

# Práctica Parte II

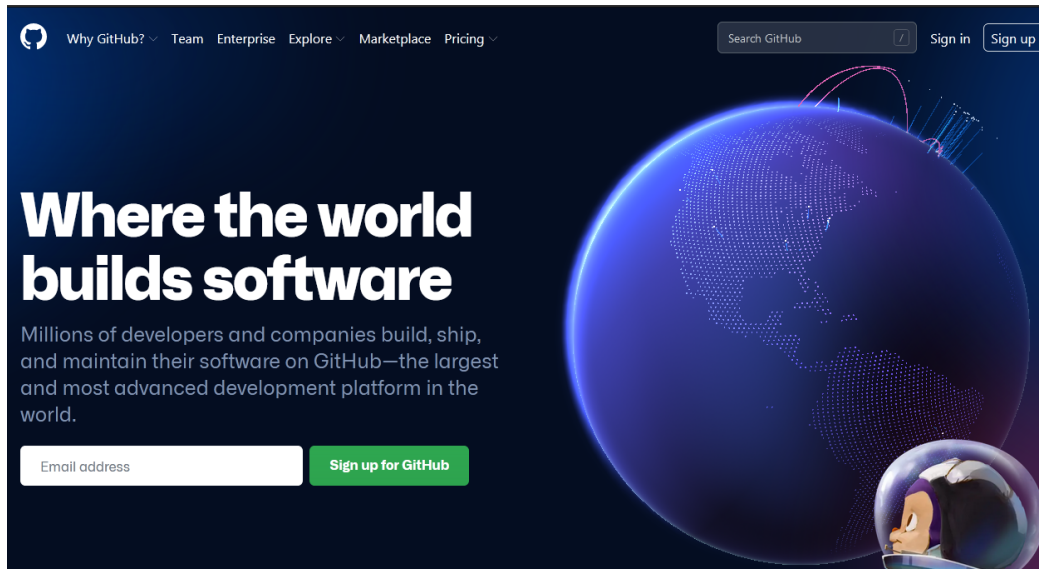
# Git & GitHub

---

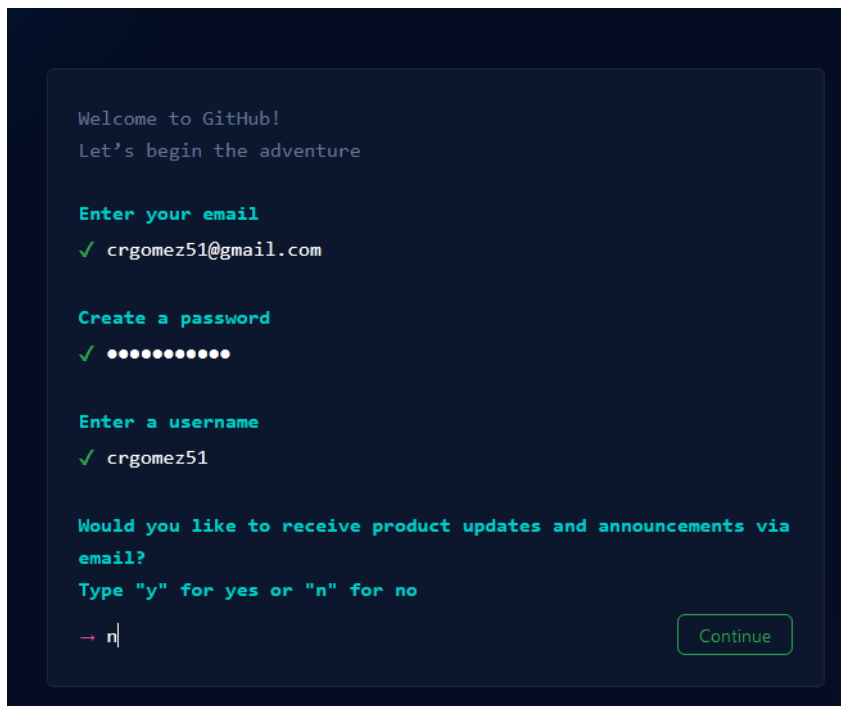


## Crear una cuenta

1. Para crear una cuenta **GitHub** vamos a seguir los siguientes pasos..

