



escuela técnica superior
de ingeniería informática

Gestión del código fuente *Source code management*

*Departamento de
Lenguajes y Sistemas Informáticos*

**Evolución y Gestión de la
Configuración**

UNIVERSIDAD DE SEVILLA

Ejemplo de otro dominio



ME QUIERO MUDAR



Ejemplo de otro dominio



FABRICAR COCHES

¿Qué hay en estos
escenarios que no
siempre hay en los
proyectos software?

Índice



Introducción

Conceptos básicos

Operaciones

Gestión de ramas

Uso de repositorios

Principios

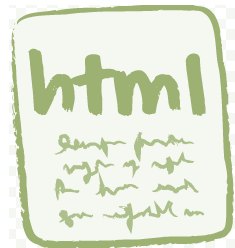
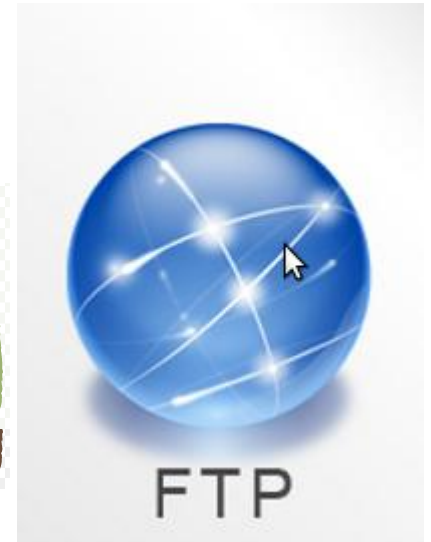
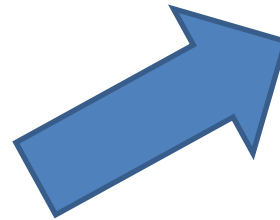
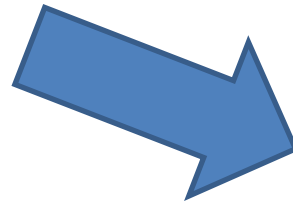
Resumen

Bibliografía

Escenarios

- ¿Podrías instalar el proyecto de IISSI en cualquier ordenador en un solo clic?
- ¿Sabrías la versión exacta del código que entregaste del proyecto en cada convocatoria?
- ¿En caso de que se detectara un *bug*, serías capaz de retomar exactamente la misma versión que entregaste, arreglar el bug y automáticamente desplegar los cambios?
- ¿Cómo gestionabas el trabajo en equipo?
¿Cómo “unías” las distintas partes de la aplicación?
- ¿Cómo gestionabas los entregables?

