

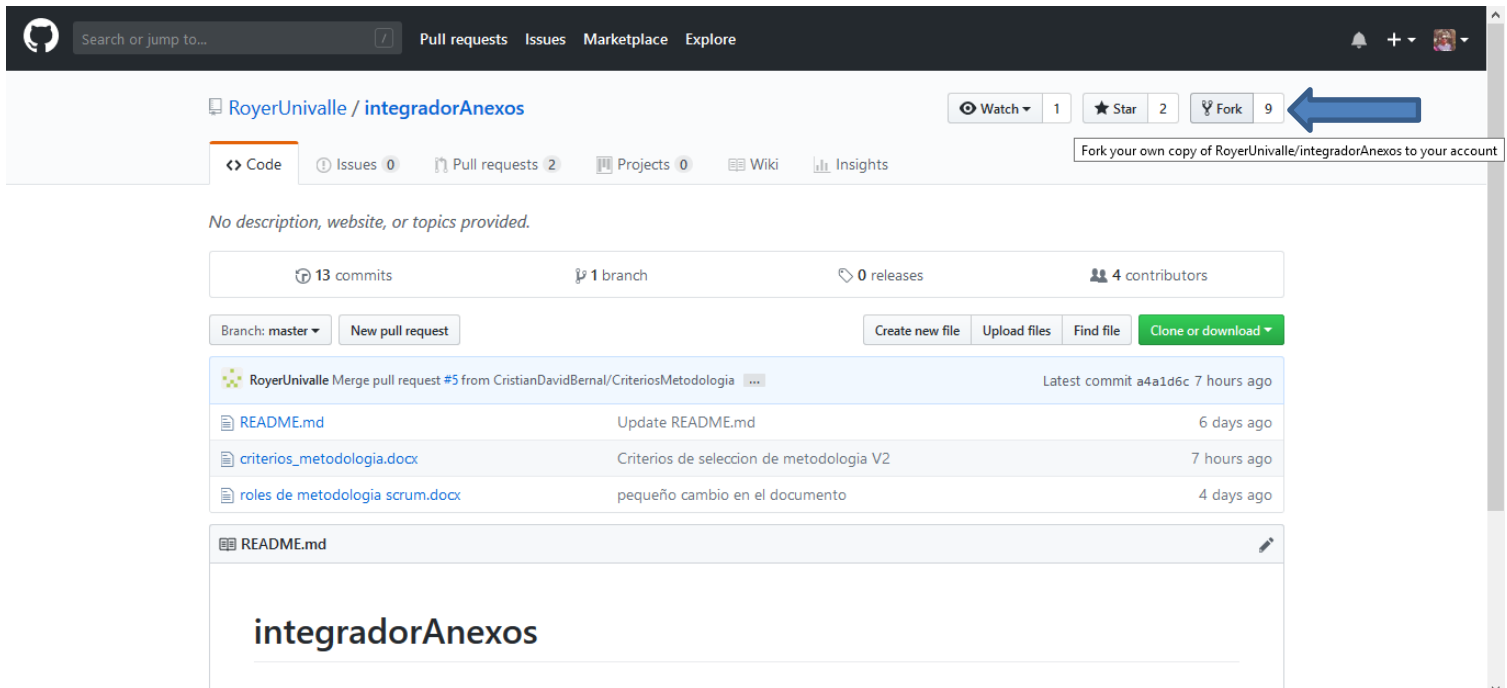
DOCUMENTO DE CONTROL DE VERSIONES

El presente documento tiene como fin proporcionar una guía para los nuevos integrantes del proyecto, en este se detallan las buenas practicas que se deben llevar acabo para realizar de manera adecuada las entregas de los archivos asignados a los colaboradores.

Pasos a seguir:

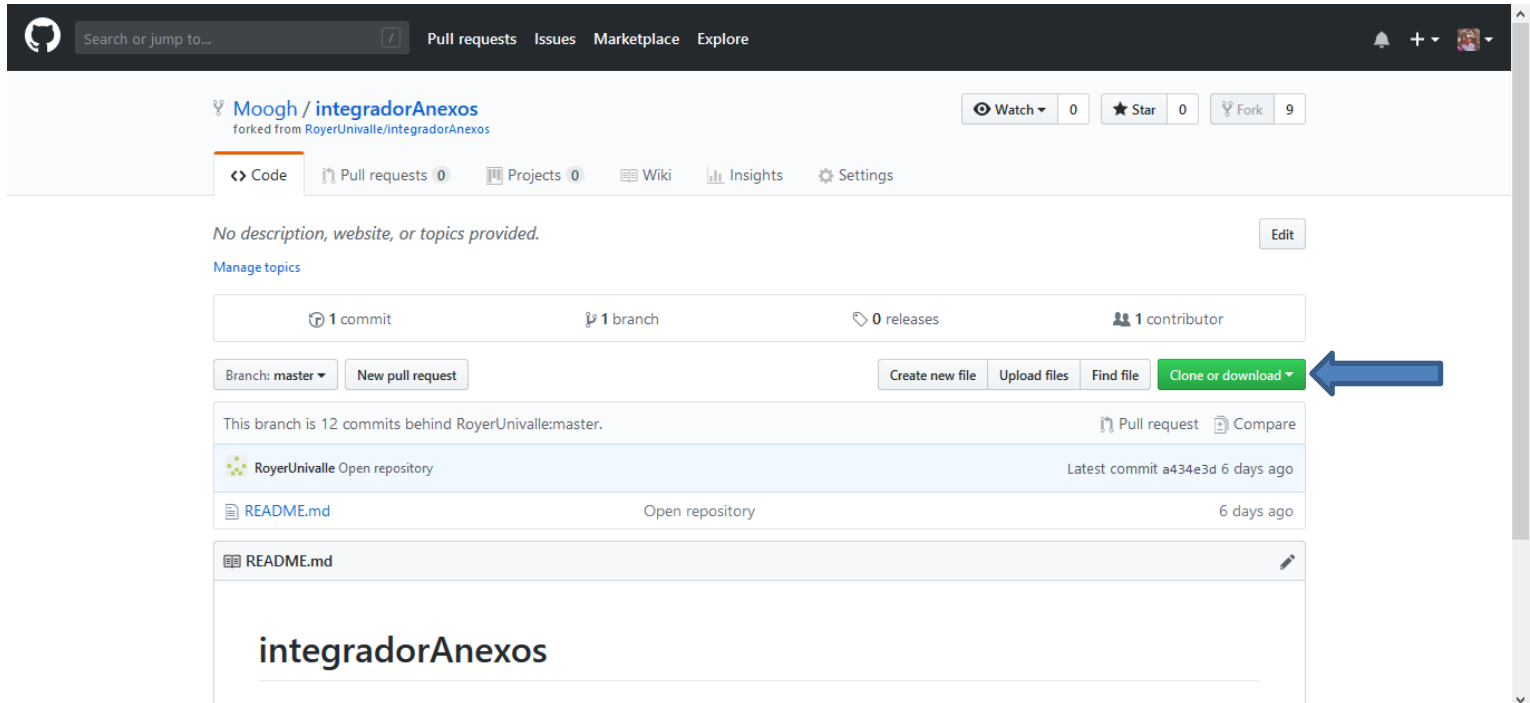
1. Los colaboradores deben situarse en el repositorio principal del cual procederán a dar click en el botón “Fork” que se encuentra en la parte superior derecha como se muestra en la figura 1.

