

## Crear y compartir repositorios privados en Github

- Enlace visibilidad repositorio:  
<https://docs.github.com/es/github/administering-a-repository/managing-repository-settings/setting-repository-visibility>
- Enlace compartir repositorio:  
<https://docs.github.com/es/github/setting-up-and-managing-your-github-user-account/managing-access-to-your-personal-repositories/inviting-collaborators-to-a-personal-repository>

## Token acceso para trabajar con repositorios

- Crear token de acceso Github (**Obligatorio: Actúa de contraseña para trabajar con repositorios**):  
<https://docs.github.com/es/github/authenticating-to-github/keeping-your-account-and-data-secure/creating-a-personal-access-token>

## Establecer usuario y email Github (solo primera vez)

```
git config --global user.name "NOMBREUSUARIO GITHUB"  
git config user.email "EMAILCUENTAGITHUB@SERVIDOR.COM"
```

- La primera vez que usamos git con Github, deberemos configurar estos parámetros. Esto indica las credenciales al conectarnos a cuentas Github para manipular repositorios.

## Usando "git clone" para clonar un repositorio

```
git clone (dirección del repositorio)
```

- Esta orden clona un repositorio de Github en tu máquina local.
  - Si el repositorio es privado te pedirá tu cuenta de usuario y una contraseña. Esa contraseña **NO ES LA CONTRASEÑA DE TU CUENTA**, sino el token personal.

## Usando "git pull" para descargar últimos cambios

```
git pull
```

- Esta orden descarga los últimos cambios de git en un repositorio de Github clonado en tu máquina local.
  - Si el repositorio es privado te pedirá tu cuenta de usuario y una contraseña. Esa contraseña **NO ES LA CONTRASEÑA DE TU CUENTA**, sino el token personal.

## Guardando cambios con "git add", "git commit" y "git push"

```
git add *  
git commit -m "Mensaje del commit"  
git push
```

- La orden "git add" añade ficheros que se almacenarán en el "commit". Aunque pueden nombrarse uno a uno, es habitual usar "git add \*" si todos son candidatos a formar parte del "commit".

-

- La orden “git commit” crea en tu repositorio local una “instantánea” del estado de tu repositorio.
  - Solo aplica a los ficheros que han sido añadidos con “git add” y además han tenido cambios desde el último commit.
  - La orden “git commit” necesita un mensaje. Se lo damos con “-m”.
- La orden “git push” sube los “commits” locales Github.
  - Al hacer push te pedirá tu cuenta de usuario y una contraseña. Esa contraseña **NO ES LA CONTRASEÑA DE TU CUENTA**, sino el token personal.

## Usando “git rm” y “git mv” para borrar renombrar ficheros/directorios

```
git rm ficheroaborrar
git mv fichero nuevonombre
```

- Estas órdenes pueden borrar o renombrar ficheros/directorios. Para que los cambios se reflejen, debe hacerse un commit y subirse, siguiendo la estructura indicada anteriormente.

## Creando ramas con “git branch” y situándose en ellas con “git checkout”

```
git branch nuevarama
git checkout nuevarama
```

- Con “git branch” creamos la rama. Crear la rama NO IMPLICA situarse en esa rama creada. Con “git checkout” nos movemos entre ramas.

## Uniendo ramas con “git merge” o “git rebase”

```
git checkout ramaProduccion
git merge ramaDesarrollo #0 si procede, git rebase ramaDesarrollo
```

- Usando esta secuencia con “git merge”:
  - Creamos un nuevo “commit” que cuelga de “ramaProducción” (sin borrar lo anterior) que incluye la unión de las ramas “ramaProducción” y “ramaDesarrollo”.
- Usando esta secuencia con “git rebase”:
  - Creamos un nuevo “commit” que cuelga de “ramaProducción” pero sin mantener a salvo la historia de la rama secundaria (“ramaDesarrollo”).
- **[OPINIÓN]** ¿Cuándo usar “git merge” y cuando “git rebase”?:
  - Opino que “git merge” debería usarse para subir cambios y nuevas features a la rama producción.
  - Opino que “git rebase” debería ser usado cuando se trata de integrar ramas secundarias en una rama.

## Más información sobre Git

- Más información de git:
  - Git, la guía sencilla: <https://rogerdudler.github.io/git-guide/index.es.html>
  - Learn git branching: <https://learngitbranching.js.org/>
  - Github + git Cheatsheet: [https://training.github.com/downloads/es\\_ES/github-git-cheat-sheet.pdf](https://training.github.com/downloads/es_ES/github-git-cheat-sheet.pdf)