



# Certified Tech Developer

The Ultimate Degree

## Infraestructura II

### Crear nuestra cuenta en GitLab

Vamos a crear nuestra cuenta en GitLab, que va a ser la herramienta utilizada para llevar a la práctica nuestros Pipelines.

#### 1 - Creamos nuestra cuenta

Para ello vamos a ir a la siguiente url: [https://gitlab.com/users/sign\\_up](https://gitlab.com/users/sign_up) (Si en algún momento la URL deja de funcionar, ir a la página oficial de GitLab e ir a la opción para registrar cuenta) y rellenaremos los campos para poder crear nuestra cuenta. (En caso de que ya tengan una personal, no es necesario volver a crear otra)



GitLab.com

First name	Last name
<input type="text"/>	<input type="text"/>
Username	
<input type="text"/>	
Email	
<input type="text"/>	
<small>We recommend a work email address.</small>	
Password	
<input type="password"/>	
<small>Minimum length is 8 characters.</small>	
<input type="button" value="Register"/>	

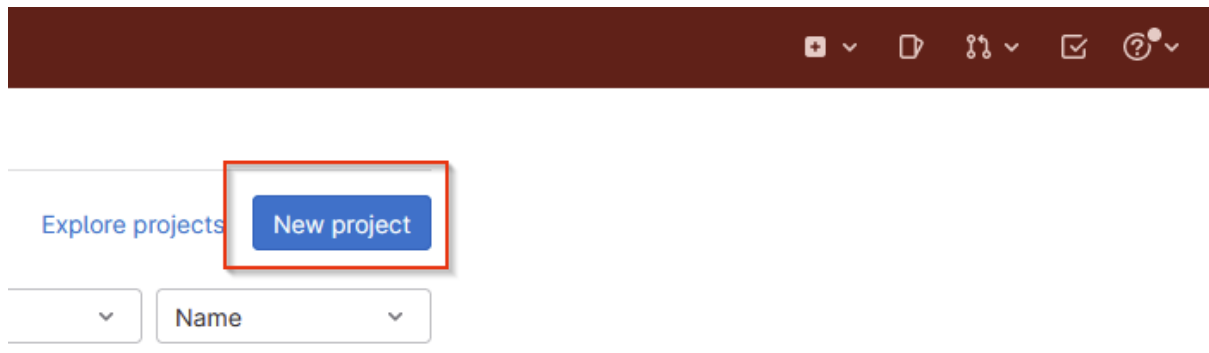
Normalmente recibiremos una confirmación y se nos informa que debemos ir a chequear nuestro email para confirmar nuestra nueva cuenta.

P/d: Revisar tanto Inbox como carpetas de Spam.

