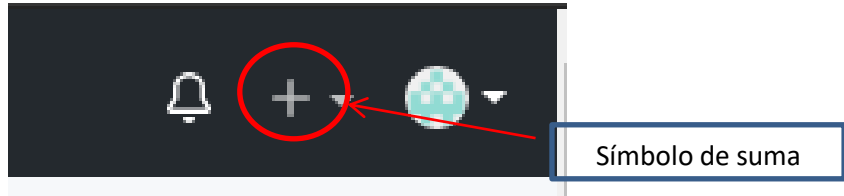


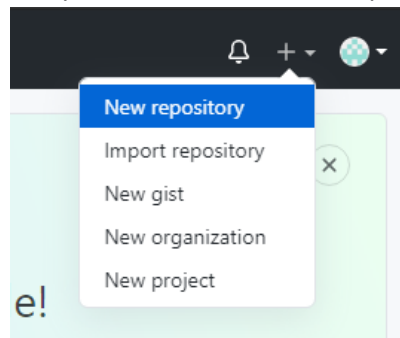
Guía para usar GitHub como herramienta de control de versiones

Sección 1.

1. Primero debemos crearnos una cuenta en GitHub, en el siguiente enlace: <https://github.com/>
2. Una vez creada la cuenta, ingresamos y nos dirigimos a la siguiente opción:



3. Se desplegarán opciones de creación y seleccionaremos new repository.




4. Para crear un repositorio tendremos las siguientes opciones:

Create a new repository

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

Owner *

 Fabricio2502 ▾

Repository name *

/

Great repository names are short and memorable. Need inspiration? How about [probable-system?](#)

Description (optional)

☒  **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.](#)

5. Establecemos un nombre a nuestro repositorio.

6. A continuación escogemos si nuestro repositorio será público o privado, si escogemos público permitiremos que otros usuarios puedan acceder y modificar a la información que se encuentre en el repositorio, pero si escogemos privado ningún otro usuario podrá acceder o modificar el contenido.
7. Debemos marcar las casillas de como deseamos inicializar nuestro repositorio:

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.](#)

Nos permite dar una descripción a cualquier documento que subamos.

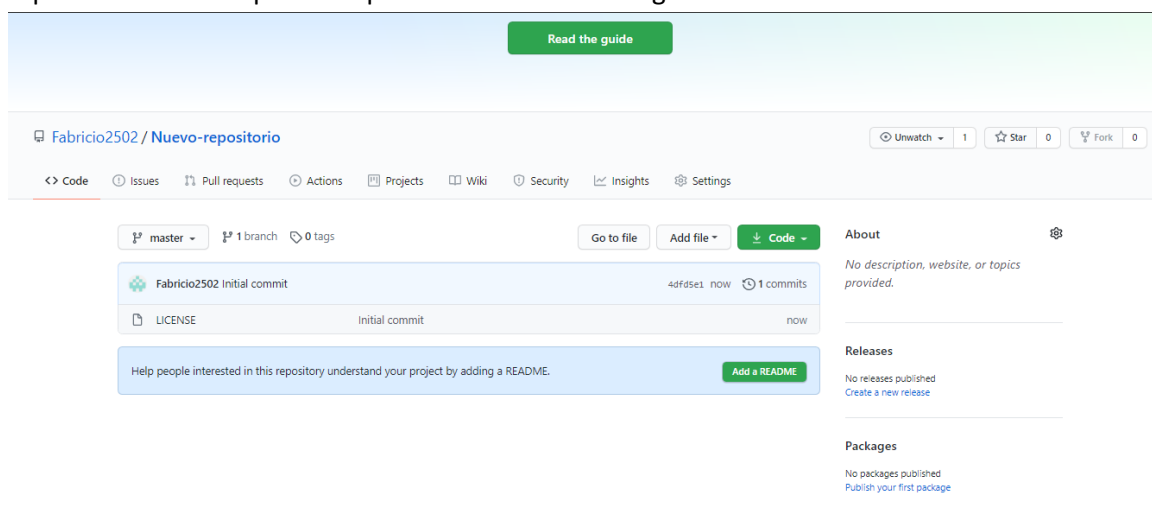
Nos permite ignorar archivos del tipo que escojamos para que no se suban al repositorio o al código.

Escoger un tipo de licencia permitirá a otros usuarios distribuir el contenido de tu repositorio libremente en el caso de que tu repositorio sea público.

