



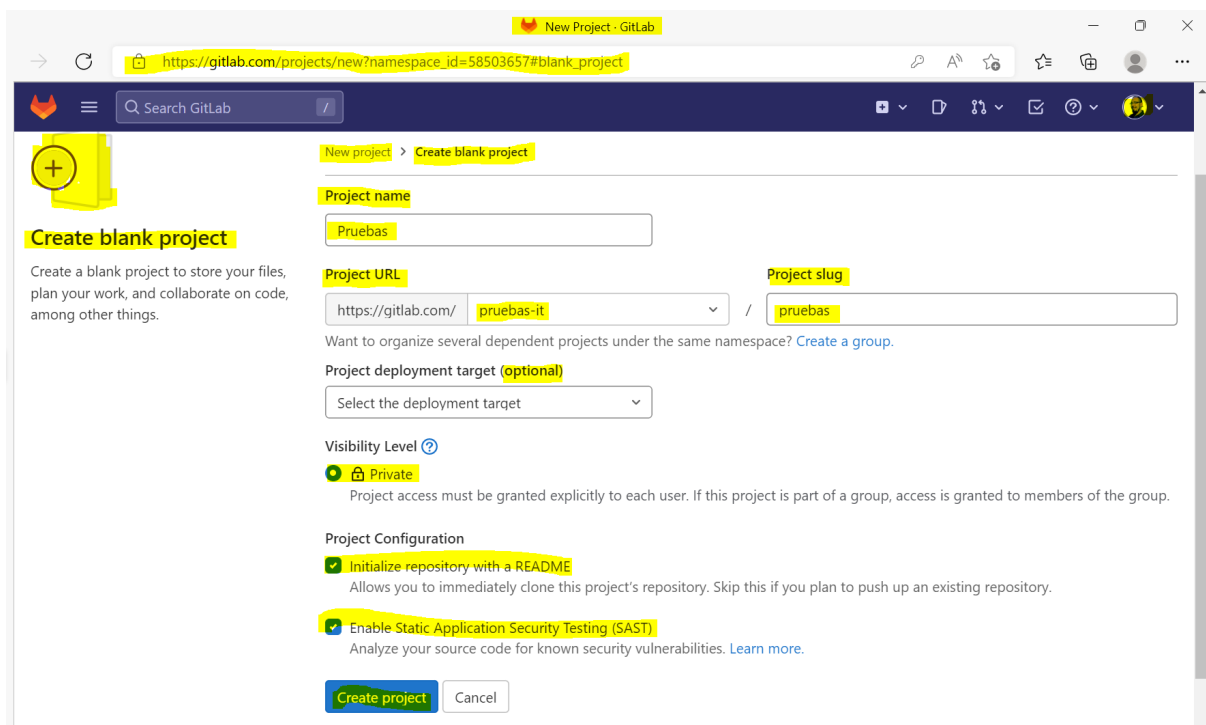
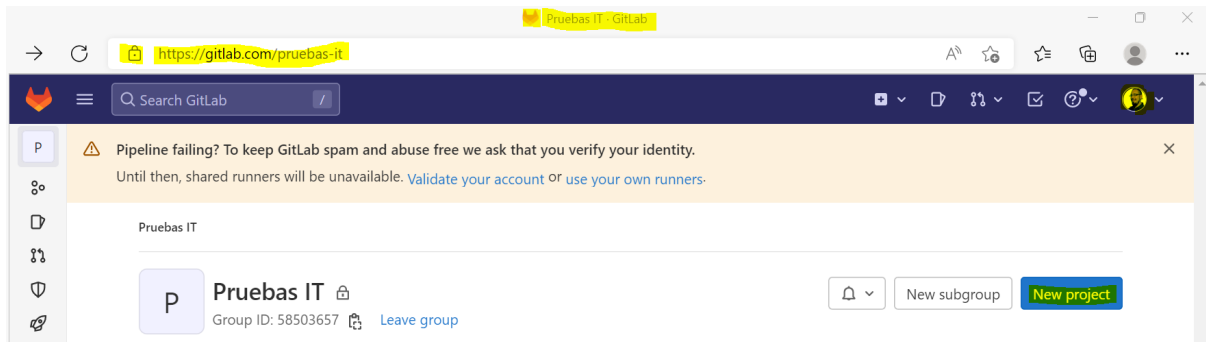
Requisitos:

- Cuenta GitLab.
- Configuración de acceso SSH por Token.
- Acceso a lectura de repositorios por proyecto.
- Acceso de escritura sobre repositorios nuevos por proyecto.

Crear:

Construir un nuevo repositorio.