Figura 1



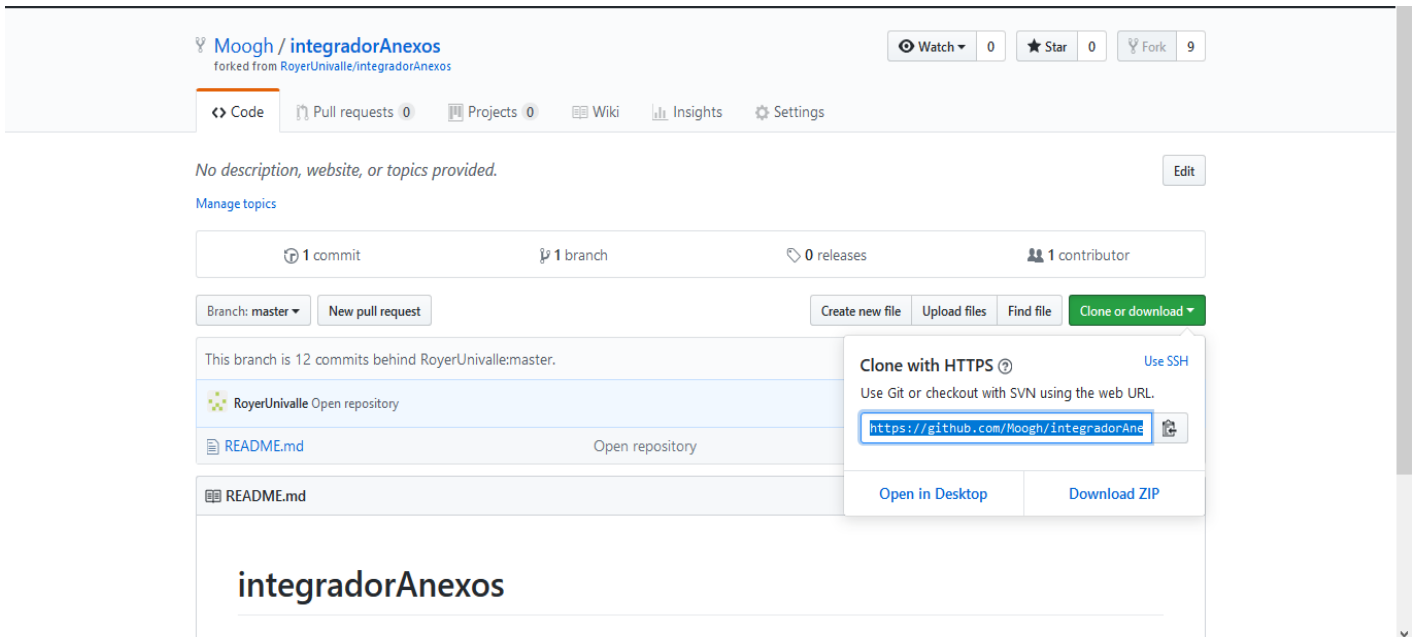
- Después se procederá a clonar el repositorio presionando la opción “Clon or download” como se muestra en la Figura 2.

Figura 2



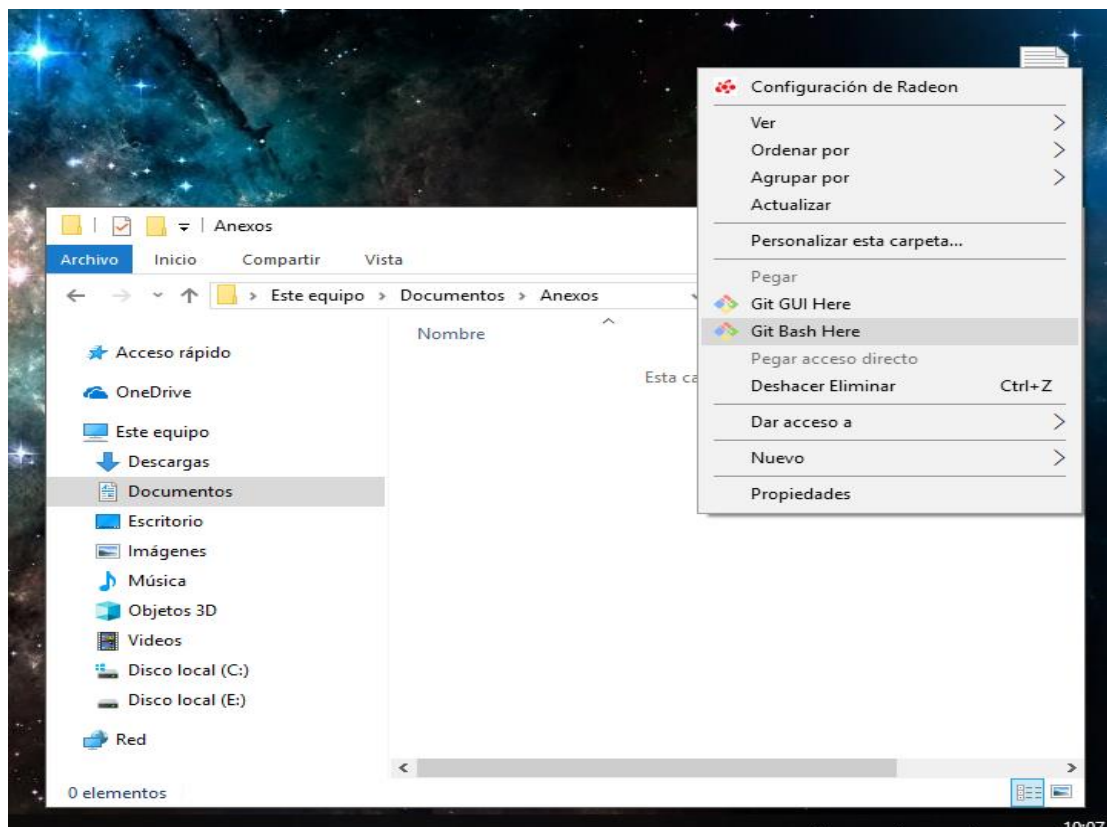
- Se copia el URL que se retorna luego de presionar la opción vista en la imagen anterior, tal y como se puede apreciar en la Figura 3.

