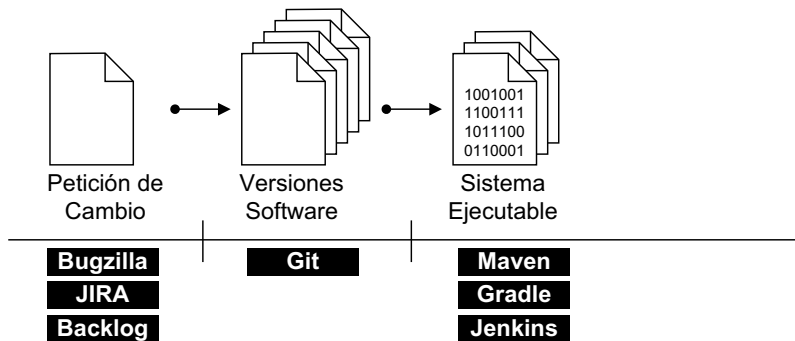
**JIRA**

**Backlog**

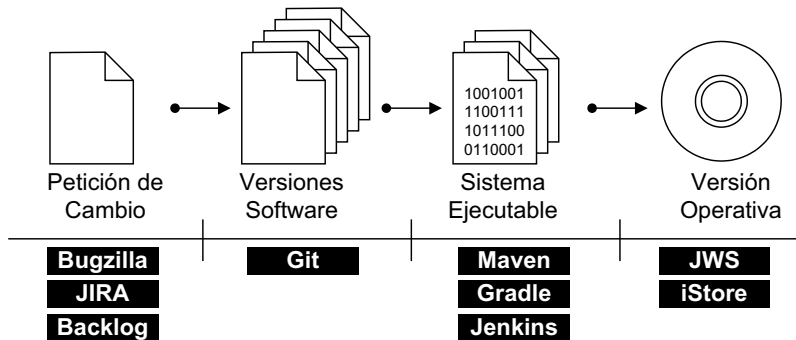
# Áreas de la Gestión de la Configuración Sw



# Áreas de la Gestión de la Configuración Sw



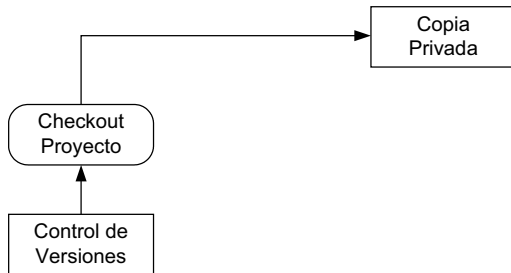
# Áreas de la Gestión de la Configuración Sw



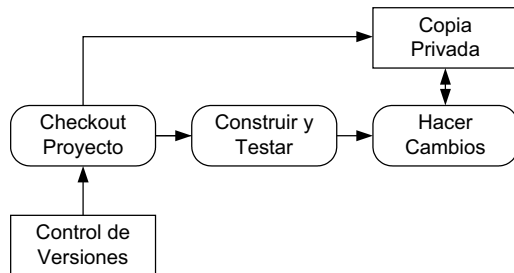
# Índice

- 1 Introducción
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
  - Gestión de la Configuración Sw
  - Integración y Entrega Continua
  - DevOps
- 4 Control de Versiones con Git
- 5 Sumario

# Integración y Entrega Continua

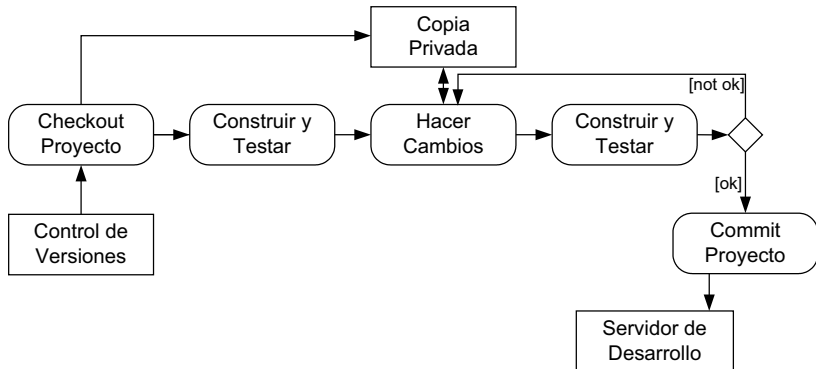


# Integración y Entrega Continua

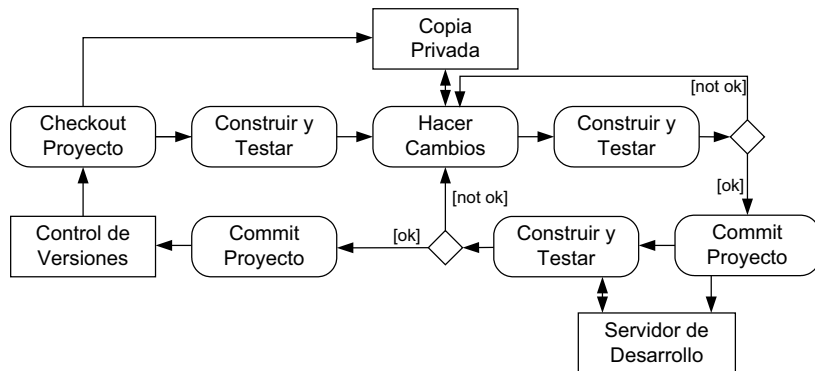




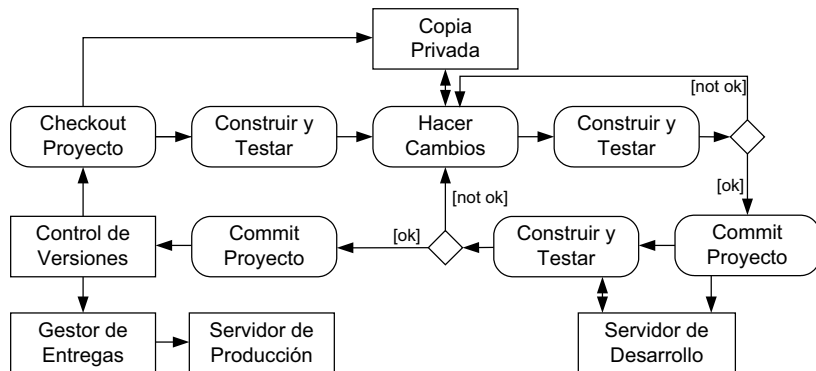
# Integración y Entrega Continua



# Integración y Entrega Continua



# Integración y Entrega Continua



# Índice

- 1 Introducción
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
  - Gestión de la Configuración Sw
  - Integración y Entrega Continua
  - DevOps
- 4 Control de Versiones con Git
- 5 Sumario

# DevOps

- 1 Unión de los equipos de desarrollo (*Development*) y operaciones (*Operations*) con el objetivo de mejorar la calidad del software.
- 2 Filosofía *you build it, you run it*.
- 3 Técnicas frecuentemente utilizadas:
  - Automatización de los procesos de desarrollo y despliegue.
  - Automatización de la infraestructura (plataforma de nube).
  - Automatización de operaciones (análisis de logs, monitoreo, alertas, etc.).

# DevOps

- 1 Unión de los equipos de desarrollo (*Development*) y operaciones (*Operations*) con el objetivo de mejorar la calidad del software.
- 2 Filosofía *you build it, you run it*.
- 3 Técnicas frecuentemente utilizadas:
  - » Desarrollo basado en microservicios
  - » Automatización de la infraestructura (IaC) con el fin de automatizar el despliegue de aplicaciones
  - » Automatización de la configuración de los entornos de desarrollo y producción

# DevOps

- ❶ Unión de los equipos de desarrollo (*Development*) y operaciones (*Operations*) con el objetivo de mejorar la calidad del software.
- ❷ Filosofía *you build it, you run it*.
- ❸ Técnicas frecuentemente utilizadas:
  - ▶ Desarrollo basado en microservicios.
  - ▶ Integración y entrega continua.
  - ▶ Automatización de la infraestructura (*platform as code*)
  - ▶ Monitorización de operaciones y análisis de datos: *observed requirements, AB tests, canary test*.

# DevOps

- ❶ Unión de los equipos de desarrollo (*Development*) y operaciones (*Operations*) con el objetivo de mejorar la calidad del software.
- ❷ Filosofía *you build it, you run it*.
- ❸ Técnicas frecuentemente utilizadas:
  - ▶ Desarrollo basado en microservicios.
  - ▶ Integración y entrega continua.
  - ▶ Automatización de la infraestructura (*platform as code*)
  - ▶ Monitorización de operaciones y análisis de datos: *observed requirements, AB tests, canary test*.



# DevOps

- ❶ Unión de los equipos de desarrollo (*Development*) y operaciones (*Operations*) con el objetivo de mejorar la calidad del software.
- ❷ Filosofía *you build it, you run it*.
- ❸ Técnicas frecuentemente utilizadas:
  - ▶ Desarrollo basado en microservicios.
  - ▶ Integración y entrega continua.
  - ▶ Automatización de la infraestructura (*platform as code*)
  - ▶ Monitorización de operaciones y análisis de datos: *observed requirements, AB tests, canary test*.

# DevOps

- ❶ Unión de los equipos de desarrollo (*Development*) y operaciones (*Operations*) con el objetivo de mejorar la calidad del software.
- ❷ Filosofía *you build it, you run it*.
- ❸ Técnicas frecuentemente utilizadas:
  - ▶ Desarrollo basado en microservicios.
  - ▶ Integración y entrega continua.
  - ▶ Automatización de la infraestructura (*platform as code*)
  - ▶ Monitorización de operaciones y análisis de datos: *observed requirements, AB tests, canary test*.

# DevOps

- ❶ Unión de los equipos de desarrollo (*Development*) y operaciones (*Operations*) con el objetivo de mejorar la calidad del software.
- ❷ Filosofía *you build it, you run it*.
- ❸ Técnicas frecuentemente utilizadas:
  - ▶ Desarrollo basado en microservicios.
  - ▶ Integración y entrega continua.
  - ▶ Automatización de la infraestructura (*platform as code*)
  - ▶ Monitorización de operaciones y análisis de datos: *observed requirements, AB tests, canary test*.