- Acceder a la cuenta GitLab.
- Ingresar al proyecto.
- Crear un nuevo proyecto.





Pruebas IT

Group ID: 58503657 [Leave group](#)

[New subgroup](#) [New project](#)

Subgroups and projects Shared projects Archived projects

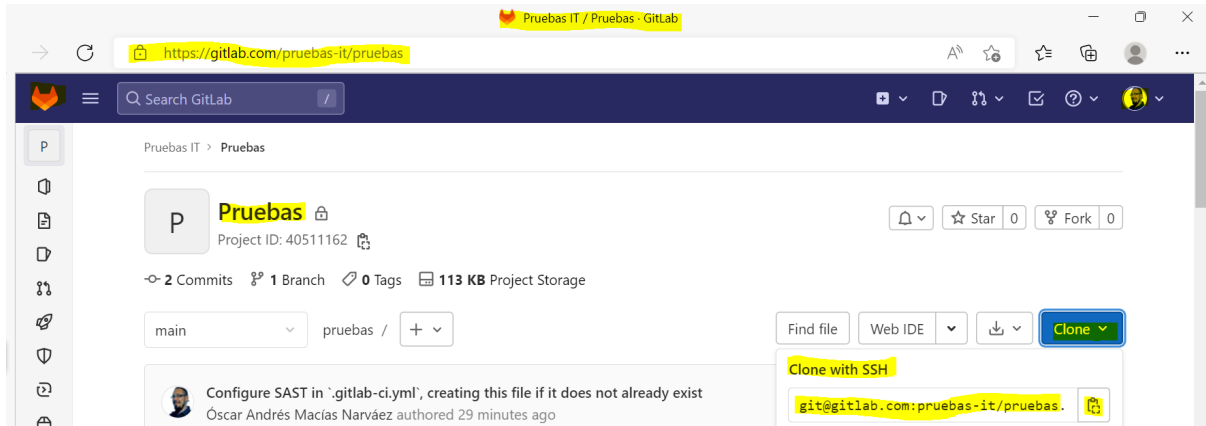
Search by name Name

Subgroup	Stars	Created
H helpers	0	5 days ago
I it_etecnic	0	5 days ago
I it_ocpi	0	5 days ago
M models	0	5 days ago
P Pruebas	0	just now

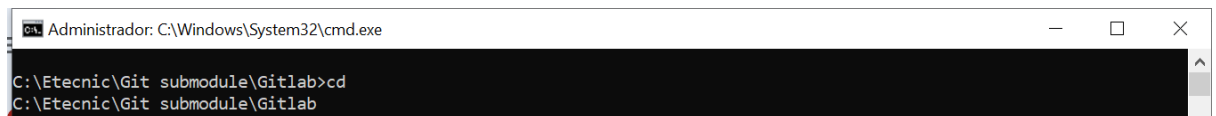
Clonar.

Replicar un nuevo repositorio.

- Ingresar al repositorio.
- Evidenciar URL del repositorio.
- Copiar enlace para clonar desde SSH.



- Acceder a la CLI desde la herramienta CMD sobre la barra de exploración de Windows 10 en la ruta absoluta donde se genera la clonación del proyecto (También puede acceder directamente desde la consola con el comando “**cd ruta\absoluta**”).
- Con el comando “**cd**” evidenciamos en que directorio actual nos encontramos posicionados (símil de “**pwd**” para Ubuntu).

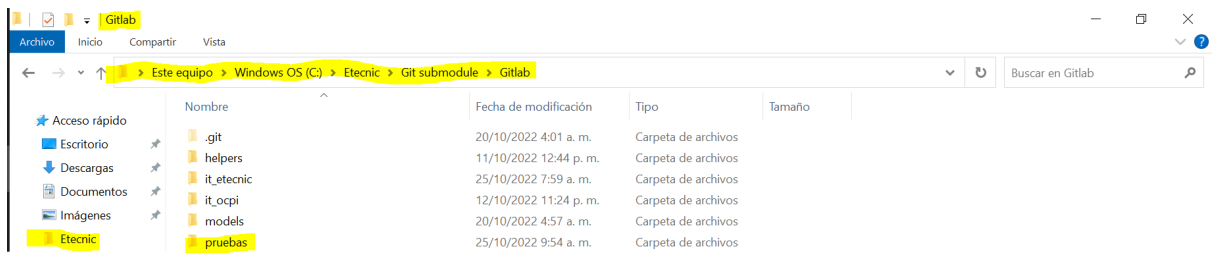


- Realizamos la descarga de la copia del repositorio global privado o remoto con la sentencia “**git clone git@gitlab:proyecto/repositorio.git**”.

```
C:\Etecnic\Git submodule\Gitlab>git clone git@gitlab.com:pruebas-it/pruebas.git
Cloning into 'pruebas'...
remote: Enumerating objects: 6, done.
remote: Counting objects: 100% (6/6), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 6 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (6/6), done.
Resolving deltas: 100% (1/1), done.
```



- Evidenciamos la salida de la sentencia anterior sobre el directorio padre.



