

Introducción a GitHub

Introducción a GitHub	1
¿Qué aprenderás?	2
Introducción	2
Introducción a GitHub	3
Configuración de GitHub	3
Subiendo y bajando cambios	4
Manejo de repositorios remotos	4
Añadiendo un repositorio remoto	5
Obteniendo información de un repositorio remoto	5
Manejo de repositorios remotos	6
Repositorios que ya están en GitHub	6
Ejercicio guiado: Utilizando GitHub	9
Contexto	9
Resumen	12



¡Comencemos!

¿Qué aprenderás?

- Aplicar el procedimiento de subida del código versionado, mediante una conexión con github desde la terminal para la mantención de un repositorio remoto.

Introducción

En la lectura anterior mencionamos la importancia de utilizar versionamiento de código y la utilidad que esta práctica tiene en nuestros proyectos. Hasta el momento, hemos visto cómo realizar esto de forma local, ahora veremos cómo utilizar un repositorio remoto como lo es Github.

La ventaja de utilizar estos recursos, es la posibilidad de trabajar con otros, obtener feedback de nuestros avances y compartir código con la comunidad de programadores, por lo que es muy importante que consideres estas herramientas al abordar tus proyectos.

¡Vamos con todo!



Introducción a GitHub

Existen varios tipos de repositorios remotos y empresas asociadas a proporcionarlos, las más usadas son: **GitHub**, Bitbucket y Gitlab. Todos funcionan de forma similar y lo importante es que soportan git, permitiéndonos de esta forma gestionar con los mismos comandos desde nuestros computadores, independiente del servicio que utilicemos. Nosotros utilizaremos GitHub.

Como mencionamos, GitHub es un gestor de repositorios remotos, lo que quiere decir que podemos almacenar una copia de nuestro código en sus servidores. Así podemos trabajar colaborativamente y respaldar nuestro trabajo.



GitHub es gratis y no tiene restricciones de la cantidad de repositorios que podamos crear. Así que si no tienes tu cuenta de GitHub creada ve a crearla a [GitHub.com](https://github.com).

¿Qué es un repositorio remoto?

Los repositorios remotos son versiones de tu proyecto que se encuentran alojados en Internet o en algún punto de la red.

Configuración de GitHub

Una parte importante de conexión con cualquier servidor es la autenticación, es decir, el procedimiento informático que permite asegurar que un usuario es auténtico o quien dice ser.

Existen varias formas de hacerlo, pero la más fácil y segura es utilizar el protocolo **SSH** y sus llaves. Las llaves, son un medio por el cual podemos identificar nuestro equipo con un servidor o página específica, incluso sin tener que ingresar una contraseña. Esto funciona mediante dos llaves, una privada y otra pública. La privada vive en nuestro equipo y la pública es la que ingresa en el lado remoto.

Si quieres conocer el procedimiento para conectar tu cuenta de Github con SSH, te invitamos a leer la [documentación oficial](#).

Subiendo y bajando cambios

Para subir los cambios al repositorio remoto debemos utilizar el comando `git push origin main`. De esta forma se suben todos los cambios registrados (comiteados) al repositorio remoto, en la branch (rama) en la que me encuentre, en este caso: `main`.

Ahora, si por el contrario necesito bajar los cambios que están en un remoto que tenemos registrado en nuestro proyecto, podremos utilizar el comando:

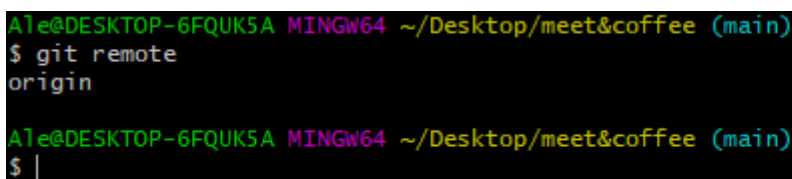
```
git pull origin main
```

Lo cual traerá a nuestro computador todos los cambios que se hayan realizado en el remoto, uniendo de forma automática los archivos que encuentre con cambios.

Recuerda que el nombre `origin` corresponderá al remoto registrado en tu proyecto y `main` se refiere a la rama main del remoto.

Manejo de repositorios remotos

Si necesitamos saber si el proyecto en que estamos trabajando ya contiene alguna referencia a un repositorio remoto, lo realizaremos con el comando `git remote`, el cual nos mostrará el nombre de los repositorios que tengamos añadidos.



