

APRENDE GIT

COMO SI ESTUVIERAS EN PRIMERO

Alejandro Barrachina Argudo
Pablo García de los Salmones

2023-10-30

PROGRAMA

1. Github
2. Git
 - a. commit
 - b. ramas
3. Ejercicio (Taller)

Github

Github nos va a servir para alojar el código de nuestros proyectos → repositorio y para colaborar con otras personas en ellos → “red social”.



Git - Introducción

Git es la herramienta de control de versiones más popular y utilizada = no te lías entre versiones cuando haces el proyecto de FP entre varios compañeros.

Git - Flujo

Debemos seguir un flujo específico si queremos trabajar correctamente:

1. `git pull` → "Descarga los cambios del repositorio"
2. `desarrollamos código` → It's a feature not a bug
3. `git add y commit` → "guardamos los cambios en un punto concreto. checkpoint"
4. `git push` → "subimos los cambios al servidor"

Fichero .gitignore

Es un fichero especial en texto plano

Le especificas las cosas a ignorar

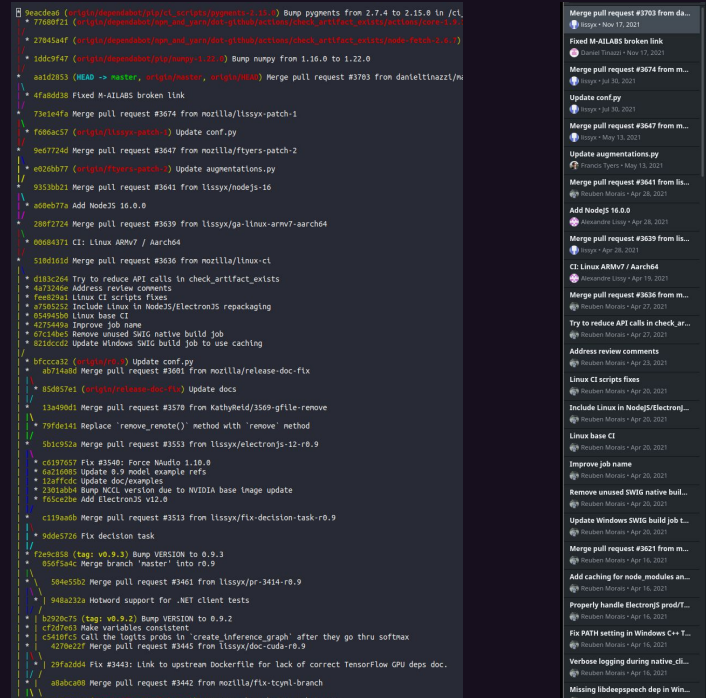
- ficheros compilados
- configuración local (IDE)
- ficheros temporales y auxiliares

Herramientas gráficas



Gitkraken

Git-Graph



git log

GH Desktop

Taller

Descargar Git

Windows

Descargar git bash
(gitforwindows.org)

Linux

Seguramente ya lo
tienes instalado

Usa apt, pacman, etc

Descargar Github Desktop

Windows

Descargar github
desktop

(<https://desktop.github.com/>)

Linux

- Ubuntu:
<https://github.com/kontr0x/github-desktop-install>
- Arch: `pacman -S github-desktop`

Si usas github desktop durante este taller y no encuentras alguna de las opciones (clonar, hacer pull o push), puedes mirar las imágenes en <https://github.com/ALK222/Charla-Git/blob/main/main.pdf>

Primera configuración de git en consola de comandos:

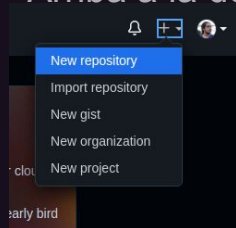
La primera vez que intentemos hacer un commit, se nos va a quejar, diciendo que debemos configurar dos variables:

```
git config --global user.name "Nombre de usuario github"  
git config --global user.email "Email de usuario github"
```

Crear repositorio en GitHub o GitLab

GitHub

Arriba a la derecha



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

Repository name *

 daviddavo /

Great repository names are short and memorable. Need inspiration? How about [verbose-adventure?](#)

Description (optional)