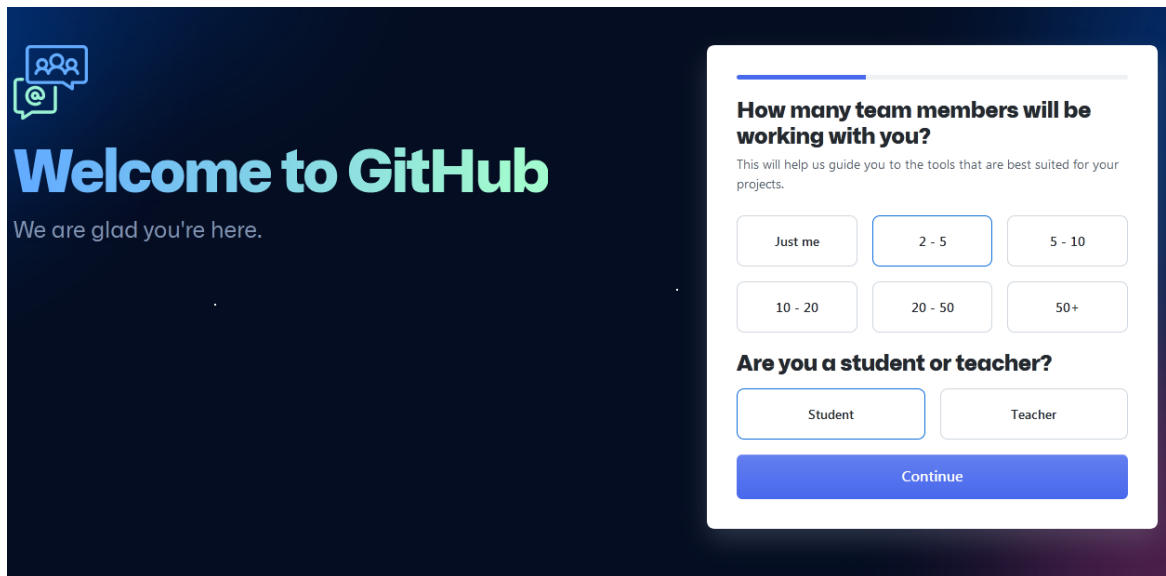


Nos registramos en **github** como se ve a continuación



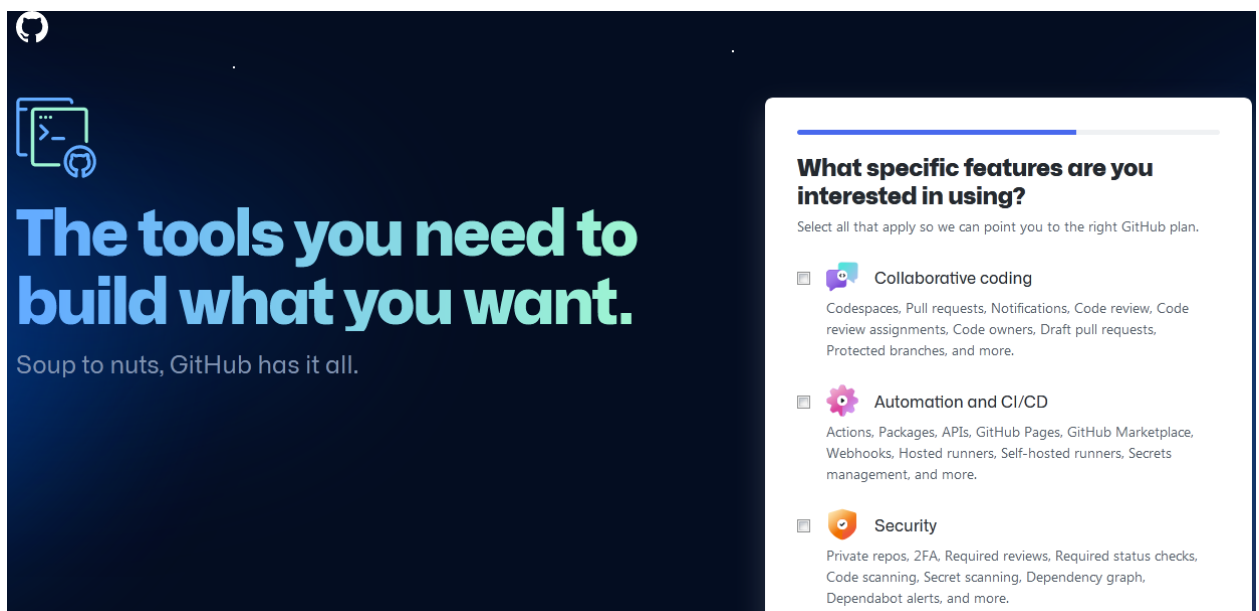
Nos llegará un **código** a nuestro correo, el cual debemos ingresar para **validar nuestra cuenta**.

