

Módulo 1. Introducción a la ciencia de datos

Git y Github

Javier Cózar

Máster en Ciencia de Datos e Ingeniería de Datos en la Nube





Control de versiones

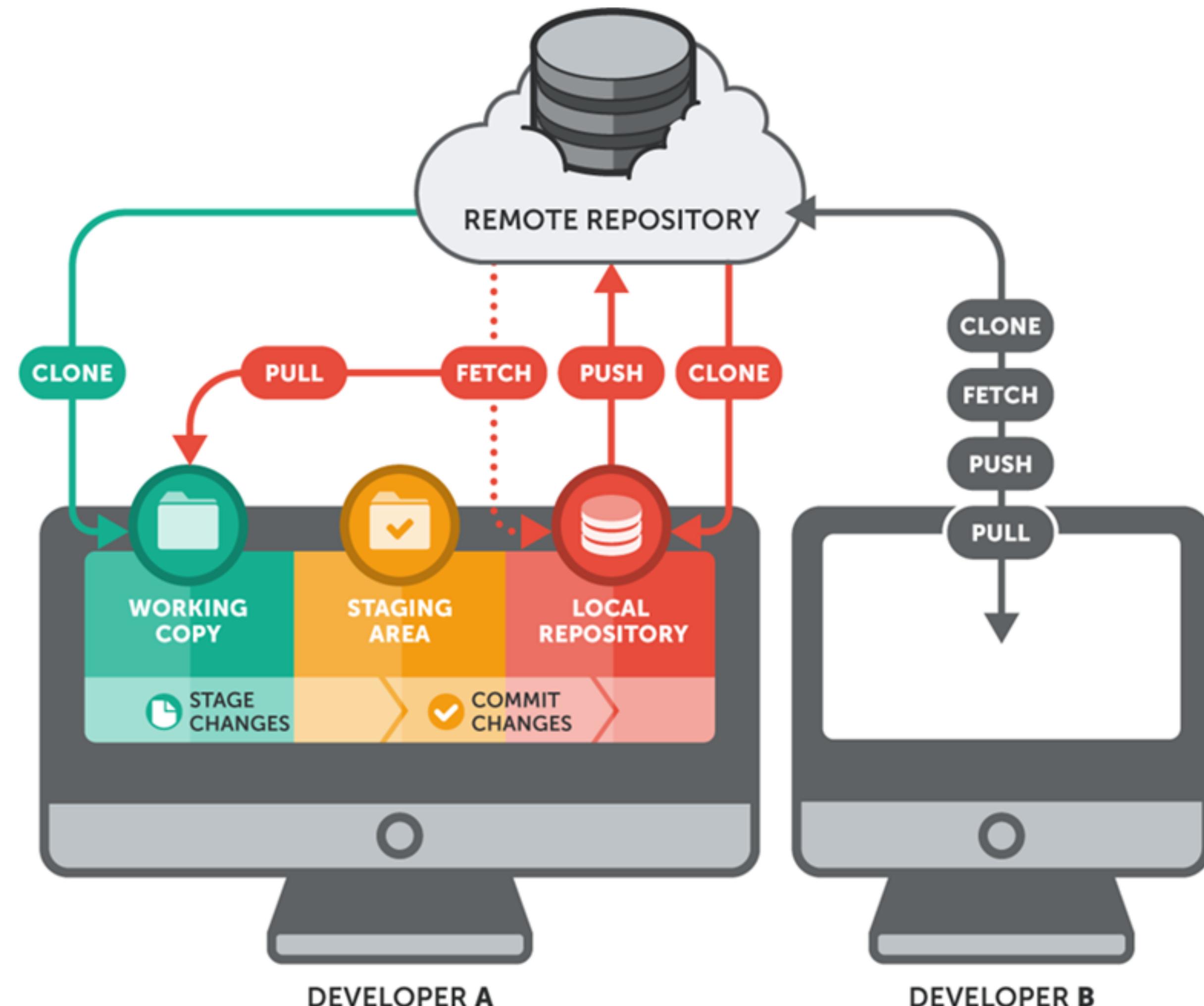
- Para controlar los cambios en un software se necesita de un CVS
- Git es una herramienta sencilla y ligera para el control de versiones
 - Gestión del histórico de cambios (**commits**)
 - Recuperación de un estado del software en un determinado momento (**checkout**)
 - Gestión de cambios concurrentes (por varios usuarios) sobre un mismo fichero



Git



- **Repositorio:** es una carpeta trackeada por git
- **Commit:** representa un conjunto de cambios realizados por una usuario en una fecha concreta
- Existen repositorios **locales** y **remotos**
- Git es una herramienta de **comandos** que nos permite registrar cambios (*commits*) y sincronizar repositorios *locales* y *remotos*





Git

- Es una herramienta **indispensable** en cualquier equipo de desarrollo
- Nosotros planteamos su uso como sistema de **comunicación de código** entre alumnos y profesores
 - Por ello, usaremos esta herramienta a un nivel muy básico
 - Si estáis interesados en aprender git, existe un tutorial práctico online [aquí](#)
 - Para usarlo, tenemos que identificarnos a través de un email

```
git config --global user.email "you@example.com"  
git config --global user.name "Your Name"
```

