



Control de Versiones

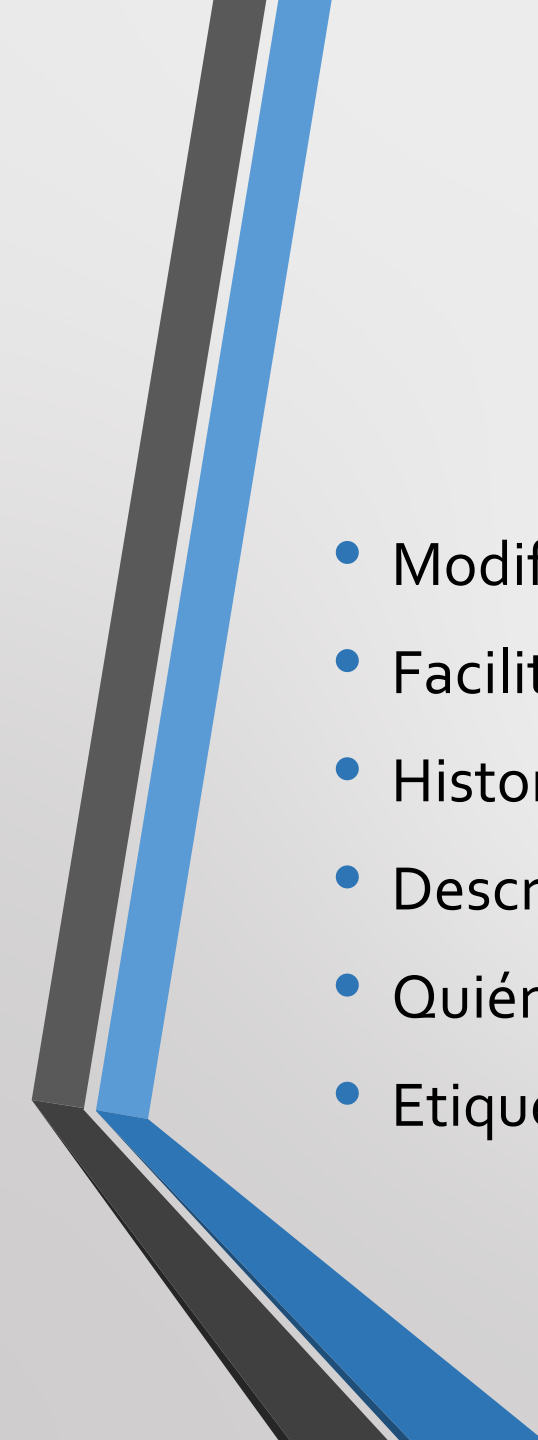
Manolo Pérez Aixendri

Desarrollo de Videojuegos- Grado de Multimedia

ETSE –Universitat de Valencia

Control de versiones


- ¿Por qué?
- Tipos según
 - Sistema de colaboración
 - Arquitectura
- Terminología
- Sistemas populares
 - SourceSafe
 - SVN
 - Git
 - Mercurial



Control de versiones

¿Por qué?


- Modificaciones y vuelta atrás (Copia de seguridad).
- Facilita desarrollo en equipo.
- Historial de cambios.
- Descripciones de cambios.
- Quién los ha realizado.
- Etiquetado de versiones.



Control de versiones

Sistema de colaboración

- Dos soluciones:
 - Bloquear-modificar-desbloquear
 - Copiar-modificar-fusionar



Control de versiones Arquitectura

- Centralizados (clientes-servidor)
- Distribuidos

Control de versiones

Terminología

- Revisión
- Repositorio
- Módulo
- Etiquetar
- Ramificar
- Desplegar (Checkout)
- Publicar
- Conflicto
- Resolver
- Cambio
- Lista de cambios
- Exportar
- Importar
- Bloquear/Desbloquear
- Integrar o fusionar
- Actualizar
- Copia de trabajo

Control de versiones

Terminología

- **Revisión:**
- Es una visión estática en el tiempo del estado de un grupo de archivos y directorios.
- Puede tener asociada metadata:
 - Quién hizo las modificaciones
 - Fecha y hora en la cual se almacenaron los cambios
 - Razón para los cambios
 - De qué revisión y/o rama se deriva la revisión
 - Palabras o términos clave asociados a la revisión

Control de versiones

Terminología

Copia de trabajo:

- El conjunto de directorios y archivos manejados por el SCV, y que están bajo edición activa.
- Está asociado a una rama de trabajo en particular.

Rama (de Trabajo/Desarrollo):

- En el caso mas simple, una rama es un conjunto ordenado de revisiones. La revisión más reciente se denomina la “cabeza” (head) o “principal” (main).
- Las ramas se pueden separar y juntar como se necesite, formando un grafo de revisión(Grafo Acíclico Directo)

Control de versiones

Terminología

Repositorio:

- Un sitio de almacenamiento de revisiones.
- Físicamente puedes usar un archivo, colección de archivos, base de datos, etc.

Conflicto:

- Ocurre cuando dos personas han hecho cambios contradictorios en un mismo documento (o grupo de documentos).
- Los SCVs sólo alertan acerca de un conflicto.

Control de versiones

Terminología

Resolución:

- El proceso de resolver un conflicto.

Cambio (o Diff o Delta):

- Modificación en un archivo bajo control de revisiones.

Integrar o Fusionar:

- Une dos grupos de cambios en un archivo (o grupo de archivos), generando una revisión unificada.

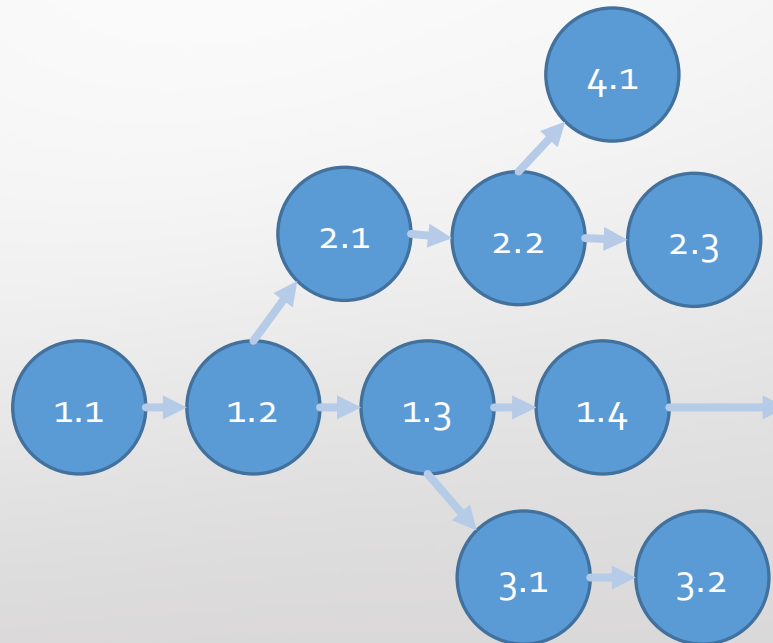
Control de versiones

Grafo de evolución

Lineal

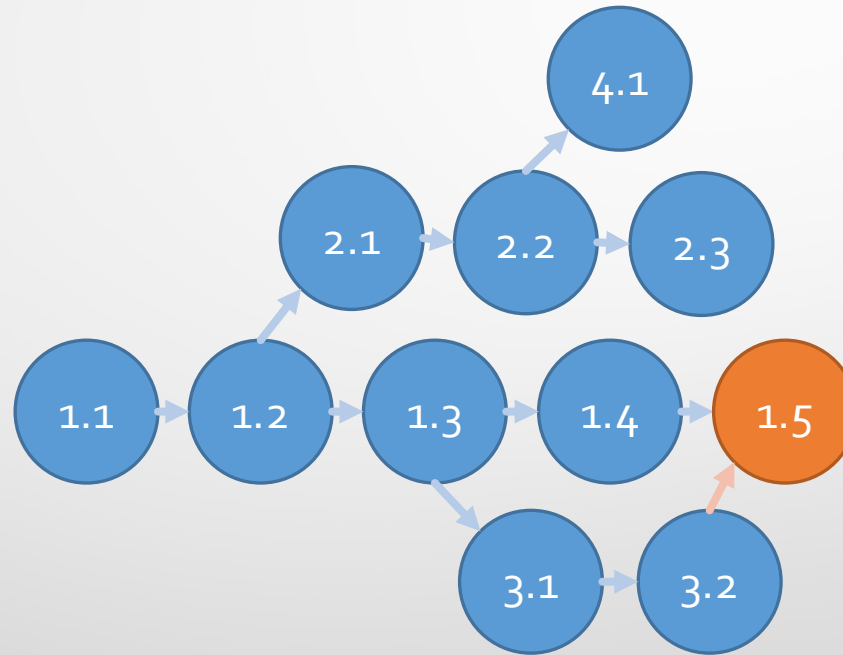


No Lineal



Control de versiones

Fusión de ramas



$$1.5 = 1.4 + (3.2 - 1.3) \text{ ó}$$

$$1.5 = 3.2 + (1.4 - 1.3)$$

pones el menos porque ya estaba antes
incluida en el camino

Control de versiones

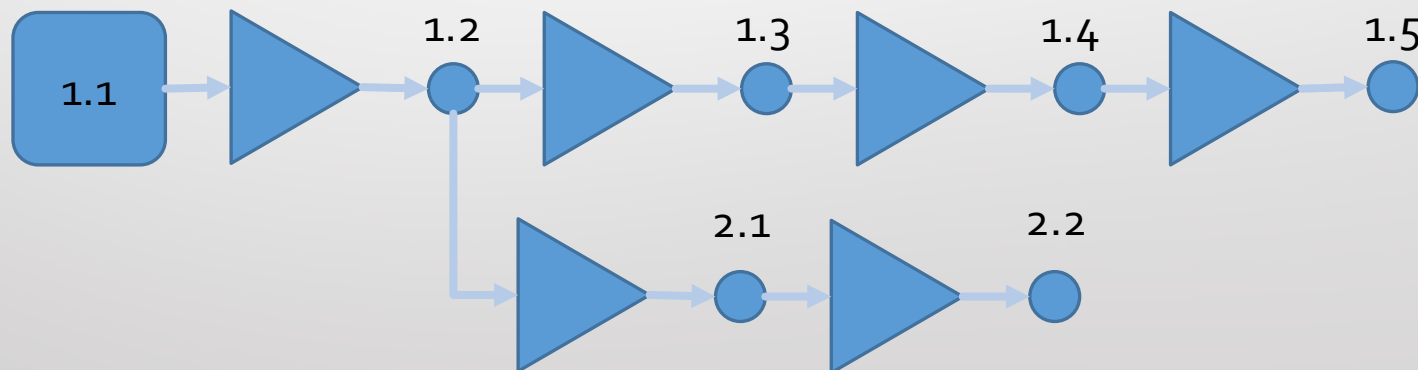
Almacenamiento de versiones

Delta directo:

- Se almacena primera versión
- Los cambios para construir versiones siguientes.