Luego de eso seleccionamos las siguientes opciones reglamentarias de github acerca de nuestra cuenta.



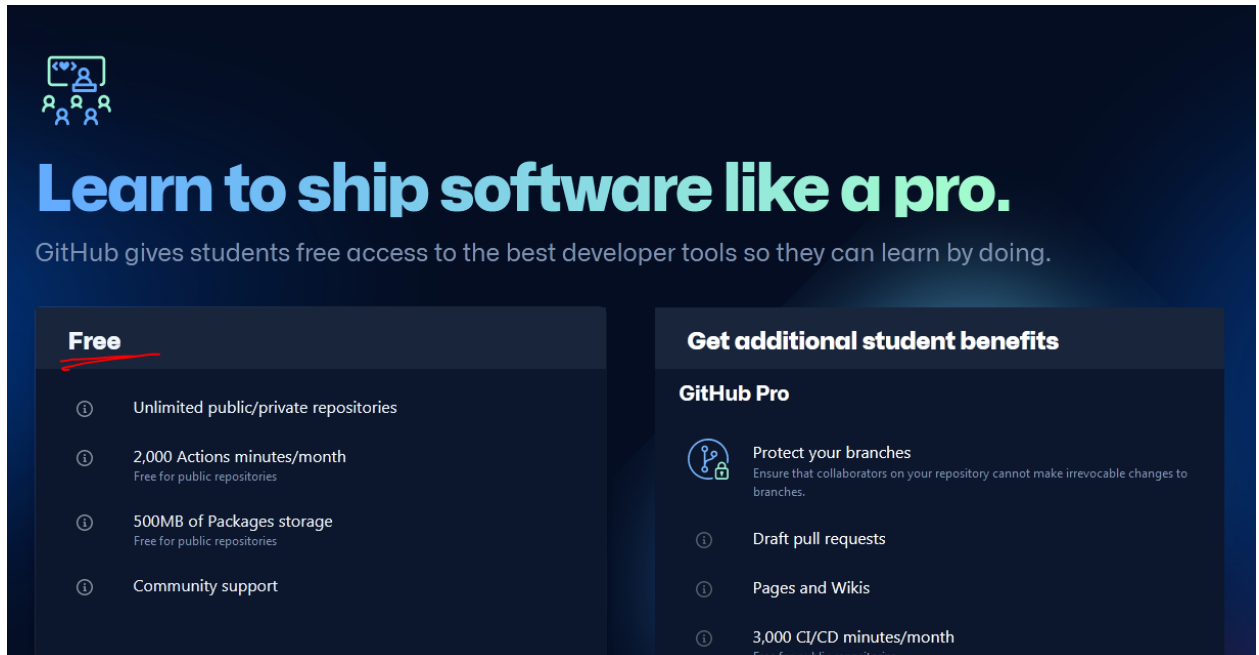
The image shows the GitHub welcome screen. On the left, there's a dark blue background with the GitHub logo (three people icons) and the text "Welcome to GitHub" in large, bold, light blue letters. Below it, in smaller white text, it says "We are glad you're here." On the right, there's a white panel with a blue border. It contains the question "How many team members will be working with you?" in bold. Below this, a small line of text says "This will help us guide you to the tools that are best suited for your projects." There are six buttons arranged in two rows: "Just me", "2 - 5", "5 - 10" in the first row, and "10 - 20", "20 - 50", "50+" in the second row. The "2 - 5" button is highlighted with a blue border. Below these buttons is the question "Are you a student or teacher?" in bold. There are two buttons: "Student" and "Teacher". The "Student" button is highlighted with a blue border. At the bottom of the white panel is a large blue button labeled "Continue".

La siguiente ventana nos muestra algunas herramientas que podríamos utilizar en el futuro, por el momento no seleccionamos ninguna de ellas y le damos en siguiente.



