

Git



TEMARIO DEL CURSO

- ¿Qué es un Sistema Control de Versiones?
- ¿Qué es Git?
- Instalación y configuración de Git
- Los tres estados de Git
- Comandos básicos
 - `git init`
 - `git clone`
 - `git add`
 - `git commit`
 - `git status`
 - `git log`
 - `git diff`
 - `git checkout --`
 - `git reset`
 - `git branch`
 - `git stash`
- .gitignore (Ignorando archivos no deseados)
- GitHub
 - ¿Qué es GitHub?
 - `git pull`
 - `git push`
 - `git merge`

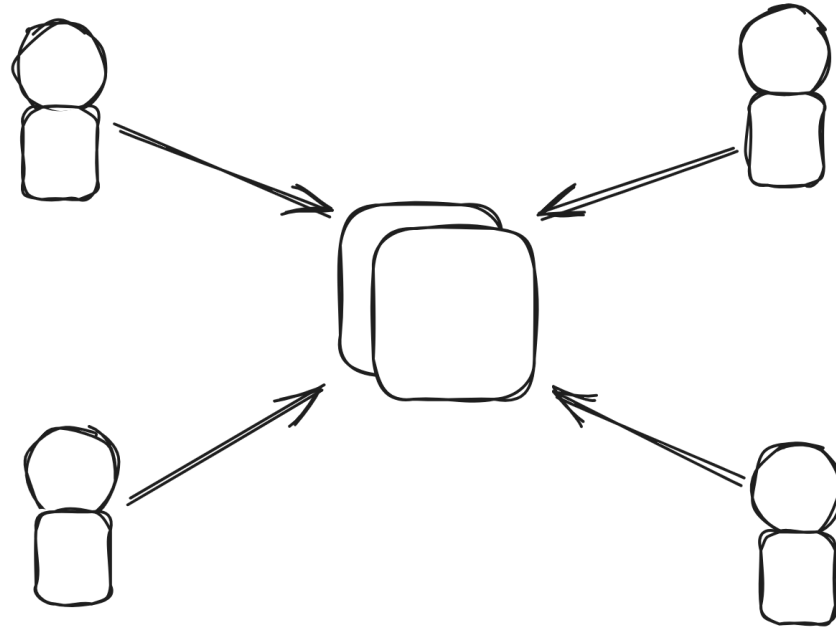
¿Qué es un Sistema Control de Versiones?

Registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo.

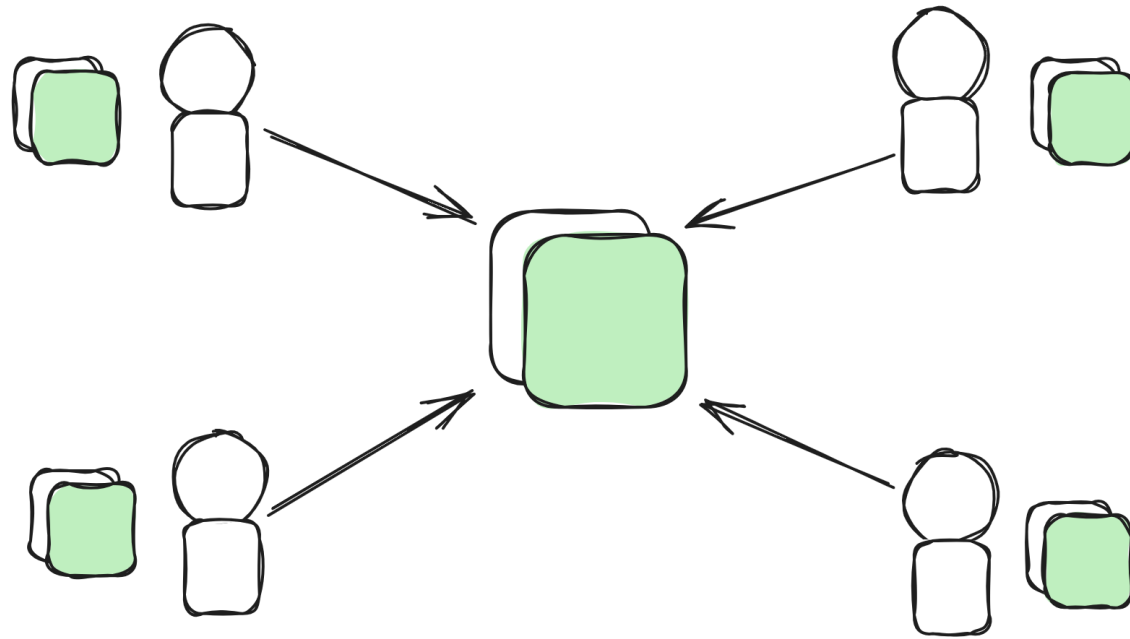
¿Qué es Git?