8. Una vez configurado nuestro repositorio, lo creamos haciendo clic en la siguiente opción:

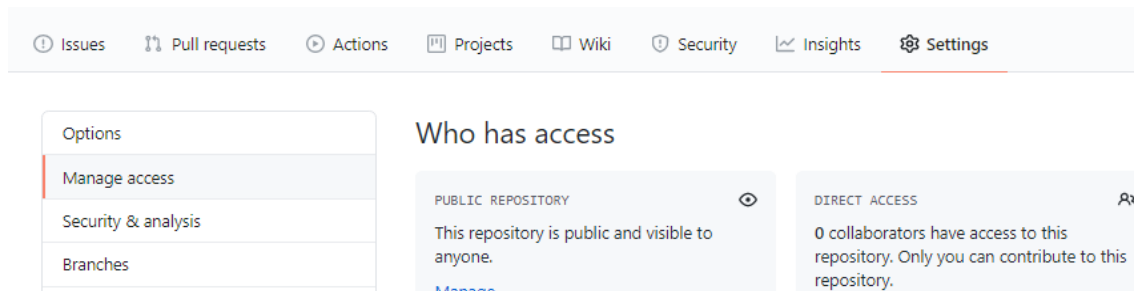
Create repository

9. Después de crear el repositorio podremos observar la siguiente ventana:

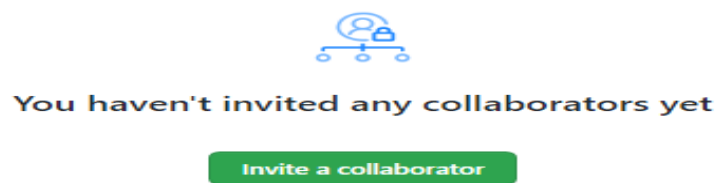


10. Para permitir que usuarios tengan acceso a nuestro repositorio y puedan modificar o descargar la información realizamos los siguientes pasos.

1. Click en Settings.
2. Nos dirigimos a manage access.



3. En la parte inferior encontraremos Invite a collaborator

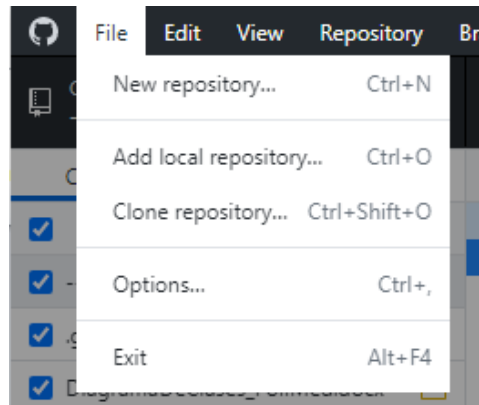


4. Agregamos a los usuarios que sean necesarios para que tengan acceso y se les enviara una solicitud de aceptación a sus respectivos correos, los invitados deberán aceptar la solicitud para tener acceso al repertorio.
11. Después de realizar todos estos pasos el dueño del repertorio y sus respectivos colaboradores agregados previamente podrán subir y descargar información del repertorio.

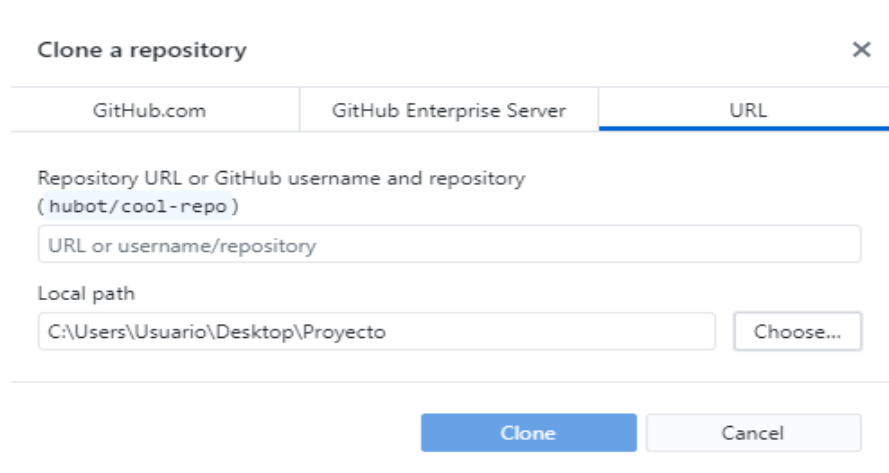
Sección 2.

GitHub también tiene su propio programa para descarga, el cual es muy útil ya que el programa nos permite usar archivos de texto plano, es decir nos facilita abrir documentos que no se abran como código en el repositorio creado en la sección 1 y de esta manera poder abrir el código, editarlo y actualizarlo quedando así las respectivas actualizaciones que se hagan a los documentos y el usuario que hizo la modificación .