The image shows the GitHub "The tools you need to build what you want." screen. On the left, there's a dark blue background with the GitHub logo (code icon) and the text "The tools you need to build what you want." in large, bold, light blue letters. Below it, in smaller white text, it says "Soup to nuts, GitHub has it all." On the right, there's a white panel with a blue border. It contains the question "What specific features are you interested in using?" in bold. Below this, a small line of text says "Select all that apply so we can point you to the right GitHub plan." There are three sections, each with a checkbox and a title: "Collaborative coding" (with a people icon), "Automation and CI/CD" (with a gear icon), and "Security" (with a shield icon). Each section has a list of features below it. The "Collaborative coding" section lists: Codespaces, Pull requests, Notifications, Code review, Code review assignments, Code owners, Draft pull requests, Protected branches, and more. The "Automation and CI/CD" section lists: Actions, Packages, APIs, GitHub Pages, GitHub Marketplace, Webhooks, Hosted runners, Self-hosted runners, Secrets management, and more. The "Security" section lists: Private repos, 2FA, Required reviews, Required status checks, Code scanning, Secret scanning, Dependency graph, Dependabot alerts, and more. All checkboxes are currently unchecked.

Después, seleccionamos el tipo de cuenta que utilizaremos. Por supuesto le daremos el plan gratuito.



The image shows the GitHub Student Developer Pack landing page. At the top, there's a logo with three people icons. Below it, the headline reads "Learn to ship software like a pro." followed by the subtext "GitHub gives students free access to the best developer tools so they can learn by doing." The page is divided into two main sections: "Free" and "Get additional student benefits".

**Free**

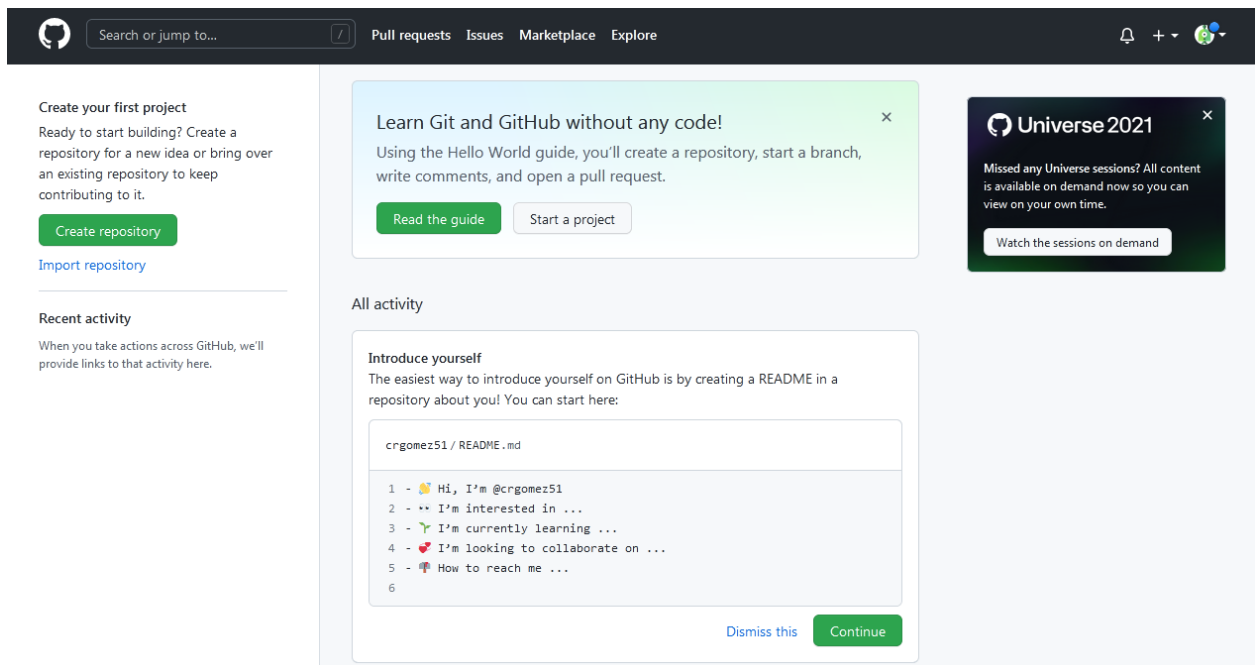
- ① Unlimited public/private repositories
- ① 2,000 Actions minutes/month  
Free for public repositories
- ① 500MB of Packages storage  
Free for public repositories
- ① Community support

**Get additional student benefits**

**GitHub Pro**

- ① **Protect your branches**  
Ensure that collaborators on your repository cannot make irrevocable changes to branches.
- ① Draft pull requests
- ① Pages and Wikis
- ① 3,000 CI/CD minutes/month  
Free for public repositories

Listo ya habremos terminado con la creación de la cuenta.