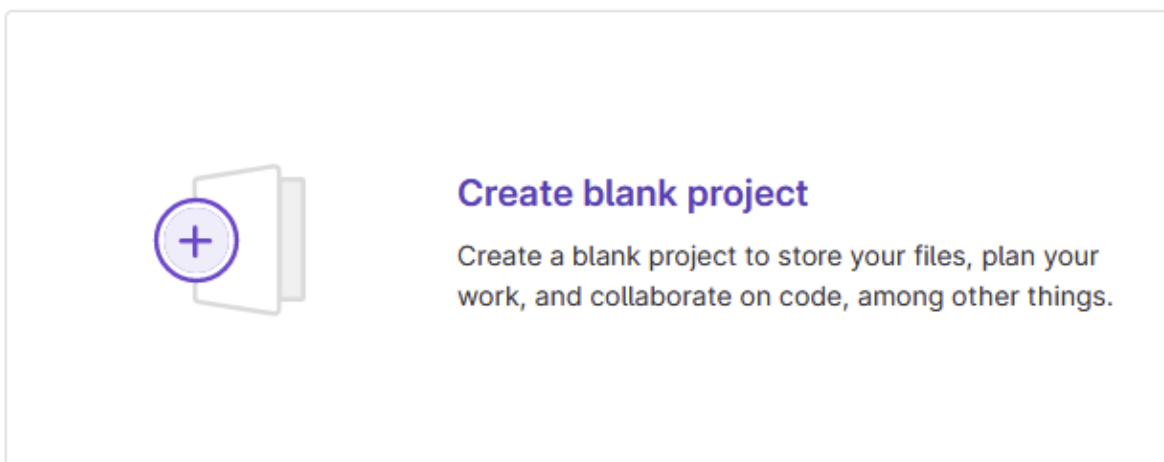
## 2 - Creamos nuestro primer proyecto

Una vez ingresados con nuestras cuentas procedemos a crear nuestro primer proyecto en donde vamos a subir algún archivo para corroborar que todo está funcionando correctamente.

Para ello vamos a hacer click en el botón "New Project"



Luego seleccionamos



Y luego rellenamos los campos con nuestros datos y luego apretamos en el botón de "Create Project". Por ejemplo:

Your work / Projects / New project / Create blank project



## Create blank project

Create a blank project to store your files, plan your work, and collaborate on code, among other things.

### Project name

Must start with a lowercase or uppercase letter, digit, emoji, or underscore. Can also contain dots, pluses, dashes, or spaces.

### Project URL

### Project slug

Want to organize several dependent projects under the same namespace? [Create a group](#).

### Project deployment target (optional)

### Visibility Level [?](#)

☒ Private

Project access must be granted explicitly to each user. If this project is part of a group, access is granted to members of the

☐ Public

The project can be accessed without any authentication.

### Project Configuration

☒ Initialize repository with a README

Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

☐ Enable Static Application Security Testing (SAST)

Analyze your source code for known security vulnerabilities. [Learn more](#).

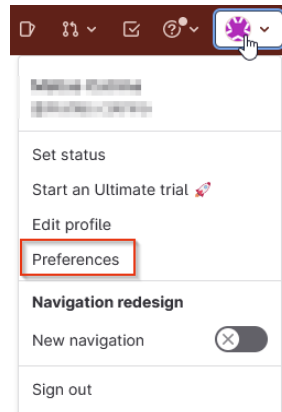
Una vez que creamos nuestro proyecto vamos a crear nuestra clave ssh que usaremos mas adelante para comunicarnos con GitLab (En nuestra terminal de GitBash en caso de que usemos Windows, sino en la Terminal propiamente dicha en Linux). Ejecutamos el comando **ssh-keygen -t rsa** sin configurar passphrase como se muestra en la foto. Es decir ejecutamos el comando antes mencionado y apretamos 3 veces ENTER.

```
MINGW64/c/Users/matia
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/c:/Users/matia/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c:/Users/matia/.ssh/id_rsa
Your public key has been saved in /c:/Users/matia/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:bxEa13Ljb1Cv9BQDbFSkhYX4u9TfHhcxbw/4pnuJWaY matia@Matias
The key's randomart image is:
+---[RSA 3072]-----+
|      +o*=      |
|      ..=+      |
| o  *oo oo      |
| B  +.o o+      |
| S  + oo+.o     |
|      . =o+=oo   |
|      o.+B+++   |
|      . .Eoo =   |
|      oo ..     |
+---[SHA256]-----+
MINGW64 ~
$
```