- Sistema de control de versiones
- Open Source
- Creado por Linus Torvalds

Centralizado



Distribuido



Uso de Git

- Historial
- Almacenar el código
- Trabajar en equipo
- Encontrar errores insertados en el código

Instalación

<https://git-scm.com/>

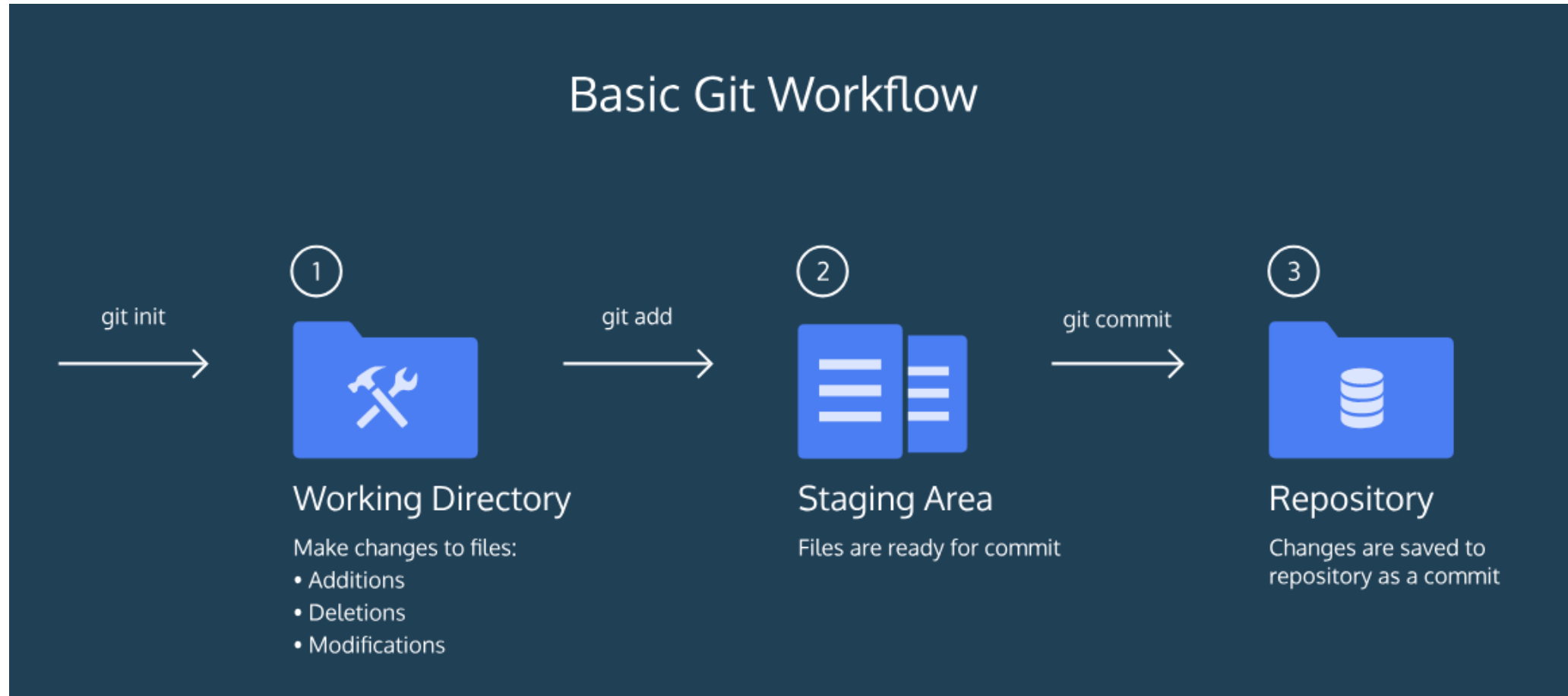
Configuración

```
git config --global user.name "John Doe"  
git config --global user.email johndoe@example.com
```

