



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



Manual de connexió a Gitlab REPO

Versió 1.1



Índex

Cóm accedir	3
Web	3
SSH	3
VPN	3
Connexió HTTPS	4
Credencials	5
On guardem les credencials?	5
Configurar credencials al Gitlab	6
Configurar l'usuari git	9
Crear projecte nou	10
Preparar espai de repositori	10
Preparar espai local	13
Unir-se a un projecte existent	16



Cóm accedir

En primer lloc s'ha de tenir en compte des d'on podem connectar-nos al Gitlab REPO. Les úniques vies habilitades son des de tot internet via Web (HTTPS), i via SSH des de les xarxes de laboratoris i TICs de la FIB, i també des de fora d'aquestes xarxes mitjançant la VPN.

Web

Via Web l'accés es realitza mitjançant l'adreça <https://repo.fib.upc.edu> on es pot fer login i gestionar l'usuari. A més, si es configura el repositori via web tenim la seguretat que cada vegada que volem fer alguna operació com *push* o *pull* ens demanarà l'usuari i contrasenya. Quan estem en equips que no son propis o que son compartits és l'opció més segura.

SSH

La connexió SSH només és possible des de un entorn segur com es la VPN de la UPC. Això implica que hem de connectar prèviament per VPN per poder accedir a la connexió SSH al servidor de Gitlab Repo. L'avantatge de connectar via SSH al servidor es que podem fer operacions com *push* o *pull* no necessitem introduir usuari i contrasenya cada cop, només tenir registrats els codis de claus als fitxers corresponents al lloc on han de ser.

VPN

Al moment de fer aquest manual el mètode de connexió VPN de Fortinet mitjançant el client Forticlient. Per més informació sobre cómo connectar per VPN adreceu-vos a la informació que trobareu al següent enllaç. <https://serveistic.upc.edu/ca/upclink>

Una de les proves definitives per comprovar que la VPN està activa i funcionant es mirar d'accedir al servei d'Aula Virtual. <https://aulavirtual.fib.upc.edu/>
Si podem connectar correctament a la web vol dir que la VPN funciona correctament.



Connexió HTTPS

Per connectar via HTTPS (web) només hem de seguir les passes següents anant directament a l'apartat “*Configurar l'usuari Git*” obviant la secció de “*Credencials*”; i en l'apartat de “*Preparar l'espai local*” hem de canviar l'adreça `git@repo.fib.upc.edu...` per l'adreça HTTPS de la figura 10. La resta de àsses son les mateixes tant per HTTPS com per SSH.

A l'apartat de “Unir-se a un projecte existent” també hem de tenir en compte que l'adreça de connexió tampoc serà `git@repo.fib.upc.edu...` si no l'adreça HTTPS de la figura 10.



Credencials

Per poder accedir a Gitlab REPO per SSH cal tindre mètodes d'enciptació de clau pública-privada. Hi han moltes formes de generar aquestes claus. La més genèrica i independent del sistema operatiu es accedir a un servei de generador de claus via web. Algunes de les webs on es poden generar aquestes claus poden ser aquestes:

<https://8gwifi.org/sshfunctions.jsp>

<https://travistidwell.com/jsencrypt/demo/>

Les claus generades tenen un aspecte semblant a això:

- Clau privada: -----BEGIN RSA PRIVATE KEY----- MIICWgIBA...
- Clau pública: -----BEGIN PUBLIC KEY----- MIGeMA0G...

La llargada de la clau privada es notòriament més llarga que la clau pública. A més, la clau privada sol acabar amb "-----END RSA PRIVATE KEY-----" i la pública no és obligatori que indiqui res.

Hi han diversos tipus de claus permeses però es recomanable fer servir la més extesa que es la RSA-2048.

S'ha de tenir en compte que les claus generades via web no es poden considerar 100% segures, però si es tracta d'un ús poc rellevant i a més passat per VPN podem estar força tranquils de fer-la servir.

També hem de tenir en compte que si generem la credencial amb contrasenya l'hauem d'introduir cada cop que vulguem connectar amb Gitlab. Posar contrasenya és opcional.