Escenario web

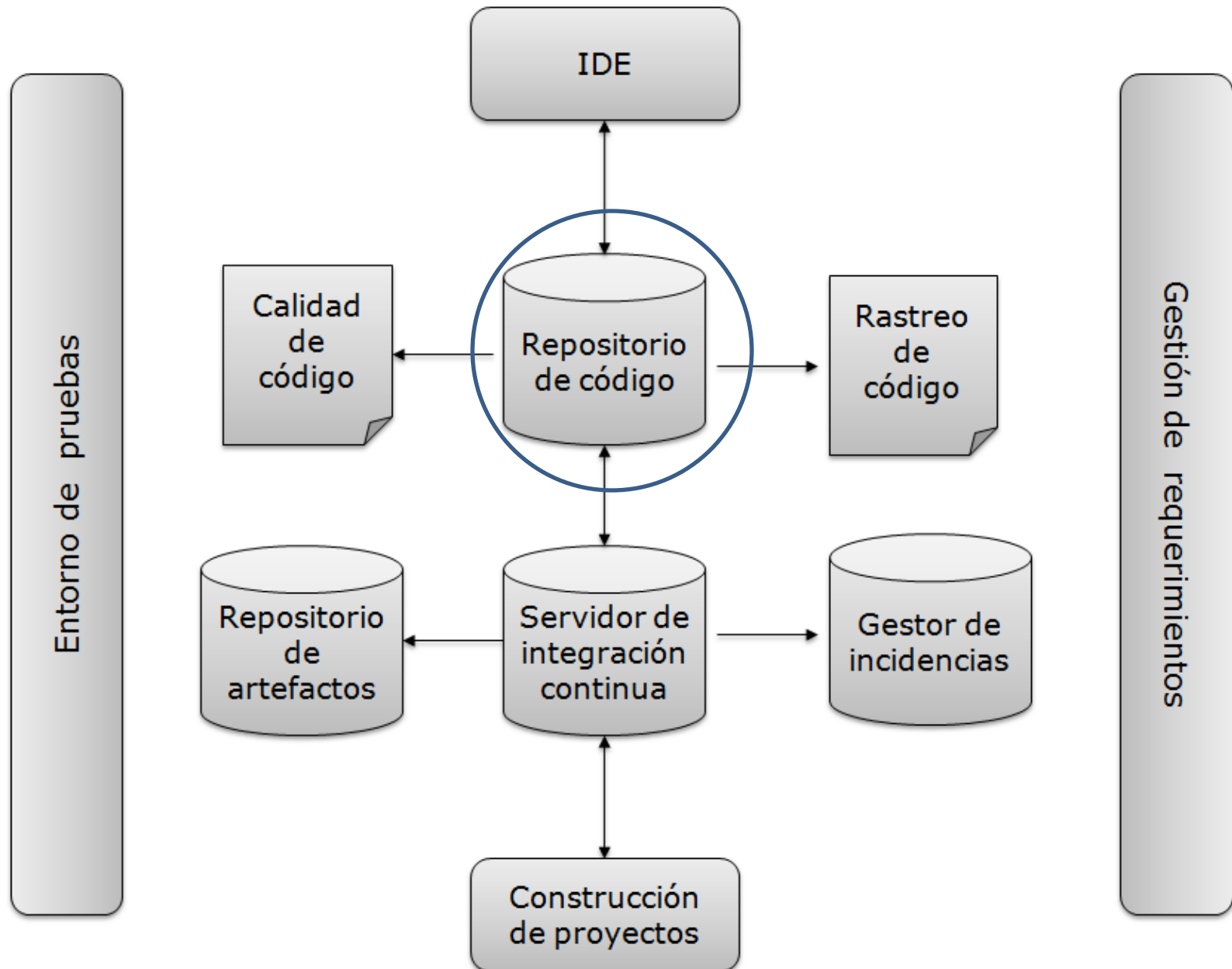


¿Cómo gestionar la colaboración?

El problema

¿Cómo gestionar la evolución y configuración del código fuente de nuestros proyectos?