Los 3 estados de GIT

confirmado (*committed*), **modificado** (*modified*) y **preparado** (*staged*)

1. Hago cambios | 2. git add | 3. git commit | 4. Vuelvo a empezar



Comandos básicos

Inicializando un repositorio

`git init` (**initialize** a new **git** repository)

`git clone` (create a **copy** of an existing repository)

Comandos básicos

Del Working Directory al Staging Area

```
git add <filename>
```

```
git add <filename1> <filenameN>
```

```
git add .
```

Comandos básicos

Viendo el estado de los archivos

```
git status
```

Comandos básicos

Del Staging Area al Repository

```
git commit -m "Commit Message"
```

Convencion estandar para los mensajes de confirmación:

- Deben ir entre comillas
- Deben escribirse en presente
- Deben ser breves (50 caracteres o menos) cuando se utiliza **-m**

Comandos básicos

Revisando las diferencias entre el working directory y el staging área (cambios desde el último commit)

```
git diff <filename>
```

Comandos básicos

Revisando la historia de nuestro proyecto

```
git log
```


Comandos básicos

Eliminado archivos

```
git log
```

Comandos básicos

Eliminado archivos

```
git rm <filename>
```

Comandos básicos

Restaurando archivos

```
git restore <filename>
```

Comandos básicos

Renombrando archivos

```
git mv <filename>
```

Comandos básicos

¿En qué branch/rama estoy?

```
git branch
```

Comandos básicos

Creando una nueva rama

```
git checkout -b "new_branch"
```

*branch names can't contain whitespaces

new_branch

new-branch

Comandos básicos

Revisando las diferencias de un archivo entre dos Branch diferentes

```
git diff branch1 branch2 <filename>
```

Comandos básicos

Revisando las diferencias de un archivo entre dos commits

```
git diff commit1 commit2 <filename>
```


Comandos básicos

RETROCESO

Descartando cambios del working directory

```
git checkout -- <filename>  
git checkout -- <filename1>  
<filenameN>  
git checkout -- .
```

Comandos básicos

RETROCESO

Deshaciendo commits

`git reset --hard <hash>` (Deshaciendo el commit
perdiendo las modificaciones)

`git reset <hash>` (Deshaciendo el commit
manteniendo las modificaciones)

`git log` (Para obtener el hash
del commit)

Comandos básicos

Moviendo a otra rama

```
git checkout "branch_name"
```

Comandos básicos

Guardado rápido provisional

```
git stash
```

```
git stash list
```

```
git stash pop
```

