

PARTE 2: INICIALIZANDO Git

PASO 2: INICIALIZAR UN REPOSITORIO DE Git

- A. Utilizar el comando `ls` para mostrar el directorio actual. Recuerde que comandos distinguen entre mayúsculas y minúsculas.

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub
$ ls
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub
$
```

- B. A continuación, configure la información de usuario que se utilizará para este repositorio local
- ```
git config --global user.name "MBP54321"
git config --global user.email "manuel.baquero2@misena.edu.co"
```

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub
$ git config --global user.name "MBP54321"
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub
$ git config --global user.email "manuel.baquero2@misena.edu.co"
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub
$ |
```

- C. En cualquier momento, puede revisar esta configuración con el comando `git config --list` (verificar por consola CMD)

```
Símbolo del sistema
C:\Users\USUARIO>git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
init.defaultbranch=master
core.editor="C:\Users\USUARIO\AppData\Local\Programs\Microsoft VS Code\Code.exe" --wait
user.name=MBP54321
user.email=manuel.baquero2@misena.edu.co
```

- D. Utilice el comando `cd` para navegar a la carpeta donde se creará el repositorio

```
MINGW64:/c/Users/USUARIO/Desktop
'DOCS SALUD PADRE'/' WhatsApp.lnk*
'EJERCICIO_CISCOPY/' desktop.ini
'IDLE (Python 3.10 64-bit).lnk'* 'ejemplos CLASE SENA'/
'MBP.lnk*' 'tipos de hongos comestibles.txt'
'Scratch 3.lnk'* '~$ASE 11-10-2021.docx'
'TALLER Git-GitHub'/' '~$PLANTILLA COSTOS.xlsx'
USUARIO@MBP MINGW64 ~/Desktop
$ cd 'TALLER Git-GitHub'/'
```

- E. Haga un directorio **git-intro** y cambie el directorio en él: con el comando **mkdir <dir-name>** (para crear directorio); con el comando **cd** (para cambiar de directorio)

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub
$ mkdir git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub
$ cd git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro
$ |
```

- F. Utilice el comando **git init** para inicializar el directorio actual (**git-intro**) como repositorio de Git. El mensaje que se muestra indica que ha creado un repositorio local dentro del proyecto contenido en el directorio oculto **.git**. Aquí es donde se encuentra todo el historial de cambios. Puedes verlo con el comando **ls -a**

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub
$ cd git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro
$ git init
Initialized empty Git repository in C:/Users/USUARIO/Desktop/TALLER Git-GitHub/g
it-intro/.git/
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$ ls -a
./ ../ .git/
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$
```

- G. A medida que trabaje en su proyecto, querrá comprobar qué archivos han cambiado. Esto es útil cuando está enviando archivos al repositorio, y no desea comprometerlos todos. El comando **git status** muestra los archivos modificados en el directorio de trabajo que se almacenan en etapas para su siguiente confirmación.

Este mensaje le dice:

- Estas en una rama maestra. (Los niveles se discuten más adelante en este laboratorio)
- El mensaje de confirmación es la confirmación inicial.
- No hay cambios que comprometer

Verá que el estado de su repositorio cambiará una vez que se agreguen archivos y se comience a realizar cambios.

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$ git status
On branch master

No commits yet

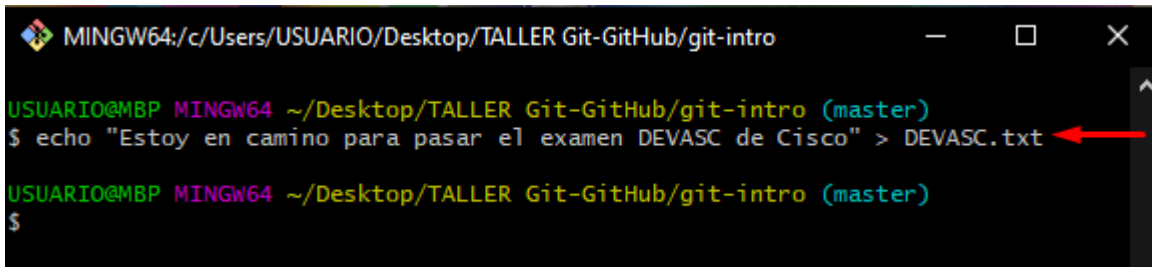
nothing to commit (create/copy files and use "git add" to track)
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
```

### PARTE 3: PREPARANDO Y CONFIRMANDO UN ARCHIVO EN EL REPOSITORIO

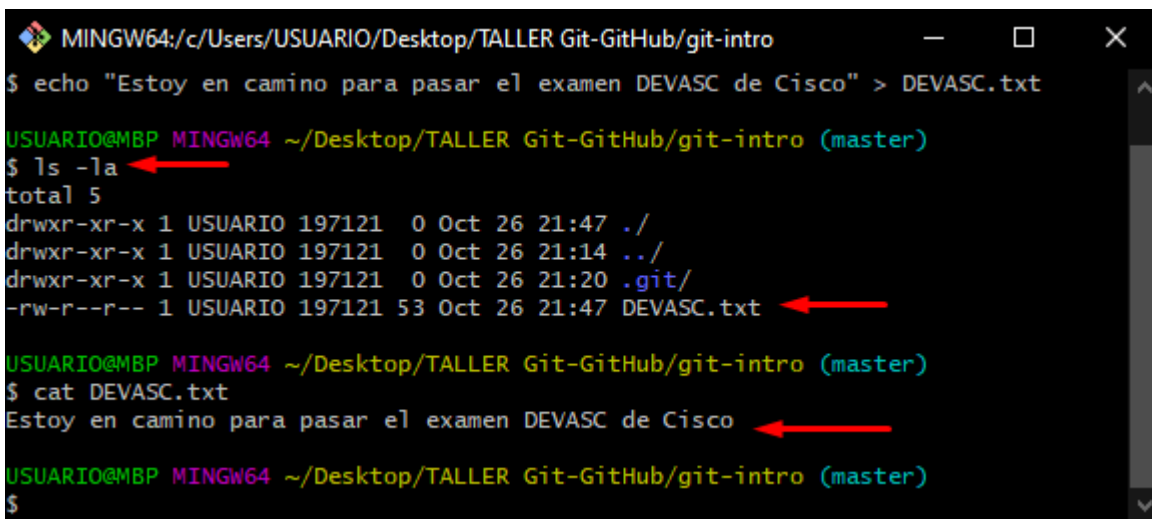
En esta parte creará un archivo, pondrá en escena ese archivo y confirmará ese archivo en el repositorio de Git.

#### PASO 1: CREAR UN ARCHIVO

- A. El repositorio **git-intro** se creó, pero está vacío. Usando el comando **echo**, crea el archivo **DEVASC.txt** con la información contenida entre comillas: **echo "Estoy en camino para pasar el examen DEVASC de Cisco" > DEVASC.txt**

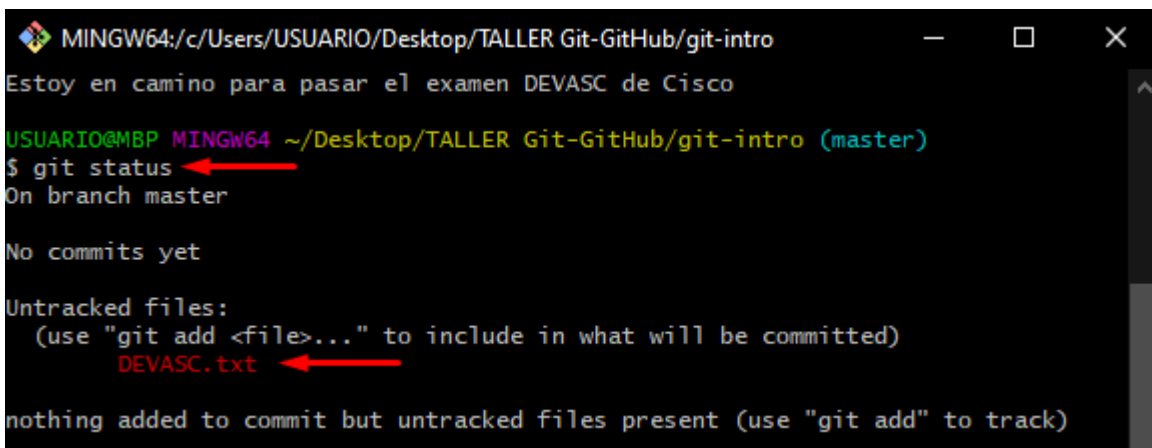
A terminal window titled 'MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro'. The prompt is 'USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)'. The command 'echo "Estoy en camino para pasar el examen DEVASC de Cisco" > DEVASC.txt' is entered and executed. A red arrow points to the command. The prompt returns to '\$'.

- B. Utilizar el comando **ls -la** para verificar el archivo, así como el directorio **.git**, que están en el directorio **git-intro**. A continuación, utilice **cat** para mostrar el contenido de **DEVASC.txt**.

A terminal window titled 'MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro'. The prompt is 'USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)'. The command 'ls -la' is entered and executed. A red arrow points to the command. The output shows the directory listing with '.git/' and 'DEVASC.txt' highlighted by red arrows. Then, the command 'cat DEVASC.txt' is entered and executed. A red arrow points to the command. The output shows the text 'Estoy en camino para pasar el examen DEVASC de Cisco' highlighted by a red arrow. The prompt returns to '\$'.

#### PASO 2: EXAMINAR EL ESTADO DEL REPOSITORIO

Examinar el estado del repositorio usando el comando **git status**. Observar que Git encontró el nuevo archivo en el directorio y sabe que no se rastreó.

A terminal window titled 'MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro'. The prompt is 'USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)'. The command 'git status' is entered and executed. A red arrow points to the command. The output shows 'On branch master', 'No commits yet', and 'Untracked files: (use "git add <file>..." to include in what will be committed) DEVASC.txt'. A red arrow points to 'DEVASC.txt'. The output ends with 'nothing added to commit but untracked files present (use "git add" to track)'. The prompt returns to '\$'.

### PASO 3: PONGA EN EJECUCIÓN EL ARCHIVO

- A. A continuación, use el comando **git add** para "ejecutar" el archivo **DEVASC.txt**. La ejecución es una fase intermedia previa a enviar un archivo al repositorio con el comando **git commit**. Este comando crea el contenido del archivo inmediatamente al mismo tiempo de que el comando se haya introducido. Cualquier cambio en el archivo requiere otro comando **git add** antes de confirmar el archivo.

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ git add DEVASC.txt
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
```