Application Lifecycle Management



Índice



Introducción

Conceptos básicos

Operaciones

Gestión de ramas

Uso de repositorios

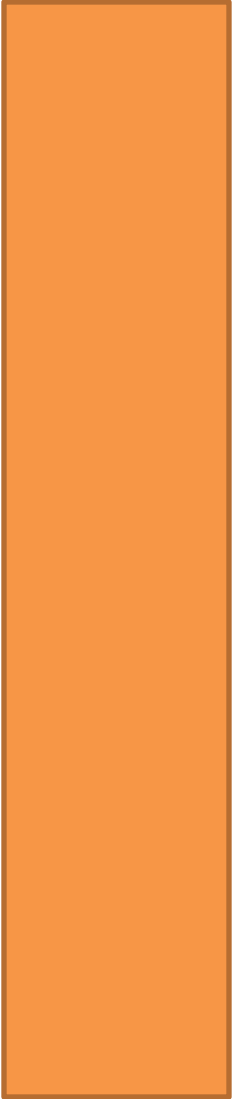
Principios

Resumen

Bibliografía



Conceptos básicos



Máquina del tiempo

Repositorio

Branch

Baseline

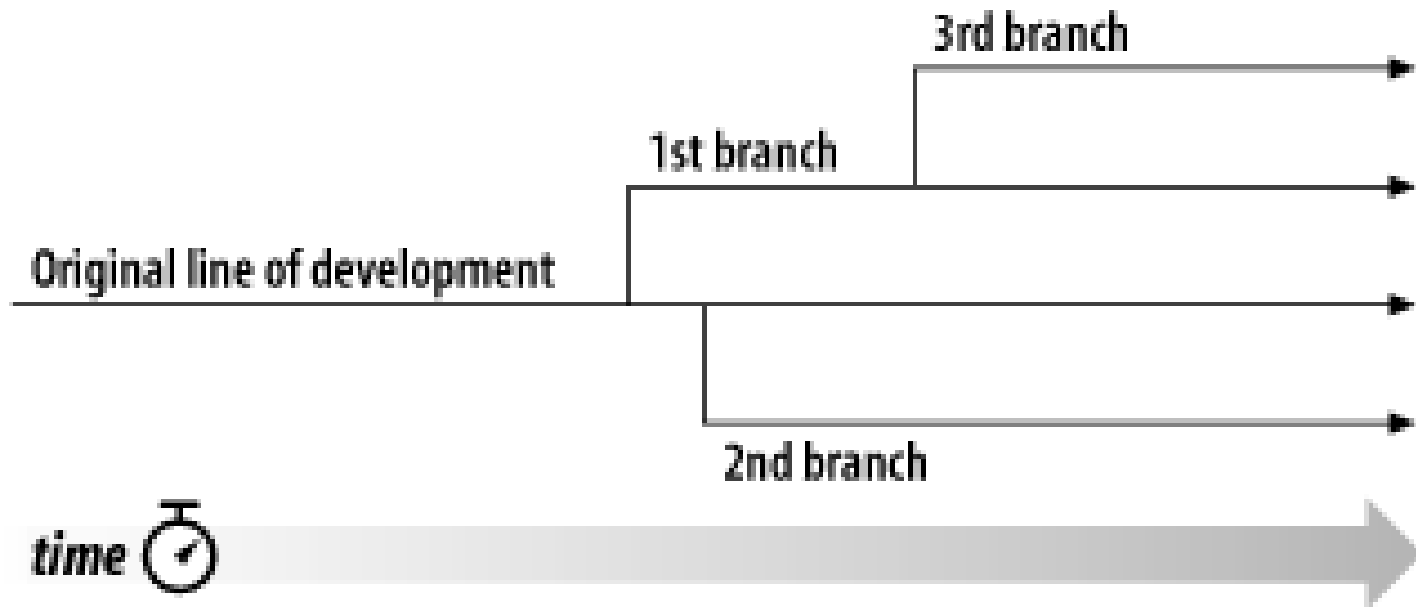
Sandbox

Repositorio de código

- Un repositorio de código es un sistema para guardar múltiples versiones de los ficheros de un proyecto de modo que cuándo se modifique un fichero se pueda acceder a las versiones anteriores
- Repositorio de código = Sistema de control de versiones = Control de código = sistema de control de revisiones = sistemas de gestión de código = "el repositorio" = "el svn", "el git", "el mercurial"...

Branch

- **Branch**: una línea de desarrollo independiente que puede compartir parte de historia común con otras Concepto de branch hija, branch padre

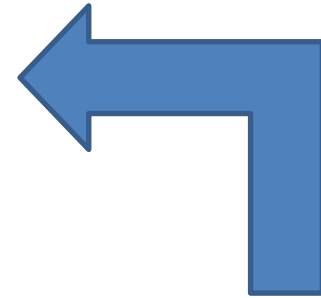


Baseline

- **Baseline:** Los cambios en el software requieren del paso por una serie de estados. Identificar los estados importantes de los artefactos de software es lo que se llama crear un “baseline” del producto.
- También se suele conocer como “tagging” “labeling” “snapshoting”
- Las “baseline” deben ser **inmutables**.
- Estados frecuentes:
 - Desarrollo -> Pruebas -> Pre-producción -> Producción.

Sandbox

- Sandbox/workspace/working copy



checkout



Pero...¿Sólo
guardamos el
código fuente? ¿qué
más?

