

Actividad 2.2: Trabajando con repositorios remotos y Github



Nicolas Primoy Pucheta

Crear una Cuenta en GitHub

Paso 1.1: Regístrate en GitHub

1. Ve a la página de GitHub en <https://github.com>.
2. Haz clic en Sign up (Regístrate) y sigue las instrucciones para crear una cuenta con tu correo electrónico.
3. Después de completar el registro, verifica tu correo electrónico si GitHub te lo solicita.

Paso 1.2: Configuración básica de tu cuenta de GitHub

1. Inicia sesión en tu cuenta de GitHub.
2. En la parte superior derecha, haz clic en tu perfil y selecciona Settings (Configuración).
3. Revisa y ajusta tus preferencias, como el idioma de la interfaz y las notificaciones de correo electrónico.

Configurar Git para Trabajar con GitHub

Paso 2.1: Configura tu nombre de usuario y correo electrónico en Git (si no lo has hecho ya). Deberías haberlo hecho en la actividad 2.1.

Paso 2.2: Configurar SSH o HTTPS para conectarte con GitHub.

GitHub permite dos métodos principales para conectarte a repositorios: HTTPS y SSH. Te recomendamos configurar SSH para mayor seguridad.

- Generar una clave SSH (si aún no tienes una): En tu terminal, ejecuta el siguiente comando para generar una nueva clave SSH:

```
ssh-keygen -t ed25519 -C "tuemail@ejemplo.com"
```

Sigue las instrucciones, presionando **Enter** cuando te pida el nombre del archivo y la ubicación. Luego, añade la clave pública a tu agente SSH:

```
eval "$(ssh-agent -s)"
```

```
ssh-add
```

```
~/.ssh/id_ed25519
```

- Añadir tu clave SSH a GitHub: Copia tu clave pública con el siguiente comando:

```
cat ~/.ssh/id_ed25519.pub
```

Entra en tu cuenta de GitHub y navega a Settings > SSH and GPG keys. Haz clic en New SSH key, ponle un nombre y pega tu clave pública en el campo correspondiente.

- Probar la conexión: Para verificar que la conexión con GitHub a través de SSH funciona correctamente, ejecuta:

```
ssh -T git@github.com
```

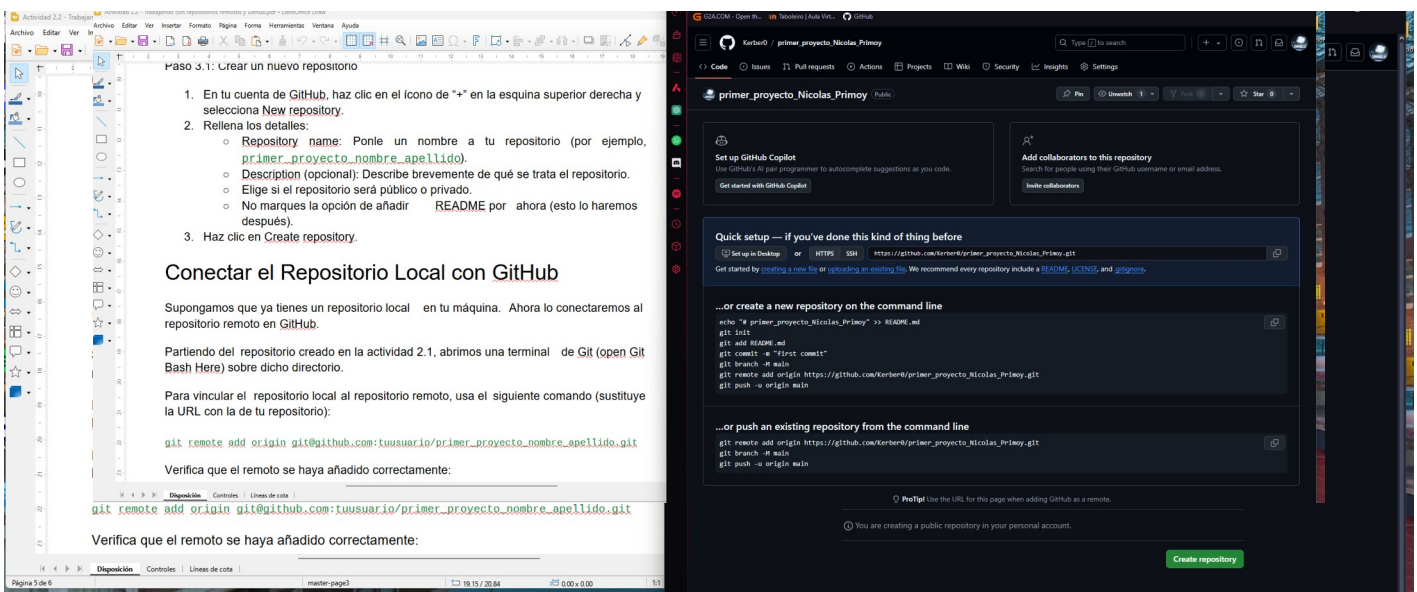
Si la configuración es correcta, deberías ver un mensaje de bienvenida.