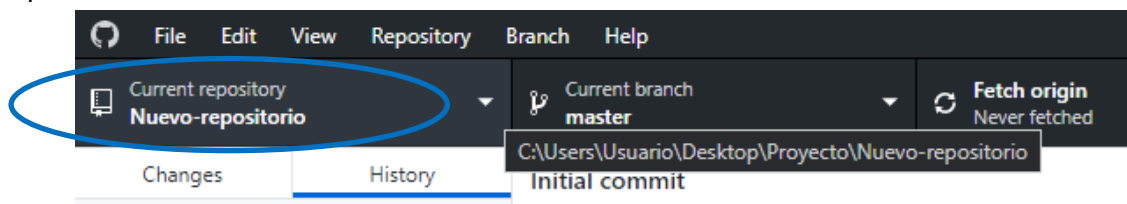
1. Para descargar GitHub Desktop podremos hacerlo desde el siguiente enlace:
<https://desktop.github.com/>
2. Una vez descargado procederemos a ingresar nuestros datos de usuario y clonaremos los repositorios que vayamos a usar.
3. Para clonar un repositorio nos dirigimos a la opciones File.



4. Seleccionamos Clone repository y podremos clonar un repositorio existente en nuestra cuenta de GitHub ingresando el URL del repositorio y en la opción de Local path podremos seleccionar el lugar donde queremos que se guarden todos los archivos de dicho repositorio.



5. Para acceder al repositorio clonado y sus respectivos archivos nos dirigimos a la siguiente opción:



Podremos ver todos los repositorios que hemos clonado

6. Para subir archivos o modificar archivos, solo abrimos el repositorio y seleccionamos cualquier archivo. A continuación crearemos un proyecto en netbeans y procederemos a subirlo como ejemplo:

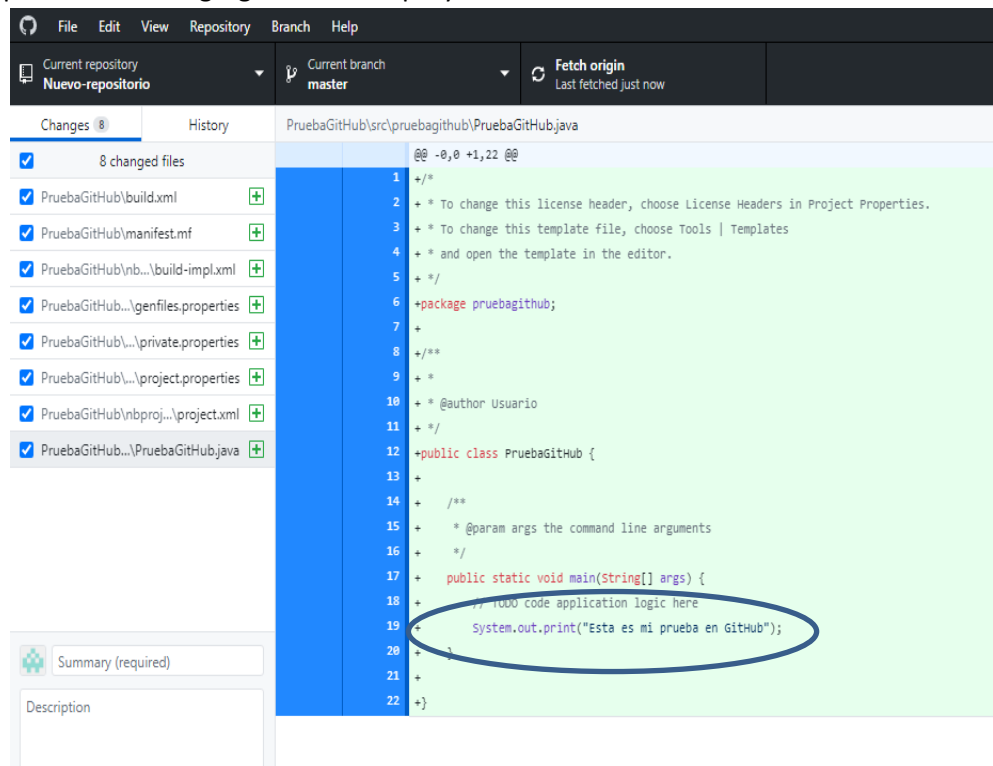
1. El nombre de nuestro proyecto en netbeans será PruebaGitHub en la cual nuestra clase main tendrá una pequeña línea de código

```

/**
 * @param args the command line arguments
 */
public static void main(String[] args) {
    // TODO code application logic here
    System.out.print("Esta es mi prueba en GitHub");
}
}

```

- Para subir el proyecto que acabamos de crear en Netbeans, debemos dirigirnos hacia el lugar donde hemos guardado nuestro repositorio clonado y copiar en esa carpeta nuestro proyecto de Netbeans, una vez copiado el proyecto veremos que en nuestro repositorio se ha agregado nuestro proyecto



Y como podemos observar se encuentra nuestra línea de código de la imagen anterior.

En el caso de que no aparezcan los archivos deberán dar clic en Fetch origin y se actualizará el repositorio.

