

Índice de contenidos

- 1 Sobre Git
 - Conceptos generales
 - Características
- 2 Operaciones básicas
 - Inicialización del repositorio
 - Guardando cambios
 - Ignorar archivos
 - Manejo de conflictos
 - Deshacer cambios
- 3 Ramas
 - Conceptos generales
 - Operaciones con ramas

Git

Definición

Git es un sistema de control de versiones gratuito y open source. Un sistema de control de versiones es un sistema que guarda los cambios de un conjunto de archivos en el tiempo con el fin de que quede registro de las versiones específicas generadas.

Otros sistemas de control de versiones

- Apache subversion (SVN)
- Mercurial

Índice de contenidos

1 Sobre Git

- Conceptos generales
- Características

2 Operaciones básicas

- Inicialización del repositorio
- Guardando cambios
- Ignorar archivos
- Manejo de conflictos
- Deshacer cambios

3 Ramas

- Conceptos generales
- Operaciones con ramas

Características

- **Snapshots en vez de diferencias** : Otros sistemas, como SVN, guarda la información como archivos y sus cambios respectivos. En cambio Git, cuando se modifica un archivo se genera un snapshot del archivo completo.

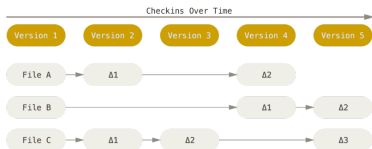


Figure: Diferencias

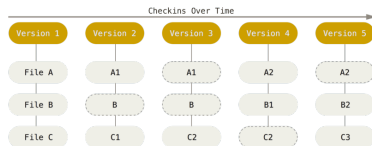


Figure: Snapshots

Características

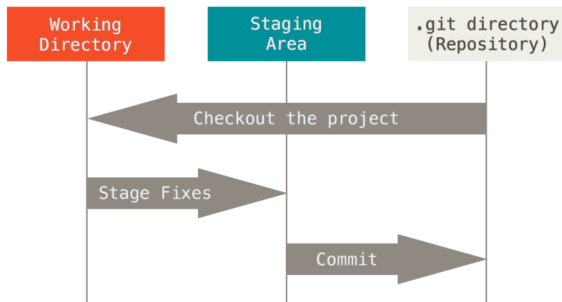
- **Tres estados para un archivo**

- ① Modificado (modified)
- ② Preparado (staged)
- ③ Confirmado (committed)

El flujo básico de git es el siguiente:

- ① Se modifican los archivos del directorio de trabajo, quedando en estado **modified**
- ② Se preparan los archivos, añadiéndolos a las **staging area**, quedando en estado **staged**
- ③ Se confirman los cambios, almacenando el snapshot permanentemente en repositorio git local. Estado **committed**

Flujo básico



Índice de contenidos

- 1 Sobre Git
 - Conceptos generales
 - Características
- 2 Operaciones básicas
 - Inicialización del repositorio
 - Guardando cambios
 - Ignorar archivos
 - Manejo de conflictos
 - Deshacer cambios
- 3 Ramas
 - Conceptos generales
 - Operaciones con ramas

Operaciones de puesta en marcha del repositorio

Creación de un repositorio local

```
mkdir <dir name>  
cd <dir name>  
git init  
touch README.md  
git add README.md  
git commit -m "<commit_message>"
```

Conectar repositorio local con remoto

```
git remote add origin <dirección remota>  
git push -u origin master
```

Clonar repositorio

```
git clone <dirección remota> [<carpeta destino>]
```

Índice de contenidos

- 1 Sobre Git
 - Conceptos generales
 - Características
- 2 Operaciones básicas
 - Inicialización del repositorio
 - **Guardando cambios**
 - Ignorar archivos
 - Manejo de conflictos
 - Deshacer cambios
- 3 Ramas
 - Conceptos generales
 - Operaciones con ramas

Guardando cambios en el repositorio

Obtener estado del repositorio

```
git status
```

Sincronizar repositorio local con repositorio remoto

```
git pull origin <rama>
```

En un principio sólo existe una única rama (master)

Preparar archivos (moverlos a la staging area)

```
git add <archivo o carpeta>
```

Hacer commit a los archivos de la staging area

```
git commit -m "<mensaje personalizado>"
```

Guardando cambios en el repositorio

Hacer commit a los archivos de la staging area

```
git commit -m "<mensaje_personalizado>"
```

Guardar cambios en repositorio remoto

```
git push origin <rama>
```

Flujo general

- Sincronizar repositorio local con remoto
- Modificar y/o añadir archivos
- Mover los archivos modificados y/o añadidos a la staging area
- Hacer commit
- Guardar cambios en el repositorio remoto

Índice de contenidos

- 1 Sobre Git
 - Conceptos generales
 - Características
- 2 Operaciones básicas
 - Inicialización del repositorio
 - Guardando cambios
 - Ignorar archivos
 - Manejo de conflictos
 - Deshacer cambios
- 3 Ramas
 - Conceptos generales
 - Operaciones con ramas

Ignorar archivos

Para ignorar archivos (no tenerlos bajo control de versión) debe crearse un archivo llamado **.gitignore** donde se especifiquen las reglas utilizadas para ignorar elementos.