Public

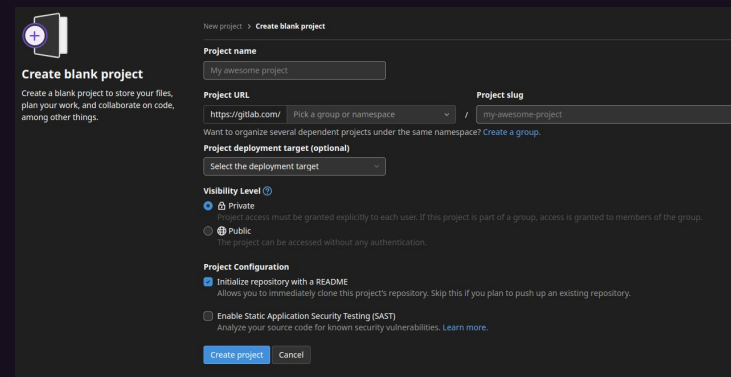
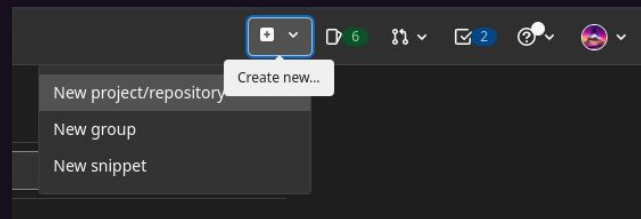
Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

GitLab



Obtener url del repositorio:

dlgeraghty / final-project-cs50 Public

Notifications Fork 0 Star 0

<> Code Issues Pull requests Actions Projects Security Insights

main 1 branch 0 tags

Go to file Code

Clone

HTTPS GitHub CLI

<https://github.com/dlgeraghty/final-project>

Use Git or checkout with SVN using the web URL.

Open with GitHub Desktop

Download ZIP

dlgeraghty finish readme contents

_pycache__	close for the day
html	fix the menu
markdown_files	fix the menu
templates	fix the menu
LICENSE	Initial commit
README.md	finish readme contents
app.py	fix the menu 3 days ago
index.db	fix the menu 3 days ago

README.md

STATIC SITE GENERATOR

About

No description, website, or topics provided.

Readme

Apache-2.0 license

0 stars

1 watching

0 forks

Releases

No releases published

Packages

No packages published

Languages

Nota: recomiendo usar https, es más "sencillo" pero puede ser mejor usar una llave ssh

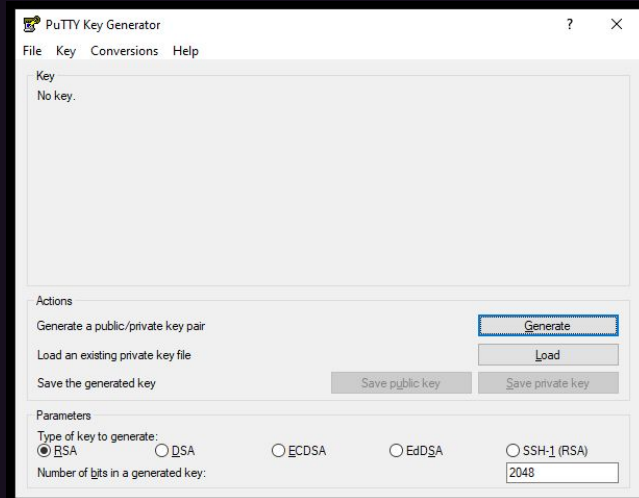
Usar clave ssh

Configurar clave SSH

Windows

<https://www.putty.org/>

- pageant.exe
- puttygen.exe



Linux

- ssh-keygen

Configurar clave SSH

<https://github.com/settings/keys>

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

Add new SSH Key

Title

Key type

Authentication Key ↕

Key

Must begin with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

- `ssh git@github.com`

Copiar o “clonar” el repo

Comando

```
git clone <dirección>
```

Se nos genera una nueva carpeta con el nombre del repositorio
Entrar en la carpeta y guardar lo que queráis en texto plano

Ejercicios propuesto:

- Puedes usar tu editor de texto favorito
- Crea un archivo `hello-wold.txt` y escribe tu nombre
- Crea o edita el archivo `README.md` usando formato markdown
- Puedes probar a crear carpetas

TIP: usa el comando `git status` frecuentemente para ver los diferentes estados en los que se encuentra el repositorio

Añadir los ficheros al commit

```
git add <fichero>
```

Tip: Usa 'git add .' para añadir todo

```
git rm <fichero>
```

“Guarda” tus cambios con commit

- `git commit -am “mi primer commit”`

Sube los cambios a github

- `git push origin main`

Visualiza los cambios en github

- Entra en tu cuenta y mira si los cambios que has hecho en local se ven reflejados en tu repositorio

Bonus

- Qué pasa si intentas clonar el repositorio de algún compañero, haces cambios y los intentas subir ?
- Qué significan las opciones del comando git push ?
<<<origin>>> y <<<main>>>
- Github actions → automatización de tareas cuando haces acciones en el repositorio
- Github pages → hostea una web estática gratis

Bonus

- Github education → Github Pro gratis y acceso gratuito a otras herramientas
- ¿Y si la cago? → ohshitgit.com
- ¿Te gusta jugar a las cartas? → [Oh My Git!](https://ohmygit.com)
- PDF Resumen con herramientas gráficas → <https://github.com/ALK222/Charla-Git>