# Índice

- 1 Introducción
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
- 4 **Control de Versiones con Git**
- 5 Sumario

# Índice

- 1 Introducción
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
- 4 Control de Versiones con Git
  - **Características de Git**
  - Acciones Básicas
  - Esquema de Versionado
  - Ramificación y Fusión
  - Stashing
  - Repositorios Remotos
  - Deshacer Cambios en Git
  - Modelos Organizativos con Git
- 5 Sumario

# Git

- 1 Distribuido y descentralizado.
- 2 Instantáneas en lugar de incrementos.
- 3 Operaciones preferentemente locales.
- 4 Verifica la integridad de los archivos.

# Git

- 1 Distribuido y descentralizado.
- 2 Instantáneas en lugar de incrementos.
- 3 Operaciones preferentemente locales.
- 4 Verifica la integridad de los archivos.

# Git

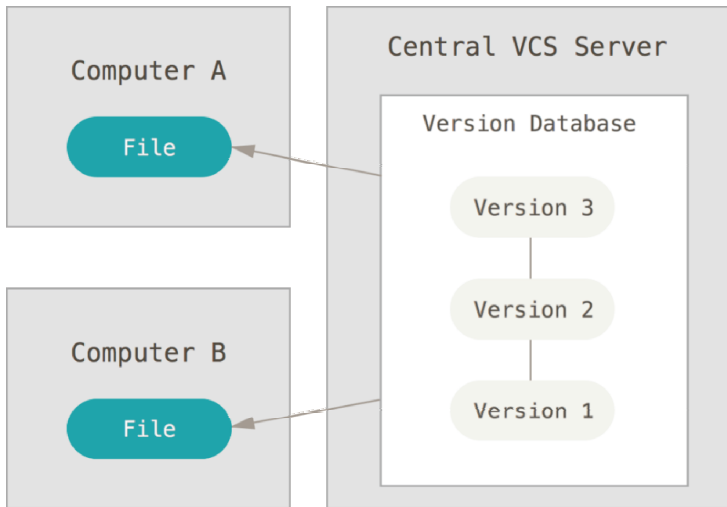
- 1 Distribuido y descentralizado.
- 2 Instantáneas en lugar de incrementos.
- 3 Operaciones preferentemente locales.
- 4 Verifica la integridad de los archivos.



# Git

- 1 Distribuido y descentralizado.
- 2 Instantáneas en lugar de incrementos.
- 3 Operaciones preferentemente locales.
- 4 Verifica la integridad de los archivos.

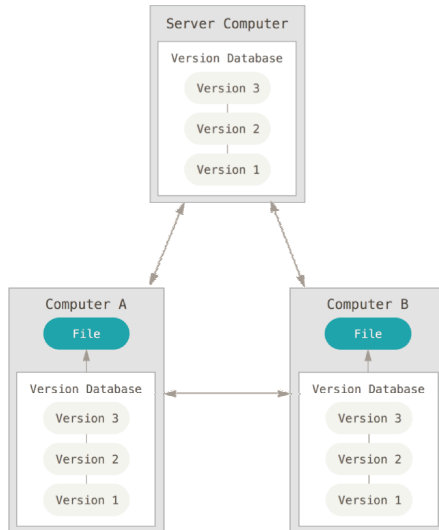
# Sistema Centralizado



# Problema Sistemas Centralizados



# Sistema Distribuido y Descentralizado



# Índice

- 1 Introducción
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
- 4 Control de Versiones con Git
  - Características de Git
  - **Acciones Básicas**
  - Esquema de Versionado
  - Ramificación y Fusión
  - Stashing
  - Repositorios Remotos
  - Deshacer Cambios en Git
  - Modelos Organizativos con Git
- 5 Sumario

# Comienzo de un Versionado con Git

**Init** Crea un repositorio en un directorio o proyecto existente.

Clone Clona un repositorio (remoto) en un directorio concreto.

# Comienzo de un Versionado con Git

**Init** Crea un repositorio en un directorio o proyecto existente.

**Clone** Clona un repositorio (remoto) en un directorio concreto.

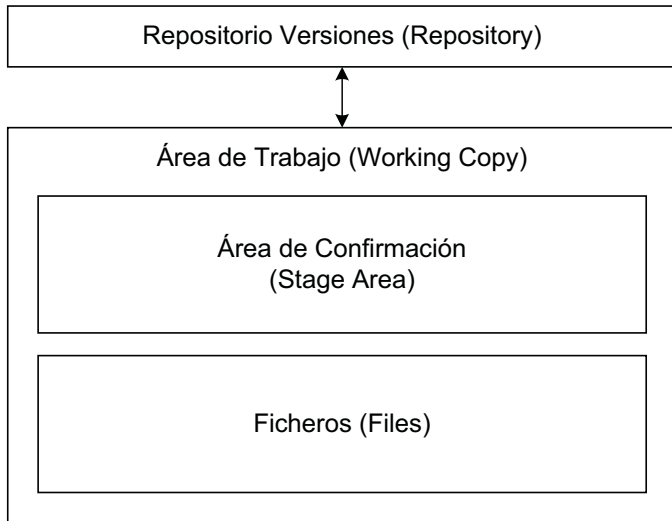
# Comienzo de un Versionado con Git

**Init** Crea un repositorio en un directorio o proyecto existente.

**Clone** **Clona** un repositorio (remoto) en un directorio concreto.



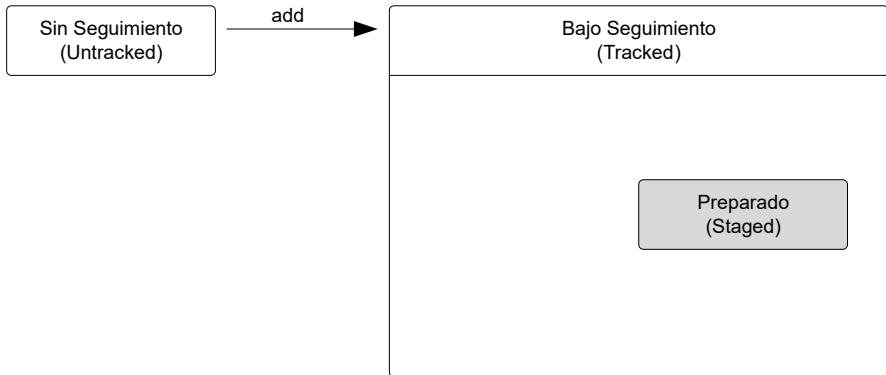
# Estructura de Proyectos en Git



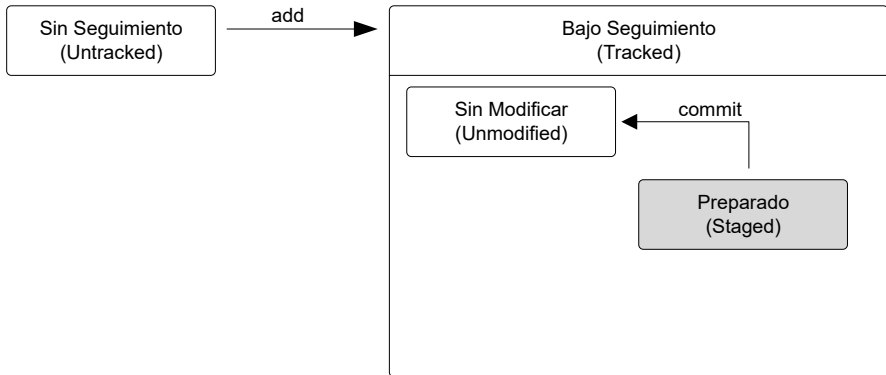
# Ciclo de Vida de Ficheros en Git

Sin Seguimiento  
(Untracked)