Figura 3



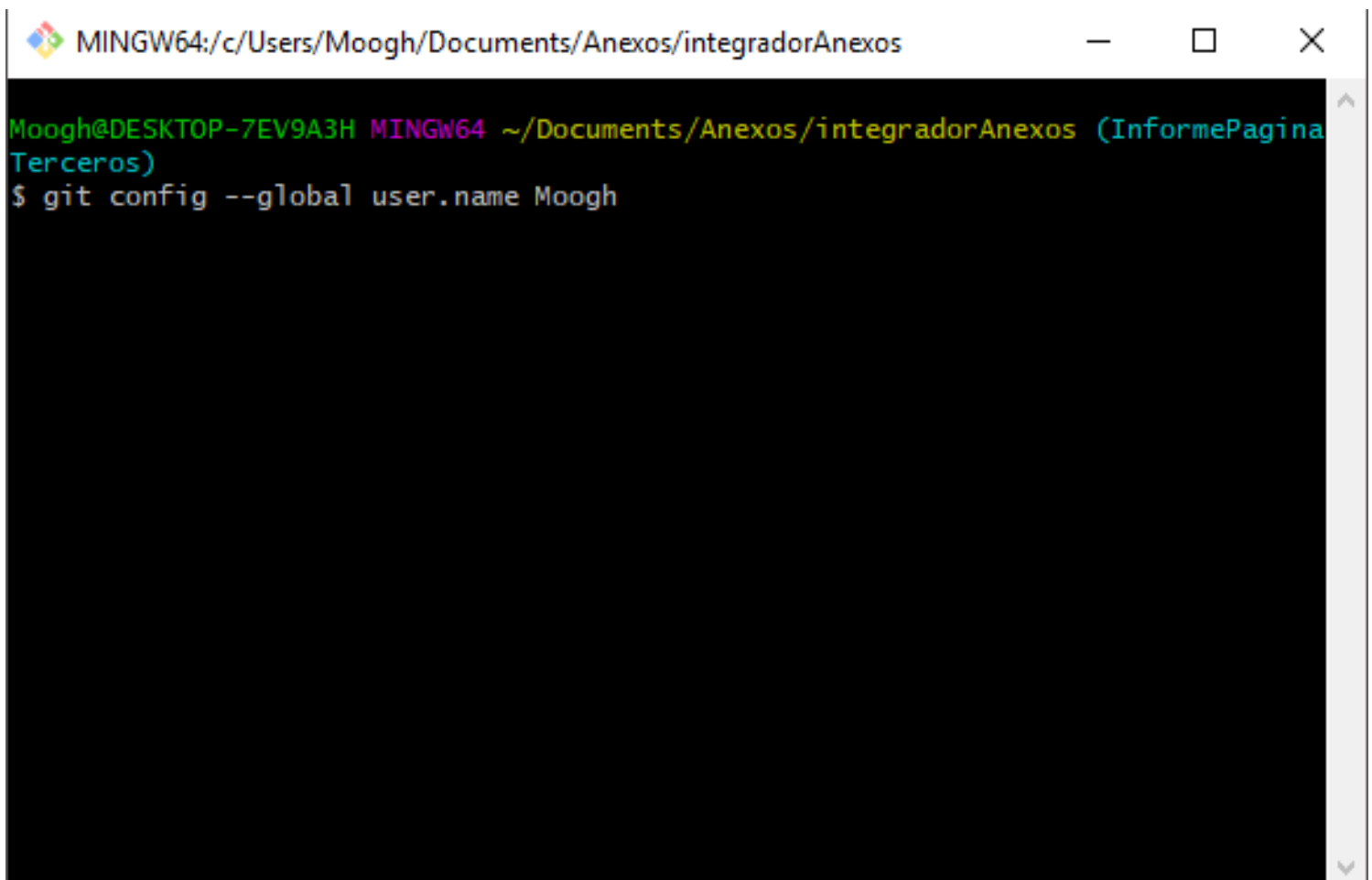
4. Se sitúa en la ubicación en la cual se va a proceder a trabajar y allí se presiona click derecho y se escoge la opción "Git Bash" la cual retornar una terminal ver Figura

Figura 4



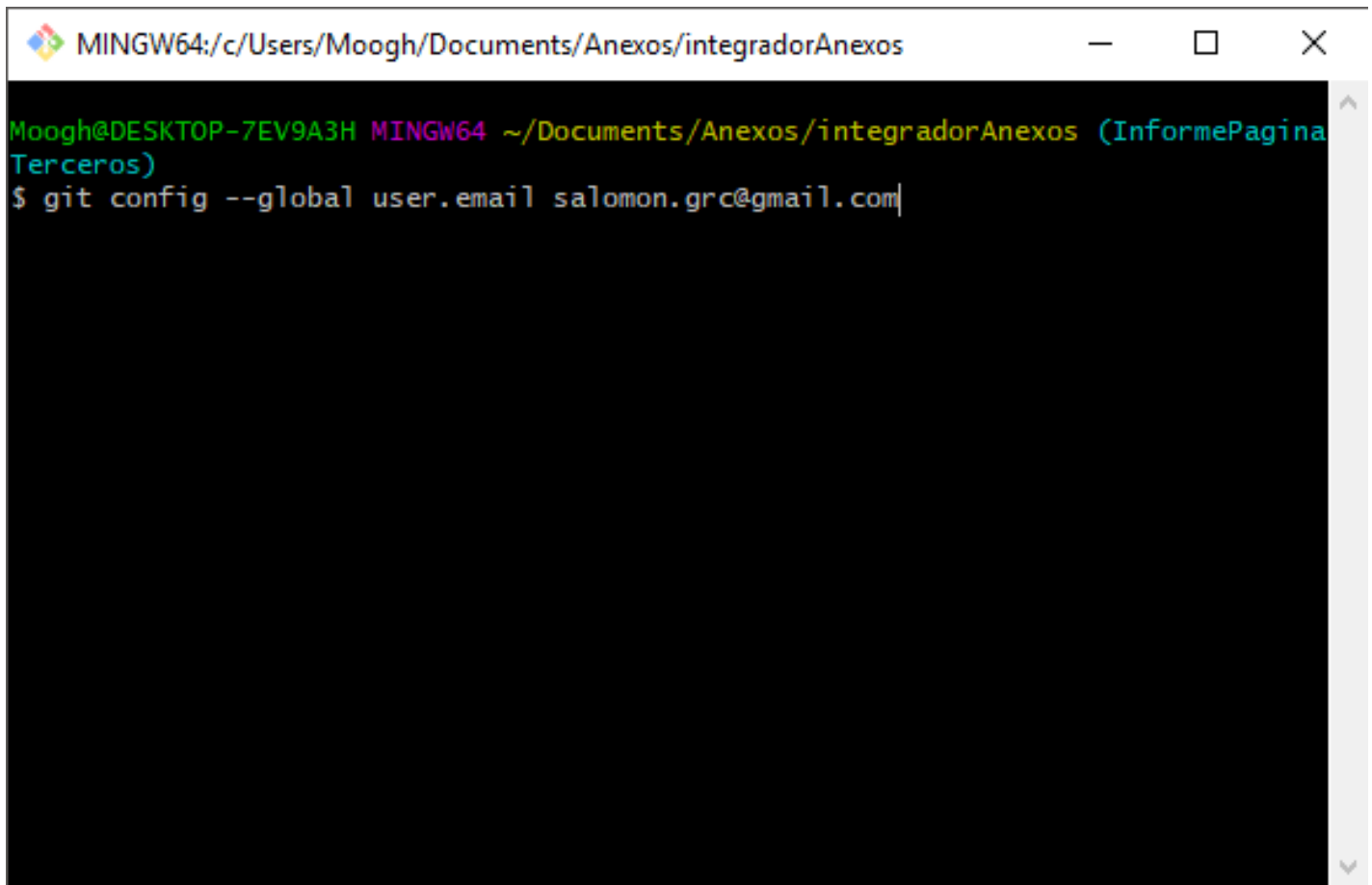
5. En la terminal se ejecuta el comando “git config --global user.name nombre_usuario” luego se presiona enter y después se ejecuta el comando “git config --global user.email correo_usuario” luego se presiona enter, esto conlleva que se vincule tu cuenta git y la terminal ver Figuras 5.

Figuras 5



The image shows a screenshot of a Windows terminal window. The title bar at the top reads "MINGW64:/c/Users/Moogh/Documents/Anexos/integradorAnexos" and includes standard window controls (minimize, maximize, close). The terminal content shows the prompt "Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePagina Terceros)" followed by the command "\$ git config --global user.name Moogh". The terminal has a black background with green and white text. A vertical scrollbar is visible on the right side of the terminal window.