- B. Usando el comando **git status** nuevamente, observe los cambios por etapas que se muestran como:

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ git status
On branch master

No commits yet

Changes to be committed:
 (use "git rm --cached <file>..." to unstage)
 new file: DEVASC.txt
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
```

### PASO 4: CONFIRMANDO UN ARCHIVO.

Ahora que ha puesto sus cambios en ejecución, tiene que confirmarlos para que Git sepa que quiere comenzar a rastrear esos cambios.

Confirmar su contenido por etapas como una nueva confirmación instantánea mediante el comando **git commit**. El mensaje **-m** permitirá agregar un mensaje explicando los cambios realizados.

Observe la combinación de número y letra resaltada en la salida. Este es el ID de confirmación. Cada confirmación se identifica mediante un hash SHA1 único. El ID de confirmación es los primeros 7 caracteres del hash de confirmación completo:

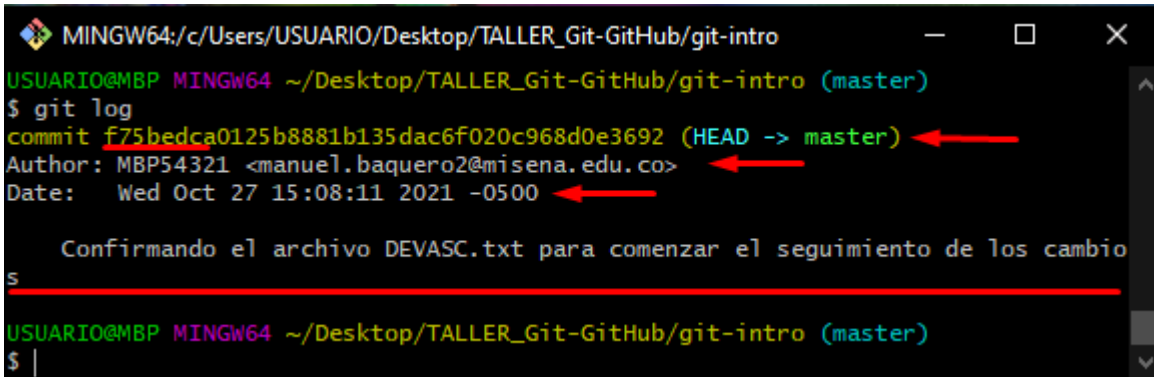
**nota:** si no funciona con comillas dobles probar con comas sencillas)

**git commit -m "Confirmando el archivo DEVASC.txt para comenzar el seguimiento de los cambios"**

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ git commit -m "Confirmando el archivo DEVASC.txt para comenzar el seguimiento de los cambios"
[master (root-commit) f75bedc] Confirmando el archivo DEVASC.txt para comenzar el seguimiento de los cambios
1 file changed, 1 insertion(+)
create mode 100644 DEVASC.txt
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$
```

## PASO 5: VER EL HISTORIAL DE CONFIRMACIÓN.

Utilice el comando **git log** para mostrar todas las confirmaciones en el historial de la rama actual. De forma predeterminada, todas las confirmaciones son realizadas en la rama principal. (Las ramas se discutirán más adelante). La primera línea es el hash de confirmación con el ID de confirmación como los primeros 7 caracteres. El archivo está comprometido con la rama maestra. A continuación, se indica su nombre y dirección de correo electrónico, la fecha de la confirmación y el mensaje que ha incluido con la confirmación.