Sencillo e intuitivo

Coste elevado para recuperar últimas versiones



Control de versiones

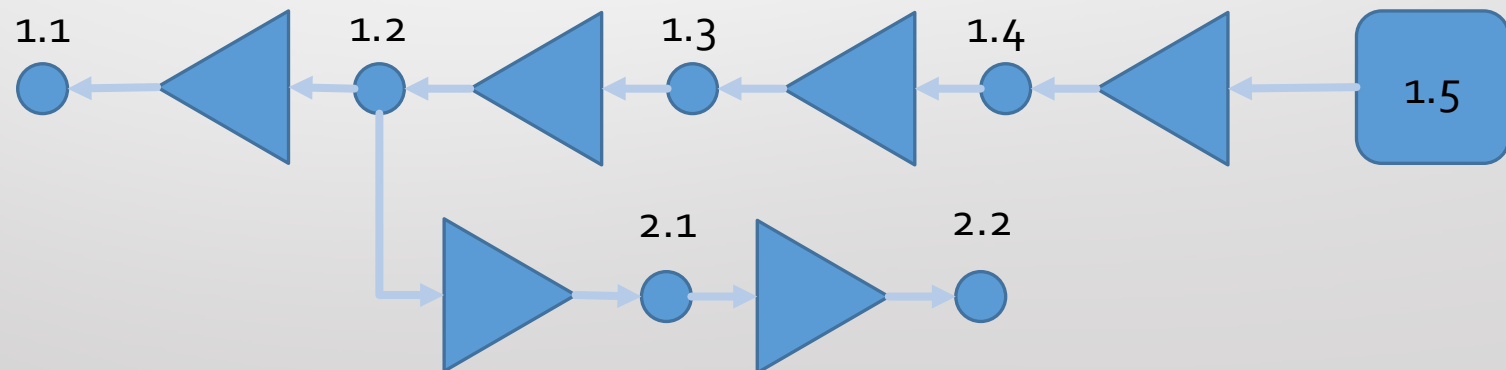
Almacenamiento de versiones

Delta indirecto:

- Se almacena última versión
- Los cambios para reproducir versiones anteriores
- En ramas delta directo

Menos costoso recuperar últimas versiones

En ramas el coste es más elevado que delta directo



Control de versiones

Almacenamiento de versiones

Marcado selectivo:

- Se almacena todas las versiones en un único fichero

Cuesta lo mismo recuperar cualquier versión

Cuanta más versiones existen más cuesta recuperar

<<1.1

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis ut placerat libero. Sed ut dolor pulvinar, congue libero et, consectetur lorem.

>>

<<1.3,1.4

Pellentesque porta vel risus ut laoreet.

>>

<<1.2,1.4

Vestibulum porta tempor imperdiet. Proin porta mattis quam, eu iaculis lorem dignissim eu.

>>

<<1.4

Phasellus metus sapien, iaculis quis felis ut, consectetur viverra nisi.

>>

<<1.1,1.2,1.3,1.4

Mauris id tellus nec velit placerat facilisis nec sed dolor. Phasellus rhoncus tempor interdum.

>>

Control de versiones

Diff

Compara ficheros línea a línea e informa de diferencias

Fichero1.txt

Hola,
Esto es un ejemplo.
Qué tal estás?
Muy bien,
Gracias

Fichero2.txt

Esto es un ejemplo.
Holaa,
Qué tal estás?
Muy bien.

diff Fichero1.txt Fichero2.txt

1d0
< Hola,
2a2
> Holaa,
4,5C4
< Muy bien,
< Gracias

> Muy bien.

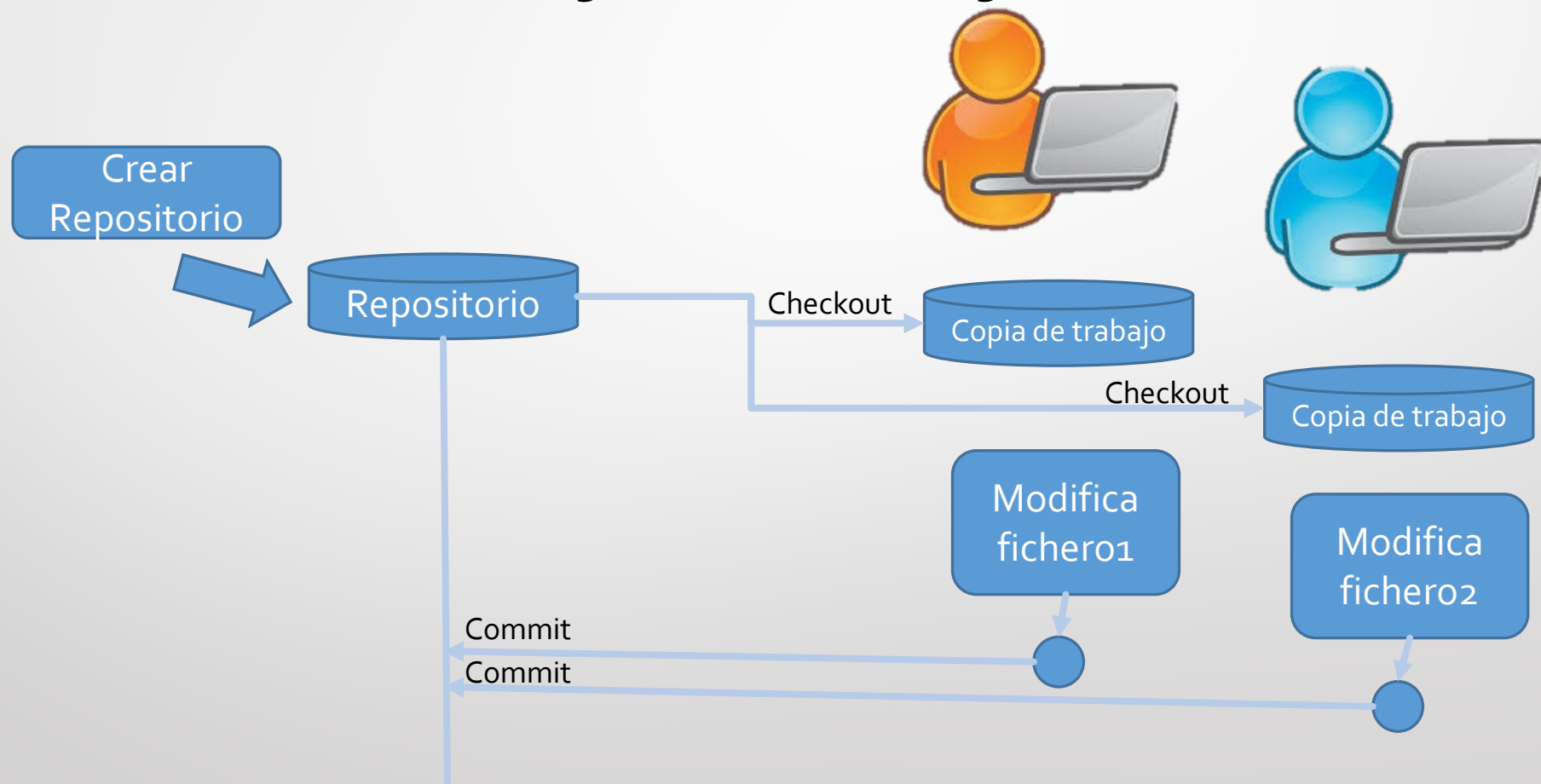
La línea 1 del primer fichero se borra y quedan sincronizados en línea 0