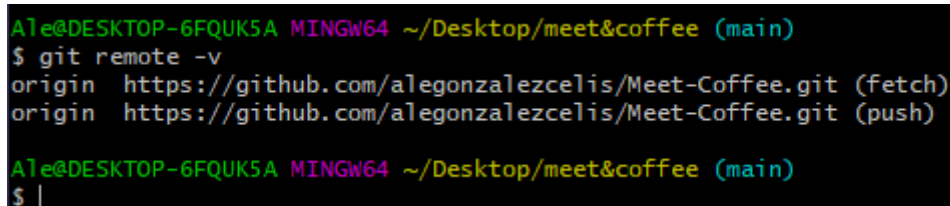
```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote
origin

Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ |
```

Imagen 5. Uso de `git remote`.

Fuente: Desafío Latam.

Si necesitamos saber las url de estos servidores, podemos utilizar `git remote -v`.



```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote -v
origin  https://github.com/alegonzalezcelis/Meet-Coffee.git (fetch)
origin  https://github.com/alegonzalezcelis/Meet-Coffee.git (push)

Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ |
```

Imagen 6. Uso de `git remote -v`.

Fuente: Desafío Latam.

Añadiendo un repositorio remoto

Para añadir un repositorio remoto, simplemente debemos usar el comando:

```
git remote add [nombre] [dirección del repositorio]
```

Si este comando se ejecuta bien, no muestra ningún resultado, solo debemos corroborarlo con `git remote` o `git remote -v`.

```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote add origin https://github.com/alegonzalezcelis/WebFundamentalsExp4.git

Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote -v
origin https://github.com/alegonzalezcelis/WebFundamentalsExp4.git (fetch)
origin https://github.com/alegonzalezcelis/WebFundamentalsExp4.git (push)
```

Imagen 7. Comprobar el correcto funcionamiento de git remote add.

Fuente: Desafío Latam

Generalmente, al añadir un repositorio remoto se utiliza el nombre "origin". Sin embargo, podemos tener más de un servidor remoto configurado, solo tendríamos que utilizar nombres distintos.

Obteniendo información de un repositorio remoto

Una vez que ya tengamos añadido un repositorio remoto en nuestro proyecto, podremos obtener información de él con el siguiente comando (cambiando el `nombre` por el de nuestro repositorio):

```
git remote show [nombre]
```

```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote show origin
* remote origin
Fetch URL: https://github.com/alegonzalezcelis/Meet-Coffee.git
Push URL: https://github.com/alegonzalezcelis/Meet-Coffee.git
HEAD branch: main
Remote branch:
  main tracked
Local branch configured for 'git pull':
  main merges with remote main
Local ref configured for 'git push':
  main pushes to main (up to date)
```

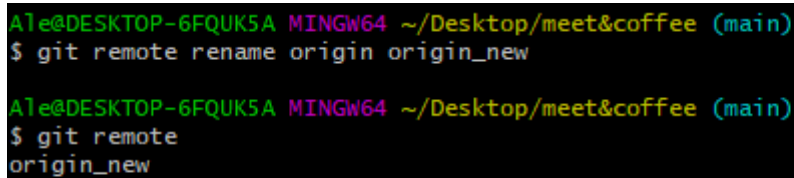
Imagen 8. Git remote show origin

Fuente: Desafío Latam

Manejo de repositorios remotos

Si necesitamos renombrar un repositorio remoto que hemos añadido, podemos realizarlo de la siguiente forma:

```
git remote rename nombreActual NuevoNombre
```



```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote rename origin origin_new

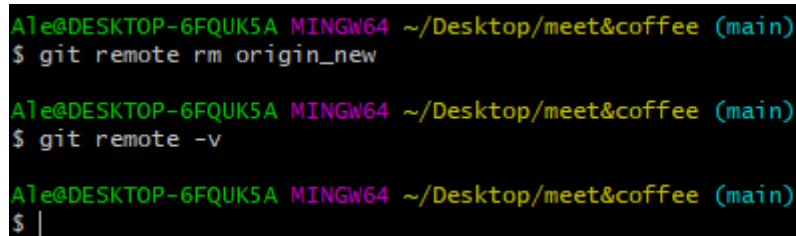
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote
origin_new
```

Imagen 9. Renombrar repositorio.

Fuente: Desafío Latam

Y, por supuesto, si necesitamos borrar un repositorio remoto, podemos realizarlo con el siguiente comando git:

```
git remote rm NombreRepositorio
```



```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote rm origin_new

Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote -v

Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ |
```

Imagen 10. Eliminar repositorio.