Añadir un nuevo módulo al repositorio.

- Acceder al repositorio local desde la CLI.

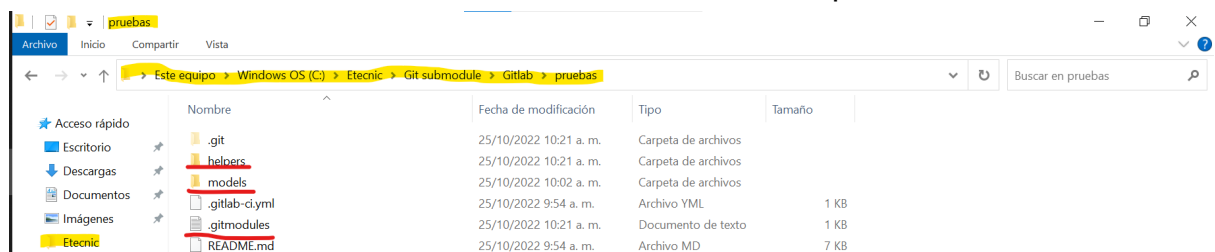
```
C:\Etecnic\Git submodule\Gitlab>cd pruebas
C:\Etecnic\Git submodule\Gitlab\pruebas>
```

- Imputar sobre la CLI (ruta absoluta donde se compartirá el repositorio) la sentencia:

git submodule add git@gitlab:proyecto/repositorio.git

```
C:\Etecnic\Git submodule\Gitlab\pruebas>git submodule add git@gitlab.com:pruebas-it/models.git
Cloning into 'C:/Etecnic/Git submodule/Gitlab/pruebas/models'...
remote: Enumerating objects: 951, done.
remote: Counting objects: 100% (110/110), done.
remote: Compressing objects: 100% (105/105), done.
remote: Total 951 (delta 6), reused 104 (delta 4), pack-reused 841
Receiving objects: 100% (951/951), 1.01 MiB | 4.20 MiB/s, done.
Resolving deltas: 100% (195/195), done.
warning: in the working copy of '.gitmodules', LF will be replaced by CRLF the next time Git touches it
```

- Evidenciamos la salida de la sentencia anterior sobre el repositorio local.



- Abrimos el archivos configuración oculto “.gitmodules”
- Evidenciamos que se añaden los módulos compartidos. la ruta y enlace de conexión.

```
.gitmodules: Bloc de notas
Archivo Edición Formato Ver Ayuda
[submodule "models"]
    path = models
    url = git@gitlab.com:pruebas-it/models.git
[submodule "helpers"]
    path = helpers
    url = git@gitlab.com:pruebas-it/helpers.git
```

Iniciar.

- Luego de haber clonado el módulo, se debe iniciar el servicio de submodule sobre git con la siguiente sentencia.

git submodule init

```
C:\Etecnic\Git submodule\Gitlab\pruebas>git submodule init
```

Actualizaciones.

- Para globalizar los cambios globales en el repositorio sobre el local se debe hacer uso de la siguiente sentencia:

git submodule update --remote

```
C:\Etecnic\Git submodule\Gitlab\it_etecnic\app>git submodule update --remote
Cloning into 'C:/Etecnic/Git submodule/Gitlab/it_etecnic/app/helpers'...
Cloning into 'C:/Etecnic/Git submodule/Gitlab/it_etecnic/app/models'...
Submodule path 'helpers': checked out '0aa146d268f304b278b73b3e4a54d5b16076999b'
Submodule path 'models': checked out '7c97b5bd414e9c4673ed3c04ae023a4a68f9d1a1'
```

Troubleshooting.



Con el uso de submodule en Git no se recomienda hacer **pull** local ya que genera un **delete** total al hacer un status del proyecto y al enviar **add -A** puede eliminar los recursos globales al enviar el **commit** y hacer **push** sobre el **origin** o **HEAD**.



Para hacer uso de un **recursive** (Actualiza todos los repositorios locales dentro de un proyecto en general, a diferencia de **update** que solo con el posicionado aplica cambios como local) en Windows se debe habilitar **WSL** para ejecutar la clonación del repositorio desde Ubuntu.



Desde Ubuntu no es práctico de primera mano hacer **login** en GitLab (las dependencias de APT no encuentran mirrors para instalar el paquete en local) sobre la Shell.