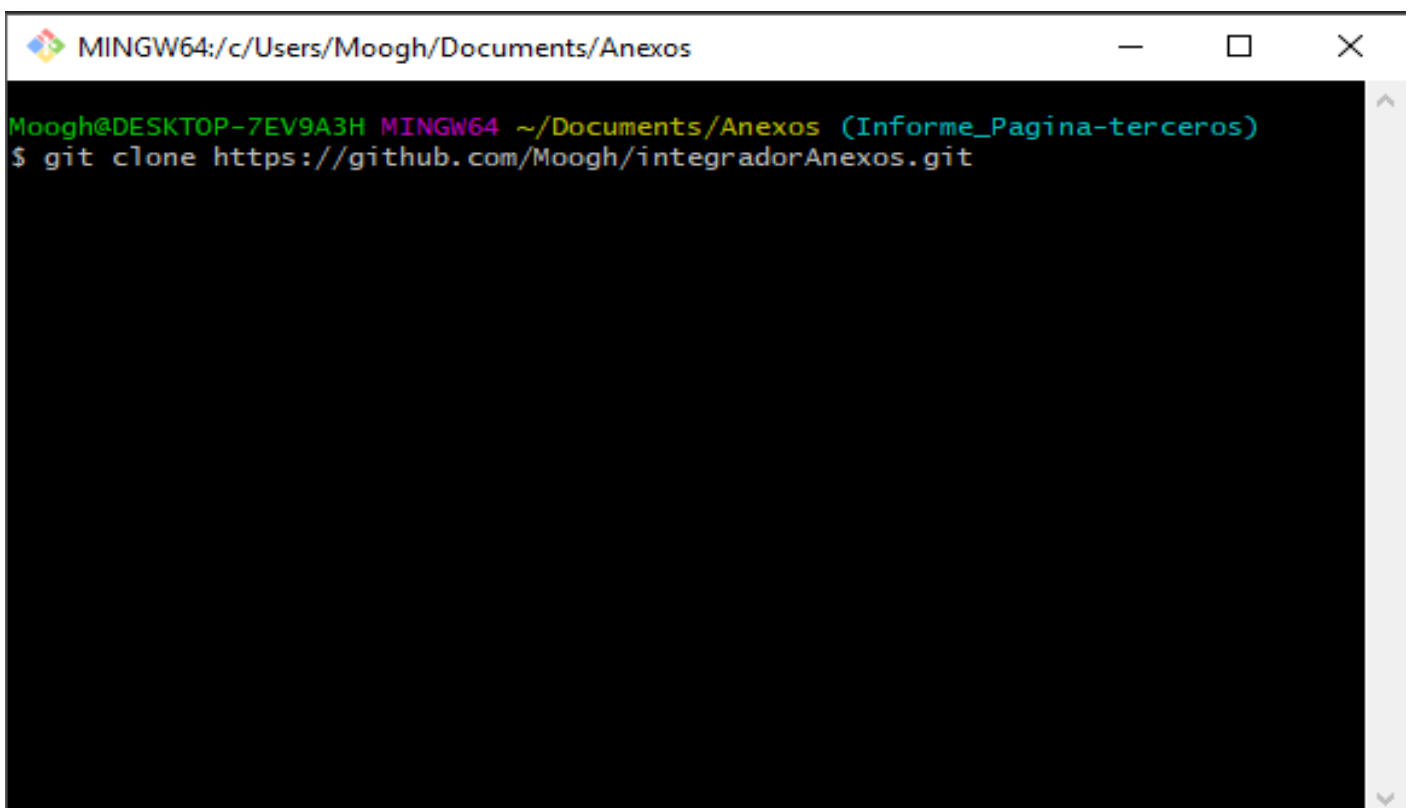
```
MINGW64:/c/Users/Moogh/Documents/Anexos/integradorAnexos
Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePagina Terceros)
$ git config --global user.name Moogh
```

A terminal window with a black background and white text. The title bar at the top reads 'MINGW64:/c/Users/Moogh/Documents/Anexos/integradorAnexos'. The prompt is 'Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePagina Terceros)'. The command entered is '\$ git config --global user.email salomon.grc@gmail.com|'.

```
MINGW64:/c/Users/Moogh/Documents/Anexos/integradorAnexos
Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePagina Terceros)
$ git config --global user.email salomon.grc@gmail.com|
```

6. Se ejecuta el comando “**git clone**” seguido de la URL que ya en pasos anteriores se ha copiado ver Figura 6.

Figura 6

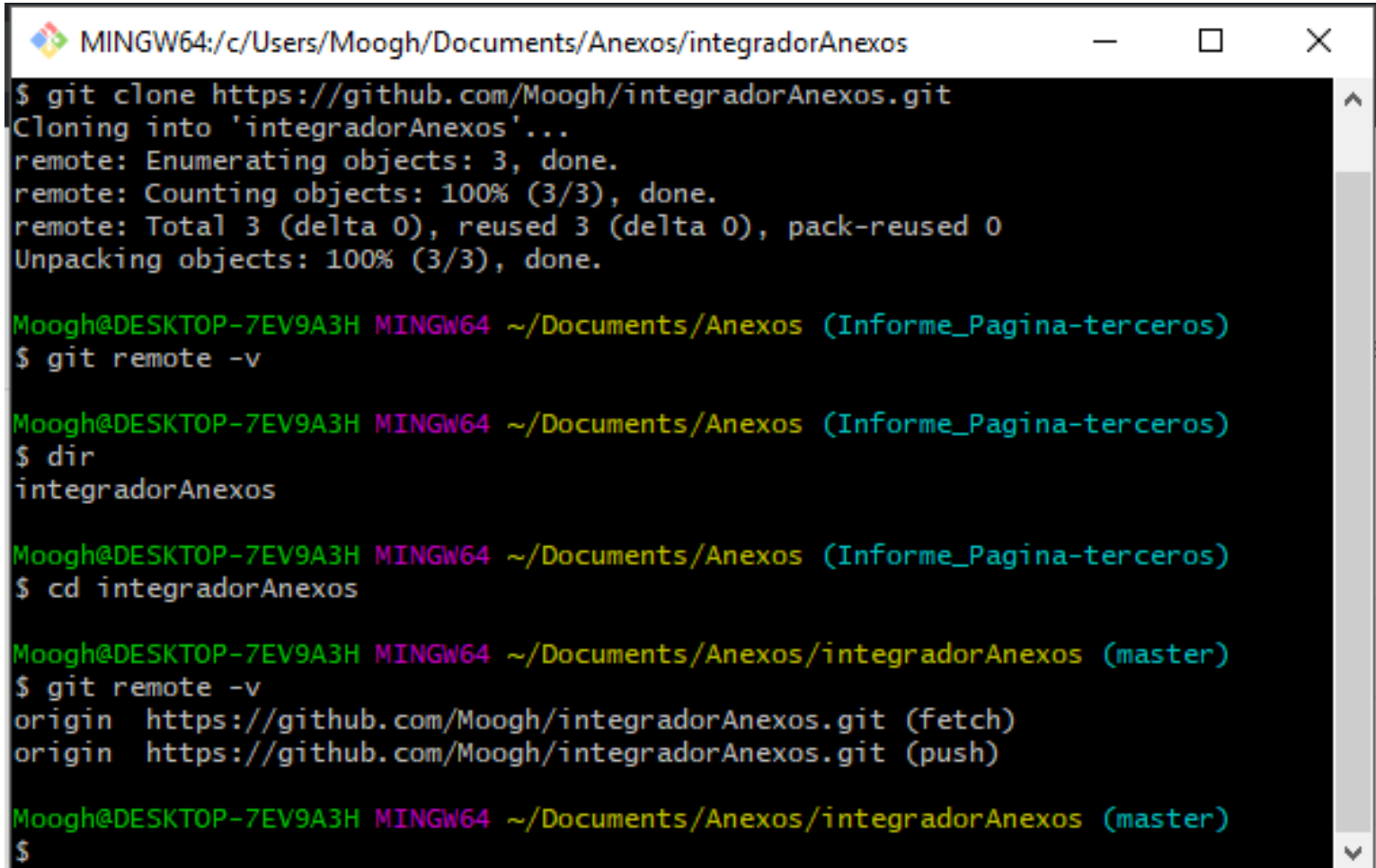
A terminal window with a black background and white text. The title bar at the top reads 'MINGW64:/c/Users/Moogh/Documents/Anexos'. The prompt is 'Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos (Informe_Pagina-terceros)'. The command entered is '\$ git clone https://github.com/Moogh/integradorAnexos.git'.

```
MINGW64:/c/Users/Moogh/Documents/Anexos
Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos (Informe_Pagina-terceros)
$ git clone https://github.com/Moogh/integradorAnexos.git
```

Nota: Al realizar este paso se debe ingresar a la carpeta que nos crea por defecto el comando anterior, para esto se utiliza el comando `cd nombre_de_la_carpeta`.