A terminal window titled 'MINGW64:/c/Users/USUARIO/Desktop/TALLER\_Git-GitHub/git-intro'. The prompt is 'USUARIO@MBP MINGW64 ~/Desktop/TALLER\_Git-GitHub/git-intro (master)'. The command '\$ git log' has been executed. The output shows a commit with hash 'f75bedca0125b8881b135dac6f020c968d0e3692' (HEAD -> master), author 'MBP54321 <manuel.baquero2@misena.edu.co>', and date 'Wed Oct 27 15:08:11 2021 -0500'. The commit message is 'Confirmando el archivo DEVASC.txt para comenzar el seguimiento de los cambios'. Red arrows point to the commit hash, the author's email, and the date. A red line separates this commit from the previous one.

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ git log
commit f75bedca0125b8881b135dac6f020c968d0e3692 (HEAD -> master)
Author: MBP54321 <manuel.baquero2@misena.edu.co>
Date: Wed Oct 27 15:08:11 2021 -0500

 Confirmando el archivo DEVASC.txt para comenzar el seguimiento de los cambios

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$
```

## PARTE 4: MODIFICACIÓN DEL ARCHIVO Y SEGUIMIENTO DE LOS CAMBIOS.

En esta parte, modificará un archivo, pondrá en ejecución el archivo, confirmará el archivo y verificará los cambios en el repositorio.

### PASO 1: MODIFICAR EL ARCHIVO.

- A. Realice un cambio en **DEVASC.txt** mediante el comando **echo**. Asegúrese de usar ">>" para agregar el archivo existente. El ">" sobrescribirá el archivo existente. Utilice el comando **cat** para ver el archivo modificado.

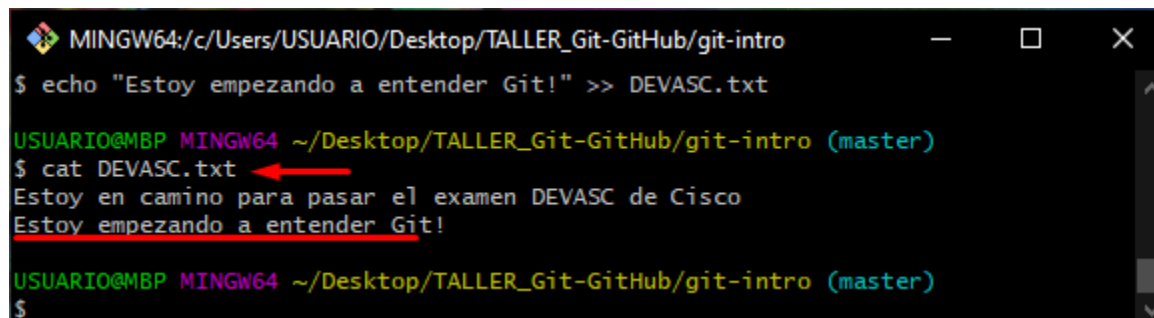
**echo "Estoy empezando a entender Git!" >> DEVASC.txt**

A terminal window showing the execution of the 'echo' command. The prompt is 'USUARIO@MBP MINGW64 ~/Desktop/TALLER\_Git-GitHub/git-intro (master)'. The command '\$ echo "Estoy empezando a entender Git!" >> DEVASC.txt' has been entered. A red arrow points to the file name 'DEVASC.txt'. The previous commit message is visible above.

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ echo "Estoy empezando a entender Git!" >> DEVASC.txt

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$
```

- B. Utilice el comando **cat** para ver el archivo modificado.

A terminal window showing the execution of the 'cat' command. The prompt is 'USUARIO@MBP MINGW64 ~/Desktop/TALLER\_Git-GitHub/git-intro (master)'. The command '\$ cat DEVASC.txt' has been entered. A red arrow points to the file name 'DEVASC.txt'. The output of the command is displayed below the prompt.

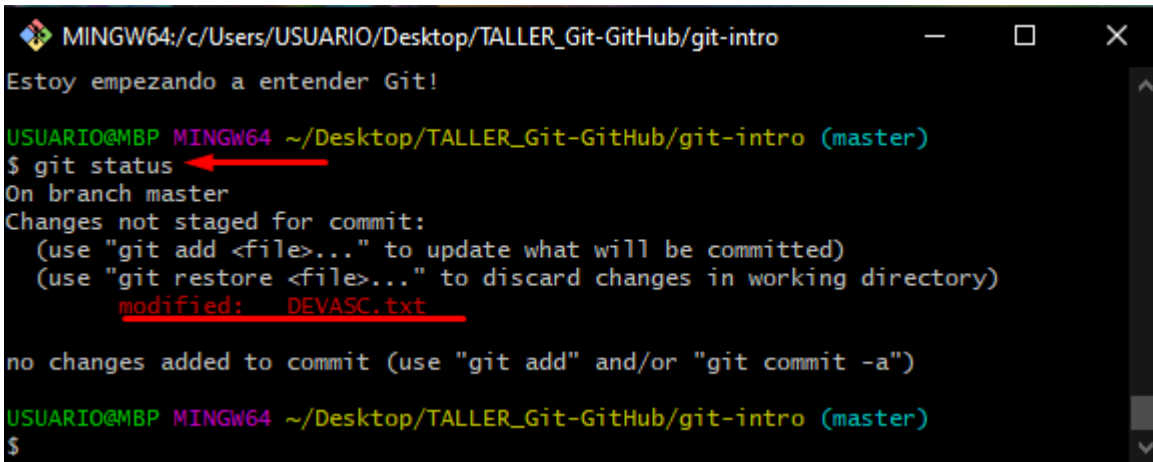
```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
$ echo "Estoy empezando a entender Git!" >> DEVASC.txt

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ cat DEVASC.txt
Estoy en camino para pasar el examen DEVASC de Cisco
Estoy empezando a entender Git!

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$
```

## PASO 2: VERIFICAR EL CAMBIO EN EL REPOSITORIO.

Verificar el cambio en el repositorio usando el comando **git status**.

A terminal window titled 'MINGW64:/c/Users/USUARIO/Desktop/TALLER\_Git-GitHub/git-intro'. The prompt is 'Estoy empezando a entender Git!'. The user runs '\$ git status'. The output shows 'On branch master', 'Changes not staged for commit:', and 'modified: DEVASC.txt' (underlined). It also says 'no changes added to commit'. The prompt returns to '\$'.

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
Estoy empezando a entender Git!