The image shows the GitHub homepage with a dark header. The main content area has a light blue background. On the left, there's a sidebar with "Create your first project" and "Recent activity" sections. The main area features several onboarding prompts and a "All activity" section.

**Create your first project**  
Ready to start building? Create a repository for a new idea or bring over an existing repository to keep contributing to it.  
[Create repository](#)  
[Import repository](#)

**Recent activity**  
When you take actions across GitHub, we'll provide links to that activity here.

**Learn Git and GitHub without any code!**  
Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.  
[Read the guide](#) [Start a project](#)

**Universe 2021**  
Missed any Universe sessions? All content is available on demand now so you can view on your own time.  
[Watch the sessions on demand](#)

**All activity**

**Introduce yourself**  
The easiest way to introduce yourself on GitHub is by creating a README in a repository about you! You can start here:

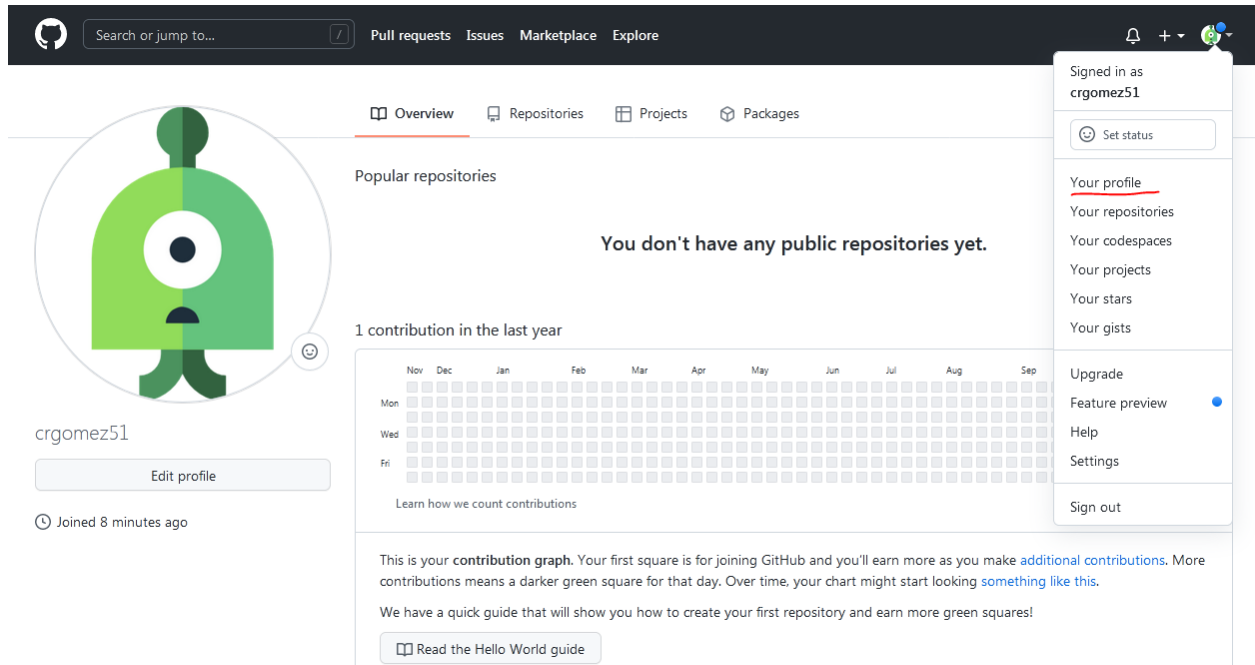
```
crngomez51 / README.md

1 - 🙋 Hi, I'm @crngomez51
2 - ** I'm interested in ...
3 - 🌱 I'm currently learning ...
4 - ❤️ I'm looking to collaborate on ...
5 - 📫 How to reach me ...
6
```

[Dismiss this](#) [Continue](#)