7. Se ejecuta el comando `"git remote -v"` para inspeccionar los repositorios los cuales se tienen configurados como se muestra en la figura 7.

Figura 7



```
MINGW64:/c/Users/Moogh/Documents/Anexos/integradorAnexos
$ git clone https://github.com/Moogh/integradorAnexos.git
Cloning into 'integradorAnexos'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos (Informe_Pagina-terceros)
$ git remote -v

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos (Informe_Pagina-terceros)
$ dir
integradorAnexos

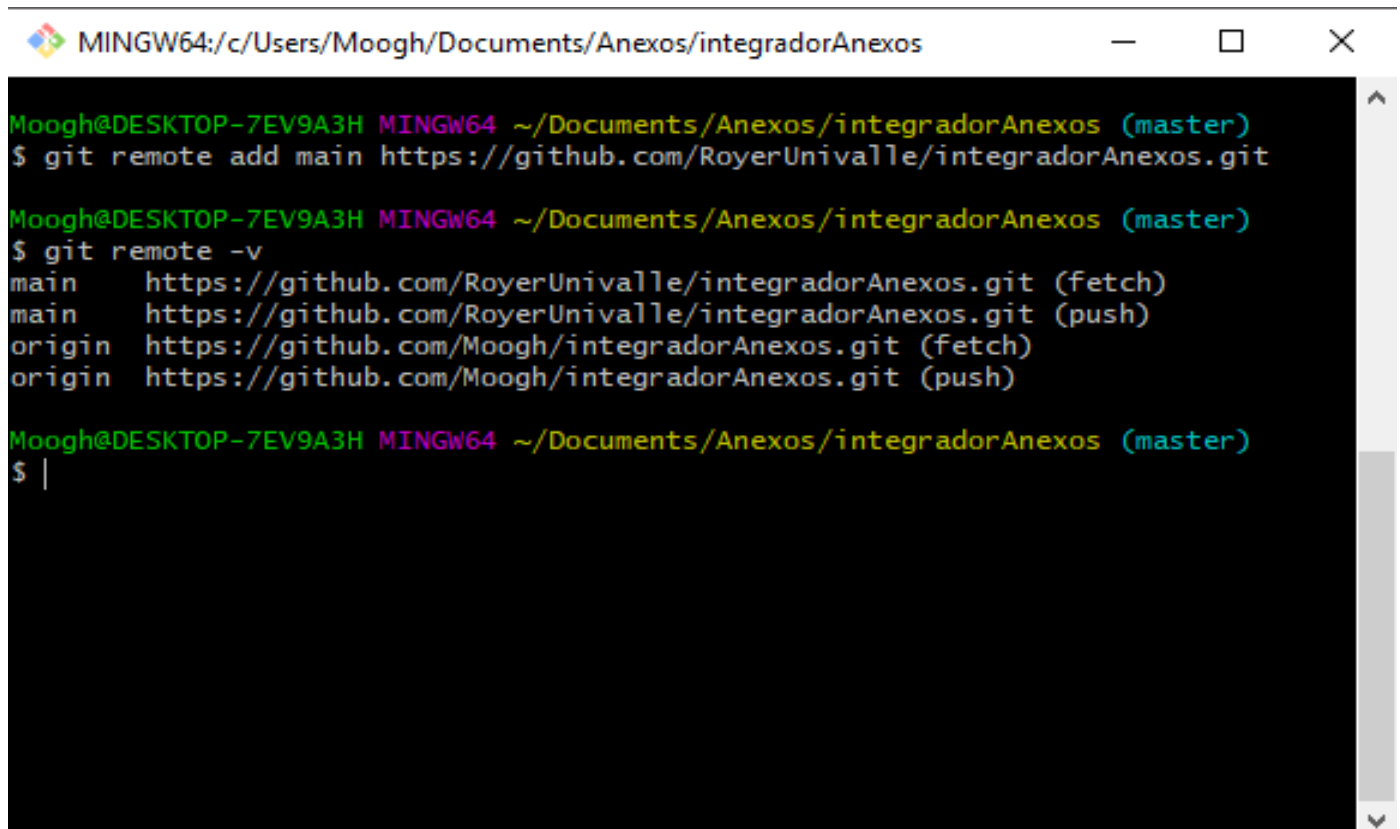
Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos (Informe_Pagina-terceros)
$ cd integradorAnexos

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (master)
$ git remote -v
origin  https://github.com/Moogh/integradorAnexos.git (fetch)
origin  https://github.com/Moogh/integradorAnexos.git (push)

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (master)
$
```

8. Se ejecuta el comando “git remote add nombre_referencia url_de_repo_original”, dicho comando lo que hace es crear una conexión remota al repositorio en el cual se trabajará, tal y como se puede ver en la Figura 8.

Figura 8



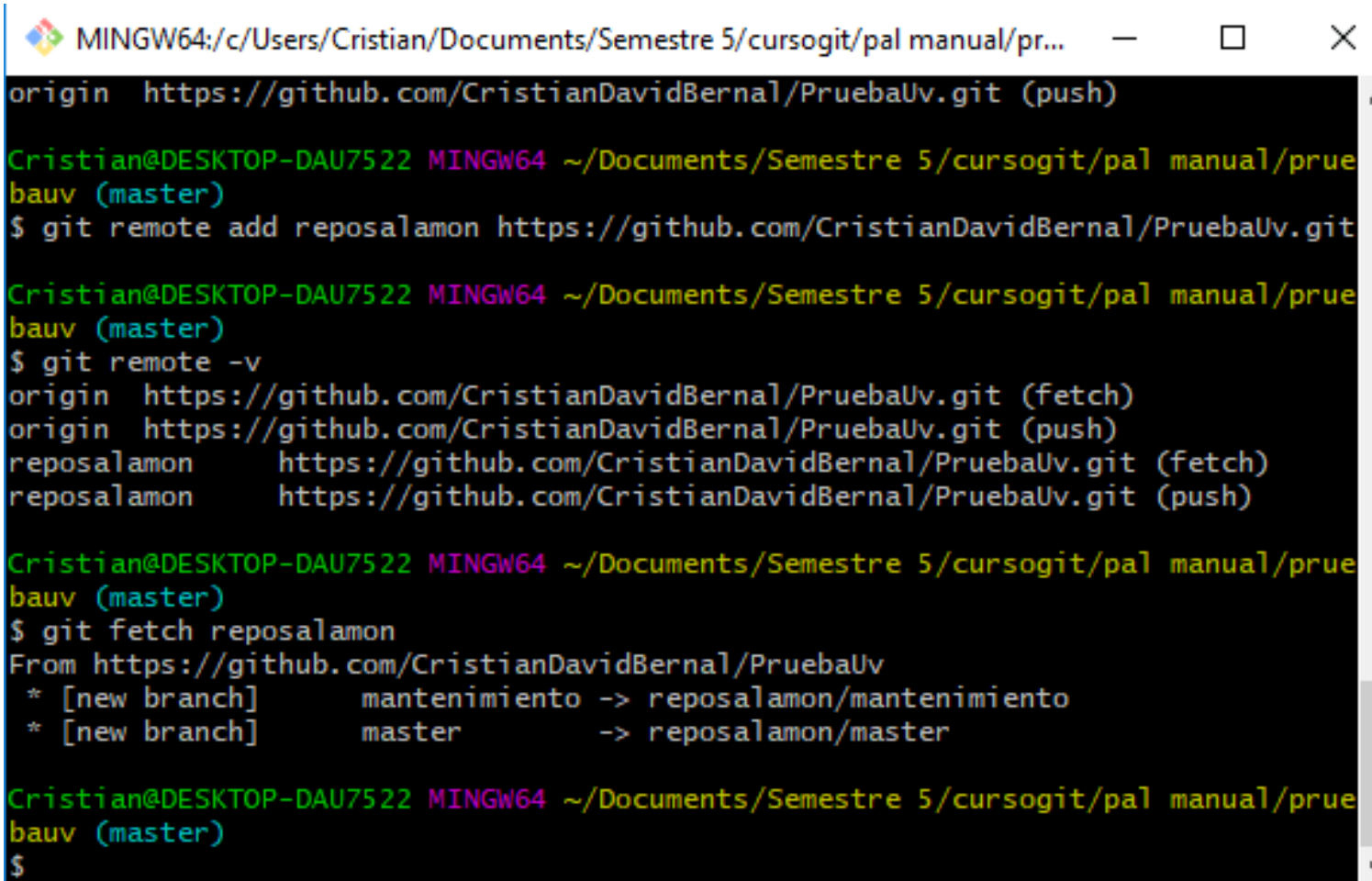
```
MINGW64:/c/Users/Moogh/Documents/Anexos/integradorAnexos
Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (master)
$ git remote add main https://github.com/RoyerUnivalle/integradorAnexos.git

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (master)
$ git remote -v
main      https://github.com/RoyerUnivalle/integradorAnexos.git (fetch)
main      https://github.com/RoyerUnivalle/integradorAnexos.git (push)
origin    https://github.com/Moogh/integradorAnexos.git (fetch)
origin    https://github.com/Moogh/integradorAnexos.git (push)

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (master)
$ |
```