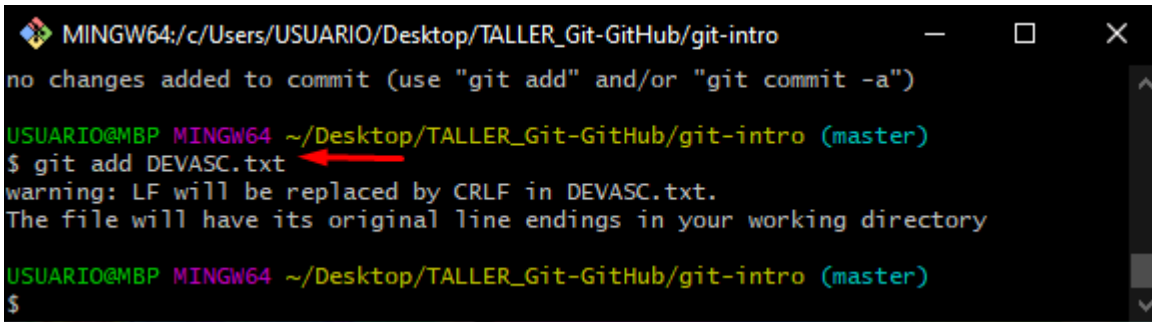
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ git status
On branch master
Changes not staged for commit:
 (use "git add <file>..." to update what will be committed)
 (use "git restore <file>..." to discard changes in working directory)
 modified: DEVASC.txt

no changes added to commit (use "git add" and/or "git commit -a")

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$
```

## PASO 3: EJECUCIÓN DEL ARCHIVO MODIFICADO.

El archivo modificado tendrá que ser puesto en escena de nuevo antes de que pueda ser confirmado usando el comando **git add** nuevamente.

A terminal window titled 'MINGW64:/c/Users/USUARIO/Desktop/TALLER\_Git-GitHub/git-intro'. The prompt is 'no changes added to commit (use "git add" and/or "git commit -a")'. The user runs '\$ git add DEVASC.txt'. The output shows a warning: 'warning: LF will be replaced by CRLF in DEVASC.txt. The file will have its original line endings in your working directory'. The prompt returns to '\$'.

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
no changes added to commit (use "git add" and/or "git commit -a")

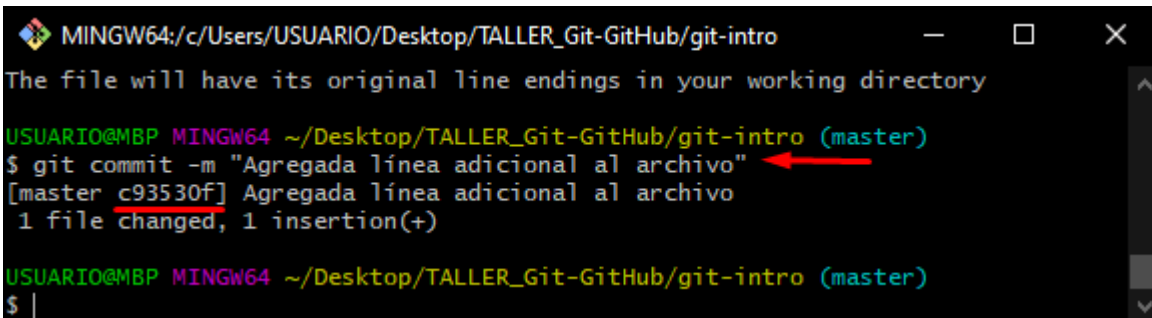
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ git add DEVASC.txt
warning: LF will be replaced by CRLF in DEVASC.txt.
The file will have its original line endings in your working directory

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$
```

## PASO 4: CONFIRMAR EL ARCHIVO POR EJECUCIÓN.

Confirmar el archivo por ejecución usando el comando **git commit**. Observe el nuevo ID de confirmación.

**git commit -m "Agregada línea adicional al archivo"**

A terminal window titled 'MINGW64:/c/Users/USUARIO/Desktop/TALLER\_Git-GitHub/git-intro'. The prompt is 'The file will have its original line endings in your working directory'. The user runs '\$ git commit -m "Agregada línea adicional al archivo"'. The output shows '[master c93530f] Agregada línea adicional al archivo' and '1 file changed, 1 insertion(+)' (underlined). The prompt returns to '\$'.

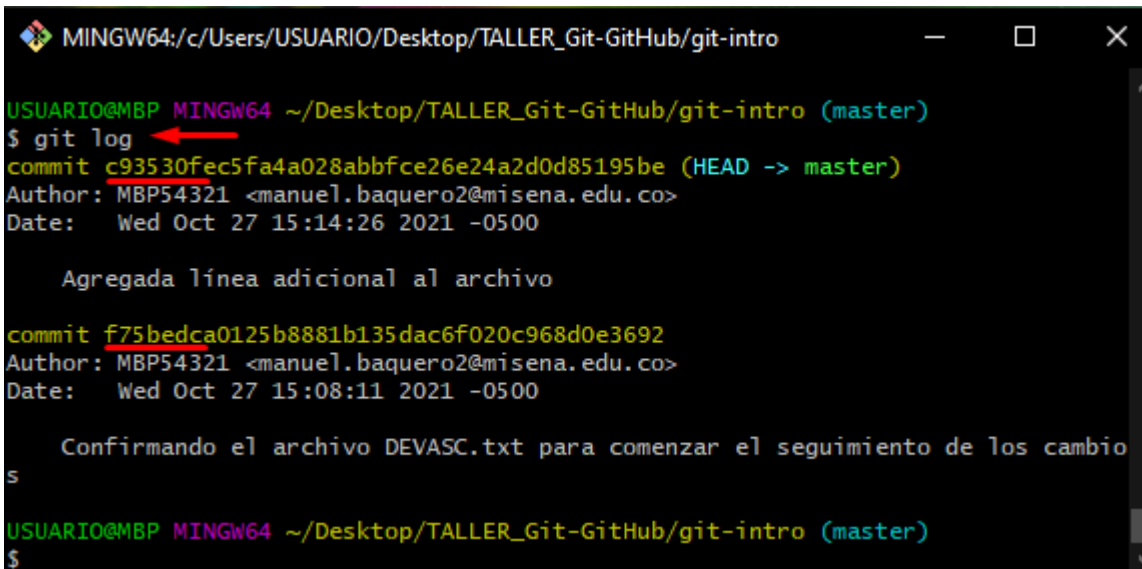
```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
The file will have its original line endings in your working directory

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ git commit -m "Agregada línea adicional al archivo"
[master c93530f] Agregada línea adicional al archivo
1 file changed, 1 insertion(+)

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$
```

## PASO 5: COMPRUEBE LOS CAMBIOS EN EL REPOSITORIO.

- A. Utilice el comando **git log** de nuevo para mostrar todas las confirmaciones. Observe que **el registro contiene la entrada de confirmación original junto con la entrada para la confirmación que acaba de realizar. La última confirmación se muestra primero**. La salida resalta el ID de confirmación (los primeros 7 caracteres del hash SHA1), la fecha/hora de la confirmación y el mensaje de la confirmación para cada entrada.