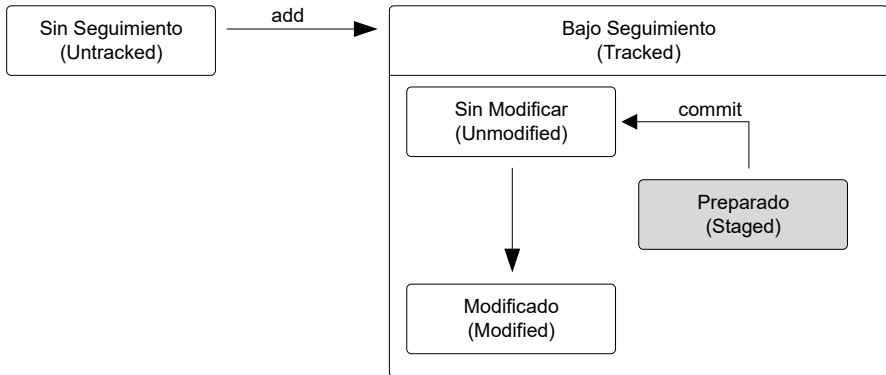
# Ciclo de Vida de Ficheros en Git



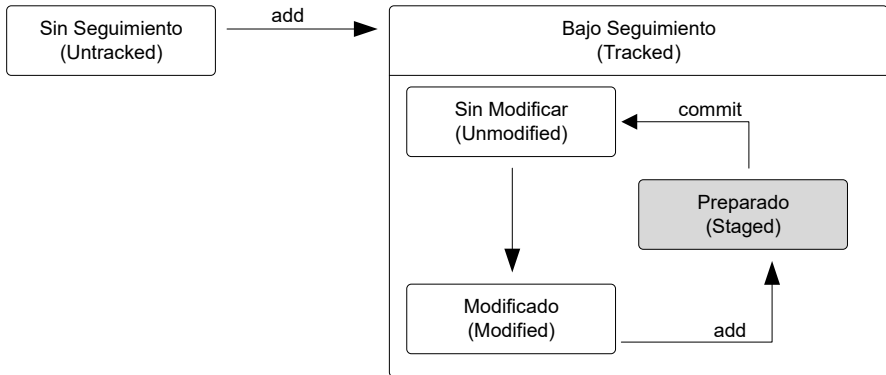
# Ciclo de Vida de Ficheros en Git



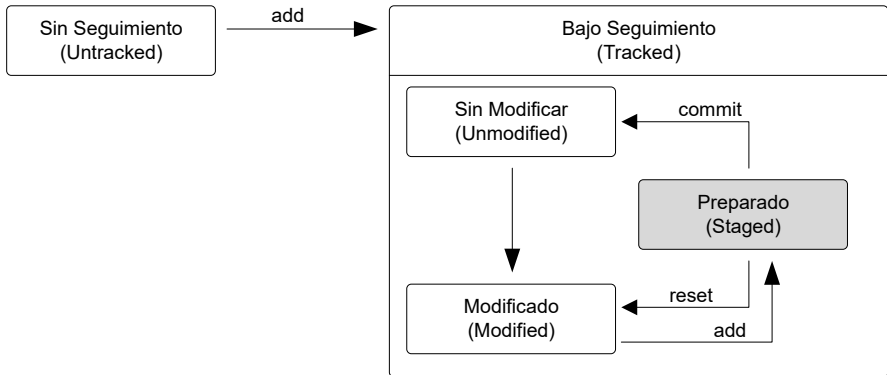
# Ciclo de Vida de Ficheros en Git



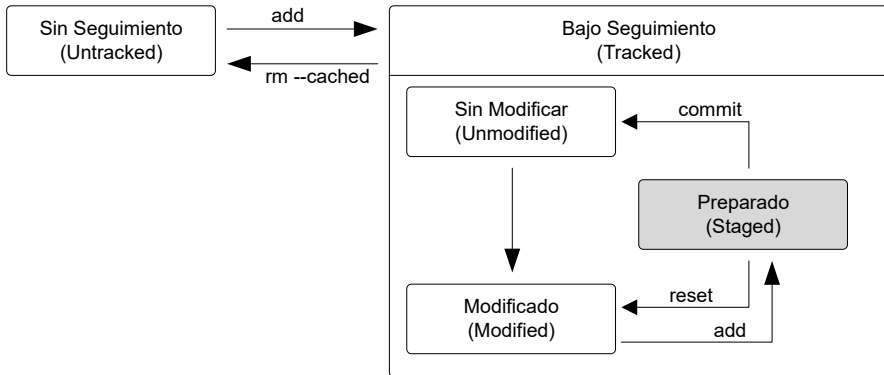
# Ciclo de Vida de Ficheros en Git



# Ciclo de Vida de Ficheros en Git

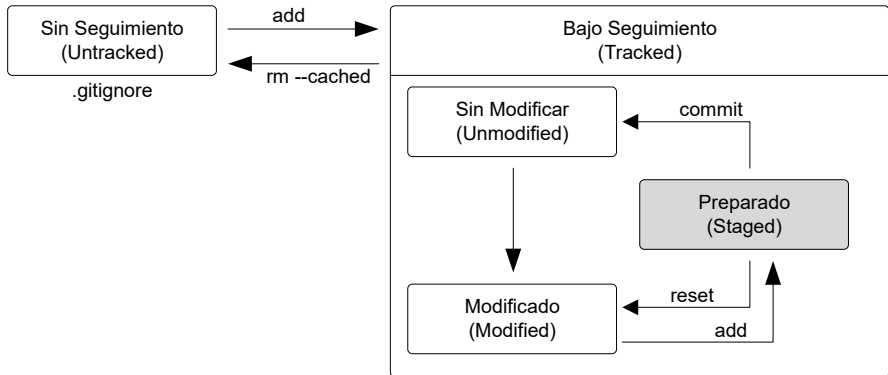


# Ciclo de Vida de Ficheros en Git





# Ciclo de Vida de Ficheros en Git



# Comandos Básicos

**status** Muestra el estado del directorio de trabajo.

**.gitignore** Fichero que contiene una lista de archivos que deben ser ignorados por el control de versiones.

**log** Muestra el historial de *commits* realizados.

# Comandos Básicos

`status` Muestra el estado del directorio de trabajo.

`.gitignore` Fichero que contiene una lista de archivos que deben ser ignorados por el control de versiones.

`log` Muestra el historial de *commits* realizados.

# Comandos Básicos

`status` Muestra el estado del directorio de trabajo.

`.gitignore` Fichero que contiene una lista de archivos que deben ser ignorados por el control de versiones.

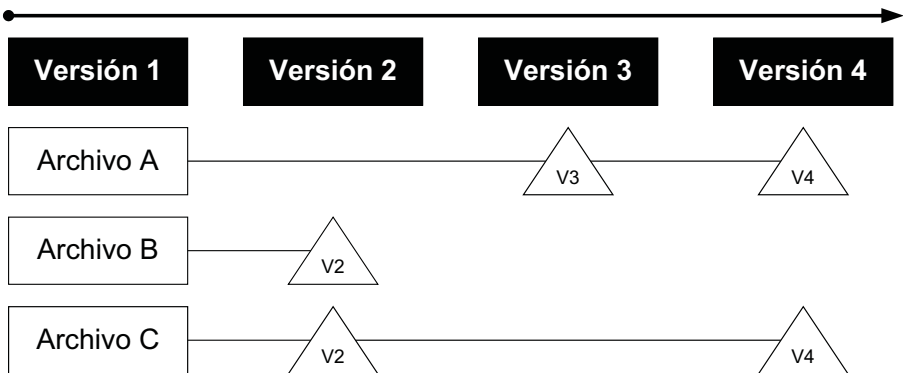
`log` Muestra el historial de *commits* realizados.

# Índice

- 1 Introducción
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
- 4 Control de Versiones con Git
  - Características de Git
  - Acciones Básicas
  - **Esquema de Versionado**
  - Ramificación y Fusión
  - Stashing
  - Repositorios Remotos
  - Deshacer Cambios en Git
  - Modelos Organizativos con Git
- 5 Sumario

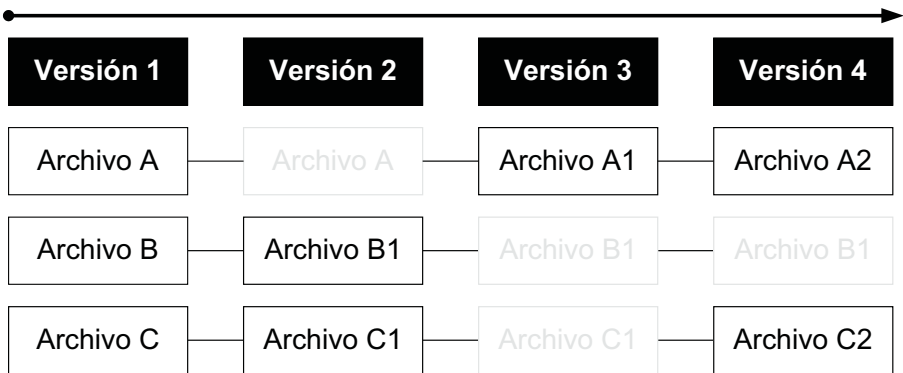
# Versionado basado en Deltas

Versionado sobre el tiempo

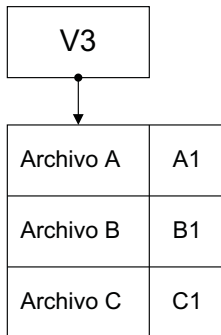


# Versionado basado en Instantáneas

Versionado sobre el tiempo

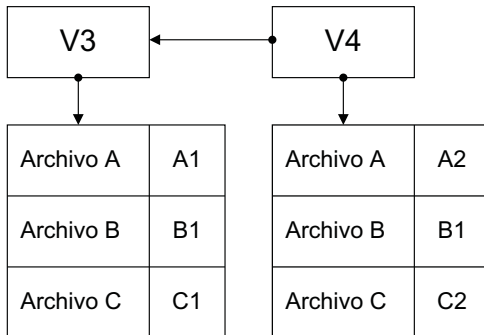


# Estructura de Versionado en Git





# Estructura de Versionado en Git



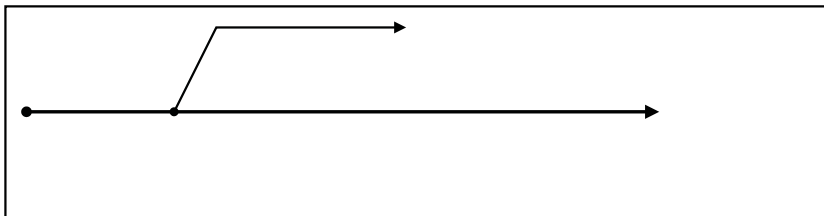
# Índice

- 1 Introducción
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
- 4 Control de Versiones con Git
  - Características de Git
  - Acciones Básicas
  - Esquema de Versionado
  - **Ramificación y Fusión**
  - Stashing
  - Repositorios Remotos
  - Deshacer Cambios en Git
  - Modelos Organizativos con Git
- 5 Sumario