On guardem les credencials?

El lloc on guardem les credencials és molt rellevant ja que el client de git te unes rutes per defecte que hem de mirar de respectar, així com el nom dels fitxers.

Per defecte, la ruta on han de estar els fitxers de credencials es `~/.ssh/`. On el símbol `~` te significats diferents en Windows i Linux. Per a Linux fa referència a la carpeta personal de l'usuari, és a dir, `/home/<nom-usuari>/`; això implica que els fitxers han d'estar a la ruta `/home/<nom-usuari>/ssh/`. Per a Windows, el client de git "Git CMD" o "Git Bash" interpreta el símbol `~` com la unitat arrel on s'estan executant les comandes. Això vol dir que si executem les comandes dins de "Git CMD" a la ruta

`Z:\users\<nom-usuari>\documents\git-project\` les claus han de deixar-se a la ruta `Z:\ssh\`.

Per "Git Bash" seria `/z/.ssh/`.

Pel que fa als noms del fitxers, hem de tenir la clau privada com `.ssh/id_rsa` i la clau pública com `.ssh/id_rsa.pub` (aquesta última només per guardar la relació entre les dos).

Pel que fa referència als sistemes operatius d'Apple podem considerar la mateixa estructura que Linux.

És possible que la ruta per a Windows sigui diferent en funció de les variables d'entorn `%HOMEDRIVE%` i `%HOMEPATH%`. S'han de verificar aquestes variables d'entorn per si alteren la referència del símbol `~`.

Configurar credencials al Gitlab

Un cop tenim les claus generades i guardades al nostre equip passarem a introduir la clau pública al nostre perfil de Gitlab REPO.

Accedim a <https://repo.fib.upc.edu/> amb el nostre usuari i contrasenya de la UPC. Anem a la secció del nostre usuari que trobarem a dalt i a la dreta de la pàgina i entrem a la secció 'Editar perfil'.

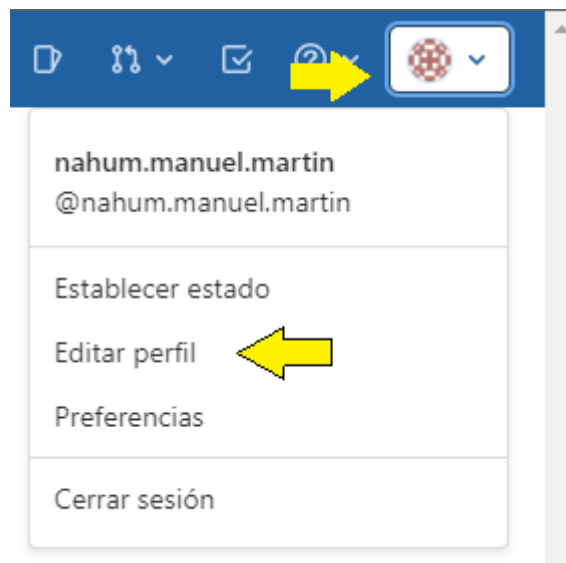


Fig. 1 - Accés a perfil i opcions del perfil.

Un cop a la secció del nostre perfil anirem a la secció de 'Clau SSH' que trobarem al menú de l'esquerra.



Fig. 2 - Accés a la secció de clau SSH del perfil.

Dintre d'aquesta secció introduïrem la clau pública amb un títol per identificar-la. Podem fer servir diferents claus per a diferents llocs de treball.

Clave

```
ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQACqN8PAV856s81eXRCHjtuju3fIX0t7DTuPw0fNQ6ET1Fd6i  
JgUoif8dClonPtCxlG4rJo40jYk+/4lx+DnqgnJmhb+D48ROxFTGV7XE05DRs9z6pN7MinTUap3co7zN  
iBzq4ihUloivGfZ187GLZdUhdL0zvQTPR/0oex9DpcQu2e3I3mB10BgDZMhJ6LJWvvhSERpTje1ad7  
KJpwegqil08KCH7a9o9Z6ZZftwPNA2UyoGTKkPzJil1ofY3kqNbmCw39Wa6tC7U9LU2fAfBgHPcrZuft  
/Apdso07LRHywVmrS9Z5nPu3GRPcKEAYWCyVZbOSDUN87U9374WWBbr
```

Begins with 'ssh-rsa', 'ssh-dss', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', o 'sk-ssh-ed25519@openssh.com'.