La línea 2 del primer fichero se añade a partir de la 2 del segundo

Las líneas 4 y 5 del primer fichero se sustituyen por la 4 del segundo

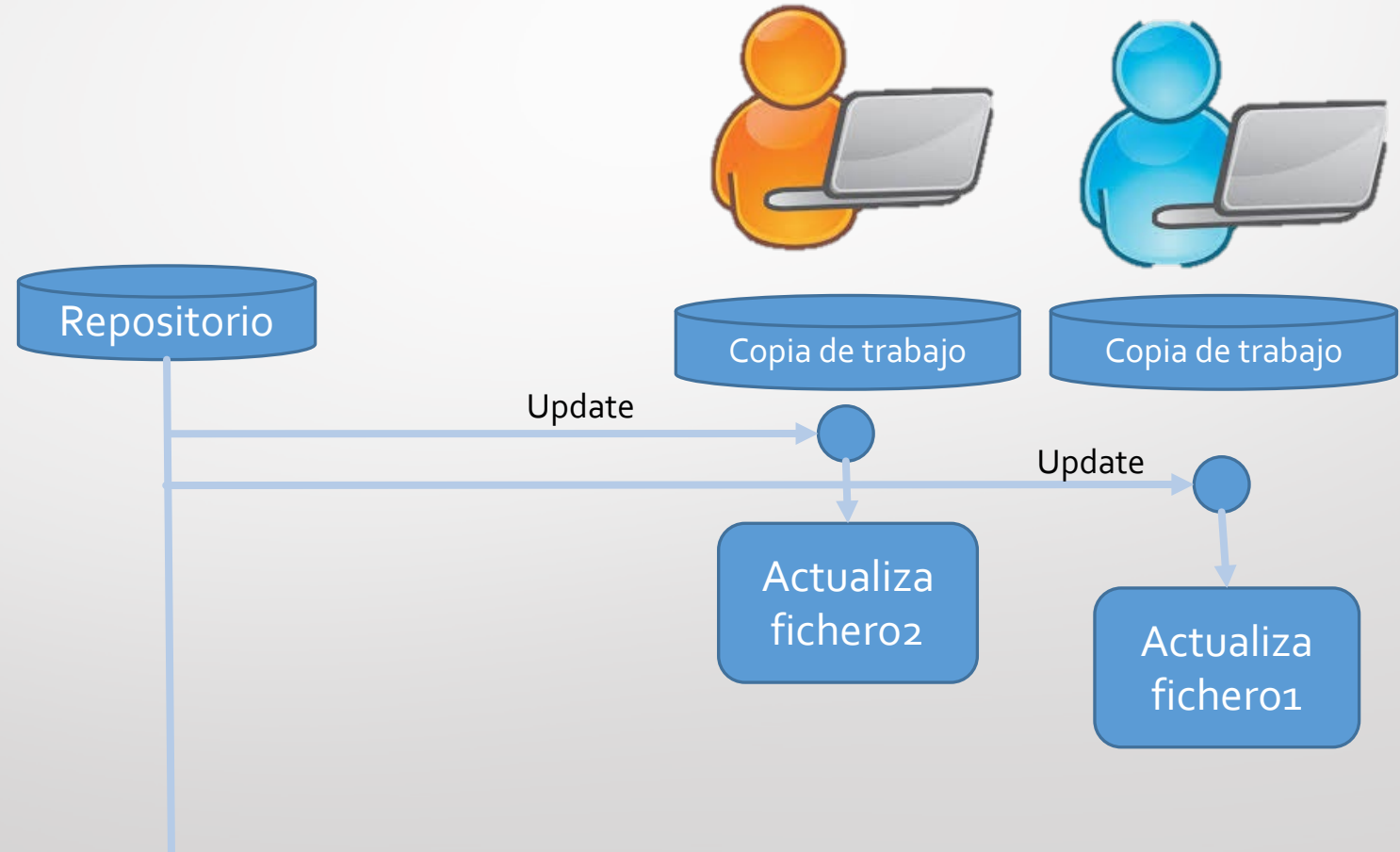
Control de versiones

Flujo de trabajo



Control de versiones

Flujo de trabajo



Control de versiones

Sistemas populares

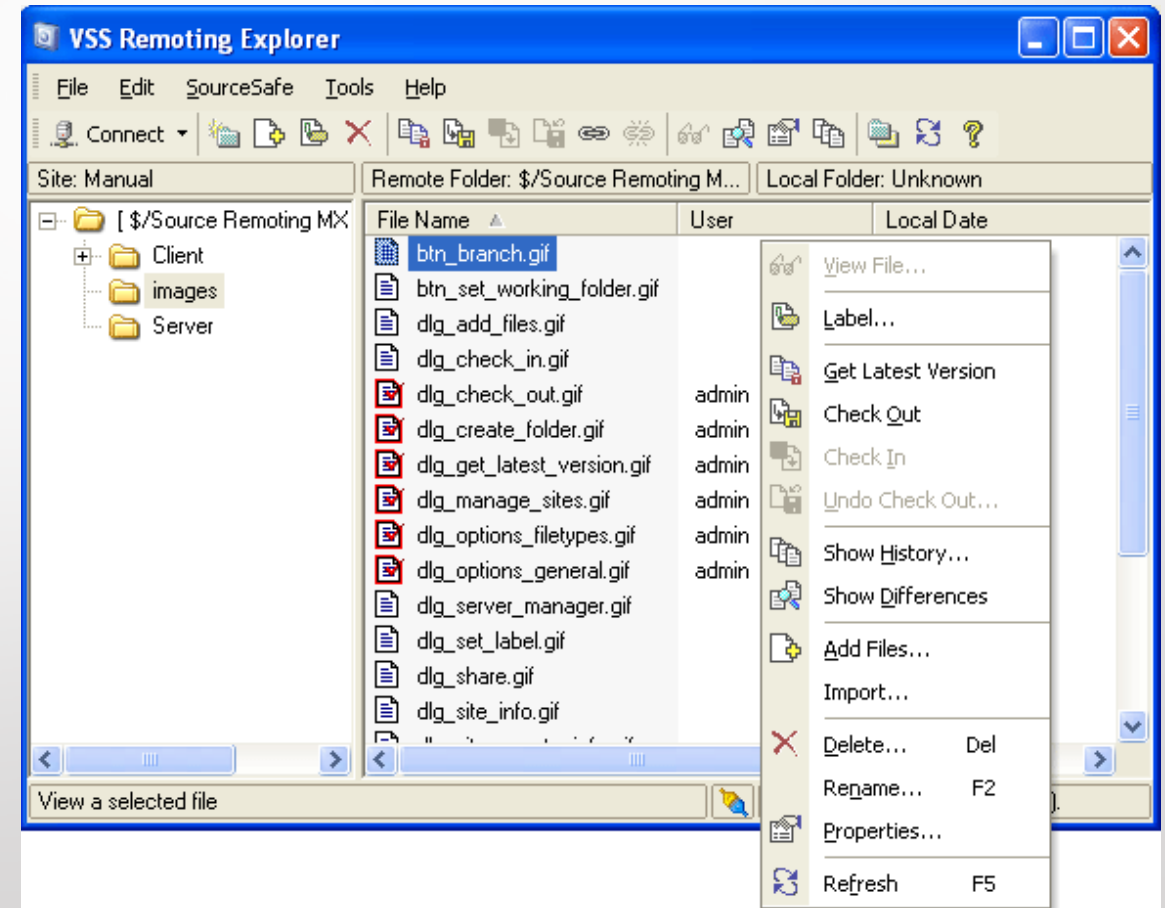
- Microsoft SourceSafe
- Concurrent Versions System (CVS)
- Subversion (SVN)
- Git
- Mercurial
- BitKeeper
- Dropbox, ...

Control de versiones

Sistemas populares

Microsoft SourceSafe

- 1994-2005
- Solución bloqueo-modificación-desbloqueo
- Muy integrado en herramientas de Microsoft



Control de versiones

Sistemas populares



- Apache Subversion (<http://subversion.apache.org>)
- Open Source (2000-actualidad)
- Copiar-modificar-fusionar
- Almacenamiento centralizado
- Línea de comandos