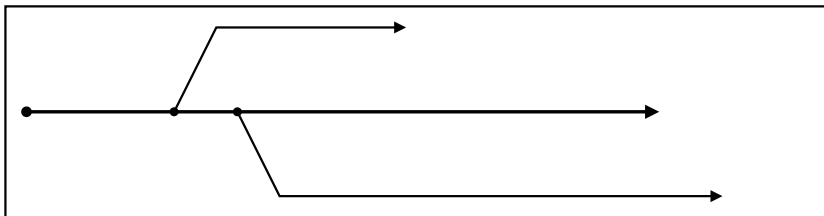
# Ramificación y Fusión



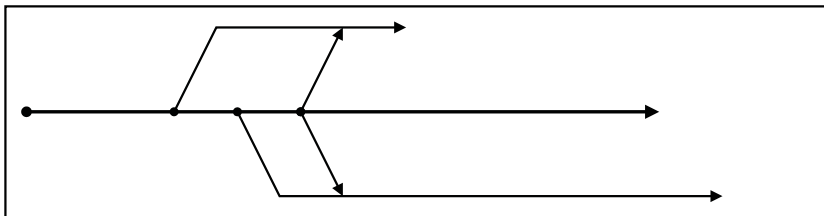
# Ramificación y Fusión



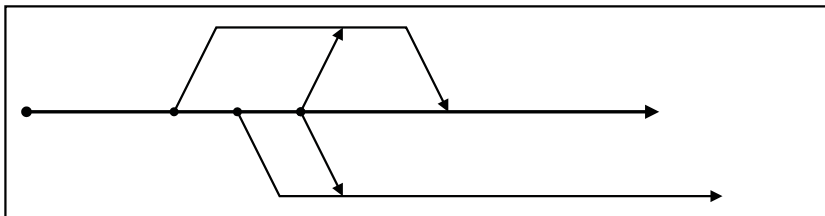
# Ramificación y Fusión



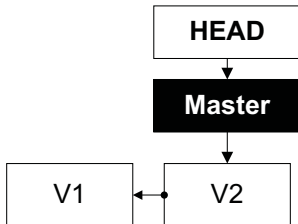
# Ramificación y Fusión



# Ramificación y Fusión

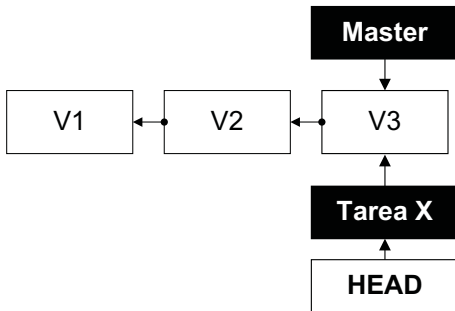


# Ramificación

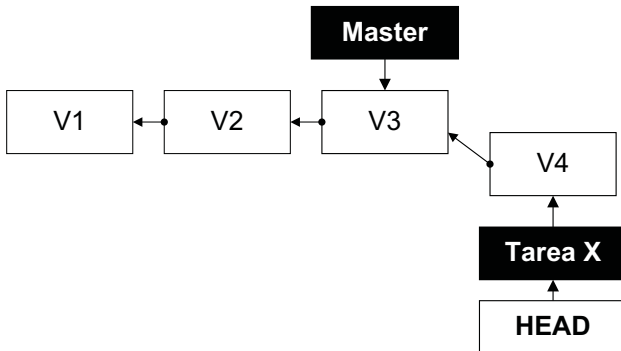




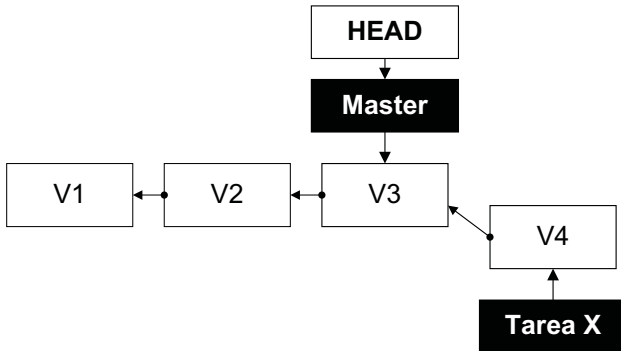
# Ramificación



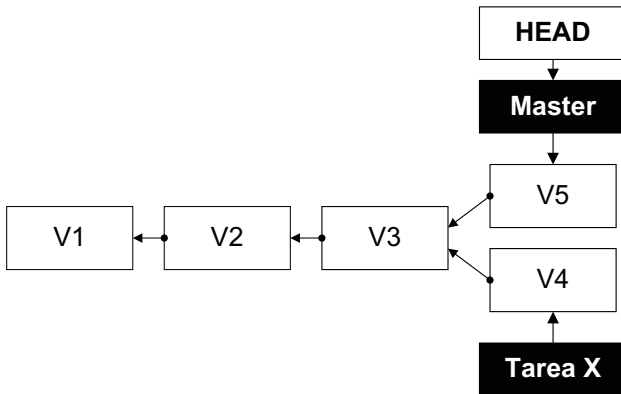
# Ramificación



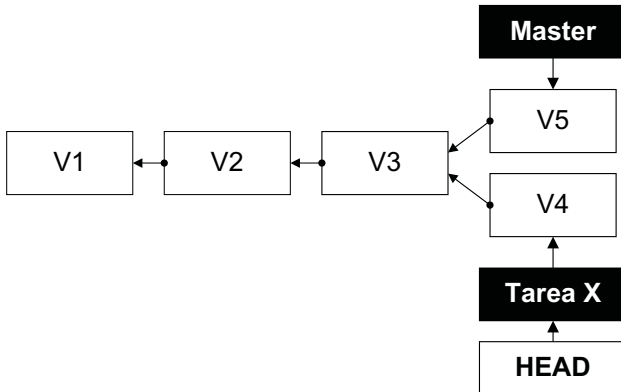
# Ramificación



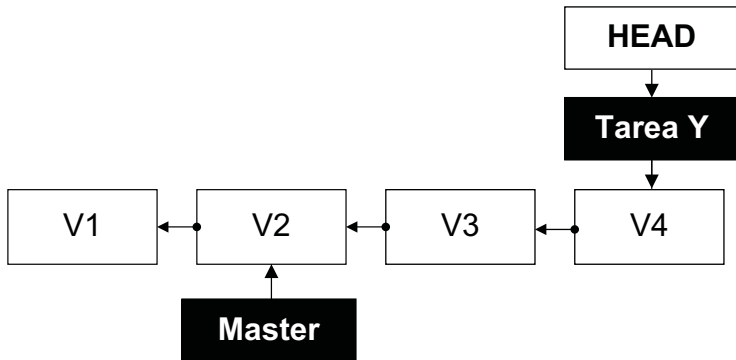
# Ramificación



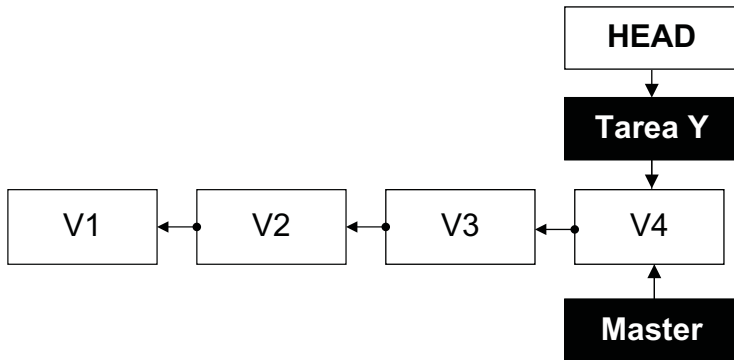
# Ramificación



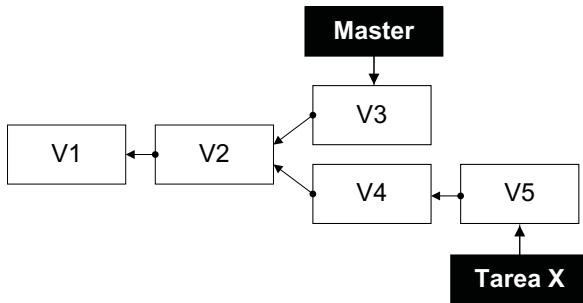
# Fusión por Avance Rápido



# Fusión por Avance Rápido

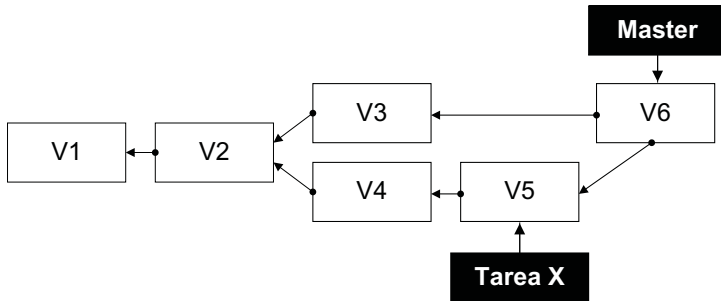


# Fusión por Recursión





# Fusión por Recursión



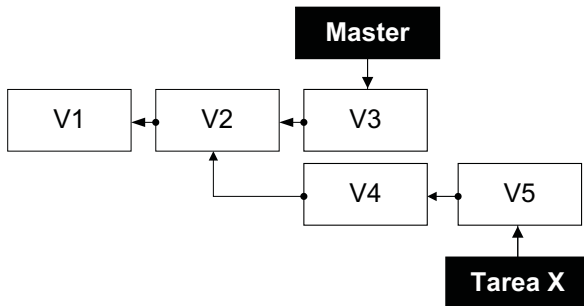
# Resolución de Conflictos

- 1 Los cambios no conflictivos (diferentes partes de un mismo archivo) simplemente se fusionan.
- 2 Los cambios conflictivos se marcan y deben resolverse manualmente.

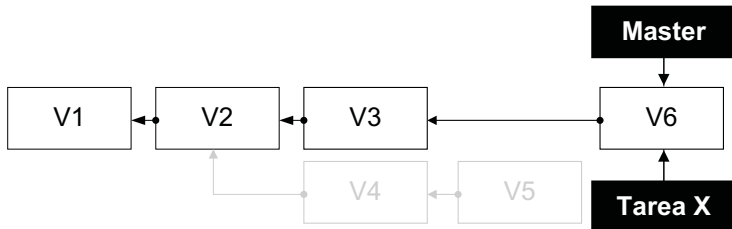
# Resolución de Conflictos

- 1 Los cambios no conflictivos (diferentes partes de un mismo archivo) simplemente se fusionan.
- 2 Los cambios conflictivos se marcan y deben resolverse manualmente.