Title

Key titles are publicly visible.

Fecha de vencimiento

Optional but recommended. If set, key becomes invalid on the specified date.

Fig. 3 - Inserció de clau SSH.

Un cop afegida ens mostrarà la informació rellevant i la podem tornar a veure a la mateixa secció a la part inferior.

Clave SSH	ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQcQN8PAV856s81eXRChjtuj3fLX0t7DTuPw0fnQ6E
Título: Título de la clau	
Creado sobre: Nov 21, 2022 12:04pm	
Caduca: Nov 21, 2023 12:00am	
Utilizado por última vez en: Nunca	

Huellas digitales

MD5: 90:25:0a:92:e6:5f:58:3a:c7:3c:5a:d6:3d:0f:a2:f7

SHA256: Zfaifjy6D3xuJ7Qn8qgr9l9TEbtMA6X451uW0mjLvmk

Eliminar

Fig. 4 - Informació de clau SSH introduïda.


2023-11-21  

Optional but recommended. If set, key becomes invalid on the specified date.

Añadir clave

Sus Claves SSH (3)

Título de la clau

 90:25:0a:92:e6:5f:58:3a:c7:3c:5a:d6:3d:0f:a2:f7

Created hace 1 minuto

Último uso: Nunca Caduca: 2023-11-21



Fig. 5 - Llistat de claus actives al perfil.

Ara que tenim les claus al Gitlab ja podem configurar el nostre projecte o importar un projecte ja creat al nostre equip.

Hem de tenir en compte que cada clau pública només serveix per un usuari, és a dir, no es poden compartir claus SSH entre usuaris. Això es deu a que cada clau identifica un usuari amb els seus permisos, i per tant no es poden identificar dos usuaris amb la mateixa clau.



Configurar l'usuari git

Si hem de fer servir connexions Https al Gitlab és important configurar nom d'usuari i correu electrònic al client git local de la següent manera:

```
git config --global user.name "username"  
git config --global user.email "nom.cognom@estudiantat.fib.upc.edu"
```

No és necessari configurar nom d'usuari i correu electrònic per fer servir claus SSH però no està de més deixar-ho configurat per si en algun moment el servei per SSH es talla.



Crear projecte nou

Per crear un projecte nou el primer que hem de fer es preparar l'espai del repositori i l'espai local on tindrem el projecte.

Preparar espai de repositori

Quan creem un nou projecte hem d'assegurar-nos de les dades més importants. El nom del projecte o repositori, la seguretat i els membres que hi treballaran.

Les dos últimes es poden fer a posteriori però el nom del repositori és clau. Si l'hem de canviar s'ha de crear un repositori nou.

Per crear un nou repositori entrarem al Gitlab Repo via web i seleccionarem la opció de "Nou projecte".

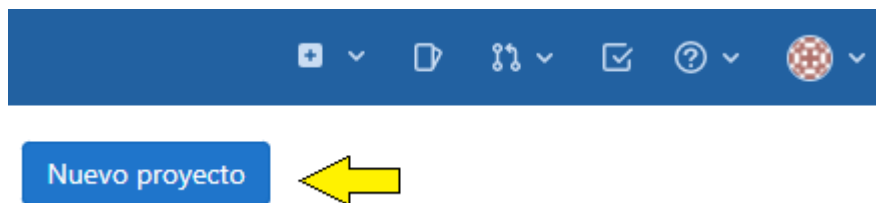


Fig. 6 -Botó de crear nou projecte

Es possible que si no tenim cap altre projecte al nostre perfil ens envii directament al següent pas. En aquest següent pas hem de seleccionar "Crear un nou projecte en blanc".