.gitignore

Ignorando archivos no deseados

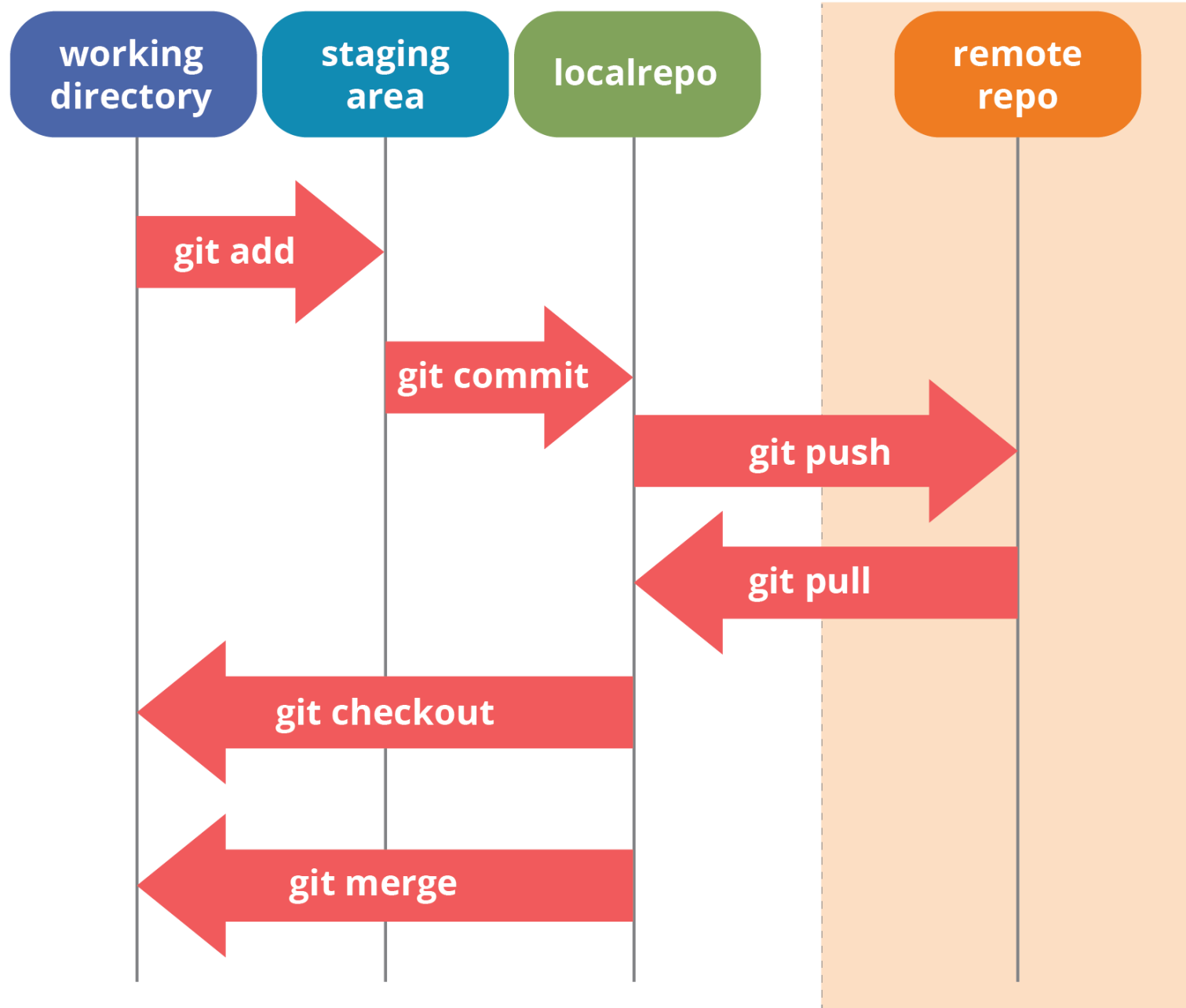


¿Qué es GitHub?

- **Plataforma de desarrollo colaborativo**
- **Aloja proyectos utilizando el sistema de control de versiones Git**
- **Público y Privado (de pago)**
- **El 4 de junio de 2018, Microsoft compró GitHub por la cantidad de 7.500 millones de dólares.**

Local

Remote



Comandos básicos

Agregando el repositorio local a remote

```
git remote add origin <url>
```


Comandos básicos

Enviando cambios al repositorio remoto

```
git push
```

```
git push -u origin <branch>
```

(Añadir un branch local a remote)

Comandos básicos

Trayendo cambios desde el repositorio remoto

```
git pull
```

Comandos básicos

Eliminando un Branch de forma local y remote

```
git push origin -delete <branch_name>
                                                    (remote)
```

```
git branch -d <branch_name>
                    (local)
```

Comandos básicos

Combinando archivos de diferentes ramas

```
git merge
```

? QUESTION 9

Which command should you use to initialize a new Git repository?

- ☐ `git bash`
- ☐ `git install`
- ☐ `git init`
- ☐ `git start`

? QUESTION 10

Which file can you configure to ensure that certain file types are never committed to the local Git repository?

- ☐ *ignore.git*
- ☐ *.gitignore*
- ☐ *gitignore.txt*
- ☐ *git.ignore*

4. We've just created a new file called `index.html` . Which of the following will stage this one file so we can commit it?

```
git add index.html
```

```
git add new
```

```
git commit index.html
```

What command do you run to view the commit history of your repository?

☐

git history

☐

git log

☐

git commit -h

☐

git past

Hoja de Referencia

https://services.github.com/on-demand/downloads/es_ES/github-git-cheat-sheet.pdf

¡A PRACTICAR!

¡A PRACTICAR!

¡A PRACTICAR!