# Fusión por Reorganización



# Fusión por Reorganización



# Comandos Ramificación y Fusión

**Branch** Permite la gestión de ramas.

**Checkout** Se desplaza la versión indicada.

**Tag** Asocia una etiqueta a una determinada version para facilitar su identificación.

**Merge** Fusiona dos ramas.

**Rebase** Reorganiza dos ramas.

# Comandos Ramificación y Fusión

**Branch** Permite la gestión de ramas.

**Checkout** Se desplaza la versión indicada.

**Tag** Asocia una etiqueta a una determinada version para facilitar su identificación.

**Merge** Fusiona dos ramas.

**Rebase** Reorganiza dos ramas.

# Comandos Ramificación y Fusión

**Branch** Permite la gestión de ramas.

**Checkout** Se desplaza la versión indicada.

**Tag** Asocia una etiqueta a una determinada version para facilitar su identificación.

**Merge** Fusiona dos ramas.

**Rebase** Reorganiza dos ramas.



# Comandos Ramificación y Fusión

**Branch** Permite la gestión de ramas.

**Checkout** Se desplaza la versión indicada.

**Tag** Asocia una etiqueta a una determinada version para facilitar su identificación.

**Merge** Fusiona dos ramas.

**Rebase** Reorganiza dos ramas.

# Comandos Ramificación y Fusión

**Branch** Permite la gestión de ramas.

**Checkout** Se desplaza la versión indicada.

**Tag** Asocia una etiqueta a una determinada version para facilitar su identificación.

**Merge** Fusiona dos ramas.

**Rebase** Reorganiza dos ramas.

# Índice

- 1 Introducción
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
- 4 Control de Versiones con Git
  - Características de Git
  - Acciones Básicas
  - Esquema de Versionado
  - Ramificación y Fusión
  - **Stashing**
  - Repositorios Remotos
  - Deshacer Cambios en Git
  - Modelos Organizativos con Git
- 5 Sumario

# Recogida y Clareado

**stash** Apila el contenido del directorio del trabajo y lo deja como tras el último *commit*.

**stash apply** Restaura un *stash* determinado.

**stash drop** Elimina un *stash* determinado.

**stash pop** Restaura un *stash* y lo borra.

# Recogida y Clareado

**stash** Apila el contenido del directorio del trabajo y lo deja como tras el último *commit*.

**stash apply** Restaura un *stash* determinado.

**stash drop** Elimina un *stash* determinado.

**stash pop** Restaura un *stash* y lo borra.

# Recogida y Clareado

**stash** Apila el contenido del directorio del trabajo y lo deja como tras el último *commit*.

**stash apply** Restaura un *stash* determinado.

**stash drop** Elimina un *stash* determinado.

**stash pop** Restaura un *stash* y lo borra.

# Recogida y Clareado

`stash` Apila el contenido del directorio del trabajo y lo deja como tras el último *commit*.

`stash apply` Restaura un *stash* determinado.

`stash drop` Elimina un *stash* determinado.

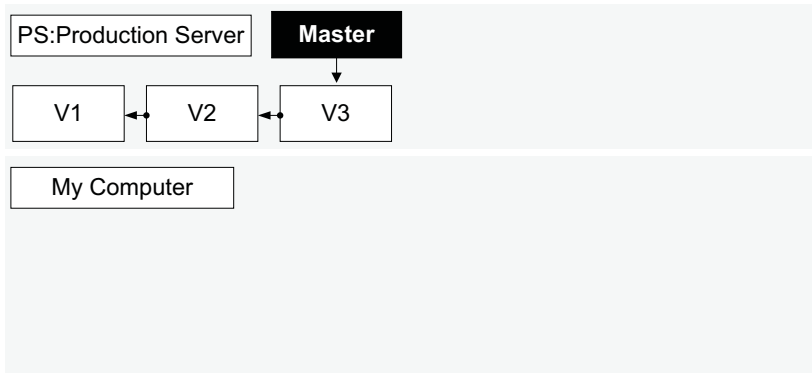
`stash pop` Restaura un *stash* y lo borra.

# Índice

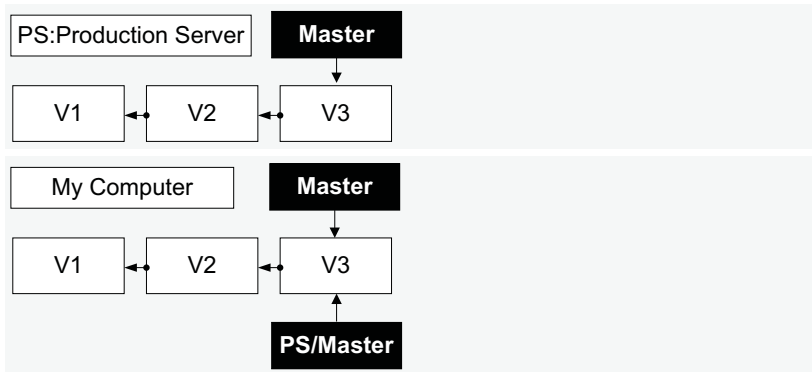
- 1 Introducción
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
- 4 Control de Versiones con Git
  - Características de Git
  - Acciones Básicas
  - Esquema de Versionado
  - Ramificación y Fusión
  - Stashing
  - Repositorios Remotos
  - Deshacer Cambios en Git
  - Modelos Organizativos con Git
- 5 Sumario