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ git log
commit c93530fec5fa4a028abbfce26e24a2d0d85195be (HEAD -> master)
Author: MBP54321 <manuel.baquero2@misena.edu.co>
Date: Wed Oct 27 15:14:26 2021 -0500

 Agregada línea adicional al archivo

commit f75bedca0125b8881b135dac6f020c968d0e3692
Author: MBP54321 <manuel.baquero2@misena.edu.co>
Date: Wed Oct 27 15:08:11 2021 -0500

 Confirmando el archivo DEVASC.txt para comenzar el seguimiento de los cambios

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$
```

- B. Cuando tiene varias entradas en el registro, puede comparar las dos confirmaciones usando el comando **git diff** agregando el ID de confirmación original primero y el último ID después: **git diff <commit ID original> <commit ID latest>**. Deberá usar sus ID de confirmación. El signo "+" al final, seguido del texto, indica el contenido que se agregó al archivo.

**git diff f75bedc c93530f**



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ git diff f75bedc c93530f
diff --git a/DEVASC.txt b/DEVASC.txt
index 665c474..14aaf45 100644
--- a/DEVASC.txt
+++ b/DEVASC.txt
@@ -1,2 @@
- Estoy en camino para pasar el examen DEVASC de Cisco
+Estoy empezando a entender Git!

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ |
```



## PARTE 5: RAMAS Y FUSIONES.

Cuando se crea un repositorio, los archivos se colocan automáticamente en una rama llamada **master**. Siempre que sea posible, se recomienda utilizar ramas alternas o secundarias, en lugar de actualizar directamente la rama maestra.

La derivación se utiliza para que pueda realizar cambios en otra área sin afectar a la rama maestra. Esto se hace para ayudar a evitar actualizaciones accidentales que podrían sobrescribir el código existente.

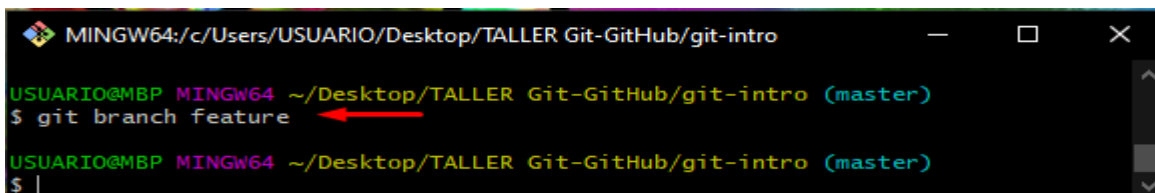
En esta parte:

- creará una nueva rama
- retirará la rama
- hará cambios en la rama
- ejecutará y confirmará la rama
- fusionará los cambios de esa rama en la rama principal
- y luego eliminará la rama.

### PASO 1: CREAR UNA NUEVA RAMA.

Cree una nueva rama llamada **feature** usando el comando **git branch <branch-name>**

**git branch feature**



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$ git branch feature
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$
```

### PASO 2: VERIFICAR LA RAMA ACTUAL.

Utilice el comando **git branch**, sin nombre de rama, para mostrar todas las ramas de este repositorio. El "\*" junto a la rama principal indica que se trata de la rama actual, la rama que está actualmente "desprotegida".



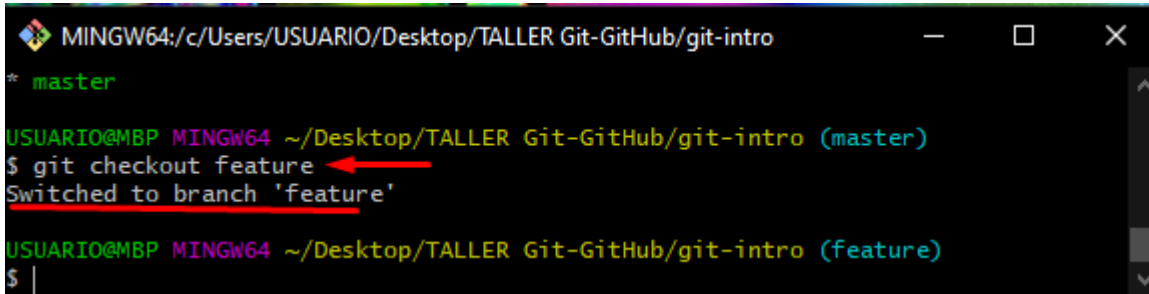
```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$ git branch
feature
* master
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$
```



### PASO 3: VERIFIQUE LA NUEVA RAMA.

Utilizar el comando **git checkout <branch-name>** para cambiar la rama de la master a la que requiera trabajar, en este caso rama feature (características en español).

**git checkout feature**



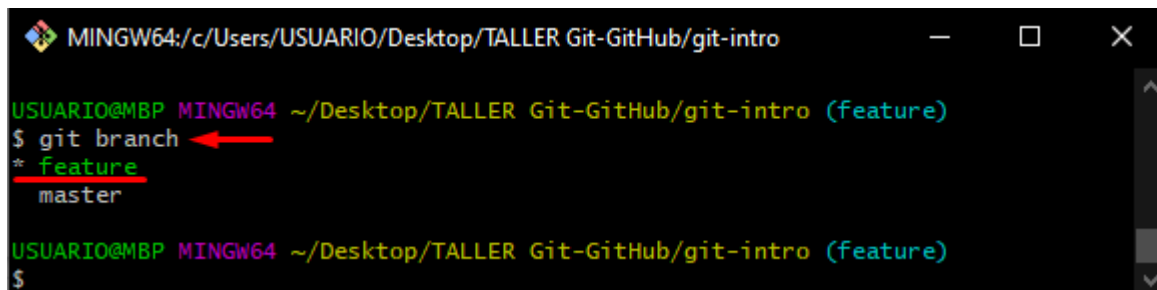
```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro
* master

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$ git checkout feature
Switched to branch 'feature'

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (feature)
$
```

### PASO 4: VERIFIQUE LA RAMA ACTUAL.

- A. Verificar que haya cambiado la **rama feature** usando el comando **git branch**. Tenga en cuenta el "\*" junto a la rama características. Esta es ahora la rama de trabajo.

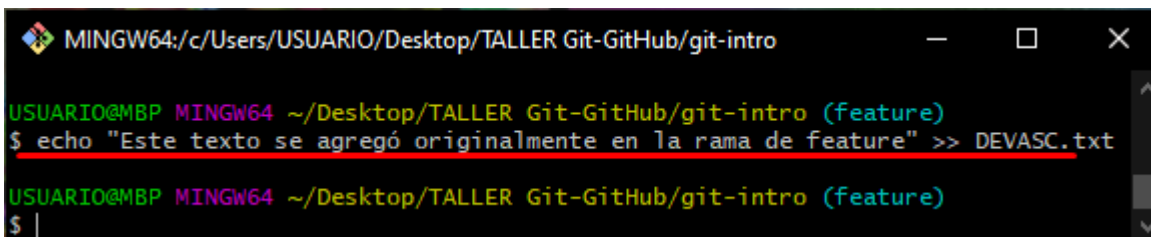


```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (feature)
$ git branch
* feature
 master