Crear nuevo proyecto

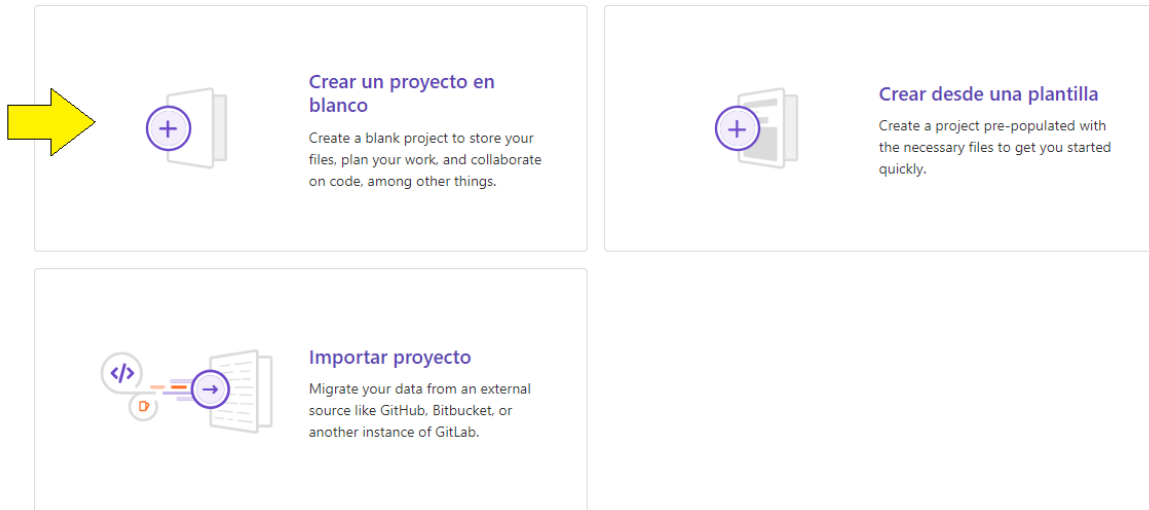
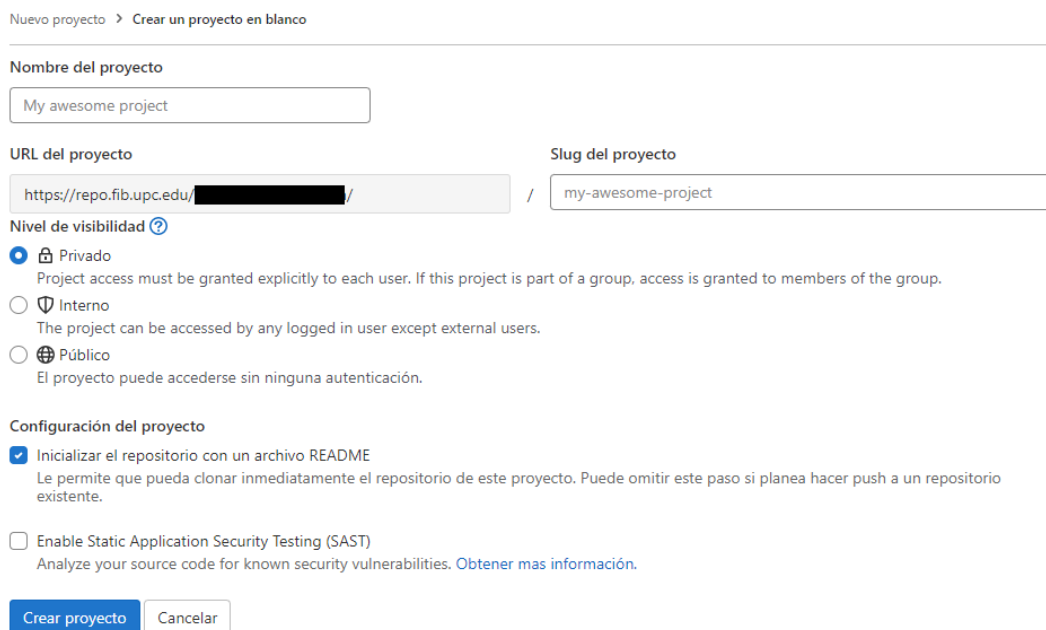


Fig. 7 - Secció de crear nou projecte en blanc.