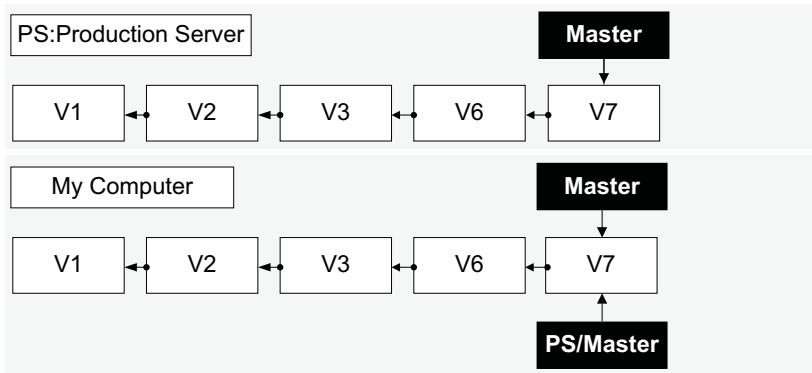
# Clonado y Seguimiento de Ramas



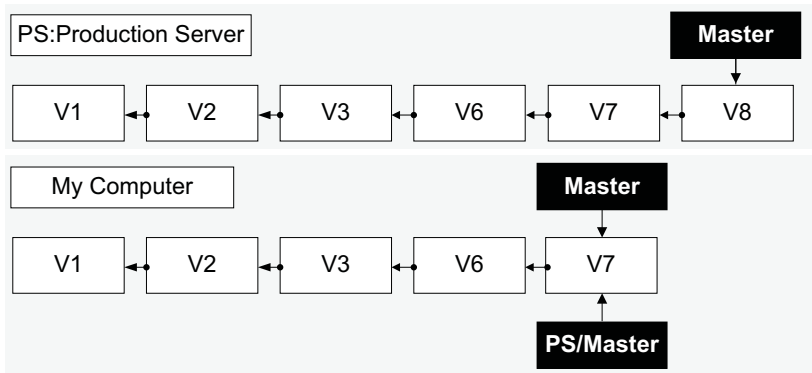
# Clonado y Seguimiento de Ramas



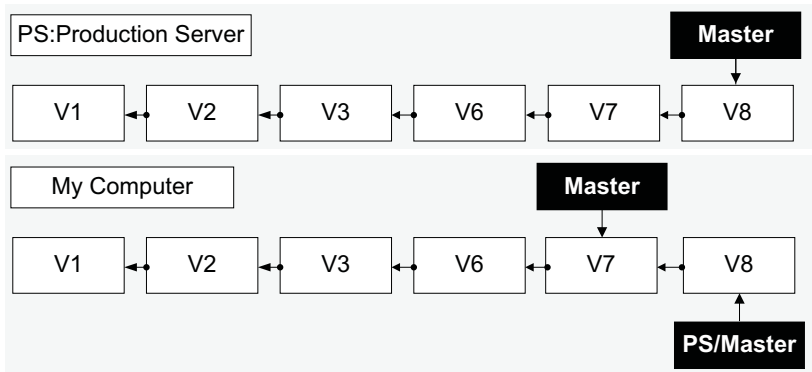
# Fetch



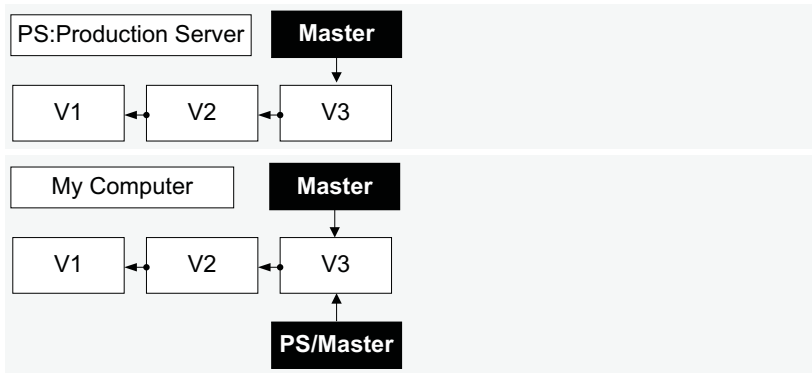
# Fetch



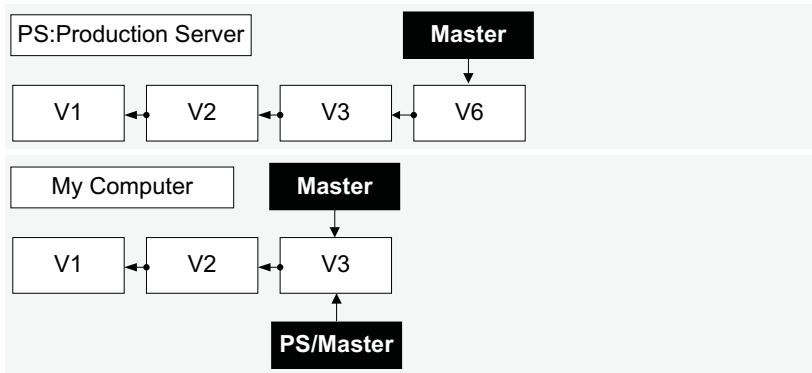
# Fetch



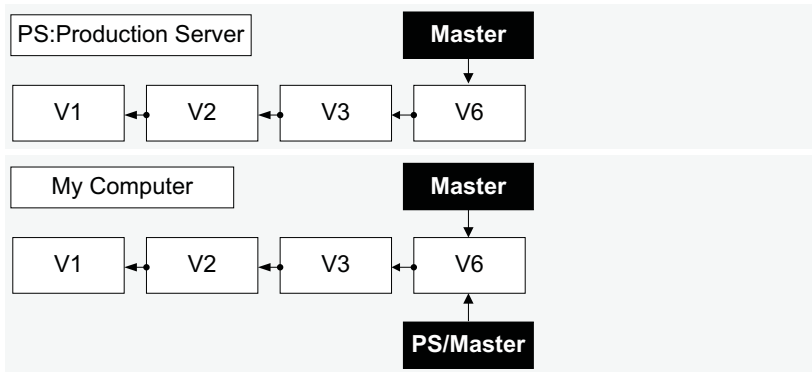
# Pull



# Pull

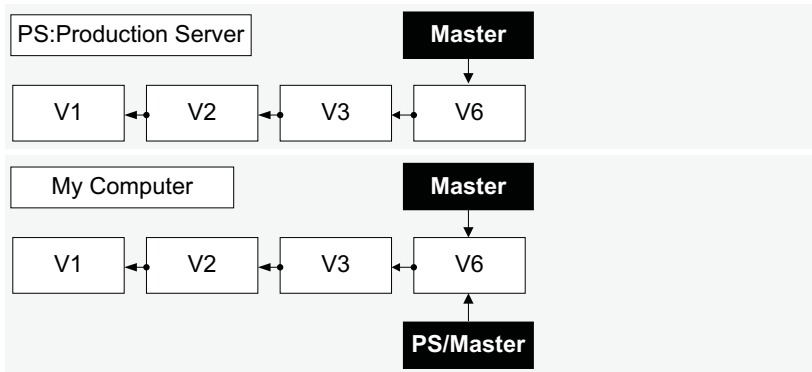


# Pull

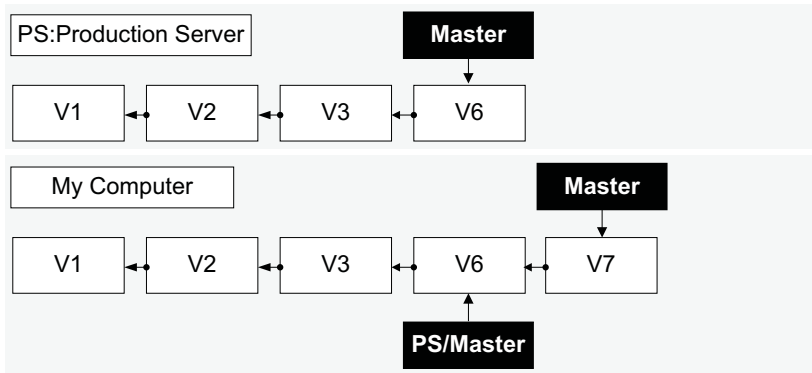




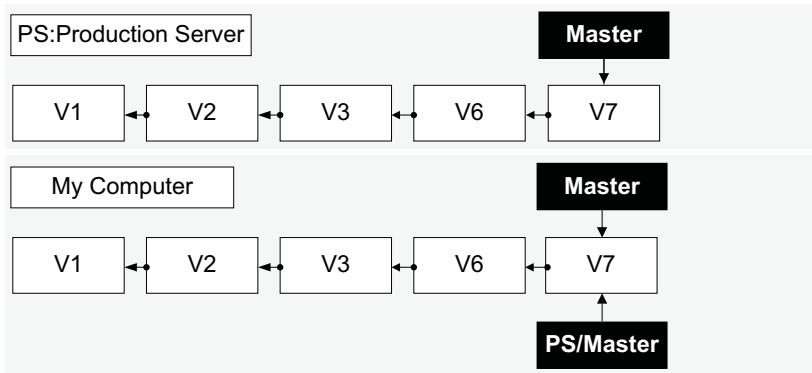
# Push



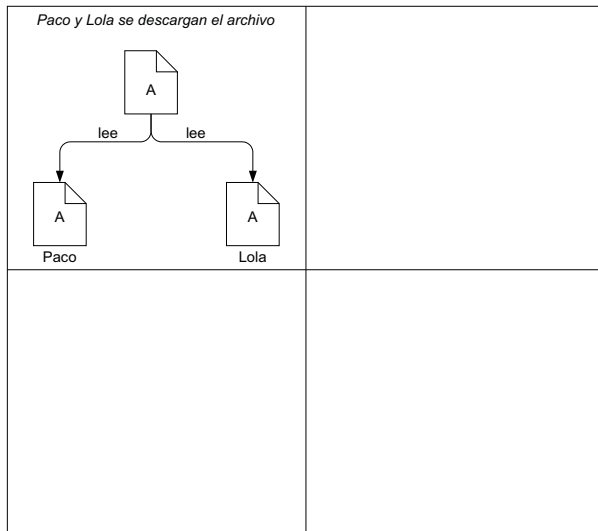
# Push



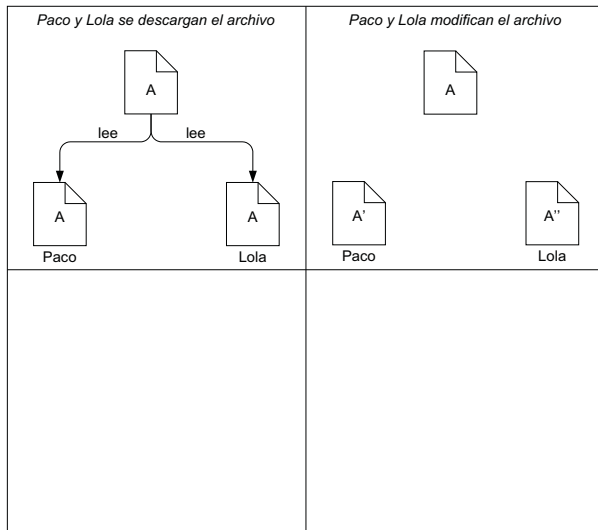
# Push



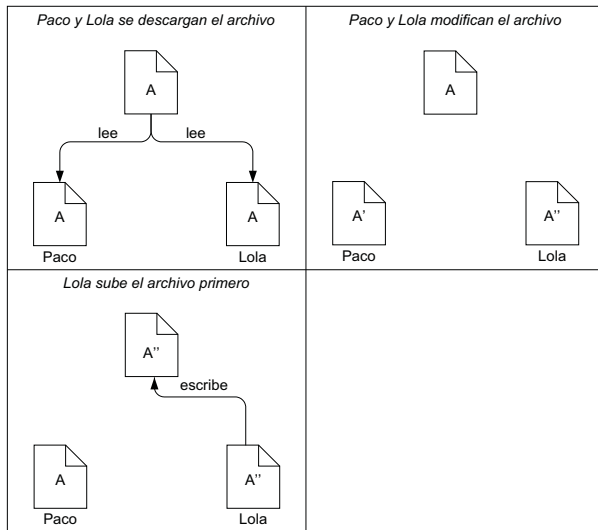
# Conflictos de Sincronización



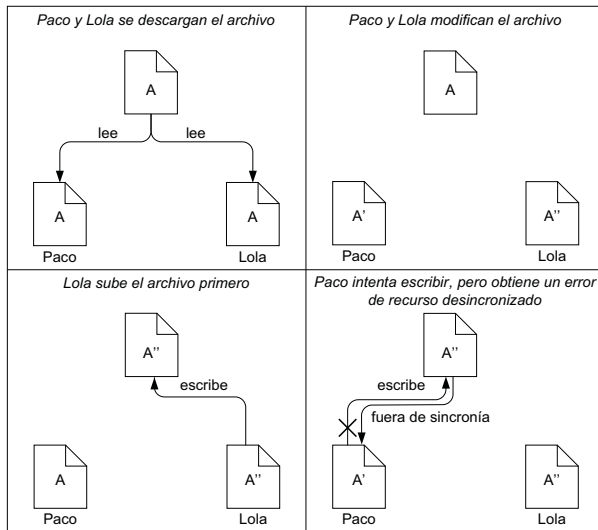
# Conflictos de Sincronización



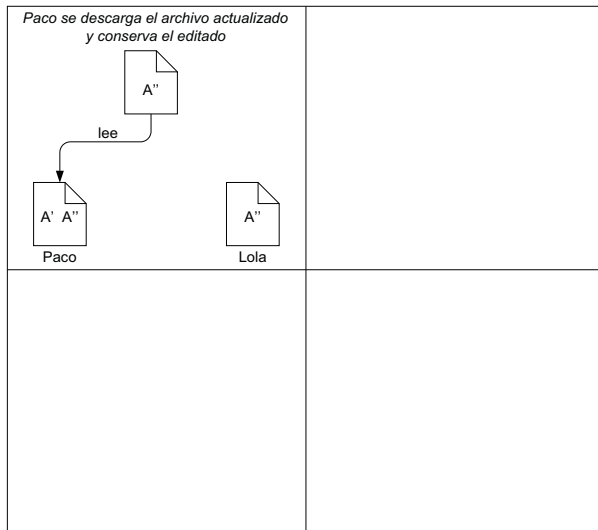
# Conflictos de Sincronización



# Conflictos de Sincronización

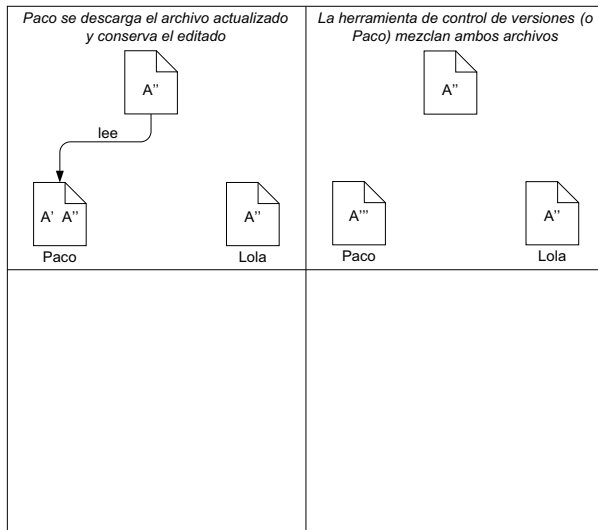


# Conflictos de Sincronización

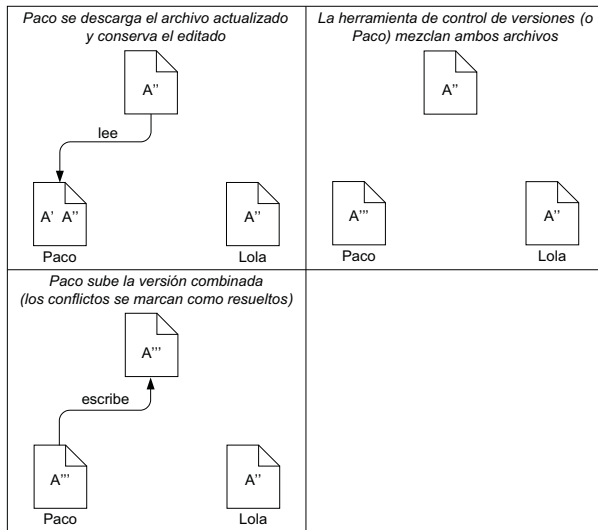




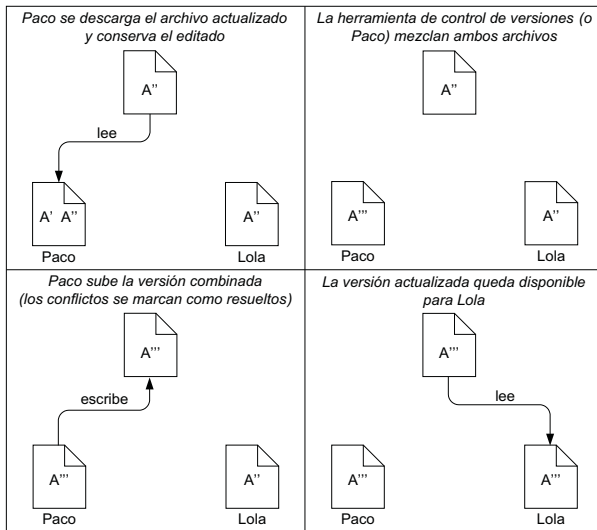
# Conflictos de Sincronización



# Conflictos de Sincronización



# Conflictos de Sincronización



# Índice

- 1 Introducción
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
- 4 Control de Versiones con Git
  - Características de Git
  - Acciones Básicas
  - Esquema de Versionado
  - Ramificación y Fusión
  - Stashing
  - Repositorios Remotos
  - **Deshacer Cambios en Git**
  - Modelos Organizativos con Git
- 5 Sumario