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (feature)
$
```

- B. Añadir una nueva línea de texto al archivo DEVASC.txt, utilizando nuevamente el comando **echo** con los signos ">>".  
**echo "Este texto se agregó originalmente en la rama de feature" >> DEVASC.txt**

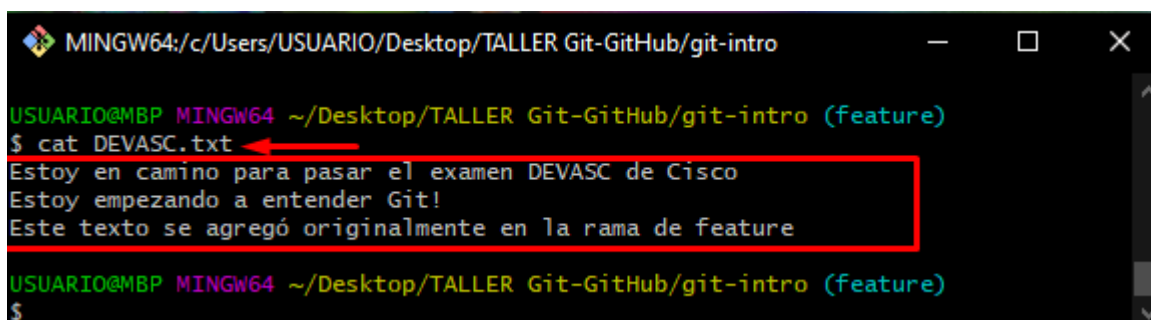


```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (feature)
$ echo "Este texto se agregó originalmente en la rama de feature" >> DEVASC.txt

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (feature)
$
```

- C. Comprobar que la línea se ha añadido al archivo mediante el comando **cat**.



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (feature)
$ cat DEVASC.txt
Estoy en camino para pasar el examen DEVASC de Cisco
Estoy empezando a entender Git!
Este texto se agregó originalmente en la rama de feature

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (feature)
$
```

## PASO 5: PRESENTE EL ARCHIVO MODIFICADO EN LA RAMA DE CARACTERÍSTICAS.

- A. Presentar el archivo actualizado a la rama de características actual, con el comando **git add**

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (feature)
$ git add DEVASC.txt

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (feature)
$ |
```

- B. Utilice el comando **git status** y observe que el archivo modificado DEVASC.txt está en la rama **feature**.

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (feature)
$ git status
On branch feature
Changes to be committed:
 (use "git restore --staged <file>..." to unstage)
 modified: DEVASC.txt

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (feature)
$ |
```

## PASO 6: CONFIRMAR EL ARCHIVO POR EJECUCIÓN EN LA RAMA DE CARACTERÍSTICAS.

- A. Confirmar el archivo por ejecución usando el comando **git commit**. Observe el nuevo ID de confirmación y su mensaje

**git commit -m "Agregada una tercera línea en la rama de feature"**

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (feature)
$ git commit -m "Agregada una tercera línea en la rama de feature"
[feature a48f60d] Agregada una tercera línea en la rama de feature
1 file changed, 1 insertion(+)

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (feature)
$ |
```

- B. Use el comando **git log** para mostrar todas las confirmaciones, incluida la confirmación que acaba de hacer en la rama **de feature**. La confirmación previa se realizó dentro de la rama principal.

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (feature)
$ git log
commit a48f60d33b25258c6ff00b6663a61f702adb5631 (HEAD -> feature)
Author: MBP54321 <manuel.baquero2@misena.edu.co>
Date: Wed Oct 27 15:26:08 2021 -0500

 Agregada una tercera línea en la rama de feature

commit c93530fec5fa4a028abbfce26e24a2d0d85195be (master)
Author: MBP54321 <manuel.baquero2@misena.edu.co>
Date: Wed Oct 27 15:14:26 2021 -0500

 Agregada línea adicional al archivo

commit f75bedca0125b8881b135dac6f020c968d0e3692
Author: MBP54321 <manuel.baquero2@misena.edu.co>
Date: Wed Oct 27 15:08:11 2021 -0500

 Confirmando el archivo DEVASC.txt para comenzar el seguimiento de los cambios

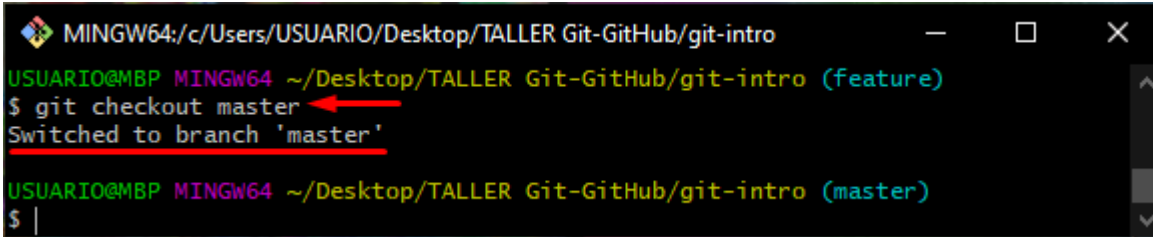
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (feature)
$ |
```

## PASO 7: VERIFICACIÓN DE LA RAMA MAESTRA.

Cambie a la rama maestra usando el comando **git checkout <branch-name>** y verifique la rama de trabajo actual usando el comando **git branch**.

```
git checkout master
```

```
git Branch
```



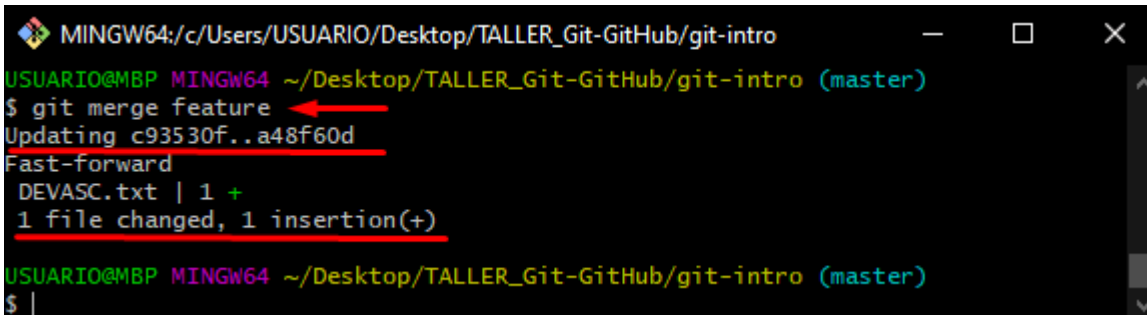
```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (feature)
$ git checkout master
Switched to branch 'master'
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$
```

## PASO 8: COMBINAR EL CONTENIDO DEL ARCHIVO DE LA RAMA FEATURE CON LA RAMA MAESTRA.

- A. Las ramas se utilizan a menudo al implementar nuevas características o correcciones. Los miembros del equipo pueden enviarlos para su revisión, y luego, una vez verificados, pueden ser arrastrados a la base de código principal: la rama maestra