Índice

Introducción

Conceptos básicos



Operaciones

Gestión de ramas

Uso de repositorios

Principios

Resumen

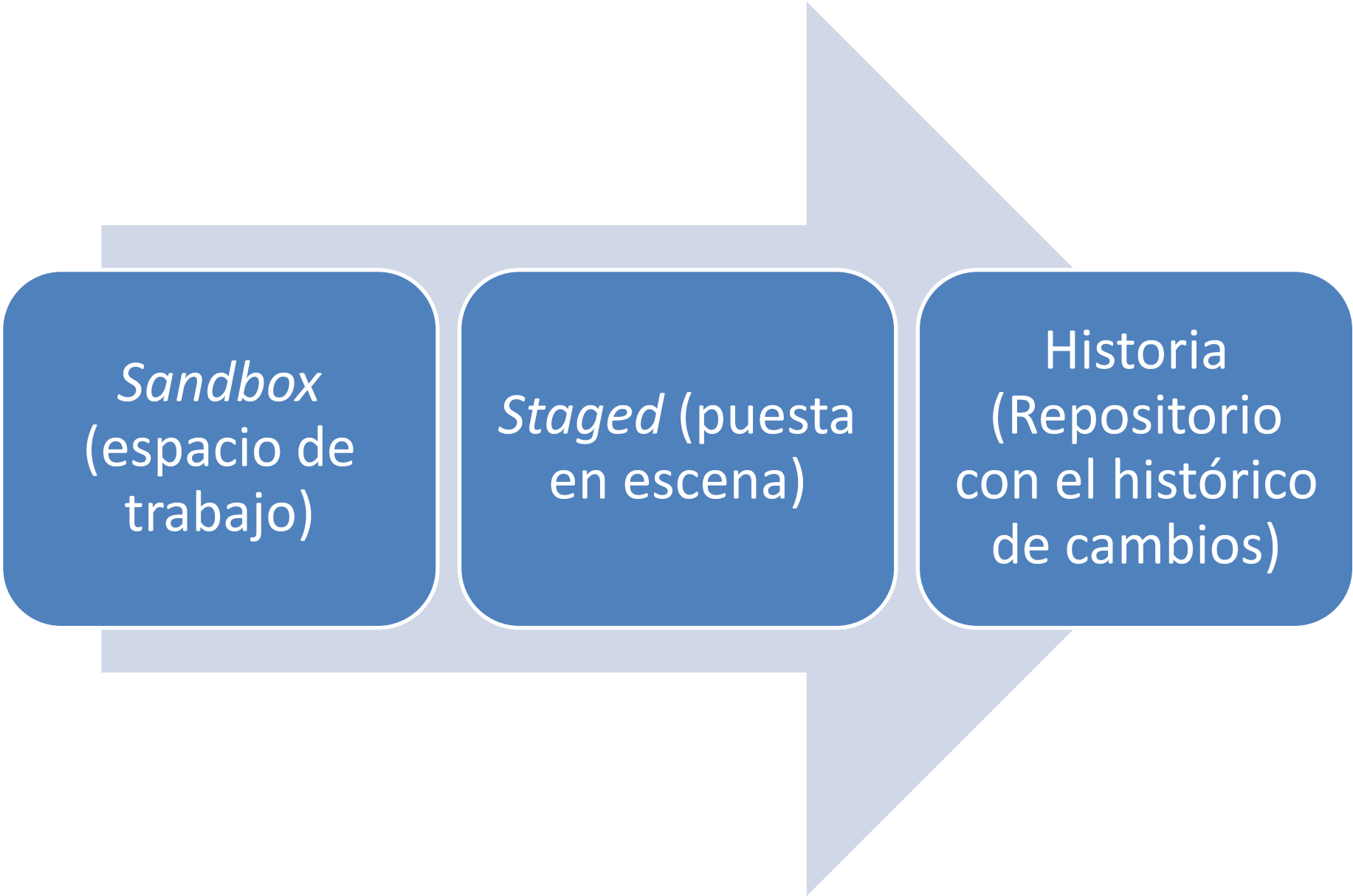
Bibliografía

Operaciones

¿Cómo hacer las operaciones básicas en una “máquina del tiempo”?



Lugares de un *configuration item*



Sandbox
(espacio de
trabajo)

The diagram consists of three blue rounded rectangular boxes arranged horizontally. Each box contains text describing a location for a configuration item. The boxes are set against a light blue background that features a large, faint arrow pointing from left to right, passing behind the boxes.