Fuente: Desafío Latam

Repositorios que ya están en GitHub

GitHub nos proporciona herramientas para utilizar repositorios que ya están creados. Para ellos existen dos herramientas que nos ayudarán: Fork y Clone.

La primera es Fork, este comando creará una bifurcación del repositorio, en otras palabras realizará una copia del proyecto completo en nuestra propia cuenta de GitHub.

Esto es útil para realizar cambios sin afectar el proyecto original o partir un nuevo proyecto desde otro como base.

Para forkear un proyecto en GitHub, debemos presionar el botón Fork ubicado en la esquina superior derecha de un repositorio.

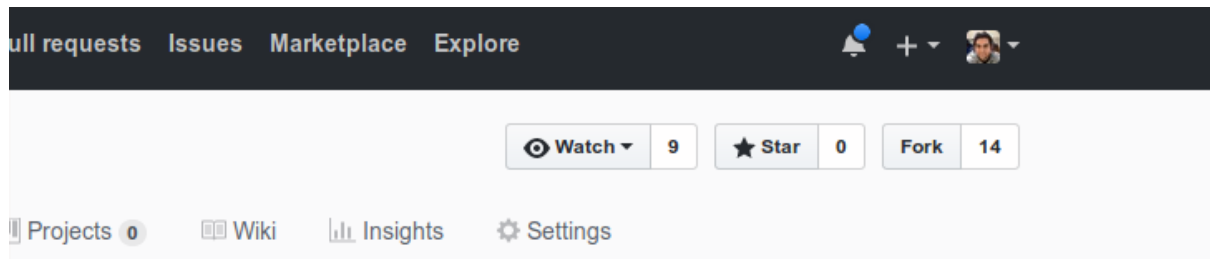


Imagen 11. Menú repositorio GitHub.
Fuente: Desafío Latam.

Esto realizará la copia del repositorio en nuestra cuenta de GitHub.

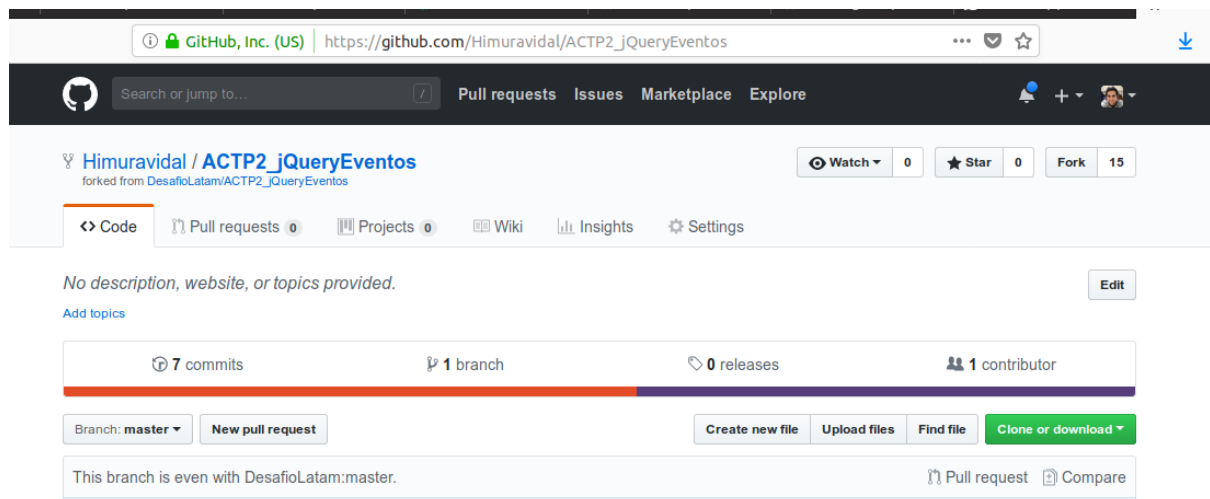


Imagen 12. Repositorio en GitHub.
Fuente: Desafío Latam.

El siguiente paso para comenzar a trabajar sería bajar el contenido de este repositorio hacia nuestro computador local. Esto lo realizaremos **clonando** el repositorio.

El comando para clonar un repositorio es el siguiente:

```
git clone [dirección del repositorio]
```

Y la dirección del remoto la podemos obtener desde el botón **Clone or download**.