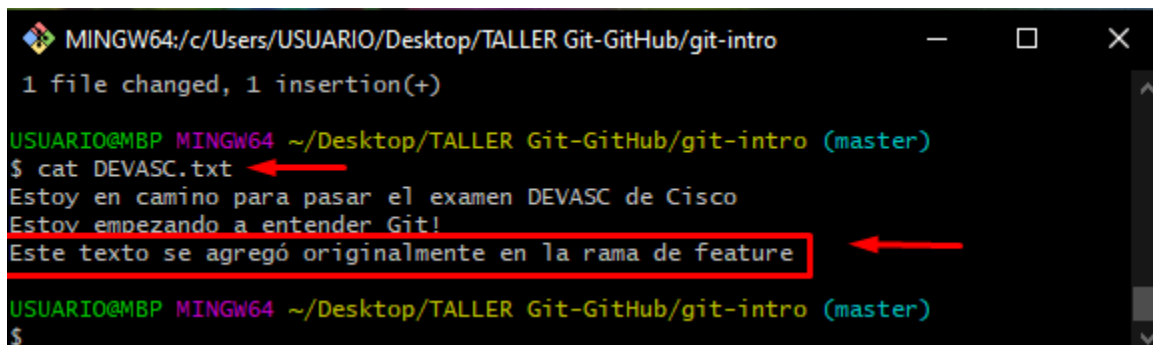
Combine el contenido (conocido como el historial) de la rama de características en la rama maestra usando el comando **git merge <branch-name>**. El nombre de la rama es la rama de la que se extraen los historiales a la rama actual. La salida muestra que se ha cambiado un archivo con una línea insertada

```
git merge feature
```



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$ git merge feature
Updating c93530f..a48f60d
Fast-forward
 DEVASC.txt | 1 +
 1 file changed, 1 insertion(+)
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$
```

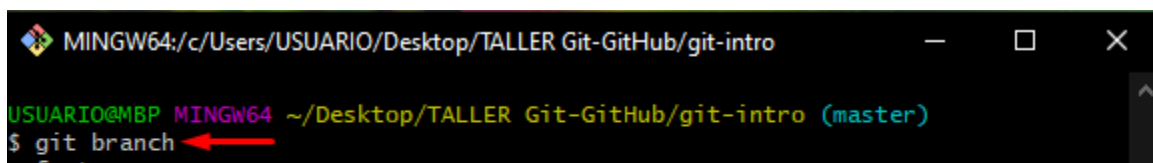
- B. Compruebe el contenido anexado al archivo **DEVASC.txt** en la rama maestra mediante el comando **cat**.



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro
1 file changed, 1 insertion(+)
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$ cat DEVASC.txt
Estoy en camino para pasar el examen DEVASC de Cisco
Estoy empezando a entender Git!
Este texto se agregó originalmente en la rama de feature
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$
```

## PASO 9: ELIMINAR UNA RAMA.

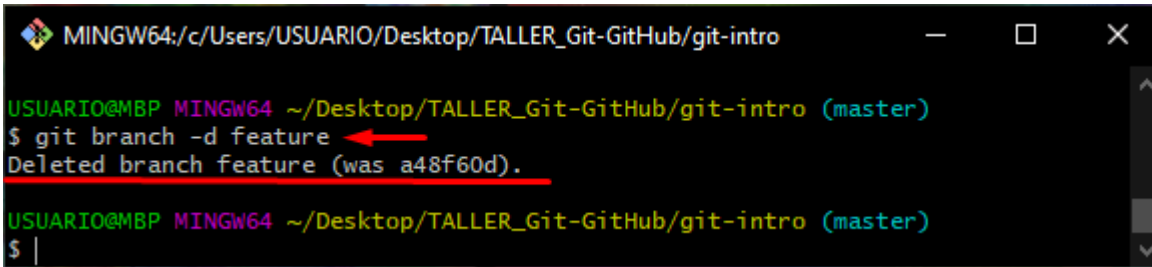
- A. Verifique que la rama de **feature** aún esté disponible usando el comando **git branch**.



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$ git branch
```

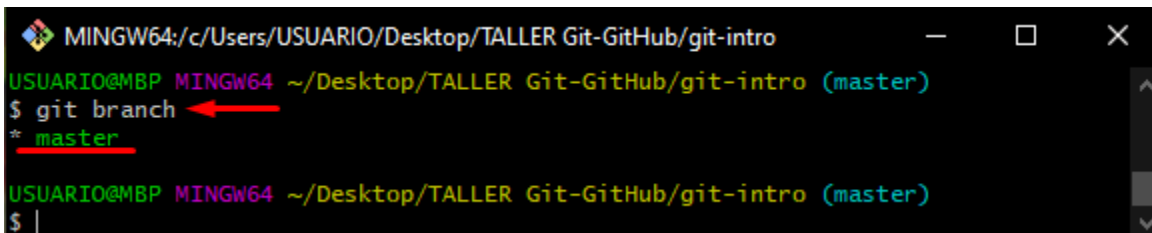
- B. Eliminar la rama de **feature** usando el comando **git branch -d <branch-name>**

**git branch -d feature**



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ git branch -d feature
Deleted branch feature (was a48f60d).
```

- C. Verifique que la rama de **feature** ya no esté disponible usando el comando **git branch**.



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ git branch
* master
```

## PARTE 6: MANEJO DE CONFLICTOS DE FUSIÓN.

A veces, puede experimentar un conflicto de fusión. Esto es cuando es posible que haya realizado cambios superpuestos en un archivo, y Git no puede combinar automáticamente los cambios.

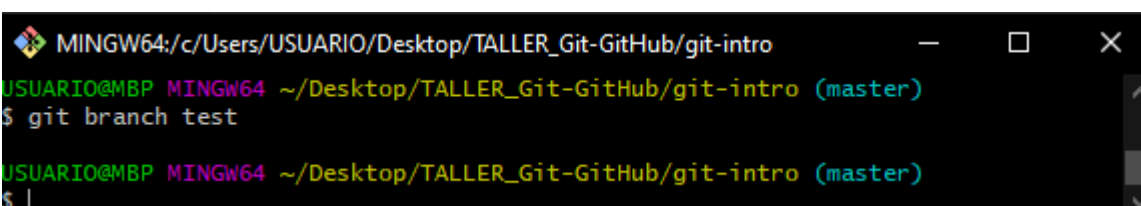
En esta parte

- creará una rama de prueba
- modificará su contenido
- pondrá en ejecución y confirmará la rama de prueba
- cambiará a la rama maestra
- modificará el contenido de nuevo
- pondrá en ejecución y comprometerá la rama maestra
- intentará fusionar ramas
- localizar y resolver el conflicto
- poner en escena y confirmar la rama maestra de nuevo
- y verificará su confirmación.

### PASO 1: CREAR UNA NUEVA PRUEBA DE RAMA.

Crear una nueva rama de **test**

**git branch test**

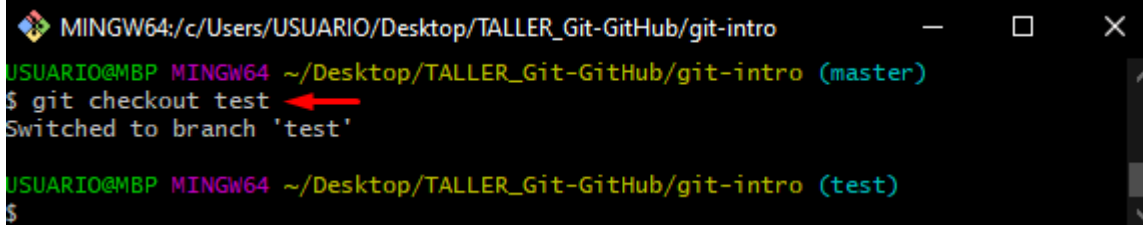