GIT

Instalación

Be creative





Git - repositorio local

- Creación de un repositorio de git

`git init`

- Añadir o borrar ficheros del sistema de trackeo de git

`git add <path>`

`git rm <path>`

- Registrar todos los cambios realizados (*necesario añadir cada cambio con git add*)

`git commit -m "mensaje que describe el cambio"`

- Se puede usar el argumento `-a` para añadir todos los cambios en el repositorio

`git commit -a -m "mensaje que describe el cambio"`

- Ver el estado actual del repositorio y el histórico de cambios

`git status`

`git log`



Git - repositorio local

- Si queremos deshacer los cambios que hemos hecho y dejar el repositorio en su estado anterior
 - `git reset --hard`
- En ocasiones los profesores cambiaremos el código en un repositorio. Cuando sincronicéis, es posible que entréis en modo edición de texto en el terminal
- El efecto es similar a realizar un commit sin el argumento `-m`
- En estos casos, lo normal es que se os abra **vim** o **nano**
 - para confirmar con vim hay que escribir `:wq` (asegurar modo comandos con `esc`)
 - para confirmar con nano **control+o** e intro para guardar, y **control+x** para salir

GIT

Comandos básicos





GitHub

- Los repositorios remotos son repositorios almacenados en un servidor
- GitHub es una de las principales plataformas (**forja**) que aloja repositorios remotos
- Éstos pueden clonarse, creando un repositorio local, copia del remoto

git clone <url>

The screenshot shows a GitHub repository page for 'Idelaossa Corrección'. At the top, there's a navigation bar with tabs: Code (highlighted), Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, it says 'master' (1 branch, 0 tags). The main area shows a file named 'Idelaossa Corrección' and a folder named 'datos'. At the bottom, it says 'Initial commit'. On the right side, there's a 'Code' dropdown menu with options: Go to file, Add file, and a green 'Code' button. A black arrow points from the text '1. Pulsando en Code' to the green 'Code' button. Another black arrow points from the text '2. url para clonar el repositorio' to the 'Clone' section of the dropdown menu. In the 'Clone' section, there are three options: HTTPS (selected), SSH, and GitHub CLI. The SSH URL is shown as 'git@github.com:cidaen/c2-c1-proyecto1-'. There's also a copy icon next to the URL.

1. Pulsando en Code

2. url para clonar el repositorio

Code

Clone

HTTPS SSH GitHub CLI

git@github.com:cidaen/c2-c1-proyecto1-



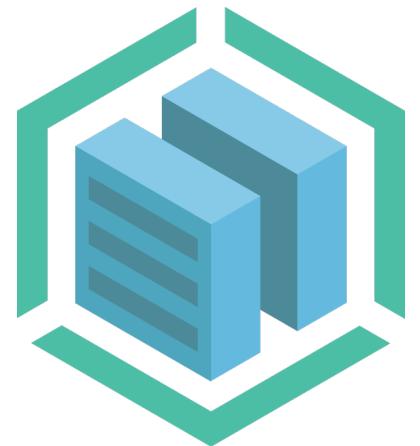
GitHub

- No se efectúan cambios en un repositorio en remoto como en un local
 - Se trabaja con el repositorio local (clonado a partir del remoto)
 - Se sincroniza desde el remoto hacia el local

git pull

- Se sincroniza desde el local hacia el remoto

git push



Free

The basics for individuals and organizations

\$ **0** per year forever

[Create a free organization](#)

- > Unlimited public/private repositories
- > Automatic security and version updates
- > 2,000 CI/CD minutes/month
Free for public repositories
- > 500MB of Packages storage
Free for public repositories
- > New issues & projects (in limited beta)
- > Community support

EXCLUSIVE ADD-ONS

- > GitHub Copilot Access

MOST POPULAR

Team

Advanced collaboration for individuals and organizations

\$ **48** \$ **44** per user/year for the first 12 months*

[Continue with Team ▾](#)

- ← Everything included in Free, plus...
- > Access to GitHub Codespaces
- > Protected branches
- > Multiple reviewers in pull requests
- > Draft pull requests
- > Code owners
- > Required reviewers
- > Pages and Wikis
- > Environment deployment branches and secrets
- > 3,000 CI/CD minutes/month
Free for public repositories

Enterprise

Security, compliance, and flexible deployment

\$ **252** \$ **231** per user/year for the first 12 months*

[Start a free trial](#) [Contact Sales](#)

- ← Everything included in Team, plus...
- > Enterprise Managed Users
- > User provisioning through SCIM
- > Enterprise Account to centrally manage multiple organizations
- > Environment protection rules
- > Audit Log API
- > SOC1, SOC2, type 2 reports annually
- > FedRAMP Tailored Authority to Operate (ATO)
- > SAML single sign-on
- > Advanced auditing

GIT

- Crear una cuenta en GitHub
- Comandos de sincronización