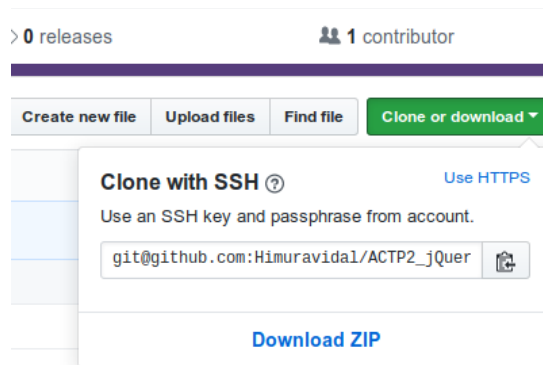


Imagen 13. Clonar o descargar repositorio.
Fuente: Desafío Latam

Debemos copiar la dirección que ahí aparece.

Cuando ejecutemos el comando git clone, veremos lo siguiente en la terminal.

```
Al@DESKTOP-6FQK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git clone git@github.com:Himuravida1/ACTP2_jQueryEventos.git
Cloning into 'ACTP2_jQueryEventos'...
Warning: Permanently added the RSA host key for IP address '140.82.114.3' to the list of known hosts.
remote: Enumerating objects: 101, done.
remote: Total 101 (delta 0), reused 0 (delta 0), pack-reused 101
Receiving objects: 100% (101/101), 21.98 MiB | 308.00 KiB/s, done.
Resolving deltas: 100% (9/9), done.
```

Imagen 14. Uso de git clone.
Fuente: Desafío Latam

La acción de clonar va a añadir por defecto el remoto desde donde estamos copiando, lo que permitirá luego, subir los cambios con el comando `git push`. Esto funcionará siempre y cuando el repositorio clonado sea nuestro o tengamos permisos de escritura en él.



Antes de continuar:

¿Existe algún concepto que no hayas comprendido?

Vuelve a revisar los conceptos que más te hayan costado antes de seguir adelante.

Ejercicio guiado: Utilizando GitHub

Contexto

Como vimos en el capítulo anterior, Git trabaja a modo de repositorio local y al finalizar cada cambio, podremos guardar los avances en el repositorio remoto de Github.

Para esto, ejercitaremos creando un repositorio remoto para el proyecto, lo añadiremos por consola y subiremos sus cambios. Sigue atentamente el paso a paso que te presentamos a continuación:

- **Paso 1:** una vez que ya tengas configurada la cuenta en GitHub, podrás generar un nuevo repositorio. Para ello, dentro de tu perfil en GitHub presiona el botón **+** y selecciona **new repository**:

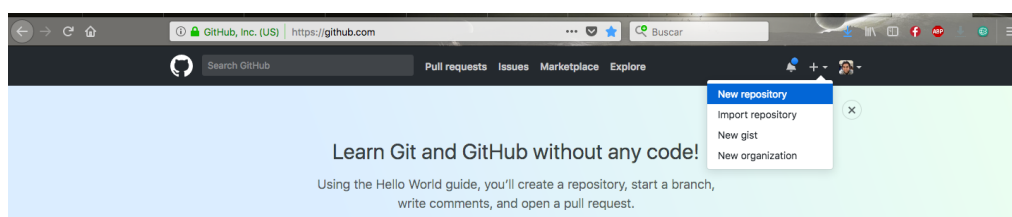


Imagen 15. Añadir nuevo repositorio.

Fuente: Desafío Latam.

Debes indicarle un nombre y una descripción:

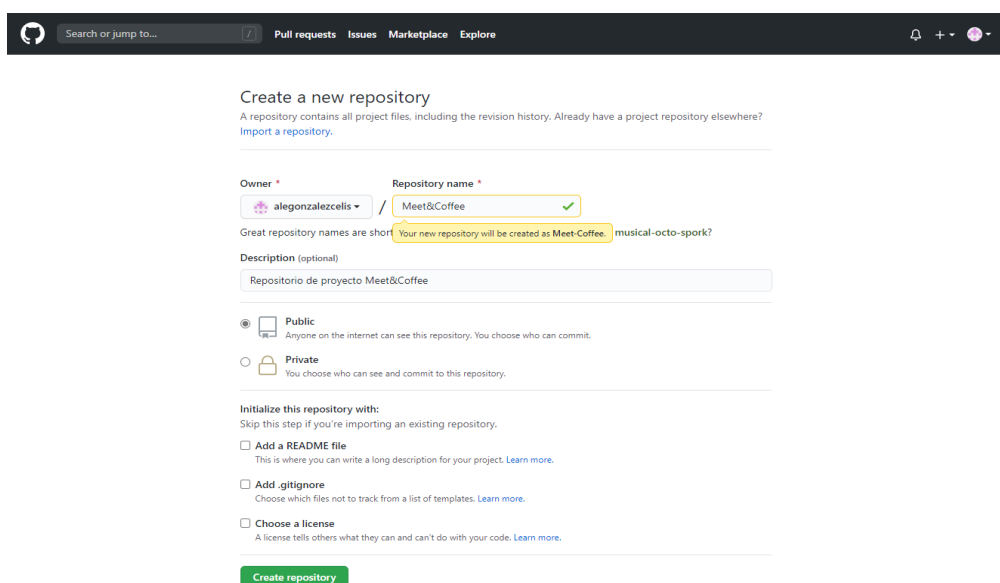


Imagen 16. Crear el nuevo repositorio.

Fuente: Desafío Latam.

- **Paso 2:** luego de crearlo, te entregará algunos ejemplos de comandos para subir el código.

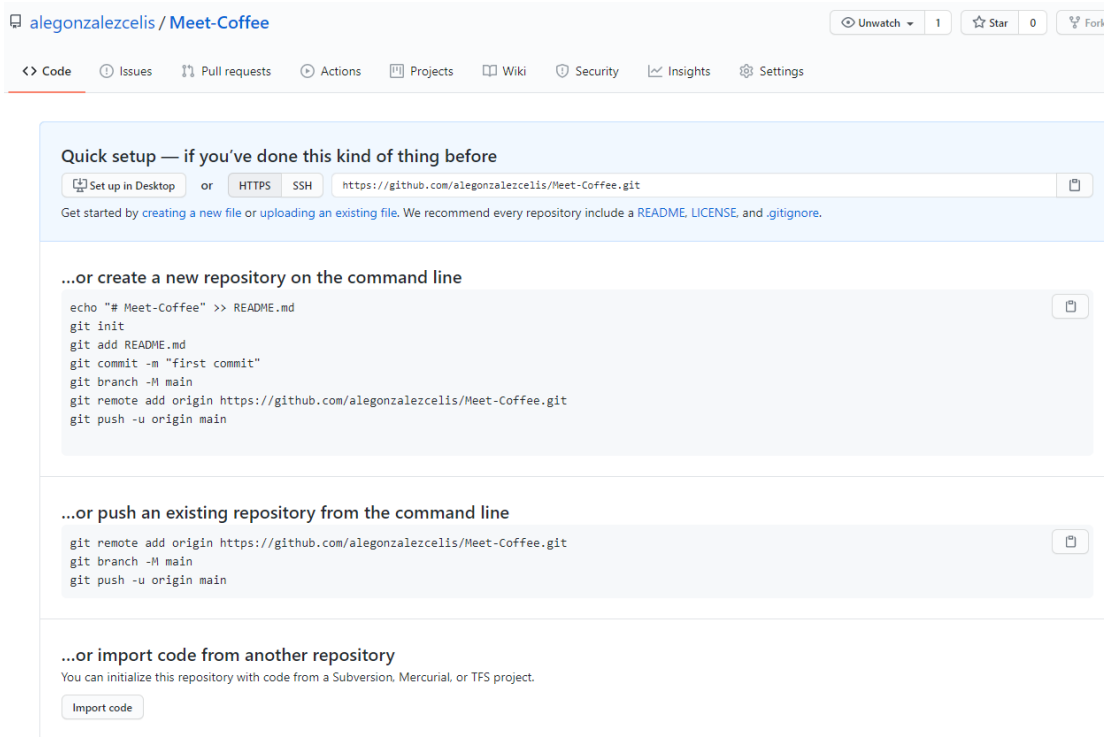


Imagen 17. Repositorio creado.

Fuente: Desafío Latam.

Si ya tienes un proyecto con git iniciado como es nuestro caso, solo debemos indicarle el nuevo repositorio remoto y subir los cambios.