Staged (puesta
en escena)

Historia
(Repositorio
con el histórico
de cambios)

Operaciones frecuentes

- **Checkout**: tomar los artefactos del repositorio y llevarlo al área de trabajo (*sandbox*) para realizar modificaciones



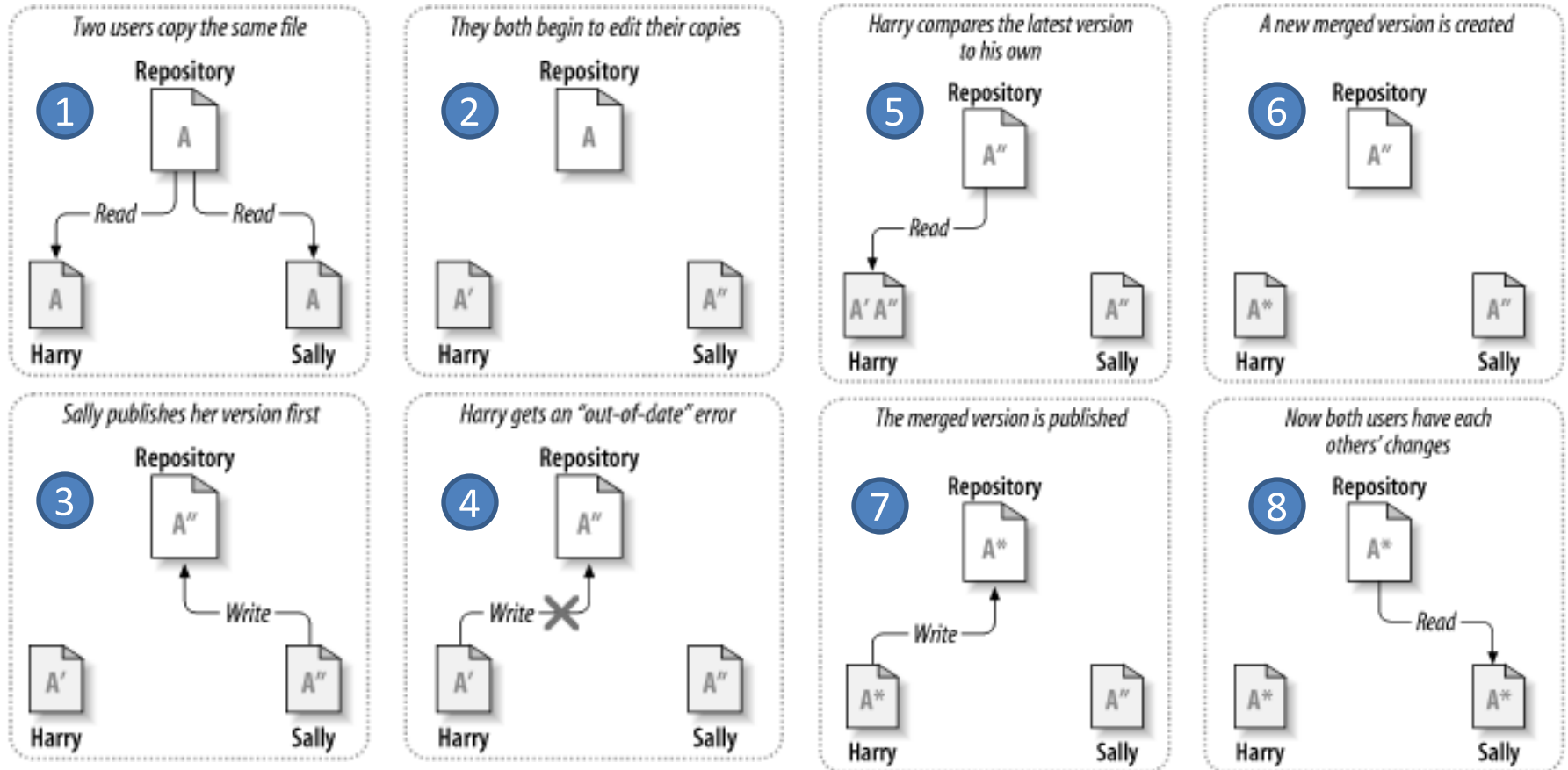