Control de versiones

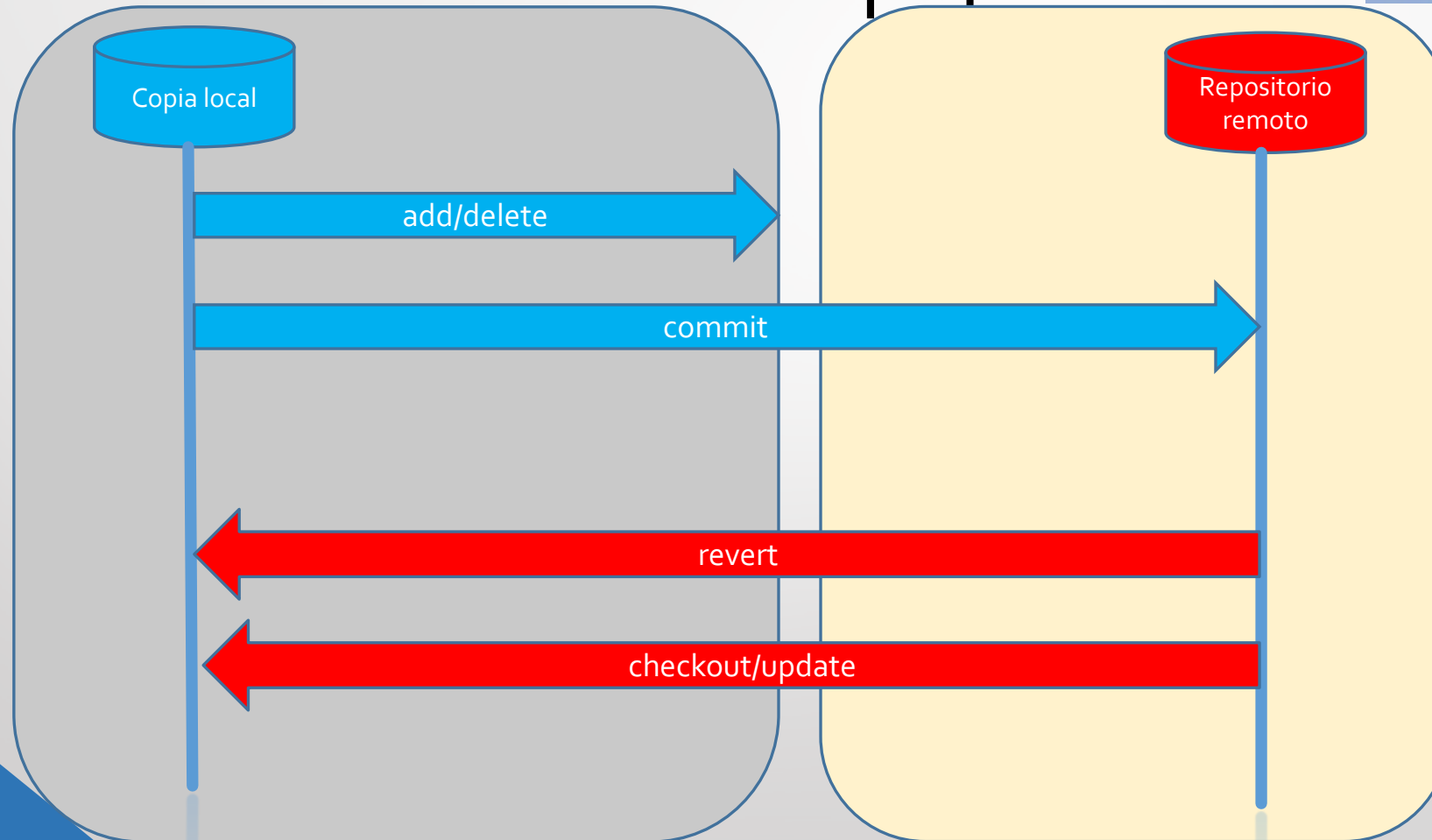
Sistemas populares



- Gran cantidad de herramientas lo integran
 - TortoiseSVN
 - Visual Studio
 - Xcode
 - ...

Control de versiones

Sistemas populares



Control de versiones

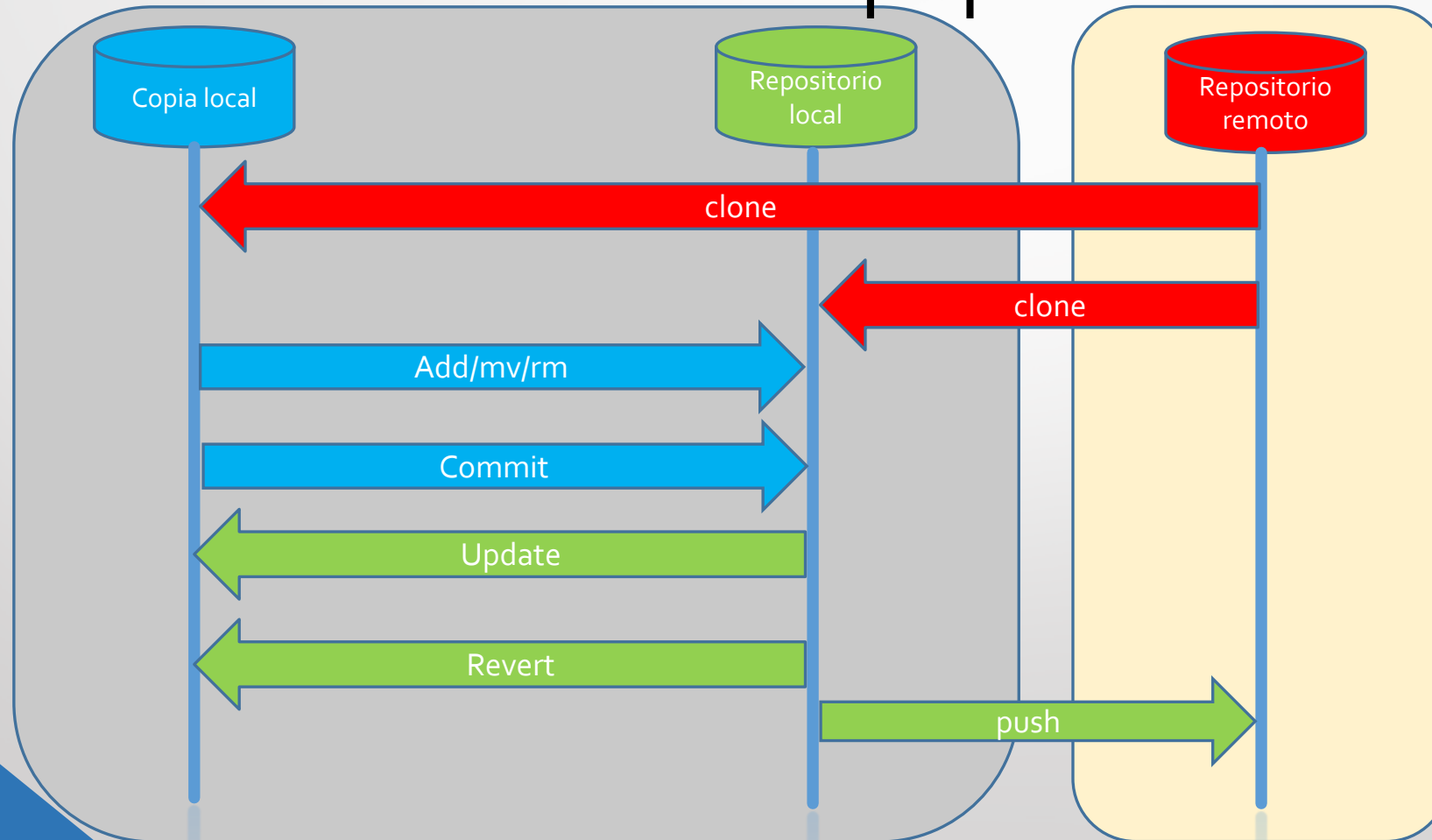
Sistemas populares



- Mercurial (<https://www.mercurial-scm.org/>)
- 2005-Actualidad
- Tras la retirada de versión gratuita de BitKeeper
- Distribuido
- Línea de comandos: Hg
- Comandos similares a Subversion +PULL+PUSH

Control de versiones

Sistemas populares



Control de versiones

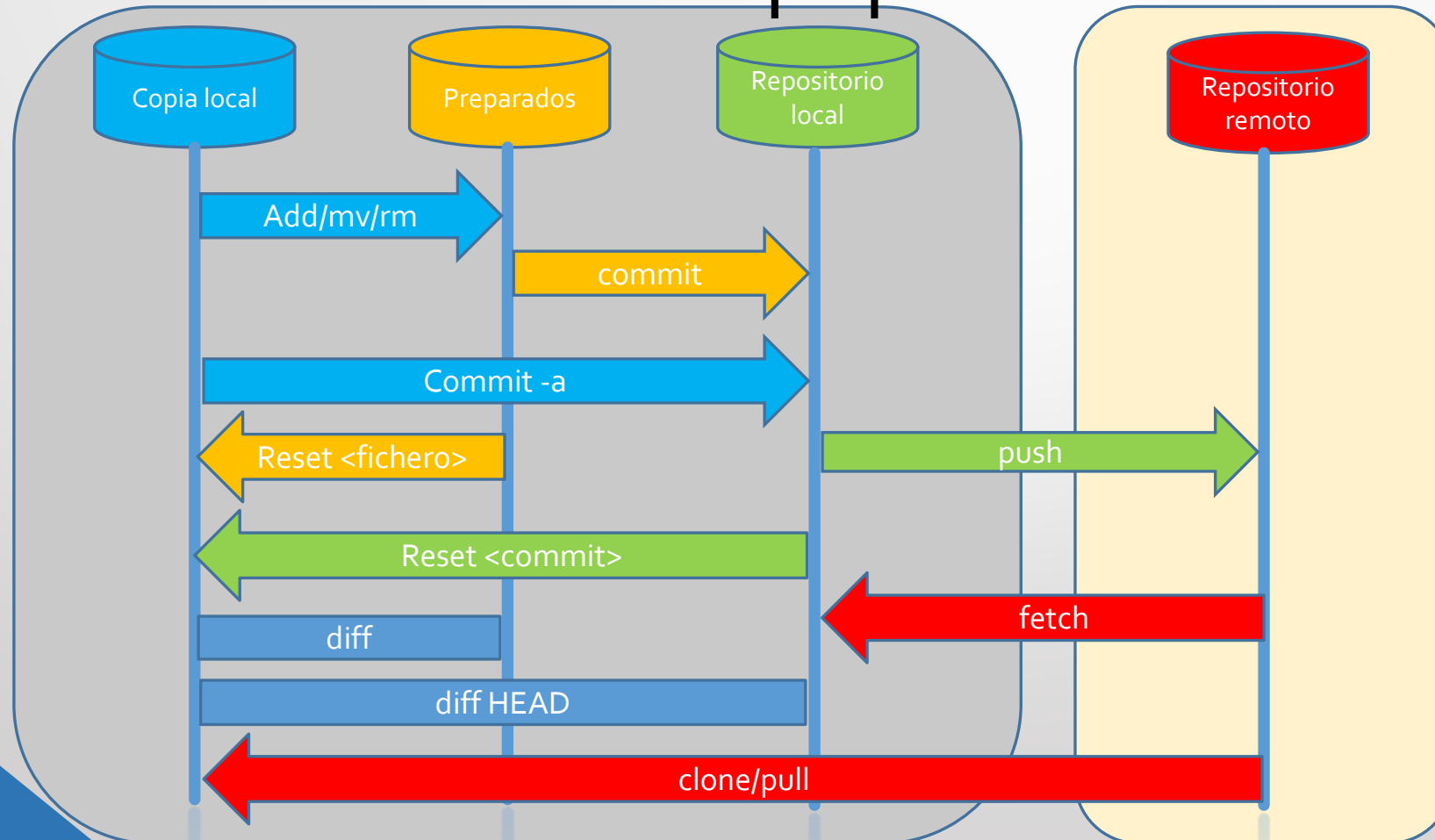
Sistemas populares



- Git (<https://git-scm.com/>)
- Linus Torvalds 2005
- OpenSource
- Nace a raíz de que BitKeeper pasa a ser de pago
- Kernel de linux
- Sistema distribuido
- Copiar-modificar-fusionar-subir
- Linea de comandos