# Recuperación de Versiones Antiguas

**reset** Retrocede hasta la versión indicada, borrando los commits intermedios.

**revert** Deshace los cambios realizados hasta volver a la versión indicada. Se crean versiones intermedias para deshacer los cambios.

# Recuperación de Versiones Antiguas

- reset** Retrocede hasta la versión indicada, borrando los commits intermedios.
- revert** Deshace los cambios realizados hasta volver a la versión indicada. Se crean versiones intermedias para deshacer los cambios.

# Borrado y Renombrado

`rm` Borra ficheros (informando al control de versiones).

`mv` Renombra un fichero (informando al control de versiones).

# Borrado y Renombrado

`rm` Borra ficheros (informando al control de versiones).

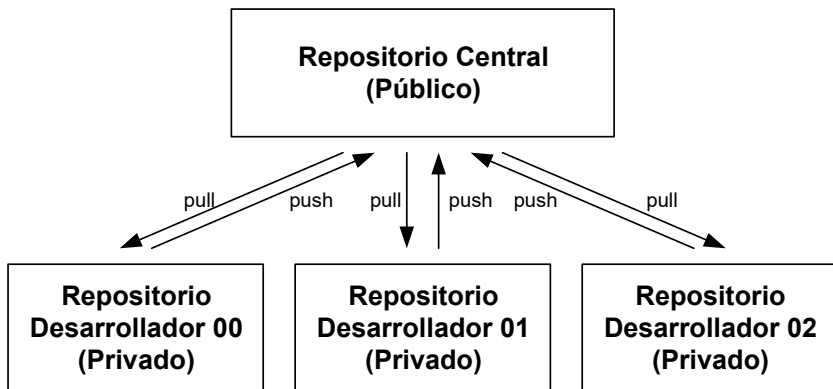
`mv` Renombra un fichero (informando al control de versiones).



# Índice

- 1 Introducción
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
- 4 Control de Versiones con Git
  - Características de Git
  - Acciones Básicas
  - Esquema de Versionado
  - Ramificación y Fusión
  - Stashing
  - Repositorios Remotos
  - Deshacer Cambios en Git
  - Modelos Organizativos con Git
- 5 Sumario

# Centralizado Clásico



# Integrador Destacado

**Repositorio Central  
(Público)**

**Integrador  
(Privado)**

# Integrador Destacado

**Repositorio Central  
(Público)**

**Repositorio Clonado 00  
(Público)**

**Repositorio Clonado 01  
(Público)**

**Integrador  
(Privado)**

# Integrador Destacado

**Repositorio Central  
(Público)**

**Repositorio Clonado 00  
(Público)**

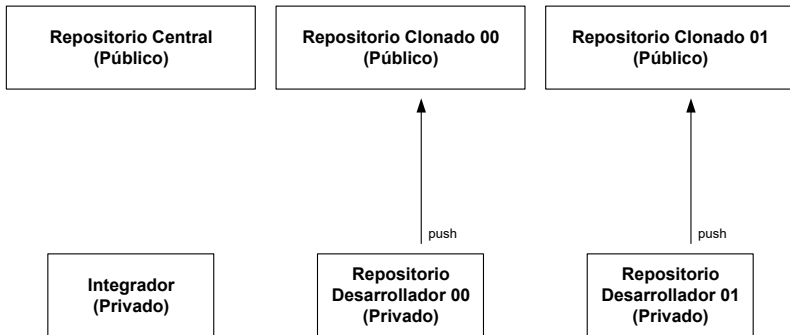
**Repositorio Clonado 01  
(Público)**

**Integrador  
(Privado)**

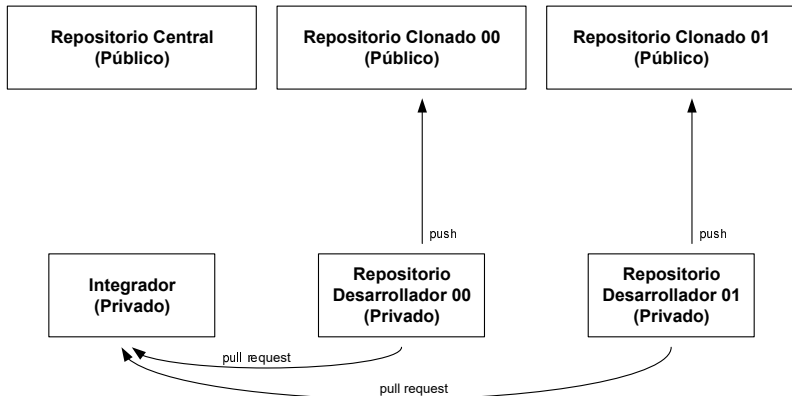
**Repositorio  
Desarrollador 00  
(Privado)**