checkout



OJO: puede haber “juguetes” nuevos

Operaciones frecuentes

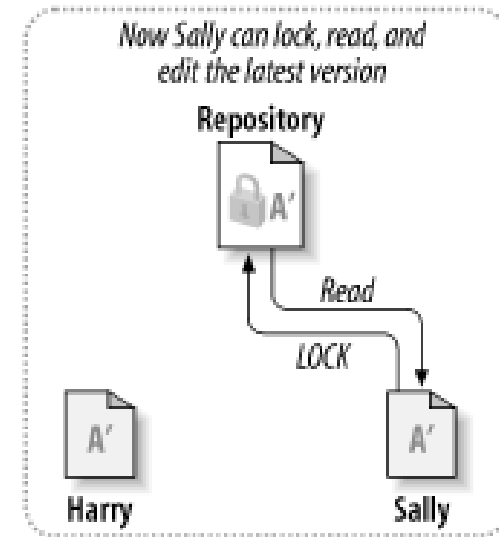
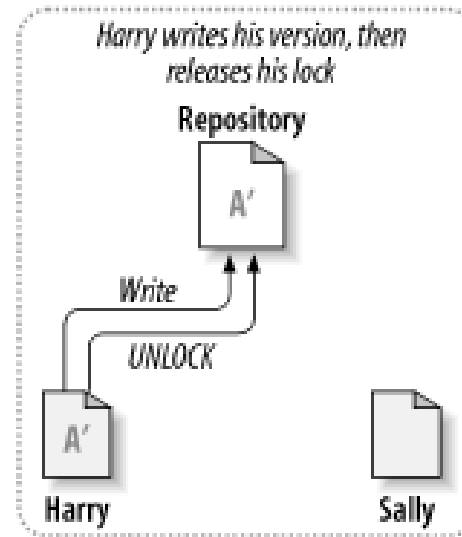
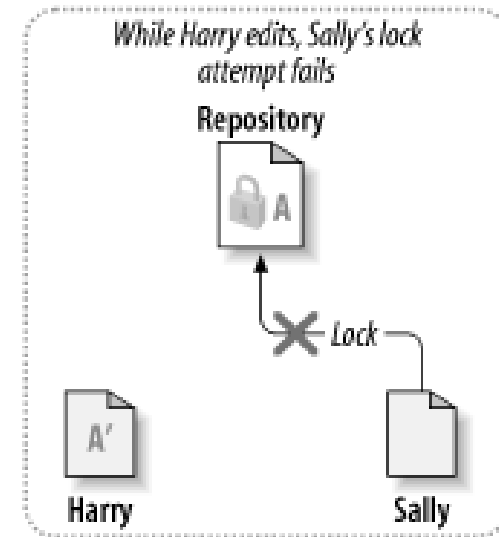
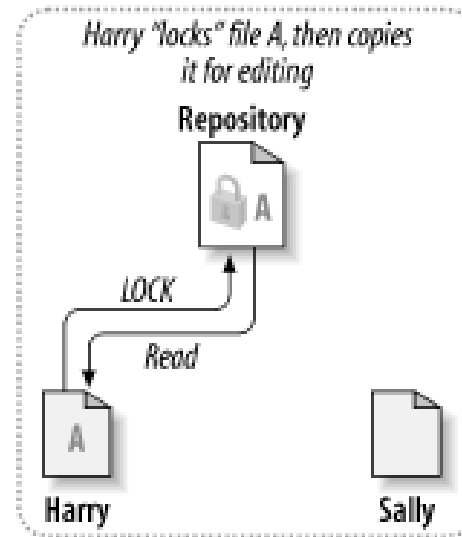
• Checkout no reservado



