

Ejercicio 1: Introducción a git y GitHub

1. Crea un compte a GitHub (Si no el tens!!!). La forma d'accedir als repositoris remots de GitHub serà per SSH, per tant has de copiar la teua clau pública RSA a GitHub, per això:

- Copia el contingut del teu fitxer `~/.ssh/id_rsa.pub`, per això: afegeix una nova clau SSH a l'apartat "SSH keys" del teu perfil a GitHub i enganxa el contingut de la teua clau pública.
- Si no teniu aquest fitxer, podeu generar una nova clau ssh pública seguint les instruccions de l'article Treballant amb claus ssh.
- Un cop finalitzat aquest exercici, hauràs de repetir-ho fent ús de tokens.

2. Crea a GitHub un repositori amb el nom `prova_el_teu_nom` (inicialitza el repositori amb un fitxer README) i la descripció Repositorio de prueba 2ASIX.

1 General

Owner * `ikerpa04` / Repository name * `prova_ikerpatino`

✓ Your new repository will be created as `prova_ikerpati-o`.
The repository name can only contain ASCII letters, digits, and the characters `.`, `-`, and `_`.

Great repository names are short and memorable. How about `fictional-happiness`?

Description
`Repositorio de prueba 2ASIX`
27 / 350 characters

2 Configuration

Choose visibility * Choose who can see and commit to this repository `Public`

Add README READMEs can be used as longer descriptions. [About READMEs](#) On ☒

Add .gitignore .gitignore tells git which files not to track. [About ignoring files](#) `No .gitignore`

Add license Licenses explain how others can use your code. [About licenses](#) `No license`

Create repository

3. Instal·la git en el teu ordinador (si no el tens instal·lado!!!).

apt-get install git

```
iker@iker-patino: ~  
iker@iker-patino:~$ sudo apt-get install git  
[sudo] contrasenya para iker:  
Leyendo lista de paquetes... Hecho  
Creando árbol de dependencias... Hecho  
Leyendo la información de estado... Hecho  
git ya está en su versión más reciente (1:2.43.0-1ubuntu7.3).  
fijado git como instalado manualmente.  
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
```

4. Configuració de git. El primer que hauries de fer quan instal·les Git és establir el teu nom d'usuari i adreça de correu electrònic (Assegura't que les dades són correctes i que has posat el teu nom complet). Això és important perquè les confirmacions de canvis (commits) a Git utilitzen aquesta informació, i és introduïda de manera immutable en els commits que envieu:

git config --global user.name "John Doe"

git config --global user.email johndoe@example.com

```
iker@iker-patino: ~  
iker@iker-patino:~$ git config --global user.name "ikerpa04"  
iker@iker-patino:~$ git config --global user.email ikepatalv@alu.edu.gva.es
```

De nou, només cal fer-ho una vegada si especifiques l'opció `--global`, ja que Git sempre utilitzarà aquesta informació per a tot el que facis en aquest sistema.

Clonar el repositori remot. Copia la url SSH del repositori (no copiïs la URL https) i clonarem el repositori al nostre ordinador.

git clone [git@github.com:xxxxxxx/xxxxxxx.git](https://github.com/ikerpa04/prova_ikerpatino.git)

```
iker@iker-patino:~$ git clone git@github.com:ikerpa04/prova_ikerpatino.git  
Clonando en 'prova_ikerpatino'...  
remote: Enumerating objects: 6, done.  
remote: Counting objects: 100% (6/6), done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 6 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
Recibiendo objetos: 100% (6/6), listo.
```

Para comprobar que tienes configurado el repositorio usando la url SSH puedes ver el fichero de configuración en `.git/config`.

```
iker@iker-patino:~$ cd prova_ikerpatino
iker@iker-patino:~/prova_ikerpatino$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
[remote "origin"]
    url = git@github.com:ikerpa04/prova_ikerpatino.git
    fetch = +refs/heads/*:refs/remotes/origin/*
[branch "main"]
    remote = origin
    merge = refs/heads/main
iker@iker-patino:~/prova_ikerpatino$
```