**Repositorio  
Desarrollador 01  
(Privado)**

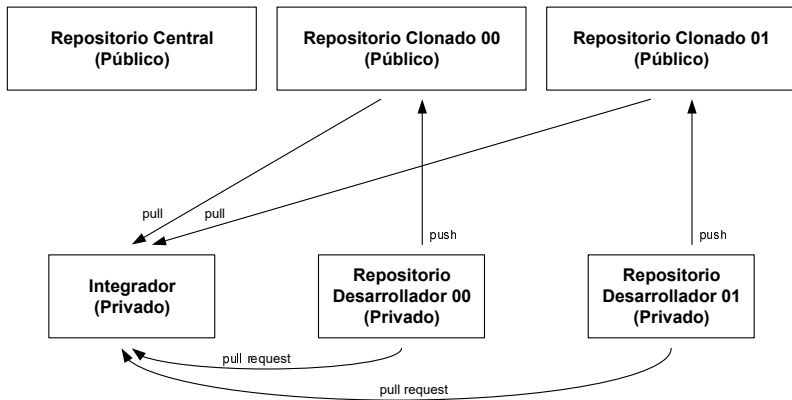
# Integrador Destacado



# Integrador Destacado

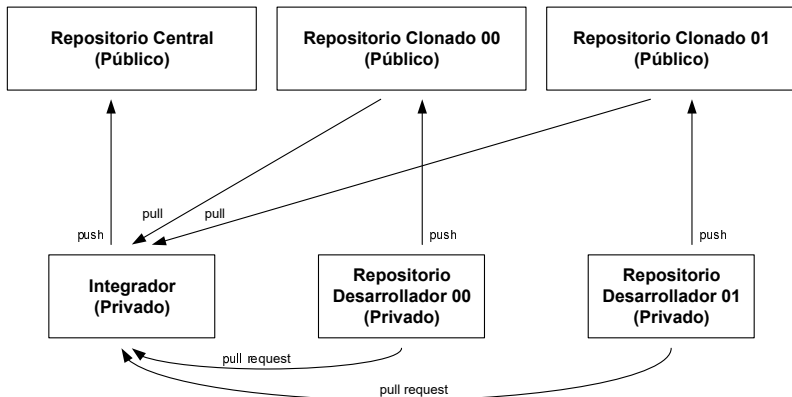


# Integrador Destacado

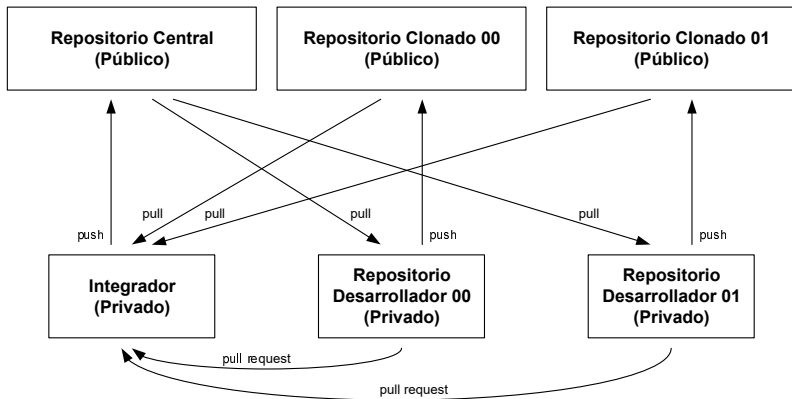




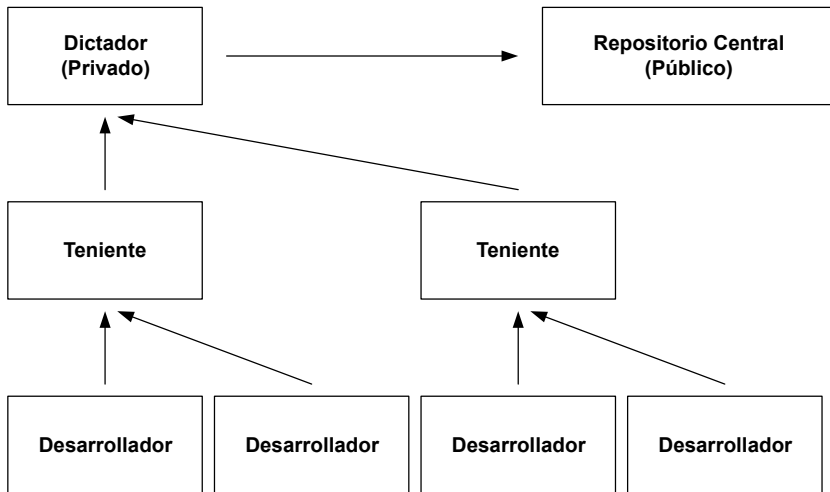
# Integrador Destacado



# Integrador Destacado



# Dictador y Tenientes



# Índice

- 1 Introducción
- 2 Terminología Básica
- 3 Procesos de Gestión de la Configuración Sw
- 4 Control de Versiones con Git
- 5 Sumario

# ¿Qué tengo que saber de todo esto?

- 1 Comprender y entender qué es la gestión de la configuración.
- 2 Conocer y comprender la terminología relacionada con la gestión de la configuración.
- 3 Conocer y comprender las áreas principales de la gestión de la configuración.
- 4 Conocer y comprender el concepto de integración y entrega continua.
- 5 Conocer y comprender el concepto de *DevOps*.

# ¿Qué tengo que saber de todo esto?

- 1 Comprender y entender qué es la gestión de la configuración.
- 2 Conocer y comprender la terminología relacionada con la gestión de la configuración.
- 3 Conocer y comprender las áreas principales de la gestión de la configuración.
- 4 Conocer y comprender el concepto de integración y entrega continua.
- 5 Conocer y comprender el concepto de *DevOps*.

# ¿Qué tengo que saber de todo esto?

- 1 Comprender y entender qué es la gestión de la configuración.
- 2 Conocer y comprender la terminología relacionada con la gestión de la configuración.
- 3 Conocer y comprender las áreas principales de la gestión de la configuración.
- 4 Conocer y comprender el concepto de integración y entrega continua.
- 5 Conocer y comprender el concepto de *DevOps*.

# ¿Qué tengo que saber de todo esto?

- 1 Comprender y entender qué es la gestión de la configuración.
- 2 Conocer y comprender la terminología relacionada con la gestión de la configuración.
- 3 Conocer y comprender las áreas principales de la gestión de la configuración.
- 4 Conocer y comprender el concepto de integración y entrega continua.
- 5 Conocer y comprender el concepto de *DevOps*.



# ¿Qué tengo que saber de todo esto?

- 1 Comprender y entender qué es la gestión de la configuración.
- 2 Conocer y comprender la terminología relacionada con la gestión de la configuración.
- 3 Conocer y comprender las áreas principales de la gestión de la configuración.
- 4 Conocer y comprender el concepto de integración y entrega continua.
- 5 Conocer y comprender el concepto de *DevOps*.

# ¿Qué tengo que saber de todo esto?

- 1 Comprender las ventajas de Git sobre otras herramientas de control de versiones.
- 2 Ser capaz de gestionar el área de trabajo de Git.
- 3 Ser capaz de almacenar, etiquetar y recuperar versiones con Git.
- 4 Ser capaz de crear ramas, gestionarlas y fusionarlas.
- 5 Ser capaz de entender por qué se producen conflictos al fusionar ramas y saber cómo solucionarlos.
- 6 Ser capaz de realizar tareas de aclarado y limpieza (*stashing*).
- 7 Ser capaz de trabajar con repositorios remotos.
- 8 Conocer y comprender las diferentes formas de organizar equipos de desarrollo utilizando Git.

# ¿Qué tengo que saber de todo esto?

- 1 Comprender las ventajas de Git sobre otras herramientas de control de versiones.
- 2 Ser capaz de gestionar el área de trabajo de Git.
- 3 Ser capaz de almacenar, etiquetar y recuperar versiones con Git.
- 4 Ser capaz de crear ramas, gestionarlas y fusionarlas.
- 5 Ser capaz de entender por qué se producen conflictos al fusionar ramas y saber cómo solucionarlos.
- 6 Ser capaz de realizar tareas de aclarado y limpieza (*stashing*).
- 7 Ser capaz de trabajar con repositorios remotos.
- 8 Conocer y comprender las diferentes formas de organizar equipos de desarrollo utilizando Git.

# ¿Qué tengo que saber de todo esto?

- 1 Comprender las ventajas de Git sobre otras herramientas de control de versiones.
- 2 Ser capaz de gestionar el área de trabajo de Git.
- 3 Ser capaz de almacenar, etiquetar y recuperar versiones con Git.
- 4 Ser capaz de crear ramas, gestionarlasy fusionarlas.
- 5 Ser capaz de entender por qué se producen conflictos al fusionar ramas y saber cómo solucionarlos.
- 6 Ser capaz de realizar tareas de aclarado y limpieza (*stashing*).
- 7 Ser capaz de trabajar con repositorios remotos.
- 8 Conocer y comprender las diferentes formas de organizar equipos de desarrollo utilizando Git.

# ¿Qué tengo que saber de todo esto?

- 1 Comprender las ventajas de Git sobre otras herramientas de control de versiones.
- 2 Ser capaz de gestionar el área de trabajo de Git.
- 3 Ser capaz de almacenar, etiquetar y recuperar versiones con Git.
- 4 Ser capaz de crear ramas, gestionarlas y fusionarlas.
- 5 Ser capaz de entender por qué se producen conflictos al fusionar ramas y saber cómo solucionarlos.
- 6 Ser capaz de realizar tareas de aclarado y limpieza (*stashing*).
- 7 Ser capaz de trabajar con repositorios remotos.
- 8 Conocer y comprender las diferentes formas de organizar equipos de desarrollo utilizando Git.

# ¿Qué tengo que saber de todo esto?

- 1 Comprender las ventajas de Git sobre otras herramientas de control de versiones.
- 2 Ser capaz de gestionar el área de trabajo de Git.
- 3 Ser capaz de almacenar, etiquetar y recuperar versiones con Git.
- 4 Ser capaz de crear ramas, gestionarlas y fusionarlas.
- 5 Ser capaz de entender por qué se producen conflictos al fusionar ramas y saber cómo solucionarlos.
- 6 Ser capaz de realizar tareas de aclarado y limpieza (*stashing*).
- 7 Ser capaz de trabajar con repositorios remotos.
- 8 Conocer y comprender las diferentes formas de organizar equipos de desarrollo utilizando Git.

# ¿Qué tengo que saber de todo esto?

- 1 Comprender las ventajas de Git sobre otras herramientas de control de versiones.
- 2 Ser capaz de gestionar el área de trabajo de Git.
- 3 Ser capaz de almacenar, etiquetar y recuperar versiones con Git.
- 4 Ser capaz de crear ramas, gestionarlasy fusionarlas.
- 5 Ser capaz de entender por qué se producen conflictos al fusionar ramas y saber cómo solucionarlos.
- 6 Ser capaz de realizar tareas de aclarado y limpieza (*stashing*).
- 7 Ser capaz de trabajar con repositorios remotos.
- 8 Conocer y comprender las diferentes formas de organizar equipos de desarrollo utilizando Git.

# ¿Qué tengo que saber de todo esto?

- 1 Comprender las ventajas de Git sobre otras herramientas de control de versiones.
- 2 Ser capaz de gestionar el área de trabajo de Git.
- 3 Ser capaz de almacenar, etiquetar y recuperar versiones con Git.
- 4 Ser capaz de crear ramas, gestionarlas y fusionarlas.
- 5 Ser capaz de entender por qué se producen conflictos al fusionar ramas y saber cómo solucionarlos.
- 6 Ser capaz de realizar tareas de aclarado y limpieza (*stashing*).
- 7 Ser capaz de trabajar con repositorios remotos.
- 8 Conocer y comprender las diferentes formas de organizar equipos de desarrollo utilizando Git.



# ¿Qué tengo que saber de todo esto?

- 1 Comprender las ventajas de Git sobre otras herramientas de control de versiones.
- 2 Ser capaz de gestionar el área de trabajo de Git.
- 3 Ser capaz de almacenar, etiquetar y recuperar versiones con Git.
- 4 Ser capaz de crear ramas, gestionarlasy fusionarlas.
- 5 Ser capaz de entender por qué se producen conflictos al fusionar ramas y saber cómo solucionarlos.
- 6 Ser capaz de realizar tareas de aclarado y limpieza (*stashing*).
- 7 Ser capaz de trabajar con repositorios remotos.
- 8 Conocer y comprender las diferentes formas de organizar equipos de desarrollo utilizando Git.