Operaciones frecuentes

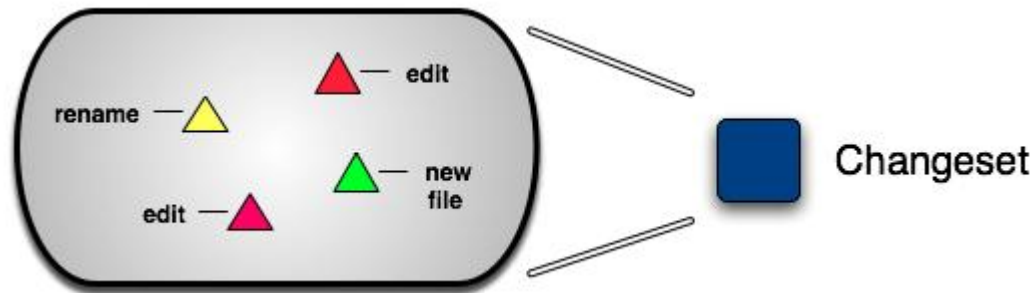
- **Checkout reservado (pesimista)** :
cuándo se hace, no puede haber más de una persona al mismo tiempo modificando el código

¿Cuándo conviene hacerlo?



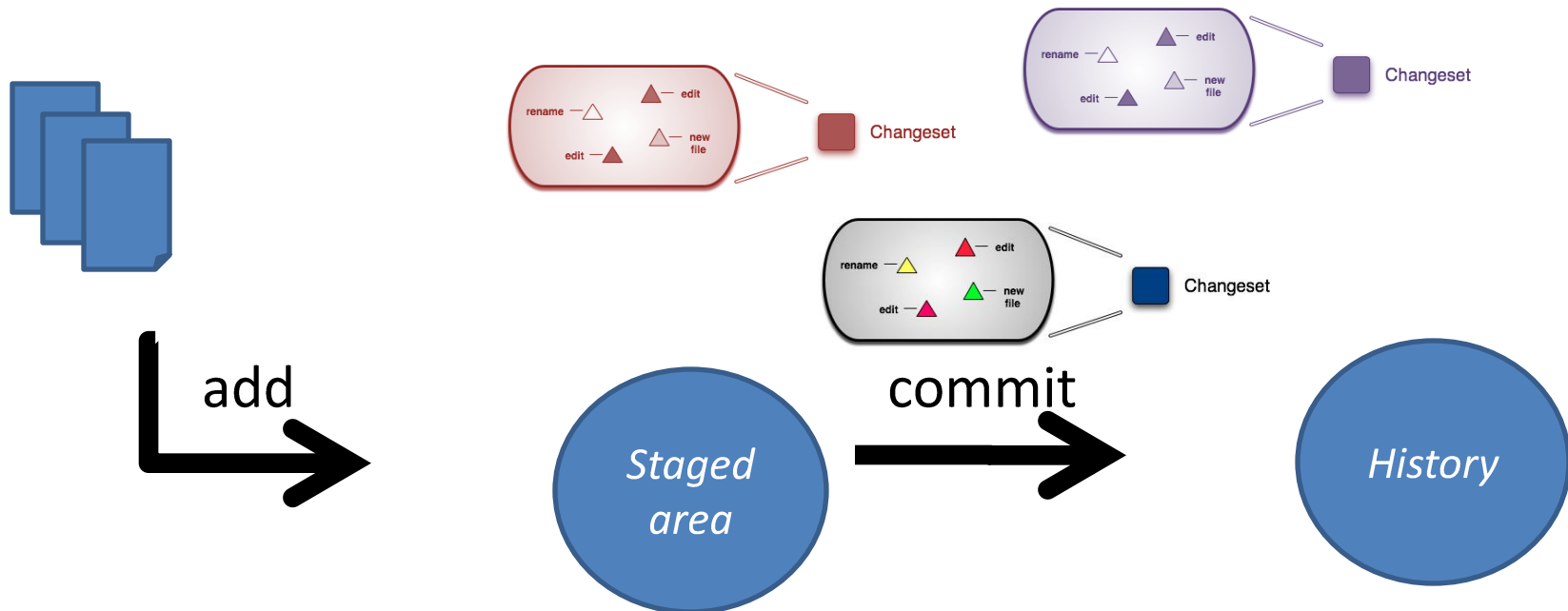
Operaciones frecuentes

- **Changeset**: conjunto de cambios que deben ser tratados como un conjunto indivisible (en svn se conoce como "revision").