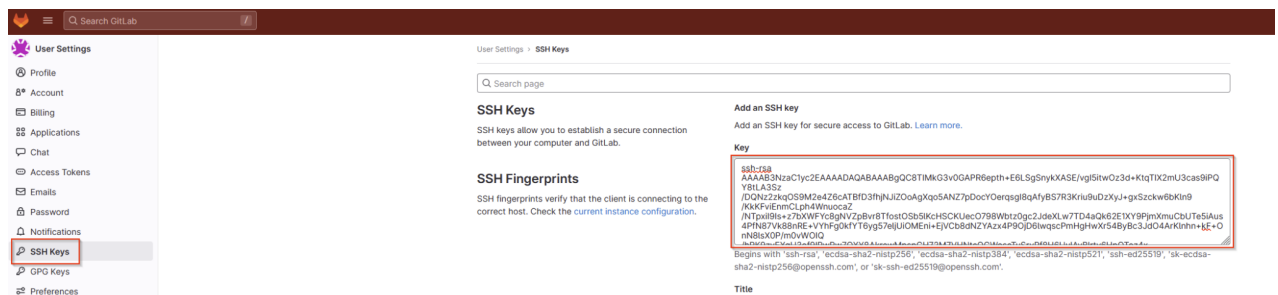
Esta clave se crea en el directorio .ssh.

```
MINGW64/c/Users/matia/.ssh
$ ls
id_rsa  id_rsa.pub  known_hosts
MINGW64 ~/.ssh
$ |
```

Luego de crear nuestro clave ssh nos vamos a nuestra cuenta y hacemos click en preferencias



Y vamos a la opción **SSH Keys** del menú izquierdo y pegamos en el campo **Key** el contenido de la clave **id\_rsa.pub**



Podemos optar por quitarle la fecha de expiración para que nos quede de la siguiente manera y no tener que renovarla después de ese periodo. (Como buena práctica es recomendable que tenga una fecha en lo posible de 1 mes máximo).

### Expiration date

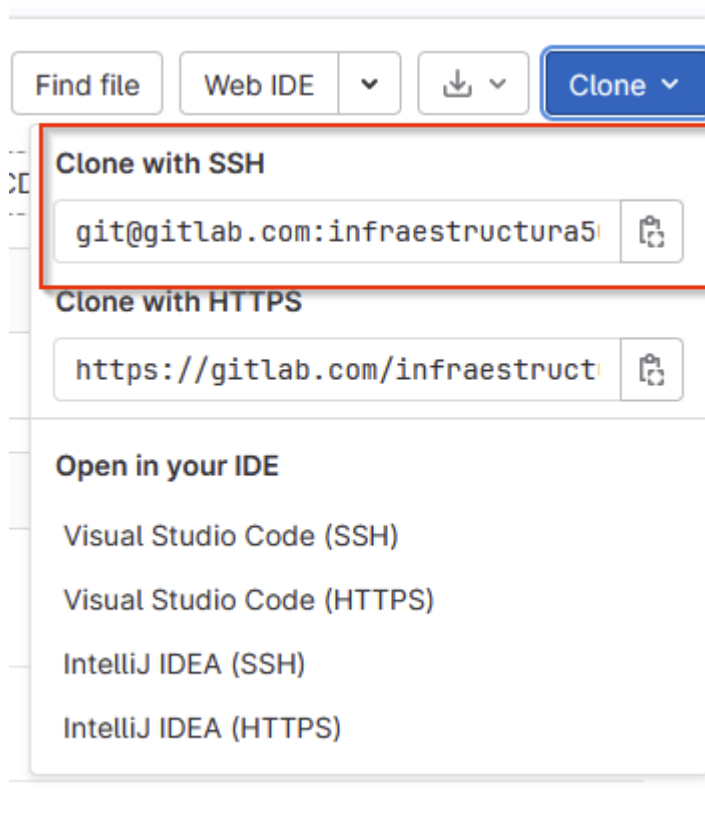
Add key

Hacemos click en el botón **Add key** y esto guardará nuestra llave en nuestra cuenta. Una vez hecho esto, nos tocaría probar que realmente podemos clonar nuestro repositorio. Para ello configuraremos primero nuestra cuenta utilizando los comandos:

```
git config --global user.name "TU NOMBRE"
```

```
git config --global user.email "TU@CORREO.COM"
```

Después vamos a nuestro proyecto y en el botón Clone hacemos click para luego copiar la dirección que figura en la imagen;



¡Ahora podremos clonar nuestro repositorio y comenzar a subir algún proyecto/archivo que tengas para visualizarlo en el repositorio de GitLab!