

Introducción a Git

Qué es Git

- Es un sistema de control de versiones.
- Mantiene un repositorio local y uno o varios remotos con los que podemos sincronizarnos.
- Permite gestionar y fusionar ramas y versiones diferentes de los mismos archivos.
- Permite a grandes equipos de desarrollo trabajar de forma coordinada sin perder información.
- ¡No confundir con GitHub!
 - *GitHub es un servicio de repositorio Git remoto donde podemos alojar nuestros proyectos.*

Instalación en local

- **En Windows:** descargando y ejecutando el instalador oficial de <http://msysgit.github.com/>
- **En Linux:** desde los repositorios de tu distribución o instalando el paquete desde <http://git-scm.com>
- **En Mac:** descargando y ejecutando el instalador oficial de <http://sourceforge.net/projects/git-osx-installer/>

Inicialización (solo la primera vez)

```
git init
```

```
git --config name "Mi nombre"
```

```
git --config email "Mi email"
```

```
git --config list
```

Flujo de trabajo habitual en local

Pasar ficheros a la stage area (INDEX)

```
git add <file>  
git add -A
```

Pasar de INDEX a HEAD (repo local) todos los ficheros

```
git commit -m "Comentario"
```

Ver estado del repositorio y el historial de cambios

```
git status  
git log  
git log --oneline
```

Configurar repositorios remotos

Crear una copia local de un repositorio remoto:

```
git clone <url-repo-remoto>
```

Configurar repositorios remotos

Muestra repositorios remotos configurados

```
git remote
```

Añade repositorio remoto

```
git remote add <nombre> <url-del-repositorio>
```

Por ejemplo, un repositorio de github se añade así:

```
git remote add mi-repo  
git://github.com/pepitoperez/mi-proyecto
```

Enviar a repositorio remoto

```
git push <nombre-repo-remoto> <rama>
```

<rama> será “master” a menos que hayamos creado ramas en nuestro proyecto.

Traer desde repositorio remoto

Trae todos los ficheros desde el repositorio remoto y los deja de momento en otra rama

```
git fetch <nombre-repo-remoto>
```

Fusiona los ficheros que hemos traído con la rama principal

```
git merge FETCH_HEAD
```

Los dos pasos anteriores se hacen a la vez así:

```
git pull
```

Flujo de trabajo habitual en local/remoto



Trabajar con ramas (1/2)

Ramificar un repositorio permite trabajar con versiones diferentes del código hasta que se decida unificarlas.

1. Crear una rama y saltar a ella

```
git branch <nombre-rama>  
git checkout <nombre-rama>
```

Los dos pasos anteriores se hacen a la vez así:

```
git checkout -b <nombre-rama>
```

Trabajar con ramas (2/2)

2. En cualquier momento podemos volver a trabajar sobre la rama original (p. ej: master) y el código de las dos ramas no se mezclará.

```
git checkout master
```

3. Cuando hayamos terminado de trabajar en la rama, podemos unirla al master así:

```
git checkout master  
git merge <nombre-rama>
```

4. Es conveniente borrar la rama cuando hayamos terminado de trabajar con ella.

```
git branch -d <nombre-rama>
```

¿Problemas al fusionar ramas?

Si se ha modificado el mismo archivo fuente en dos ramas distintas, git no fusionará el archivo automáticamente, sino que lo marcará como "sin fusionar" para que lo hagas tú a mano.

Puedes ver qué archivos no se han fusionado así:

```
git status
```

Al editar los archivos sin fusionar, verás que git te indica qué zonas del código tienen problemas. Por ejemplo:

```
<<<<<< HEAD:index.html
<div id="footer">contact : email.support@github.com</div>
=====
<div id="footer">
  please contact us at support@github.com
</div>
>>>>>> mi-rama:index.html
```

Este es el
código de la rama
master

Y este es el de la
rama <mi-rama>

Interfaces gráficas

- **gitk y git-gui:** Muy básicas, distribuidas con git.
- **GitHub Desktop:** cliente gráfico de GitHub, solo disponible para Windows.
- **GitKraken:** muy popular, propietario, completo y multisistema.
- **SmartGit:** propietario, completo y multisistema.
- **git-cola:** ligero, libre, completo y multisistema.
- **La que venga con tu IDE preferido.** Los principales IDE tiene soporte para Git de forma nativa o a través de plugins.

Más info:

<https://git-scm.com/book/es/v2>