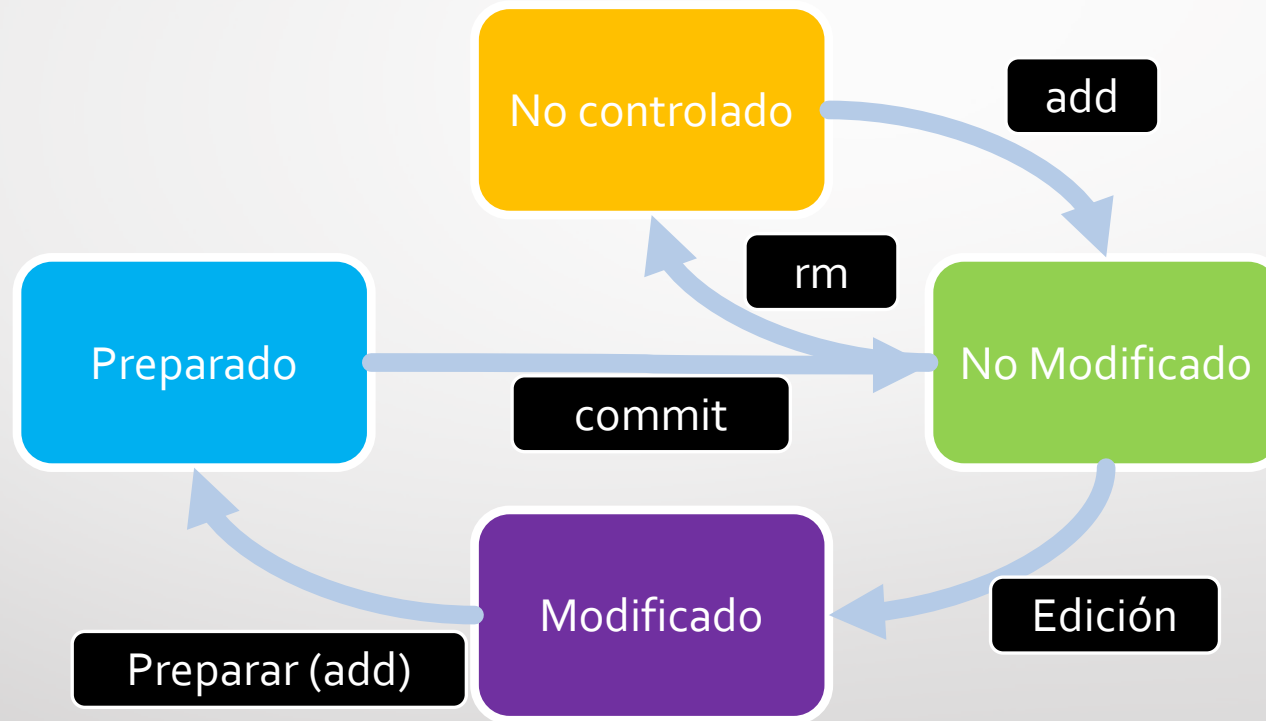
Control de versiones

Sistemas populares



Control de versiones

Sistemas populares



Control de versiones

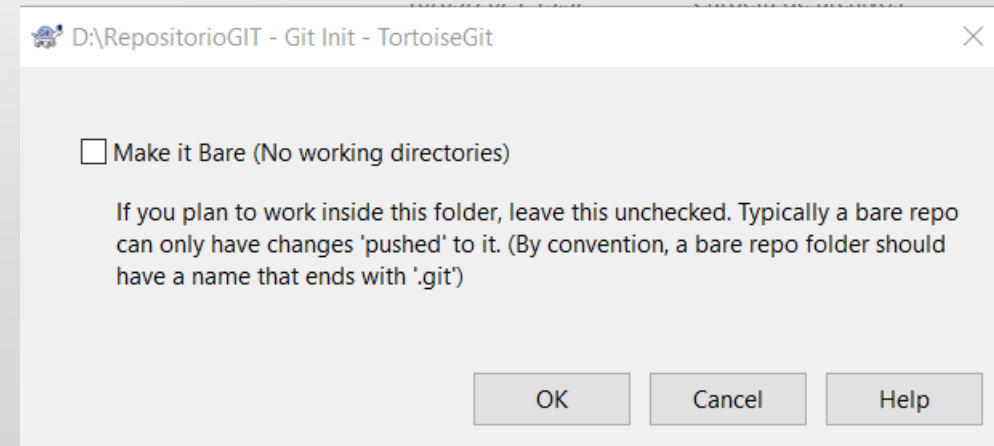
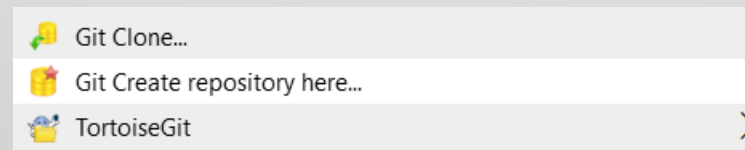


- Flujo de trabajo
 - Crear repositorio
 - Crear copia local/Importar proyecto existente a repositorio
 - Incluir ficheros en sistema de control de versiones
 - Trabajar sobre proyecto
 - Comprobar estado
 - (A veces...)Deshacer cambios
 - Actualizar versión local (resolver conflictos si es necesario)
 - Subir cambios a repositorio local
 - En ocasiones propagar cambios a repositorio remoto

Control de versiones



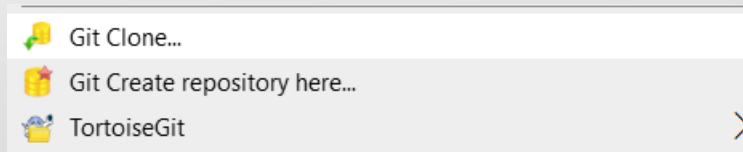
- Acompañado servidor apache, excepto uso individual
 - Creamos carpeta para contener repositorio (P.ej. RepositorioGIT)
 - Por línea de comandos:
 - `git init --bare` si el repositorio no va a contener copia de trabajo
 - `git init` si el repositorio además será copia de trabajo
 - Mediante TortoiseGIT:
 - Menú contextual sobre la carpeta RepositorioGIT
 - Git Create repository here



Control de versiones



- Crear carpeta de la copia local (P.Ej proyecto1)
- Por línea de comandos:
 - `git clone <url repositorio> <path local>`
 - `git clone file:///c:/RepositorioGIT proyecto1`
- Mediante TortoiseGIT:
 - Menú contextual sobre proyecto1
 - `git clone`

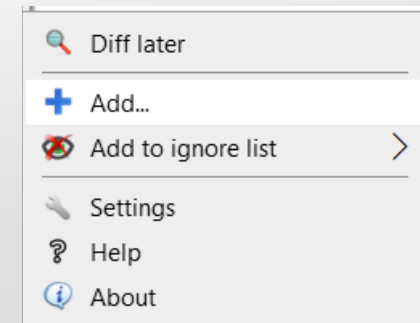


Control de versiones

Incluir ficheros en el SCV



- Por línea de comandos:
 - `git add <path fichero/s>`
 - **ATENCIÓN:** Esto **NO** sube nada al repositorio. Sólo queda marcado para el siguiente commit
- Mediante TortoiseGIT:
 - Menú contextual sobre el fichero a incluir
 - TortoiseGIT ->Add
- NOTA: Existen comandos similares para **delete**, **copy** y **move**



Control de versiones

Trabajar sobre el proyecto



- Trabajaremos con las herramientas de cada proyecto:
 - Word, Notepad, Photoshop, Unity, etc...

Control de versiones

Subir cambios al repositorio local



- Por línea de comandos:
 - `git commit path -m <descripción del commit>`
- Mediante TortoiseGIT:
 - Menú contextual sobre la carpeta a subir (puede ser la raíz o no)
 - GIT Commit -> <rama>