9. Se ejecuta el comando "git fetch nombre_referencia" el cual trae tanto los archivos subidos como las ramas creadas dentro del repositorio tal y como se muestra en la Figura 9

Figura 9



```
MINGW64:/c/Users/Cristian/Documents/Semestre 5/cursogit/pal manual/prue...
origin https://github.com/CristianDavidBernal/PruebaUv.git (push)

Cristian@DESKTOP-DAU7522 MINGW64 ~/Documents/Semestre 5/cursogit/pal manual/prue
bauv (master)
$ git remote add reposalamon https://github.com/CristianDavidBernal/PruebaUv.git

Cristian@DESKTOP-DAU7522 MINGW64 ~/Documents/Semestre 5/cursogit/pal manual/prue
bauv (master)
$ git remote -v
origin https://github.com/CristianDavidBernal/PruebaUv.git (fetch)
origin https://github.com/CristianDavidBernal/PruebaUv.git (push)
reposalamon https://github.com/CristianDavidBernal/PruebaUv.git (fetch)
reposalamon https://github.com/CristianDavidBernal/PruebaUv.git (push)

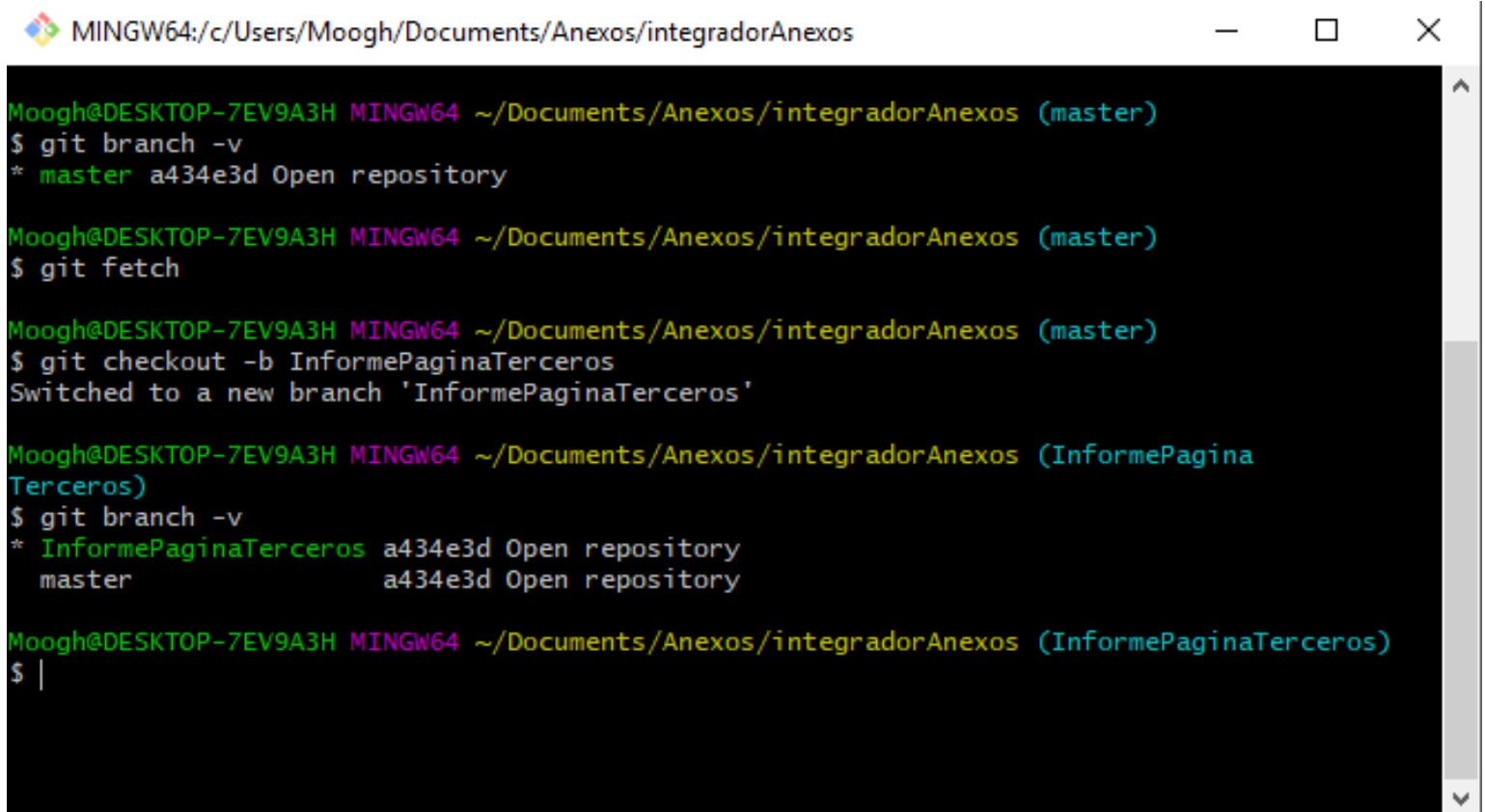
Cristian@DESKTOP-DAU7522 MINGW64 ~/Documents/Semestre 5/cursogit/pal manual/prue
bauv (master)
$ git fetch reposalamon
From https://github.com/CristianDavidBernal/PruebaUv
* [new branch]      mantenimiento -> reposalamon/mantenimiento
* [new branch]      master       -> reposalamon/master

Cristian@DESKTOP-DAU7522 MINGW64 ~/Documents/Semestre 5/cursogit/pal manual/prue
bauv (master)
$
```


10. Se crea una rama en la cual se ponen los archivos correspondientes a las tareas asignadas ejecutando el comando “git checkout -b nombre_de_la_rama”, tal y como se muestra en la Figura 10.

Nota: Este comando permite crear la rama y se posiciona en la misma, si solo necesita crearla se ejecuta el comando “git branch nombre_de_la_rama”.