Comprova que dins del repositori que hem creat hi ha el fitxer `README.md`, en aquest fitxer podem posar la descripció del projecte.

```
iker@iker-patino:~/prova_ikerpatino$ ls
README.md
```

Crearem un nou fitxer, l'afegirem al nostre repositori local i després ho sincronitzarem amb el nostre repositori remot de GitHub. Cada vegada que fem una modificació en un fitxer ho podem assenyalar creant un commit. Els missatges dels commits són fonamentals per explicar l'evolució d'un projecte.

Un commit ha de ser un petit conjunt de canvis dels fitxers del projecte amb una certa coherència.

```
iker@iker-patino:~/prova_ikerpatino$ echo "Açò és una prova" > exemple.txt
iker@iker-patino:~/prova_ikerpatino$ git add exemple.txt
```

```
iker@iker-patino:~/prova_ikerpatino$ git commit -m "He creat el fitxer exemple.txt"
[main 0a2990f] He creat el fitxer exemple.txt
1 file changed, 1 insertion(+)
create mode 100644 exemple.txt
```

```
iker@iker-patino:~/prova_ikerpatino$ git push
Enumerando objetos: 4, listo.
Contando objetos: 100% (4/4), listo.
Compresión delta usando hasta 16 hilos
Comprimiendo objetos: 100% (2/2), listo.
Escribiendo objetos: 100% (3/3), 313 bytes | 313.00 KiB/s, listo.
Total 3 (delta 0), reusados 0 (delta 0), pack-reusados 0
To github.com:ikerpa04/prova_ikerpatino.git
6487cb5..0a2990f  main -> main
```

Per comprovar l'estat del meu repositori local:

```
iker@iker-patino:~/prova_ikerpatino$ git status
En la rama main
Tu rama está actualizada con 'origin/main'.

nada para hacer commit, el árbol de trabajo está limpio
```

Busca informació per a crear un nou repositori anomenat `prova2_el_teu_nom`. En aquesta ocasió, crea primer el repositori local (usant `git init`) i després cerca informació per sincronitzar-lo i crear el repositori remot a GitHub.

```
iker@iker-patino:~$ mkdir prova2_ikerpatino
iker@iker-patino:~$ cd prova2_ikerpatino/
iker@iker-patino:~/prova2_ikerpatino$ git init
ayuda: Usando 'master' como el nombre de la rama inicial. Este nombre de rama pr
edeterminado
ayuda: está sujeto a cambios. Para configurar el nombre de la rama inicial para
usar en todos
ayuda: de sus nuevos repositorios, reprimiendo esta advertencia, llama a:
ayuda:
ayuda:  git config --global init.defaultBranch <nombre>
ayuda:
ayuda: Los nombres comúnmente elegidos en lugar de 'master' son 'main', 'trunk'
y
ayuda: 'development'. Se puede cambiar el nombre de la rama recién creada median
te este comando:
ayuda:
ayuda:  git branch -m <nombre>
Inicializado repositorio Git vacío en /home/iker/prova2_ikerpatino/.git/
```

Ahora añadimos texto al README.md.

```
iker@iker-patino:~/prova2_ikerpatino$ echo "Esto es una segunda prueba" > README
.md
```

```
iker@iker-patino:~/prova2_ikerpatino$ git commit -m "Actualización"
[master (commit-raíz) 42ac1f2] Actualización
1 file changed, 1 insertion(+)
create mode 100644 README.md
iker@iker-patino:~/prova2_ikerpatino$
```

```
iker@iker-patino:~/prova2_ikerpatino$ git remote add origin https://github.com/i
kerpa04/prova2_ikerpatino.git
iker@iker-patino:~/prova2_ikerpatino$ git branch -M main
iker@iker-patino:~/prova2_ikerpatino$ git push -u origin main
Username for 'https://github.com': ikerpa04
```