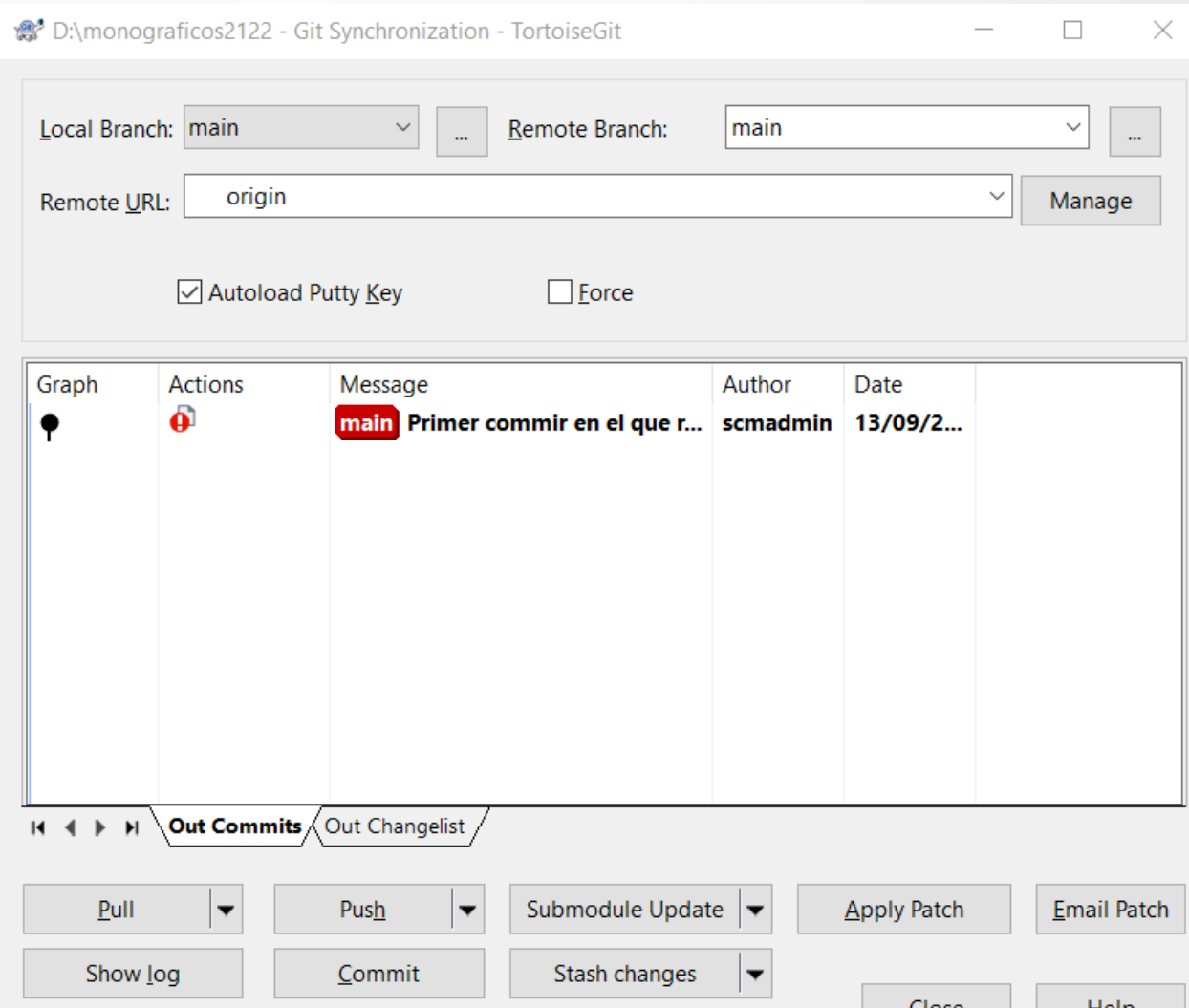


Control de versiones

Subir cambios al repositorio remoto



- Por línea de comandos:
 - git push
- Mediante TortoiseGIT:
 - Menú contextual sobre la carpeta a subir
 - GIT Sync

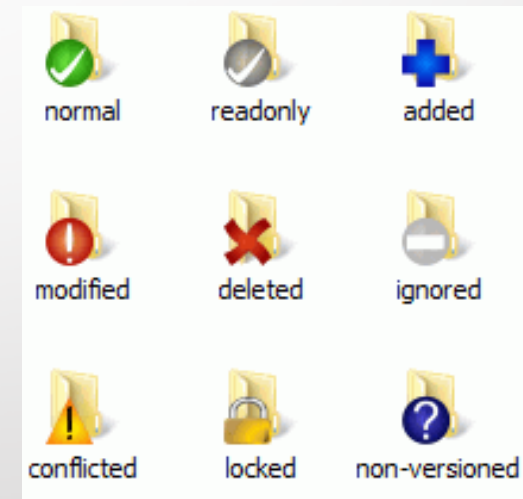
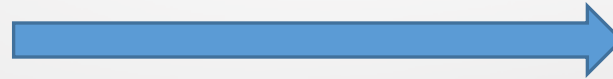


Control de versiones

Comprobar estado



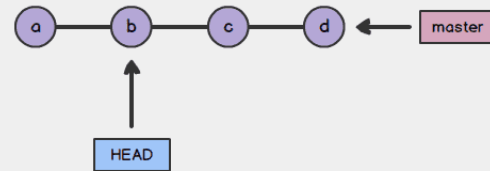
- Por línea de comandos:
 - git status
- Mediante TortoiseGIT:





Control de versiones Deshacer cambios

git checkout, git revert y git reset

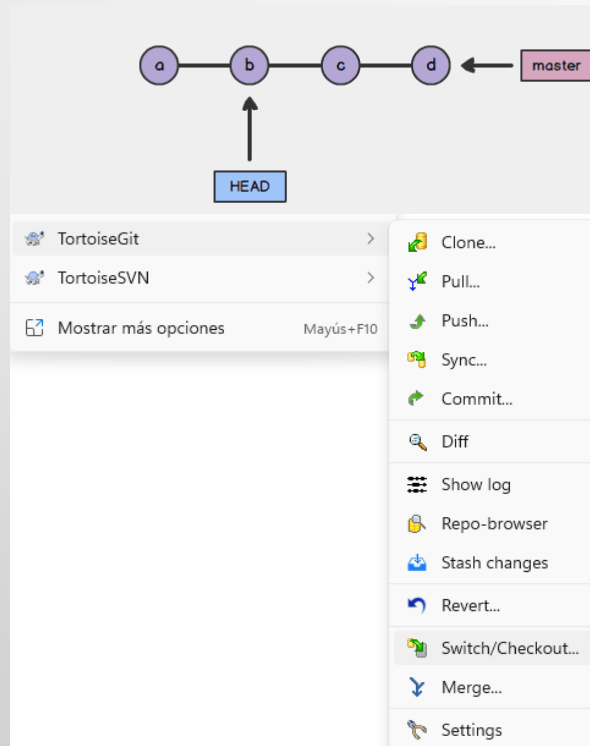


git checkout b

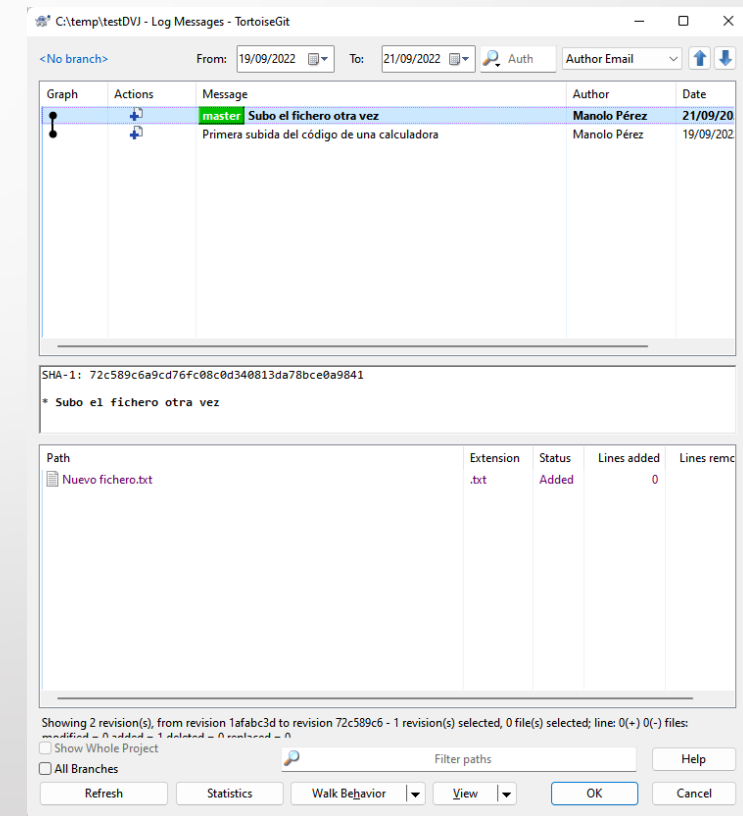
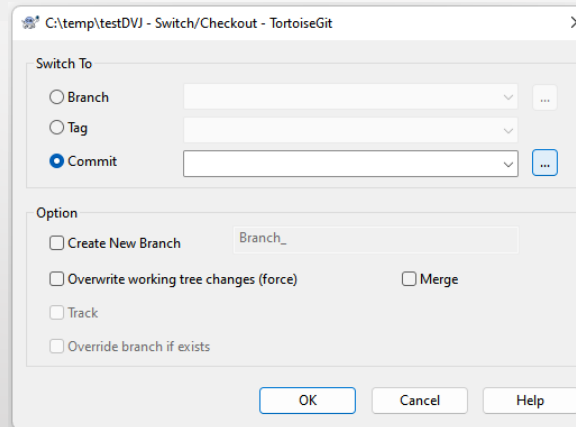
Control de versiones