Operaciones frecuentes

- **Add**: Añadir un elemento a la “máquina del tiempo” de modo que su estado y versiones sean controlados por al misma
- **Commit**: Guardar en la máquina del tiempo un conjunto de “changesets” Pueden ser atómicos o no.



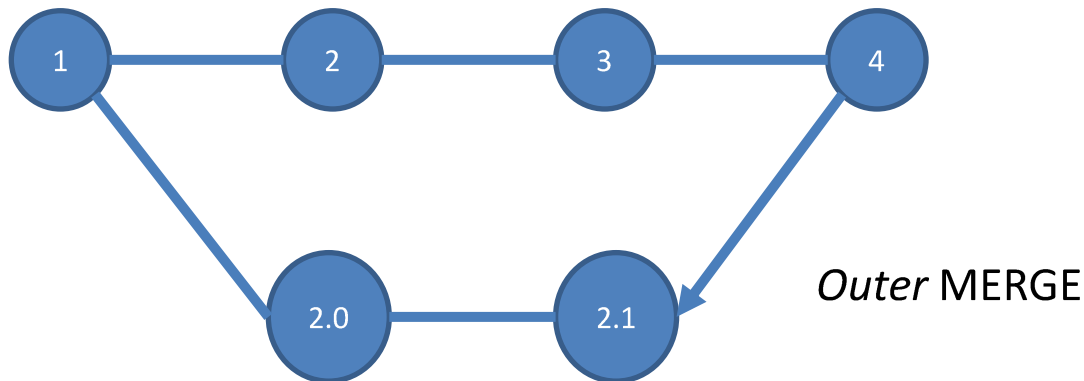
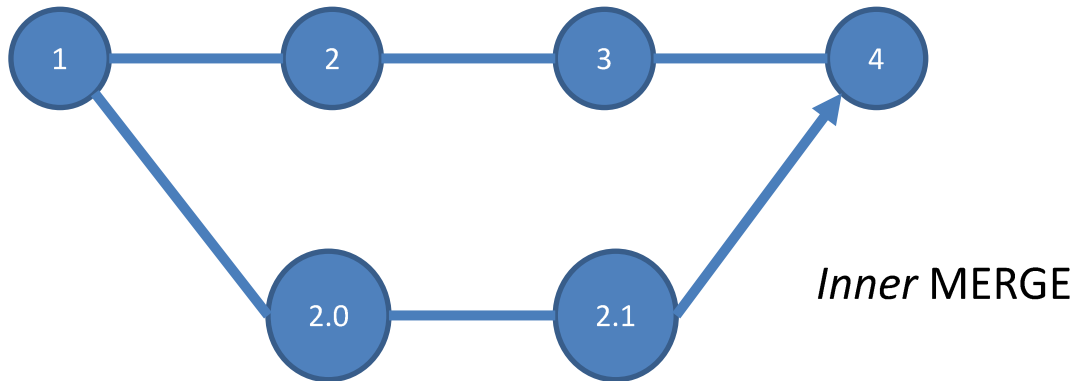
Operaciones básicas

- **Crear ramas/branches**
- **Copybranches vs deltas:** Algunos de los gestores de código requieren que una rama tenga un duplicado de todos los artefactos (copybranch). Otras herramientas en cambio, sólo usan una “subrama” que contiene sólo los artefactos que han cambiado. A estas subramas se les suele llamar en este caso, “deltas”.

¿Veis algún problema en una u otra aproximación?

Operaciones frecuentes

- **Merging:** Es el proceso de mezclar dos o más ramas de artefactos



Operaciones frecuentes

- **Resolucion de conflictos**. Cuando se hace *merge* pueden aparecer conflictos. Algunos pueden resolverse de manera automática, otros requieren intervención manual. ***Resolver conflictos es duro.***
- **Diff**. Hacer *diff* significa mirar la diferencia que hay entre artefactos de un repositorio. A partir de un *diff* se puede generar un **patch**, es decir, un conjunto de cambios a realizar a un conjunto de artefactos (ver <http://es.wikipedia.org/wiki/Diff> probar con <http://www.quickdiff.com>). Cuándo un conjunto de cambios se aplican a una serie de artefactos, se dice que se ha aplicado un *patch*.

Conflictos
sintácticos vs
conflictos
semánticos