The image shows two side-by-side screenshots. The left screenshot is from a PDF document titled '2.2 - Trabajando con repositorios remotos y GitHub.pdf'. It contains a list of steps for configuring Git and GitHub, followed by a section 'Configurar Git para Trabajar con GitHub' with sub-steps 'Paso 2.1' and 'Paso 2.2'. The right screenshot is a screenshot of the GitHub 'Settings' page for a user named 'Kerber0'. The 'SSH and GPG keys' section is active, showing a list of SSH keys. One key is listed with the name 'Pc-Instituto' and a public key. There are buttons for 'New SSH key' and 'Delete'. Below this, the 'GPG keys' section is shown with a 'New GPG key' button. The 'Vigilant mode' section is also visible, indicating that unsigned commits are flagged as unverified.

Crear un Repositorio en GitHub

Paso 3.1: Crear un nuevo repositorio

1. En tu cuenta de GitHub, haz clic en el ícono de "+" en la esquina superior derecha y selecciona New repository.
2. Rellena los detalles:
 - Repository name: Ponle un nombre a tu repositorio (por ejemplo, `primer_proyecto_nombre_apellido`).
 - Description (opcional): Describe brevemente de qué se trata el repositorio.
 - Elige si el repositorio será público o privado.
 - No marques la opción de añadir README por ahora (esto lo haremos después).
3. Haz clic en Create repository.



Conectar el Repositorio Local con GitHub

Supongamos que ya tienes un repositorio local en tu máquina. Ahora lo conectaremos al repositorio remoto en GitHub.

Paso 4.1: Inicializa el repositorio local (si no lo has hecho ya). Deberías haberlo hecho en la actividad 2.1. Abre tu terminal de git y navega al directorio del proyecto local:

```
cd ruta/a/tu/proyecto
```

Si el repositorio aún no está inicializado en Git, inicialízalo:

```
git init
```

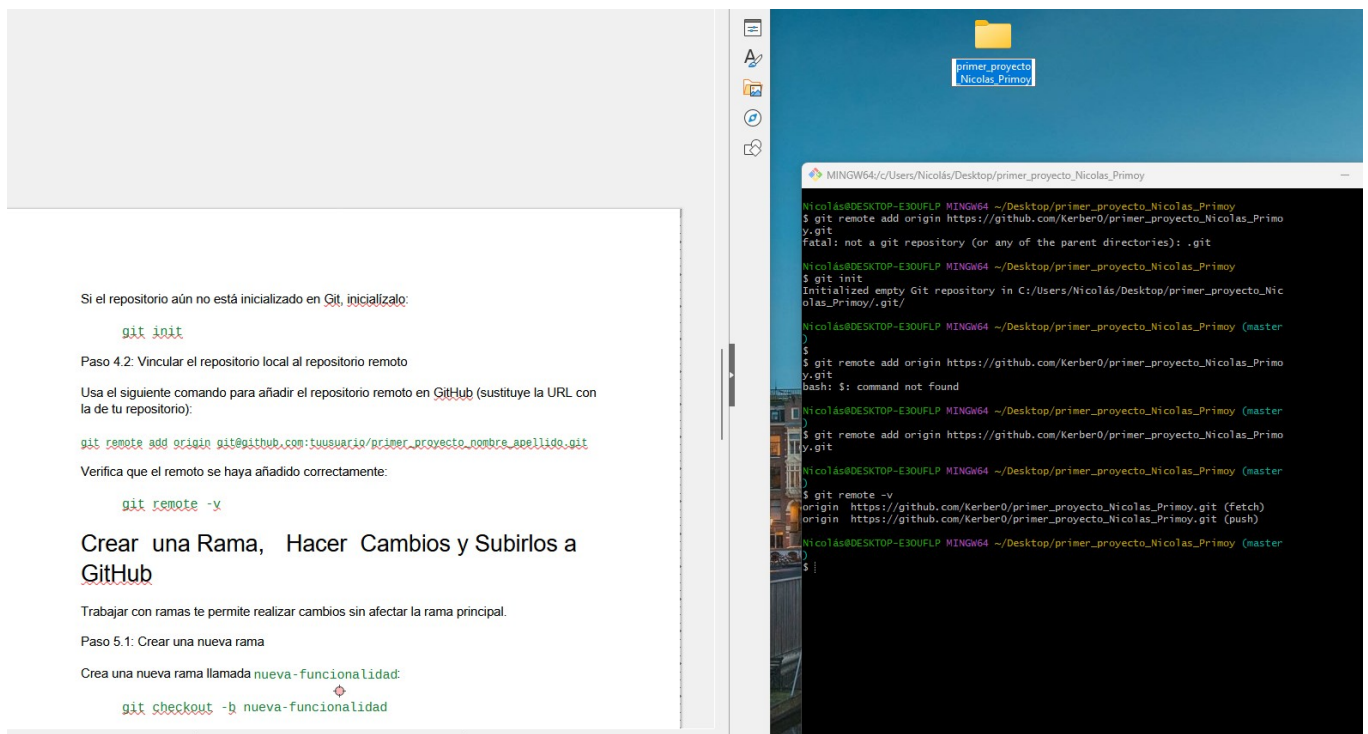
Paso 4.2: Vincular el repositorio local al repositorio remoto

