

Flux d'un repositori Git

Taula de Continguts

- [Creació del repositori](#)
- [Edició del document](#)
- [Gestió de canvis](#)
- [Importació del repositori a Github](#)
- [Sincronització amb GitHub](#)

Creació del repositori

Obrim el nostre terminal preferit i creem el directori arrel del repository

```
$ mkdir 'Tasques_DAW'
```

Després, inicialitzem el repositori

```
$ git init
```

Git ens tornarà el missatge d'èxit

```
hint: Using 'master' as the name for the initial branch. This default
hint: branch name
hint: is subject to change. To configure the initial branch name to use in
hint: all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:     git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this
hint: command:
hint:
hint:     git branch -m <name>
Initialized empty Git repository in /home/manpop/Tasques DAW/.git/`
```

Després, creem el nostre document

```
$ nano Documentació_flux_git.md
```

Edició del document

Per editar el document he triat l'editor MarkText, perquè renderitza el document a la par que es va editant, però es pot editar inclús amb nano mateix

Gestió de canvis

Després de guardar els canvis, afegim tots els arxius del directori del repositori al àrea *staging* amb este comandament :

```
$ git add .
```

Finalment, fem efectiu el canvi amb un *commit*

```
$ git commit -m "Primer commit, creada documentació"
```

Importació del repositori a Github

Per importar el repositori a GitHub, seguirem aquests passos 1. Al nostre perfil de GitHub creem un nou repositori

Import a repository.' and a note 'Required fields are marked with an asterisk (*)'. The form is divided into two sections: '1 General' and '2 Configuration'. In the 'General' section, there's a dropdown for 'Owner' (currently 'Leprechaun19-null') and a text input for 'Repository name'. Below this is a hint: 'Great repository names are short and memorable. How about [special-octo-giggle?](#)'. There's a 'Description' text area with a character count '0 / 350 characters'. The '2 Configuration' section has a 'Choose visibility' dropdown (currently 'Public'), a toggle for 'Add README' (currently 'Off'), a dropdown for 'Add .gitignore' (currently 'No .gitignore'), and a dropdown for 'Add license' (currently 'No license'). At the bottom right is a green 'Create repository' button."/>

New repository

Search Type [] to search

Create a new repository

Repositories contain a project's files and version history. Have a project elsewhere? [Import a repository.](#)

Required fields are marked with an asterisk (*).

1 General

Owner * Repository name *

Leprechaun19-null /

Great repository names are short and memorable. How about [special-octo-giggle?](#)

Description

0 / 350 characters

2 Configuration

Choose visibility * Public

Choose who can see and commit to this repository

Add README Off

READMEs can be used as longer descriptions. [About READMEs](#)

Add .gitignore No .gitignore

.gitignore tells git which files not to track. [About ignoring files](#)

Add license No license

Licenses explain how others can use your code. [About licenses](#)

Create repository

Després, al quick setup, executem aquests comandaments:

```
$ git remote add origin <url del repositori>
```

```
$ git branch -M main
```

```
$ git push -u origin main
```

Sincronització amb GitHub

A partir d'ara, cada volta que fem commits, haurem d'executar

```
$ git push -u origin main
```

Així garantim la sincronització entre la nostra còpia local del repositori i la de GitHub.