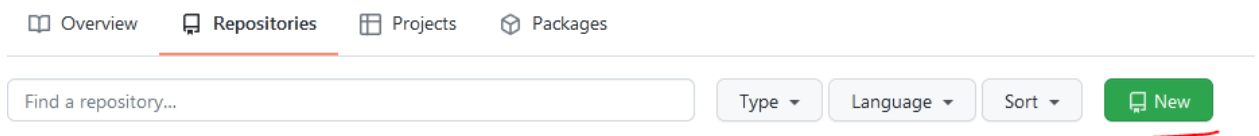
## Crear un nuevo repositorio

1. Vamos al icono de nuestra **cuenta** y le damos nuestro **perfil**



Ahí podremos ver toda nuestra actividad y nuestros repositorios.

2. En la sección de **repositorios** nos vamos a crear un **nuevo repositorio**




crgomez51 doesn't have any public repositories yet.

### 3. Creamos el siguiente repositorio con los siguientes datos

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*

 crgomez51 ▾

Repository name \*

/ prueba ✓

Great repository names are short and memorable. Need inspiration? How about **redesigned-system**?

Description (optional)

Este es un repositorio de prueba



**Public**

Anyone on the internet can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.

**Initialize this repository with:**

Skip this step if you're importing an existing repository.

☐ **Add a README file**

This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**


Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**

A license tells others what they can and can't do with your code. [Learn more.](#)

4. Nos debería salir el setup inicial de nuestro repositorio desde git

**Quick setup — if you've done this kind of thing before**

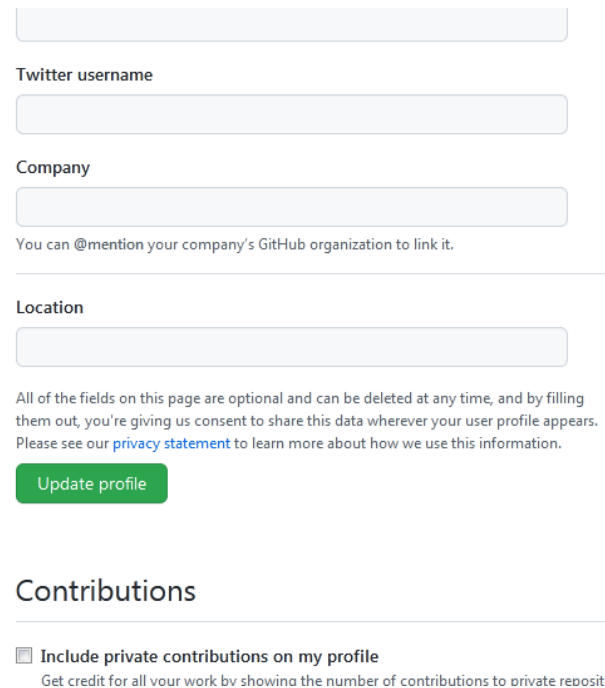
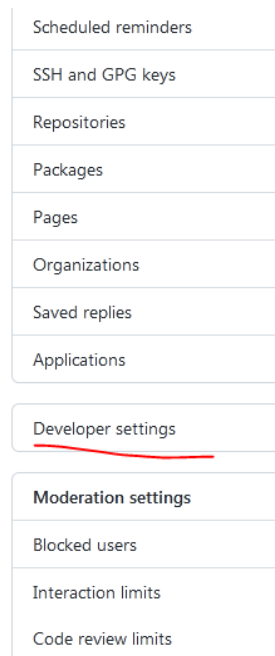
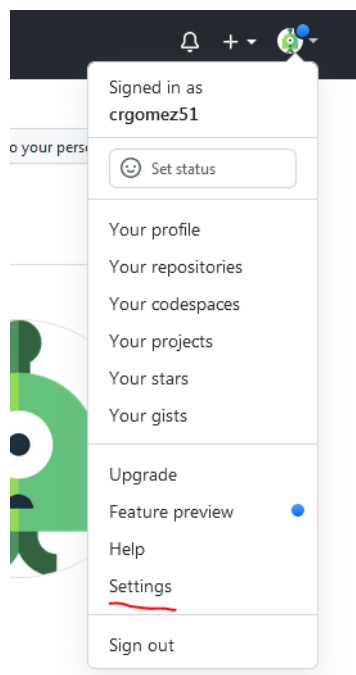
 Set up in Desktop or **HTTPS** **SSH** <https://github.com/crgomez51/prueba.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