Ejemplo de archivo .gitignore

```
# Logs and databases #
#####
*.log
*.sql
*.sqlite

# OS generated files #
#####
.DS_Store
.DS_Store?
.*
._
.Spotlight-V100
.Trashes
ehthumbs.db
Thumbs.db
```

1 Sobre Git

- Conceptos generales
- Características

2 Operaciones básicas

- Inicialización del repositorio
- Guardando cambios
- Ignorar archivos
- Manejo de conflictos
- Deshacer cambios

3 Ramas

- Conceptos generales
- Operaciones con ramas

Manejo de conflictos

- Cuando 2 o más personas trabajan concurrentemente en el mismo repositorio, es común que se produzcan conflictos si han modificado el mismo archivo
- Al sincronizar con el repositorio remoto, git es capaz de detectar inconsistencias con el repositorio local, haciéndoselo notar al usuario en el mismo archivo donde esto se produce

Ejemplo de archivo con conflicto

Contenido común sin conflictos

<<<<<< HEAD

Contenido presente en la rama local

=====
Contenido presente en la rama remota

>>>>>> f3cea776f66f61b22b0296e8b0c62399cc373e27

Contenido común sin conflictos

Índice de contenidos

- 1 Sobre Git
 - Conceptos generales
 - Características
- 2 Operaciones básicas
 - Inicialización del repositorio
 - Guardando cambios
 - Ignorar archivos
 - Manejo de conflictos
 - Deshacer cambios
- 3 Ramas
 - Conceptos generales
 - Operaciones con ramas

Deshacer cambios

Rehacer commit incompleto

```
git commit —amend
```

Sacar archivos de la staging area

```
git reset HEAD <archivo>
```

Volver archivo a su estado inicial (antes de ser modificado)

```
git checkout — <archivo>
```

Índice de contenidos

1 Sobre Git

- Conceptos generales
- Características

2 Operaciones básicas

- Inicialización del repositorio
- Guardando cambios
- Ignorar archivos
- Manejo de conflictos
- Deshacer cambios

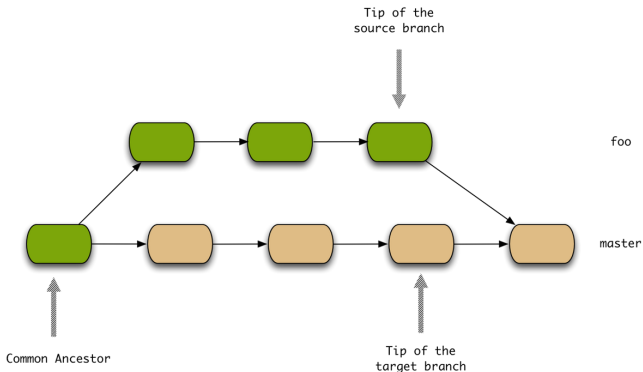
3 Ramas

- Conceptos generales
- Operaciones con ramas

Ramas (Branches)

- Crear una rama significa que a partir de un punto en el tiempo el trabajo diverge de la rama principal (master) y se continúa sin modificarla (y potencialmente introducir errores).
- Esta estrategia es particularmente útil en ambientes reales de desarrollo, puesto que es posible dejar la versión estable de un software en la rama master e introducir cambios a partir de nuevas ramas (se abordará con más detalle más adelante en el curso).

Ramas(Branches)



Índice de contenidos

1 Sobre Git

- Conceptos generales
- Características

2 Operaciones básicas

- Inicialización del repositorio
- Guardando cambios
- Ignorar archivos
- Manejo de conflictos
- Deshacer cambios

3 Ramas

- Conceptos generales
- Operaciones con ramas

Operaciones con ramas

Crear rama a partir de la actual

```
git branch <nueva rama>
```

Listar ramas

```
git show-branch --list
```

Posicionarse en una rama específica

```
git checkout <rama>
```

Obtener del repositorio remoto posibles nuevas ramas

```
git fetch
```


Operaciones con ramas

Si se quieren incluir cambios de una rama en otra, se debe seguir el siguiente flujo.

- Posicionarse en la rama destino.
- Ejecutar un comando **git merge** para recibir los cambios de la otra rama.
- Resolver conflictos si los hubiese.

Ejemplo: Recibir cambios de rama "branch-1" hacia rama "master"

```
git checkout master  
git merge branch-1
```