Ara ens apareixerà l'apartat de creació del projecte on hem d'omplir el "Nom del projecte", "Slug del projecte" si el volem canviar, seleccionar el tipus de visibilitat del projecte i decidir si volem que es creï automàticament el fitxer README. Un cop omplert piquem a "Crear projecte".



The screenshot shows the 'Crear un proyecto en blanco' form. It includes the following fields and options:

- Nombre del proyecto**: Input field containing "My awesome project".
- URL del proyecto**: Input field containing "https://repo.fib.upc.edu/[redacted] /".
- Slug del proyecto**: Input field containing "my-awesome-project".
- Nivel de visibilidad**: Radio buttons for "Privado" (selected), "Interno", and "Público".
- Configuración del proyecto**: Checkboxes for "Inicializar el repositorio con un archivo README" (checked), "Enable Static Application Security Testing (SAST)", and "Obtener mas información".
- Buttons**: "Crear proyecto" and "Cancelar".

Fig. 8 - Secció de creació de projecte.



Un cop creat tenim la informació del projecte que acabem de crear. En aquesta pantalla necessitem l'adreça del projecte per poder connectar per SSH, per això farem clic a l'opció de "Clonar".

el meu projecte

Proyecto 'el meu projecte' fue creado satisfactoriamente.

E el meu projecte 🔒 🔔 Marcar como favorito 0 🍴 Fork 0

ID de proyecto: 756 🔗

1 Commit 1 Branch 0 Tags 61 KB Project Storage

main el-meu-projecte / +

Buscar archivo Web IDE 📄 Clonar

Initial commit a4745711 🔗 Autor solo ahora

README Auto DevOps habilitado Añadir LICENSE Añadir CHANGELOG Añadir CONTRIBUTING

Añadir clúster de Kubernetes Configurar integraciones

Nombre	Último cambio	Última actualización
README.md	Initial commit	solo ahora

Fig. 9 - Vista del projecte i botó de clonar.

Aquí ens apareixen les dos opcions de connexió, SSH i HTTPS. Aquí hem de copiar l'adreça per SSH. Si fem clic al botó del costat ens ho copia al portapapers per poder enganxar-ho a un altre lloc.

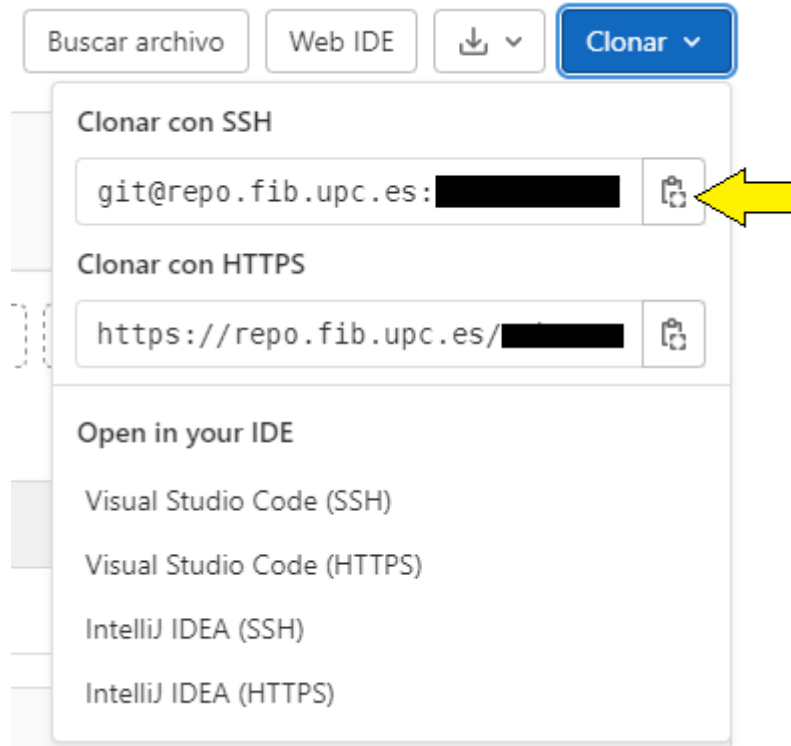


Fig. 10 - Adreces de connexió SSH i HTTPS.