Usa el siguiente comando para añadir el repositorio remoto en GitHub (sustituye la URL con la de tu repositorio):

```
git remote add origin  
git@github.com:tuusuario/primer_proyecto_nombre_apellido.git
```

Verifica que el remoto se haya añadido correctamente:

```
git remote -v
```



Crear una Rama, Hacer Cambios y Subirlos a GitHub

Trabajar con ramas te permite realizar cambios sin afectar la rama principal. Paso 5.1: Crear una nueva rama

Crea una nueva rama llamada `nueva-`

```
funcionalidad: git checkout -b  
nueva-funcionalidad
```

Ahora estarás trabajando en esta nueva rama.

Paso 5.2: Hacer cambios en tu proyecto. Modifica o añade algún archivo en tu proyecto. Por ejemplo, crea un archivo `README.md` con una descripción del proyecto.

Añade el archivo al staging area y haz un commit:

```
git add README.md  
  
git commit -m "Añadir archivo README.md"
```

Paso 5.3: Subir la nueva rama al repositorio remoto

Envía la nueva rama al repositorio remoto en

GitHub:

```
git push origin nueva-funcionalidad
```

Trabajar con ramas te permite realizar cambios sin afectar la rama principal.

Paso 5.1: Crear una nueva rama

Crea una nueva rama llamada `nueva-funcionalidad`:

```
git checkout -b nueva-funcionalidad
```

Ahora estarás trabajando en esta nueva rama.

Paso 5.2: Hacer cambios en tu proyecto. Modifica o añade algún archivo en tu proyecto. Por ejemplo, crea un archivo `README.md` con una descripción del proyecto.

Añade el archivo al staging area y haz un commit:

```
git add README.md
```

```
git commit -m "Añadir archivo README.md"
```

Paso 5.3: Subir la nueva rama al repositorio remoto

Envía la nueva rama al repositorio remoto en [GitHub](#):

```
git push origin nueva-funcionalidad
```