Deshacer cambios

git checkout, git revert y git reset



git checkout b



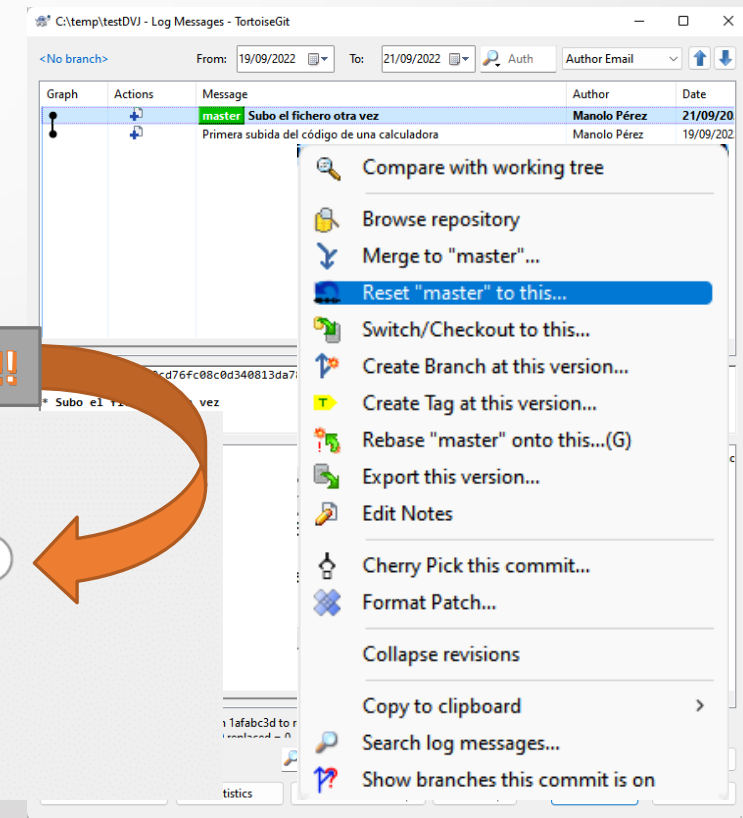
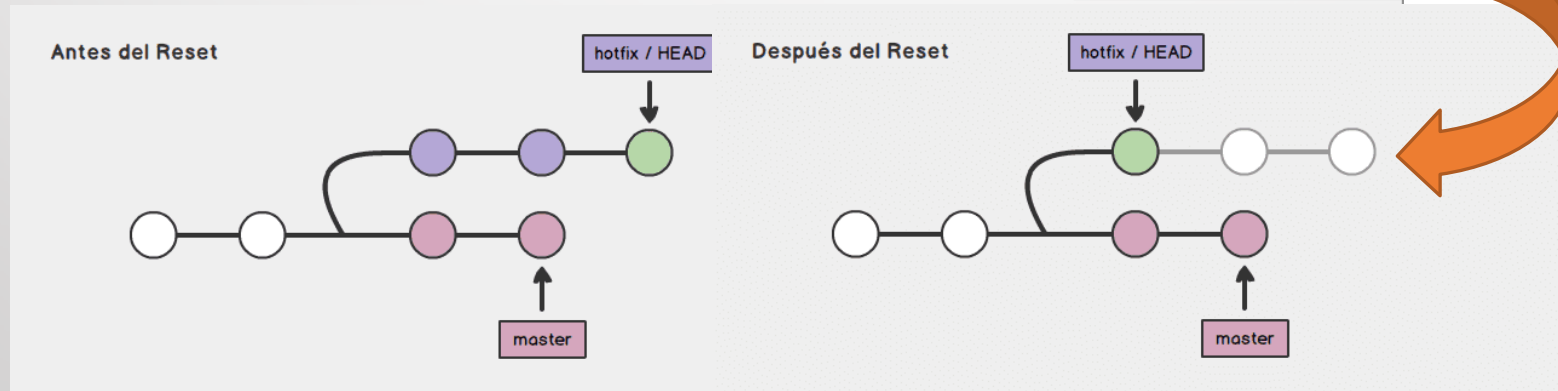
Control de versiones

Deshacer cambios



git checkout, git reset y git revert

git reset HEAD~2



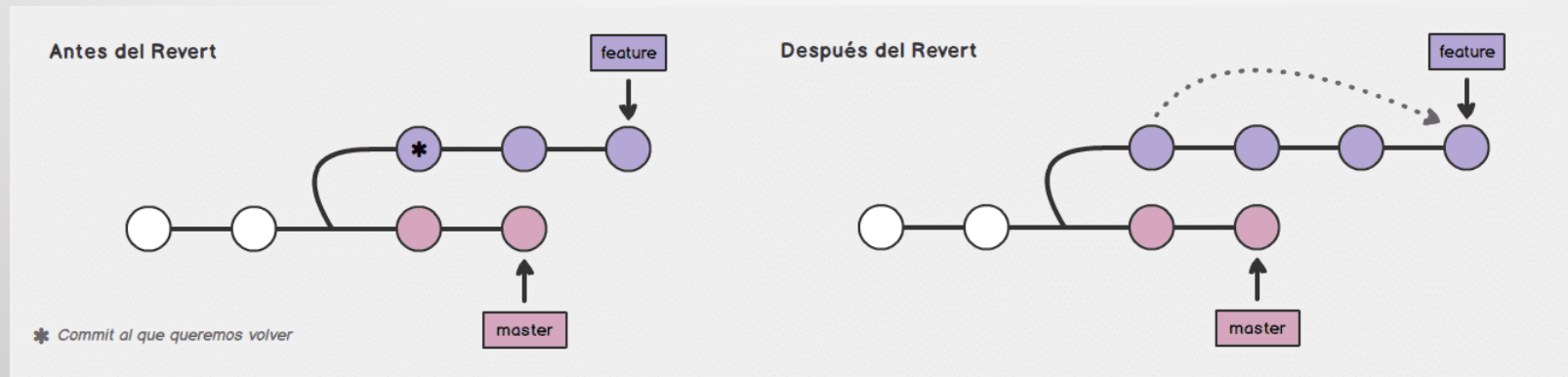


Control de versiones

Deshacer cambios

git checkout, git revert y git reset

git revert HEAD~2

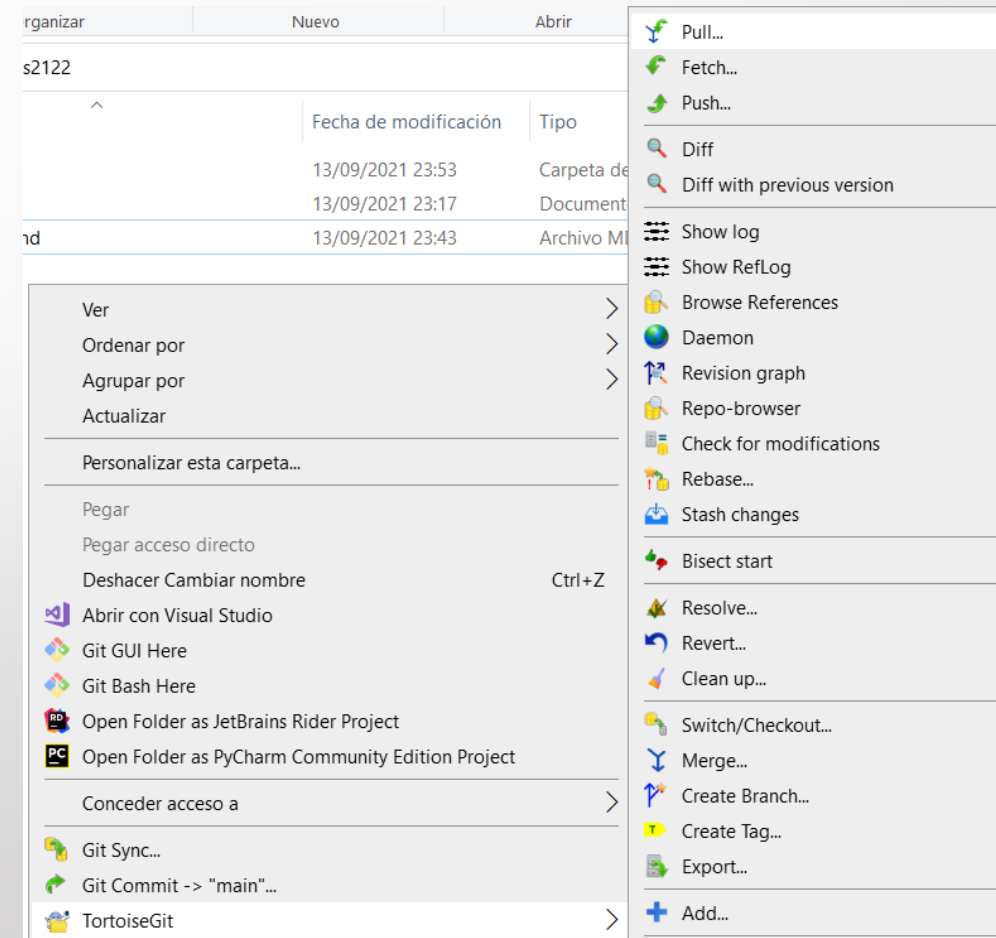


Control de versiones

Actualizar versión local con el contenido de un repositorio remoto



- Por línea de comandos:
 - `git pull`
- Mediante TortoiseGIT:
 - Menú contextual el objeto (fichero o carpeta) sobre el que queremos actualizar