Preparar espai local

Ara hem de passar a la creació de l'espai local, si és amb Linux podem fer-ho directament per terminal però per Windows hem de fer servir el client "Git CMD" o "Git Bash".

En primer lloc hem de crear la carpeta del projecte, es recomanable que es digui com el nom que es va posar al "Slug del projecte".

```
mkdir el-meu-projecte
```

Entrem a la carpeta.

```
cd el-meu-projecte
```

I generem el fitxer de configuració de git per aquest nou projecte.

```
git init
```



Mirarem de crear un nou fitxer per a fer el primer commit, aquest fitxer nou el podem dir README, LICENSE o qualsevol altre que necessitem.

```
touch LICENSE
```

Afegirem el nou fitxer al pròxim commit.

```
git add LICENSE
```

Prepararem el primer commit del projecte.

```
git commit -m 'primer commit'
```

Afegirem l'adreça del projecte que hem copiat de la web com hem explicat a la secció anterior.

```
git remote add origin git@repo.fib.upc.edu:<nom.usuari o grup>/el-meu-projecte.git
```

Per finalitzar i verificar l'accés via SSH farem la primera pujada (push)

```
git push -u origin master
```

Hem de tenir en compte que ara estem creant una nova branca (branch) que es dirà 'master', i ara tindrem la branca 'main' i 'master'. En qualsevol moment que tinguem una versió funcional la podem fusionar dintre de l'accés web al repositori fent sol·licituds de fusió (merge) cap al 'main'.

Podem veure els resultats de algunes d'aquestes comandes a la següent imatge:

```
$ git add LICENSE

MINGW64 /f/e1-meu-projecte (master)
$ git commit -m 'primer commit'
[master (root-commit) 298f659] primer commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 LICENSE

MINGW64 /f/e1-meu-projecte (master)
$ git remote add origin git@repo.fib.upc.es: /e1-meu-projecte.git

MINGW64 /f/e1-meu-projecte (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 211 bytes | 35.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: To create a merge request for master, visit:
remote: https://repo.fib.upc.es/ /e1-meu-projecte/--merge_requests/new?merge_request%5Bsource_branch%5D=master
remote:
To repo.fib.upc.es: /e1-meu-projecte.git
* [new branch] master -> master
branch 'master' set up to track 'origin/master'.
```

Fig. 11 - Mostra de resultats de creació de projecte.

Unir-se a un projecte existent

Per unir-se a un projecte és molt més senzill que el pas de creació. Només hem d'anar allà on volem crear el directori que tindrà el nostre projecte i executar la comanda:

```
git clone git@repo.fib.upc.edu:<nom.usuari o grup>/el-meu-projecte.git
```

A partir d'aquí ja tindrem tota la configuració i dades del projecte a la carpeta que ens crearà. Per defecte ens entra a la branca 'main'. Podem canviar de branca amb la instrucció 'git checkout <branca>'.

Podem veure un exemple de tot això a la següent imatge:

```
$ git clone git@repo.fib.upc.edu:~/el-meu-projecte.git
Cloning into 'el-meu-projecte'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 10 (delta 2), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (10/10), done.
Resolving deltas: 100% (2/2), done.

MINGW64 /f
$ cd el-meu-projecte/

MINGW64 /f/el-meu-projecte (main)
$ ls
README.md

MINGW64 /f/el-meu-projecte (main)
$ git checkout master
Switched to a new branch 'master'
branch 'master' set up to track 'origin/master'.

MINGW64 /f/el-meu-projecte (master)
$ ls
LICENSE
```

Fig. 12 - Mostra d'unió a un projecte existent.