Figura 10



```
MINGW64:/c/Users/Moogh/Documents/Anexos/integradorAnexos

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (master)
$ git branch -v
* master a434e3d Open repository

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (master)
$ git fetch

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (master)
$ git checkout -b InformePaginaTerceros
Switched to a new branch 'InformePaginaTerceros'

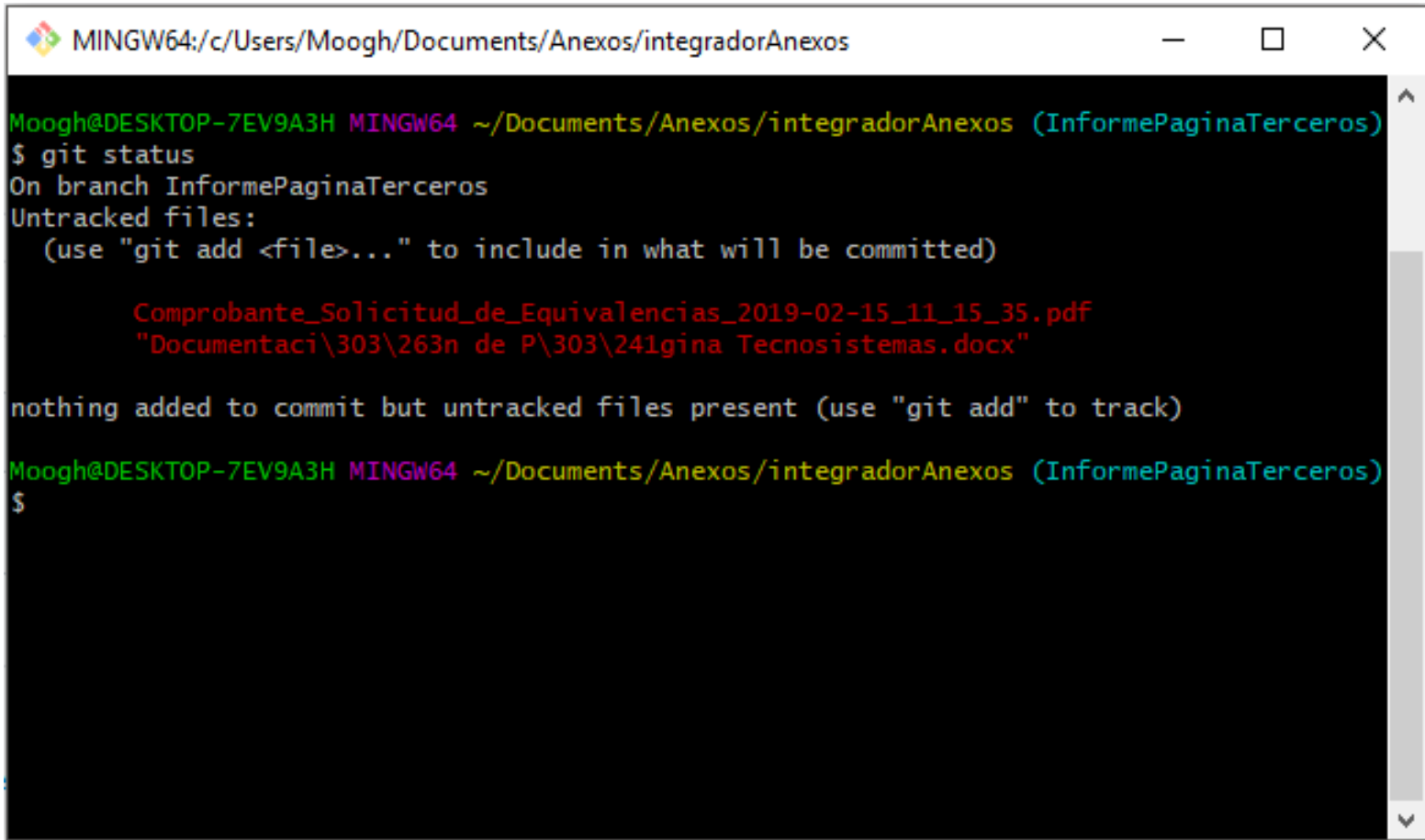
Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePaginaTerceros)
$ git branch -v
* InformePaginaTerceros a434e3d Open repository
  master a434e3d Open repository

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePaginaTerceros)
$ |
```

11. Se procede a poner los archivos que se necesitan subir en la carpeta en la que se está trabajando.

12. Luego de hacer el paso anterior, se revisa en qué estado están los archivos para poder prepararlos ejecutando el comando “git status”, tal y como se muestra en la Figura 12.

Figura 12



```
MINGW64:/c/Users/Moogh/Documents/Anexos/integradorAnexos

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePaginaTerceros)
$ git status
On branch InformePaginaTerceros
Untracked files:
  (use "git add <file>..." to include in what will be committed)

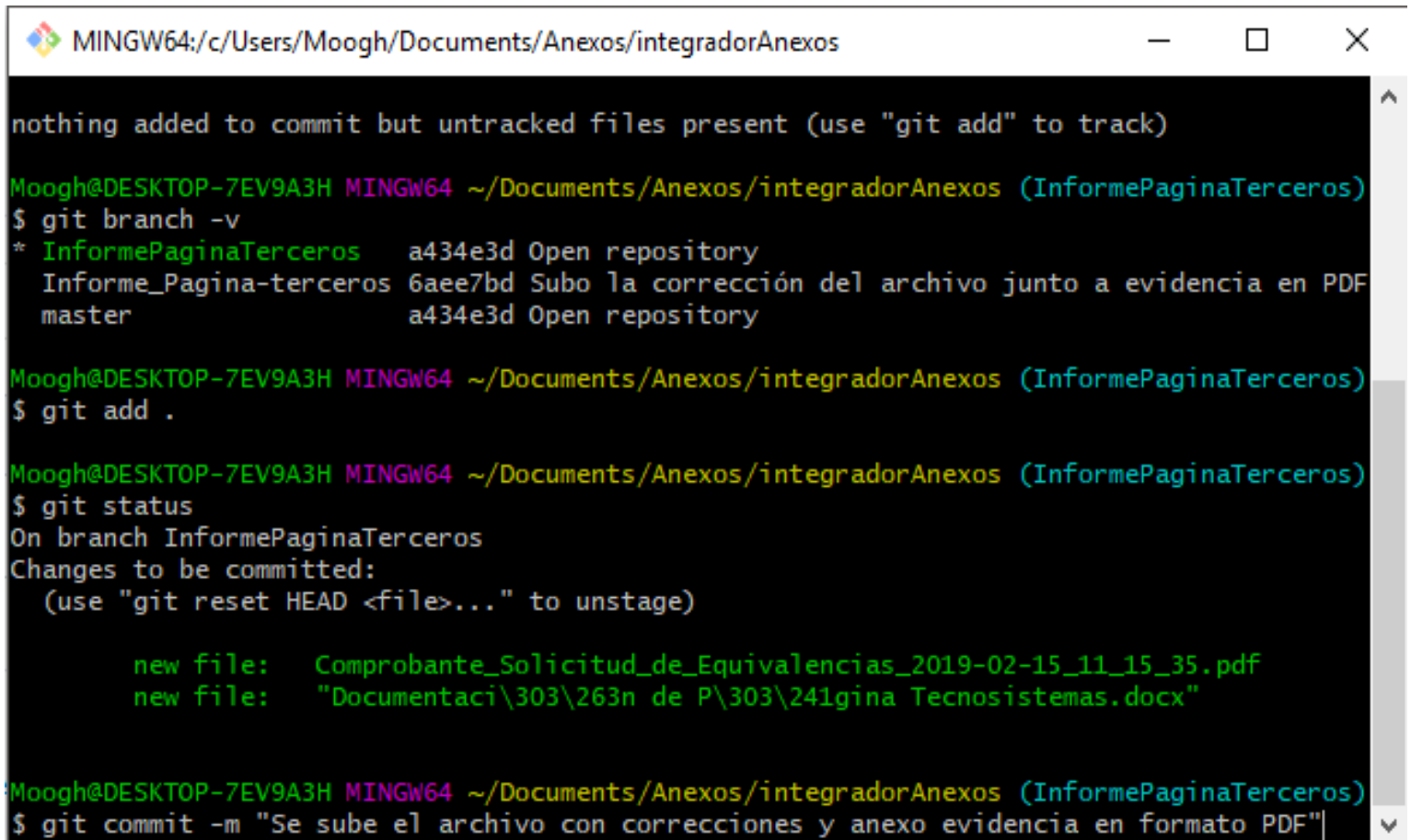
    Comprobante_Solicitud_de_Equivalencias_2019-02-15_11_15_35.pdf
    "Documentaci\303\263n de P\303\241gina Tecnosistemas.docx"

nothing added to commit but untracked files present (use "git add" to track)

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePaginaTerceros)
$
```

13. En este paso se preparan los archivos, para ello se usa el comando "git add nombre_del_archivo" o si se va a preparar varios cambios al mismo tiempo, se utiliza el comando "git add .", tal y como se muestra en la siguiente Figura.