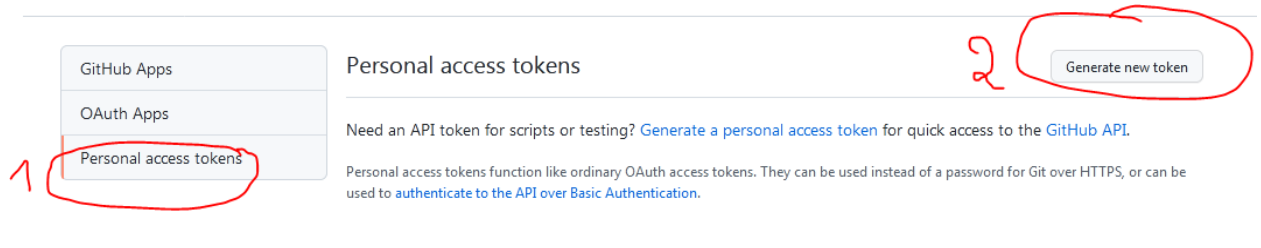
**...or create a new repository on the command line**

```
echo "# prueba" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/crgomez51/prueba.git
git push -u origin main
```

5. Ahora nos vamos **al icono de nuestra cuenta** y nos vamos a **configuraciones (settings)**, Luego vamos a la opción de **developer settings**



6. Nos debería salir la siguiente ventana y le damos a **Personal Access tokens**



7. Luego le damos a **generate new token**, e ingresamos el nombre (puede ser el nombre de su computadora) y seleccionamos en expiración **No expiration**

**New personal access token**

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

**Note**

mi-pc

What's this token for?

**Expiration \***

No expiration The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help keep your information secure. [Learn more](#)

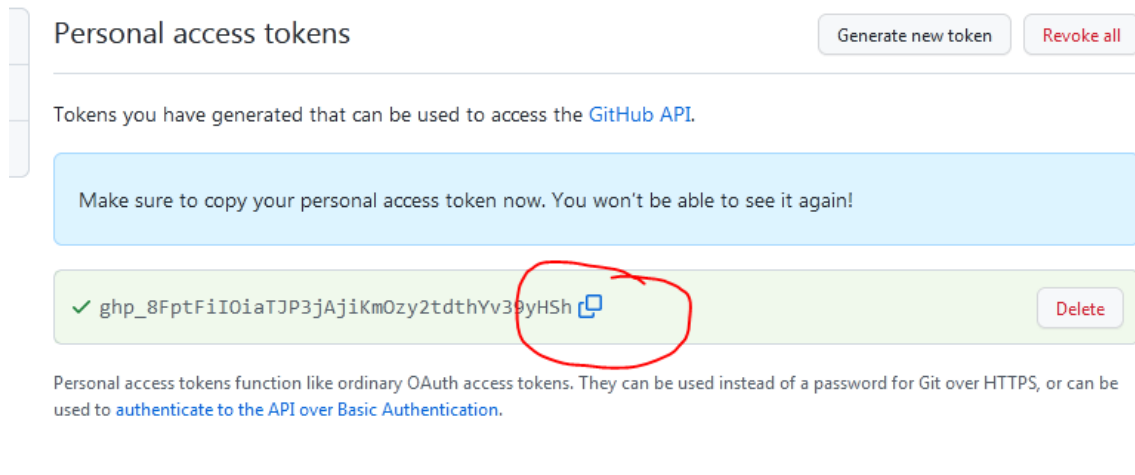
**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> <b>repo</b>	Full control of private repositories
<input checked="" type="checkbox"/> repo:status	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repo:invite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> <b>workflow</b>	Update GitHub Action workflows



8. Luego nos saldrá nuestros **tokens**, le damos al icono de copiar y lo copiamos en un **bloc de notas**, para que esté a la mano.



9. Ahora nos vamos a nuestro proyecto de la práctica anterior e ingresamos los siguientes comandos para cambiar nuestro repositorio local a remoto.

```
windows@Windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ git remote add origin https://github.com/crgomez51/prueba.git
error: remote origin already exists.

windows@Windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ git remote
origin

windows@Windows-PC MINGW64 ~/Desktop/MiProyecto (master)
$ git push origin master
```

10. Les pedira que ingrese su usuario y contraseña (aqui debemos pegar el **token** )