Control de versiones

Resolver conflictos

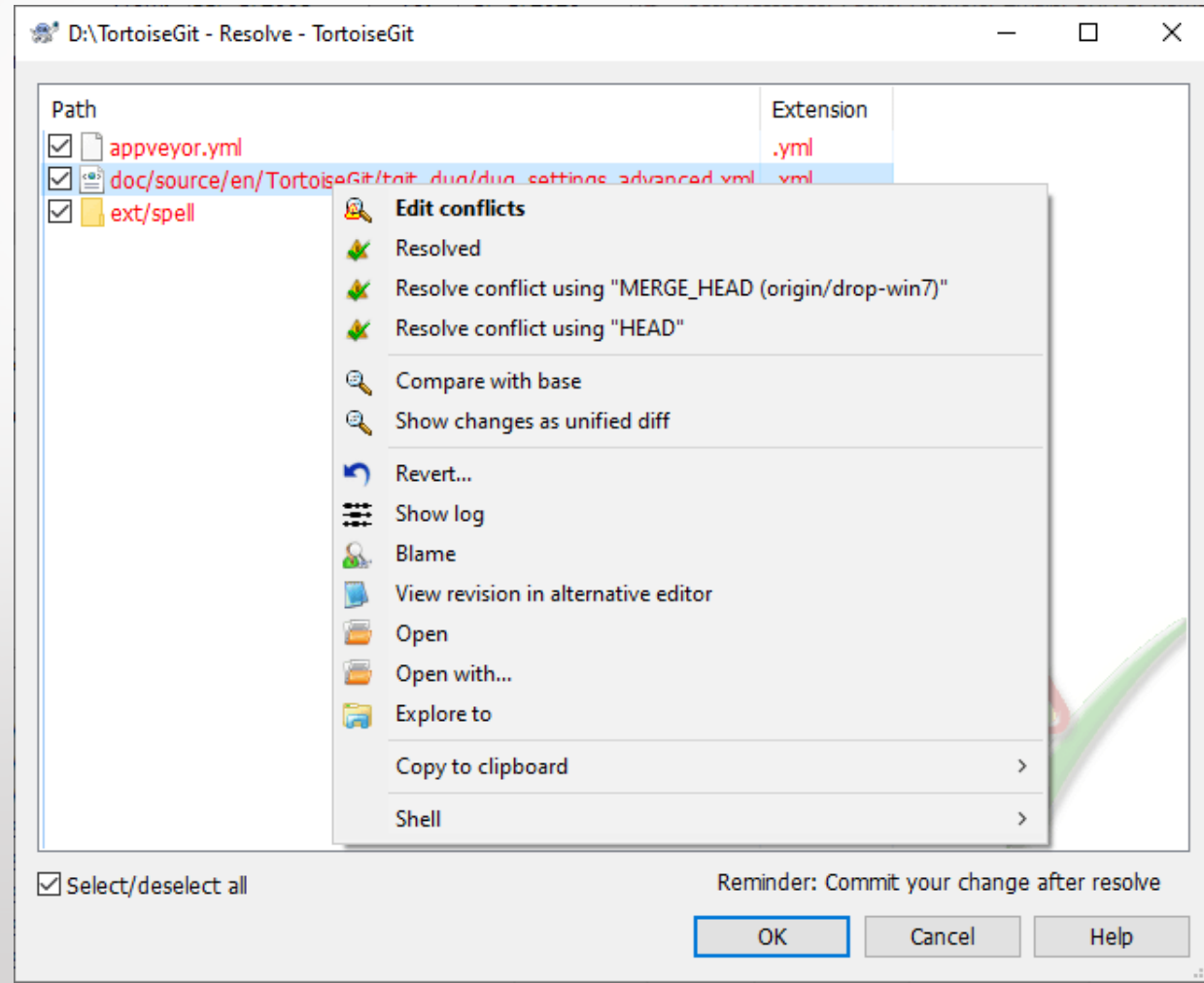
- Por línea de comandos:
 - Durante el pull pueden aparecer conflictos. Al aparecer tenemos varias opciones:
 - Select: (p) postpone, (df) diff-full, (e) edit, (mc) mine-conflict, (tc) theirs-conflict, (s) show all options: s
 - (e) edit – Abrir en editor
 - (df) diff-full – Muestra las diferencias en formato diff
 - (r) resolved – Marca como resuelto (normalmente se hace después de resolver los conflictos con un editor)
 - (dc) display-conflict – muestra los conflictos
 - (mc) mine-conflict – En caso de conflictos, el código que queda es el local
 - (tc) theirs-conflict – En caso de conflictos, el código que queda es el del repositorio
 - (mf) mine-full – El fichero local completo se toma como válido
 - (tf) theirs-full – El fichero del repositorio completo se toma como válido
 - (p) postpone – El conflicto se resolverá más tarde
 - (l) launch – Lanza herramienta externa para resolver conflicto
 - (s) show all – Muestra la lista anterior

Control de versiones

Resolver conflictos



- Mediante TortoiseGIT:
 - Durante el update pueden aparecer conflictos. Estos aparecerán en rojo en la ventana de diálogo del update

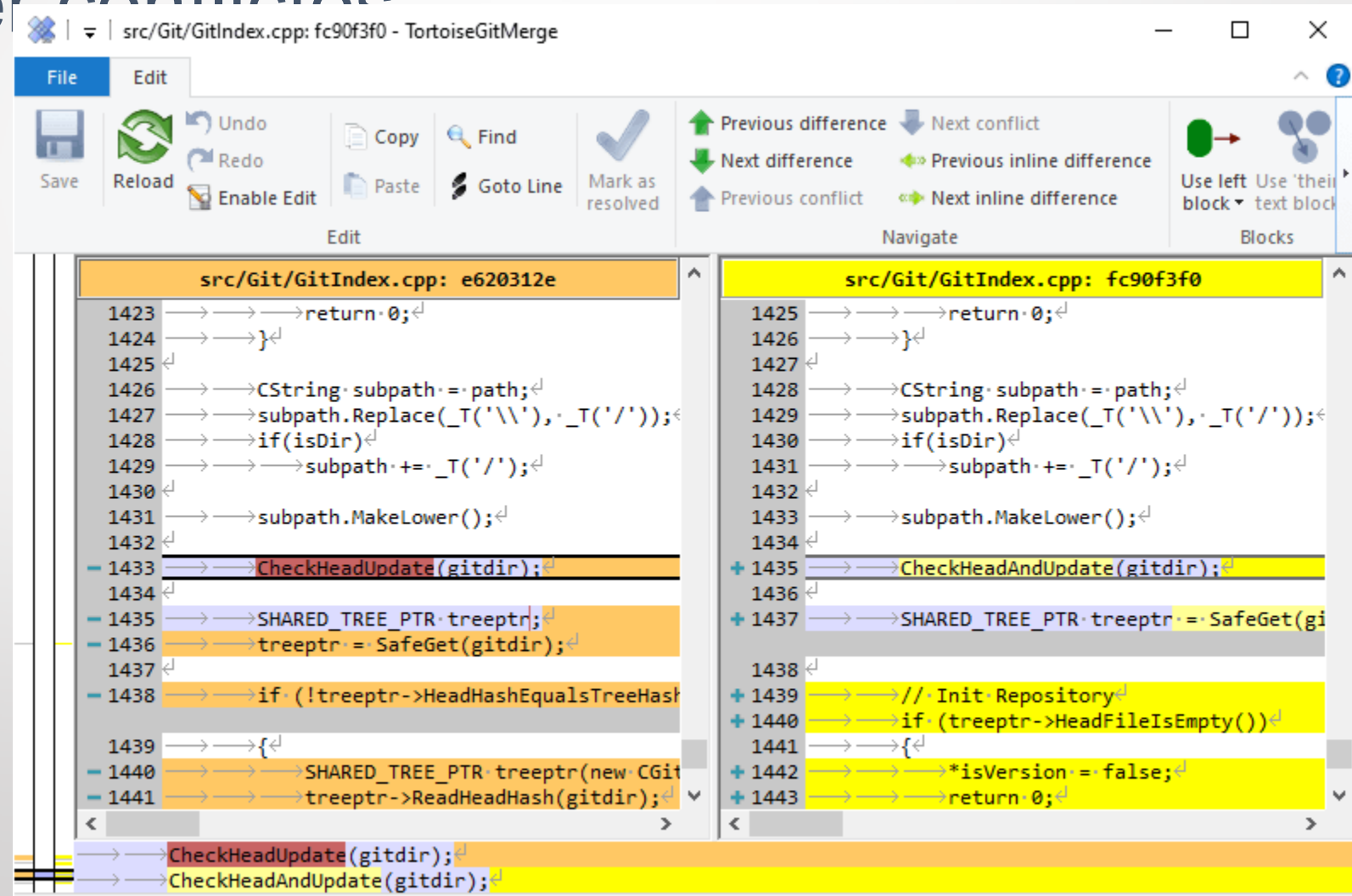


Control de versiones

Resolver conflictos



- TortoiseMerge:

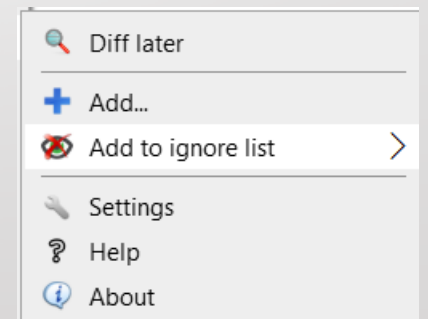


Control de versiones

Ignore List



- Generalmente hay ficheros que no queremos versionar:
 - Ejecutables
 - Compilados
 - Temporales
 - ...
- GIT permite marcarlos mediante listas de ignorados.



Control de versiones

Ignore List



- Es un sistema recursivo. Conforme encuentre ficheros .gitignore irá incluyendo más recursos que no se subirán al repositorio.
- En Unity utilizaremos el siguiente fichero:
 - [gitignore/Unity.gitignore at master · github/gitignore](https://github.com/gitignore/Unity.gitignore)

Control de versiones

Ayuda



- [Git - Book \(git-scm.com\)](https://git-scm.com/book)
- [TortoiseGit – Documentation – Windows Shell Interface to Git](#)
- [NDP Software :: Git Cheatsheet](#)
- [LearningGitBranching](#)

Control de versiones

Ejercicios



- Descargar el repositorio varios.
 - Crea un fichero txt con tu DNI como nombre de éste.
 - Dentro del fichero escribe tu nombre y fecha de nacimiento.
 - Sube este fichero al repositorio de diversion.
 - Consigue todos los ficheros del resto de tus compañeros.

Control de versiones

Ejercicios



- En el mismo repositorio y por parejas:.
 - Cread una carpeta con el nombre de los dos usuarios concatenados
 - Añadid al control de versiones un fichero logo.jpg (cada alumno de manera individual seleccionará una imagen como logo de su ejercicio) y copiad dentro de vuestra carpeta el fichero cuestiones.txt que se encuentra en la raíz del repositorio.
 - Responded de manera individual a las preguntas y llegad a una versión final del fichero con las respuestas correctas para cada pareja.
 - Llegad también a un acuerdo en cuanto al logo de la pareja

Control de versiones

Ejercicios



- Escribir en HTML5 la lista de clase ordenada alfabéticamente.
 - Cada alumno debe escribir su nombre en la posición adecuada.