Nota: Al ejecutar de nuevo el comando "git status" los nombres de los archivos deben estar en color verde, esto significa que están listos para confirmar los cambios.



```
MINGW64:/c/Users/Moogh/Documents/Anexos/integradorAnexos

nothing added to commit but untracked files present (use "git add" to track)

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePaginaTerceros)
$ git branch -v
* InformePaginaTerceros    a434e3d Open repository
  Informe_Pagina-terceros  6aee7bd Subo la corrección del archivo junto a evidencia en PDF
  master                  a434e3d Open repository

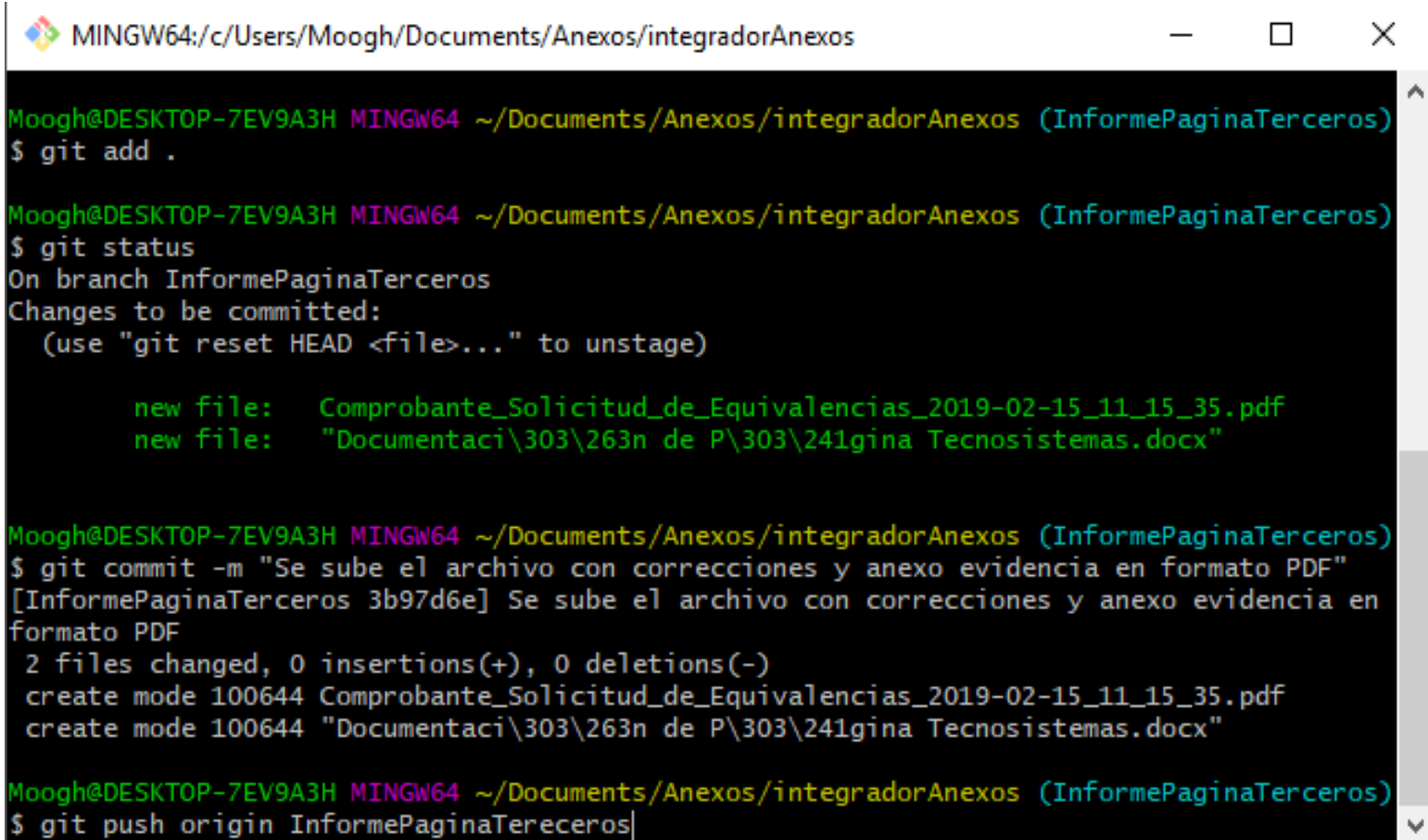
Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePaginaTerceros)
$ git add .

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePaginaTerceros)
$ git status
On branch InformePaginaTerceros
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   Comprobante_Solicitud_de_Equivalencias_2019-02-15_11_15_35.pdf
        new file:   "Documentaci\303\263n de P\303\241gina Tecnosistemas.docx"

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePaginaTerceros)
$ git commit -m "Se sube el archivo con correcciones y anexo evidencia en formato PDF"
```

14. Ya para terminar se ejecuta el comando "git commit -m "mensaje" ", el cual confirma los cambios que se realizaron con un mensaje en el cual se destaque que es lo que se está modificando al repositorio, por último se ejecuta el comando "git push origin nombre_de_la_rama" de esta manera se subirá con éxito las asignaciones (archivos), tal y como se muestra en la siguiente Figura.



```
MINGW64:/c/Users/Moogh/Documents/Anexos/integradorAnexos
Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePaginaTerceros)
$ git add .

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePaginaTerceros)
$ git status
On branch InformePaginaTerceros
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   Comprobante_Solicitud_de_Equivalencias_2019-02-15_11_15_35.pdf
        new file:   "Documentaci\303\263n de P\303\241gina Tecnosistemas.docx"

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePaginaTerceros)
$ git commit -m "Se sube el archivo con correcciones y anexo evidencia en formato PDF"
[InformePaginaTerceros 3b97d6e] Se sube el archivo con correcciones y anexo evidencia en
formato PDF
2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Comprobante_Solicitud_de_Equivalencias_2019-02-15_11_15_35.pdf
create mode 100644 "Documentaci\303\263n de P\303\241gina Tecnosistemas.docx"

Moogh@DESKTOP-7EV9A3H MINGW64 ~/Documents/Anexos/integradorAnexos (InformePaginaTerceros)
$ git push origin InformePaginaTereceros|
```

Si se requiere hacer la revisión de la entrega de algún integrante se debe hacer remoto hacia el Fork ejecutando el comando “git fetch **nuevo_remoto** ruta_fork”, luego se deben bajar los punteros ejecutando el comando “git fetch nuevo_remoto”, por último se debe de posicionar en la rama la cual se quiere revisar.

Después de revisar se debe seguir el siguiente diagrama de flujo.