Fusionar la Rama con la Rama Principal

Una vez que tu trabajo en la rama `nueva-funcionalidad` está listo, es momento de fusionar los cambios con la rama principal (`main`).

```
$ git remote add origin https://github.com/Kerber0/primer_proyecto_Nicolas_Primo  
y.git  
bash: $: command not found  
  
Nicolas@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primo (master)  
$ git remote add origin https://github.com/Kerber0/primer_proyecto_Nicolas_Primo  
y.git  
  
Nicolas@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primo (master)  
$ git remote -v  
origin https://github.com/Kerber0/primer_proyecto_Nicolas_Primo.git (fetch)  
origin https://github.com/Kerber0/primer_proyecto_Nicolas_Primo.git (push)  
  
Nicolas@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primo (master)  
$ git checkout -b nueva-funcionalidad  
Switched to a new branch 'nueva-funcionalidad'  
  
Nicolas@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primo (nueva-funcionalidad)  
$ git add README.md  
  
Nicolas@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primo (nueva-funcionalidad)  
$ git status  
On branch nueva-funcionalidad  
  
No commits yet  
  
Changes to be committed:  
  (use "git rm --cached <file>..." to unstage)  
    new file:   README.md  
  
Nicolas@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primo (nueva-funcionalidad)  
$ git commit -m "Añadir archivo README.md"  
[nueva-funcionalidad (root-commit) 222a4d5] añadir archivo README.md  
1 file changed, 1 insertion(+)  
 create mode 100644 README.md  
  
Nicolas@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primo (nueva-funcionalidad)  
$ git push origin nueva-funcionalidad  
Enumerating objects: 3, done.  
Counting objects: 100% (3/3), done.  
Writing objects: 100% (3/3), 260 bytes | 260.00 KiB/s, done.  
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
To https://github.com/Kerber0/primer_proyecto_Nicolas_Primo.git  
 * [new branch] nueva-funcionalidad -> nueva-funcionalidad  
  
Nicolas@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primo (nueva-funcionalidad)  
$
```

Fusionar la Rama con la Rama Principal

Una vez que tu trabajo en la rama `nueva-funcionalidad` está listo, es momento de fusionar los cambios con la rama principal (`main`).

principal al repositorio remoto.

```
git push origin main
```

Conclusión

Al finalizar esta actividad habrás aprendido a:

- Crear una cuenta en [GitHub](#) y configurar una conexión [SSH](#).
- Crear un repositorio remoto en [GitHub](#) y conectarlo con un repositorio local.
- Trabajar con ramas en [Git](#) para desarrollar nuevas funcionalidades de forma aislada.
- Subir ramas al repositorio remoto y fusionar los cambios con la rama principal.

Este es el flujo de trabajo básico pero poderoso que te permitirá colaborar eficientemente en proyectos de software y mantener tu trabajo organizado.

```
Nicolás@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primoy (nueva-funcionalidad)
$ git commit -m "añadir archivo README.md"
[nueva-funcionalidad (root-commit) 222a4d5] añadir archivo README.md
1 file changed, 1 insertion(+)
create mode 100644 README.md

Nicolás@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primoy (nueva-funcionalidad)
$ git push origin nueva-funcionalidad
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 260 bytes | 260.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Kerber0/primer_proyecto_Nicolas_Primoy.git
 * [new branch]      nueva-funcionalidad -> nueva-funcionalidad

Nicolás@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primoy (nueva-funcionalidad)
$ git checkout main
error: pathspec 'main' did not match any file(s) known to git

Nicolás@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primoy (nueva-funcionalidad)
$ git checkout master
error: pathspec 'master' did not match any file(s) known to git

Nicolás@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primoy (nueva-funcionalidad)
$ git checkout master
error: pathspec 'master' did not match any file(s) known to git

Nicolás@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primoy (nueva-funcionalidad)
$ git checkout main
error: pathspec 'main' did not match any file(s) known to git

Nicolás@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primoy (nueva-funcionalidad)
$ git checkout primer_proyecto_Nicolas_Primoy
error: pathspec 'primer_proyecto_Nicolas_Primoy' did not match any file(s) known to git

Nicolás@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primoy (nueva-funcionalidad)
$ git checkout -b master
Switched to a new branch 'master'

Nicolás@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primoy (master)
$ git merge nueva-funcionalidad
Already up to date.

Nicolás@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primoy (master)
$ git push origin master
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'master' on GitHub by visiting:
remote:   https://github.com/Kerber0/primer_proyecto_Nicolas_Primoy/pull/new/master
remote:
To https://github.com/Kerber0/primer_proyecto_Nicolas_Primoy.git
 * [new branch]      master -> master

Nicolás@DESKTOP-E30UFLP MINGW64 ~/Desktop/primer_proyecto_Nicolas_Primoy (master)
$
```

Paso 6.1: Ir a la rama `main`. Cambia a la rama principal (`main` o `master`):

```
git checkout main
```

Paso 6.2: Fusionar los cambios de la nueva rama. Fusiona la rama `nueva-funcionalidad` con la rama principal:

```
git merge nueva-funcionalidad
```

Si no hay conflictos, los cambios se integrarán automáticamente.

Paso 6.3: Subir los cambios fusionados a GitHub. Finalmente, sube los cambios de la rama principal al repositorio remoto:

```
git push origin main
```