- **Paso 3:** abre la terminal y dirígete a la carpeta del proyecto:

```
cd Desktop/meet\&coffee/
```



Nota: GitHub en sus inicios definió la rama `master` como su rama principal o por defecto, sin embargo, desde octubre de 2020, han implementado cambios en su rama principal, sustituyendo gradualmente la rama `master` por la nueva rama `main`, este cambio afecta directamente a los nuevos repositorios. Según se puede apreciar en su documentación oficial, el cambio responde a la implementación de un nombre más corto y a la interpretación más óptima en otros lenguajes. Para más información acerca de esta actualización visita el siguiente link: <https://github.com/github/renaming>

Es por eso que desde ahora comenzaremos a trabajar en nuestra rama `main`:

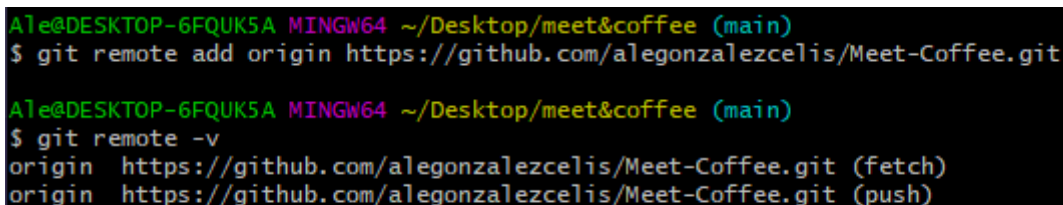
- **Paso 4:** renombra la rama `master` por `main` con el siguiente comando:

```
git branch -M main
```

- **Paso 5:** añade el repositorio remoto recién creado al proyecto desde la terminal. Reemplaza dirección con la URL o git del repositorio recién creado.

```
git remote add origin [dirección del repositorio remoto]
```

Al presionar enter, no recibirás ninguna confirmación, pero si quieres ver si está ingresado en el repositorio remoto, puedes usar el comando : `git remote -v` y verás algo como esto:



```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote add origin https://github.com/alegonzalezcelis/Meet-Coffee.git

Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git remote -v
origin https://github.com/alegonzalezcelis/Meet-Coffee.git (fetch)
origin https://github.com/alegonzalezcelis/Meet-Coffee.git (push)
```

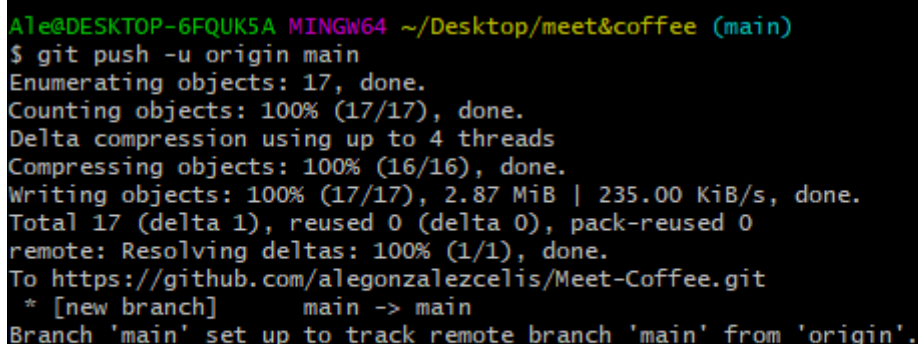
Imagen 18. Uso de git remote

Fuente: Desafío Latam

- **Paso 6:** ahora puedes subir el proyecto repositorio remoto:

```
git push -u origin main
```

De esta forma todo lo que tienes en local, será subido al repositorio recién creado y ya estás compartiendo tu código con el mundo.



```
Ale@DESKTOP-6FQUK5A MINGW64 ~/Desktop/meet&coffee (main)
$ git push -u origin main
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 4 threads
Compressing objects: 100% (16/16), done.
Writing objects: 100% (17/17), 2.87 MiB | 235.00 KiB/s, done.
Total 17 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), done.
To https://github.com/alegonzalezcelis/Meet-Coffee.git
* [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Imagen 19. Uso de Git Push.

Fuente: Desafío Latam.

Resumen

- GitHub es un gestor de repositorios remotos, lo que quiere decir que podemos almacenar una copia de nuestro código en sus servidores. Así podemos trabajar colaborativamente y respaldar nuestro trabajo.
- Las llaves, son un medio por el cual podemos identificar nuestro equipo con un servidor o página específica, incluso sin tener que ingresar una contraseña.
- Para subir los cambios al repositorio remoto debemos utilizar el comando `git push origin main`.
- Si necesitamos saber si el proyecto en que estamos trabajando ya contiene alguna referencia a un repositorio remoto, lo realizaremos con el comando `git remote`, el cual nos mostrará el nombre de los repositorios que tengamos añadidos.
- Fork es un comando que creará una bifurcación del repositorio, en otras palabras realizará una copia del proyecto completo en nuestra propia cuenta de GitHub.