```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ git branch test
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$
```

## PASO 2: REVISAR LA RAMA DE PRUEBA

- A. Cambiar a la rama **test**

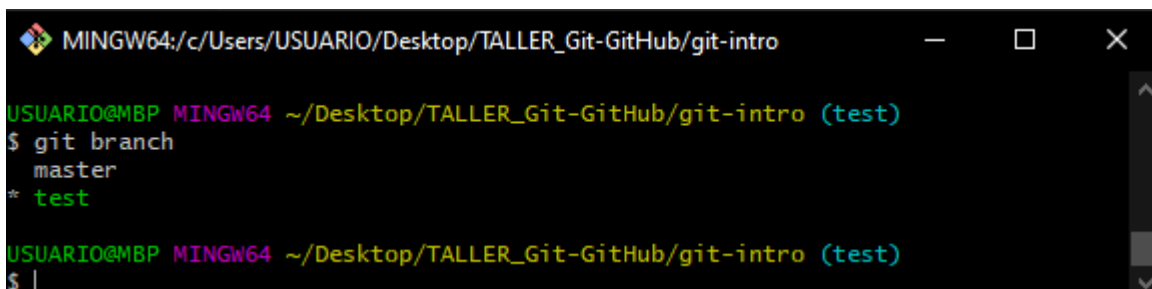
**git checkout test**



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ git checkout test
Switched to branch 'test'
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (test)
$
```

- B. Verificar que la rama de trabajo sea la rama **test**

**git branch**

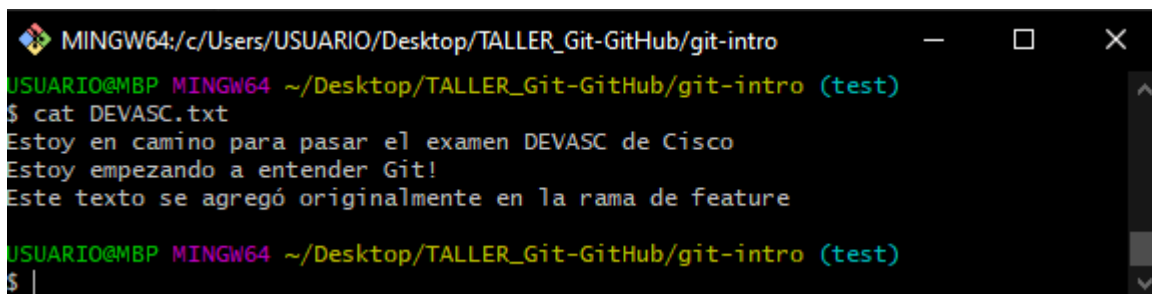


```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (test)
$ git branch
 master
* test
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (test)
$
```

## PASO 3: COMPROBAR EL CONTENIDO ACTUAL DE DEVASC.txt

Comprobar el contenido actual del archivo **DEVASC.txt**. Observar que la primera línea incluya la palabra **"Cisco"**.

**cat DEVASC.txt**

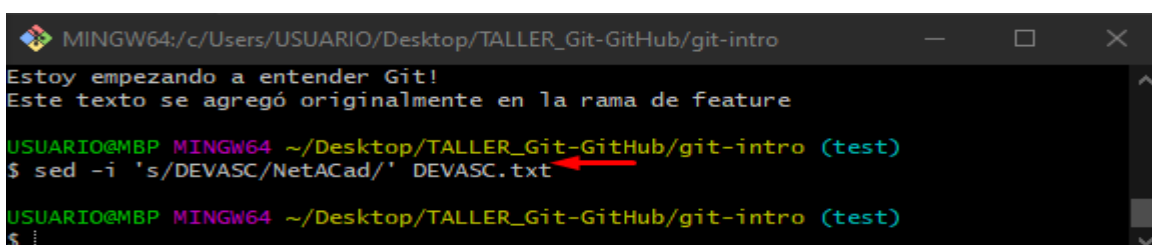


```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (test)
$ cat DEVASC.txt
Estoy en camino para pasar el examen DEVASC de Cisco
Estoy empezando a entender Git!
Este texto se agregó originalmente en la rama de feature
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (test)
$
```

## PASO 4: MODIFICAR EL CONTENIDO DE DEVASC.txt EN LA RAMA prueba.

Utilizar el comando **sed** para cambiar la palabra **"Cisco"** a **"NetACad"** en el archivo **DEVASC.txt**.

**sed -i 's/Cisco/NetACad/' DEVASC.txt**



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
Estoy empezando a entender Git!
Este texto se agregó originalmente en la rama de feature
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (test)
$ sed -i 's/DEVASC/NetACad/' DEVASC.txt
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (test)
$
```

## PASO 5: COMPROBAR EL CONTENIDO DEL DEVASC.txt MODIFICADO EN LA RAMA test.

Comprobar el cambio en el archivo **DEVASC.txt**: **cat DEVASC.txt**

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (prueba)
$ cat DEVASC.txt
Estoy en camino para pasar el examen DEVASC de NetACad
Estoy empezando a entender Git!
Este texto se agregó originalmente en la rama de feature

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (prueba)
$
```

## PASO 6: PONGA EN ESCENA Y CONFIRME LA RAMA test.

Ponga en escena **git add** y confirme el archivo con un solo comando **git commit -a**. La opción **-a** sólo afecta a los archivos modificados y eliminados. No afecta a los archivos nuevos.

**git commit -a -m "Cambiar Cisco a NetACad"**

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (test)
$ git commit -a -m "Cambiar Cisco a NetACad"
[test 349d744] Cambiar Cisco a NetACad
1 file changed, 1 insertion(+), 1 deletion(-)

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (test)
$
```

## PASO 7: REVISAR LA RAMA PRINCIPAL.

A. Cambiar a la rama principal: **git checkout master**

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (test)
$ git checkout master
Switched to branch 'master'

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$
```

B. Compruebe que la rama principal es la rama de trabajo actual: **git branch**

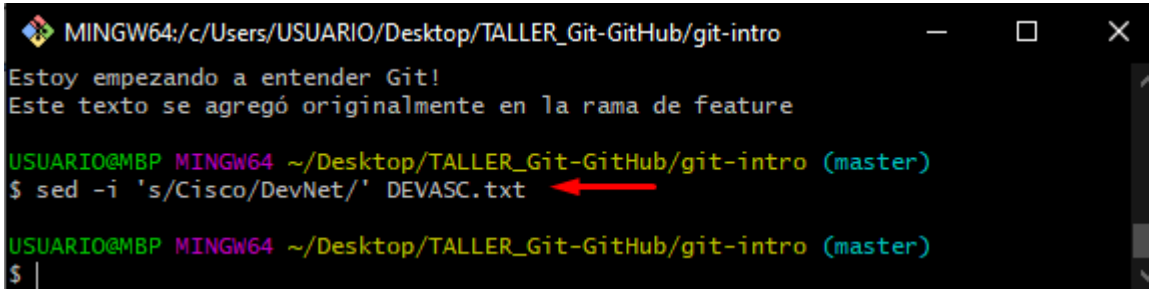
```
MINGW64:/c/Users/USUARIO/Desktop/TALLER Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$ git branch
* master
test

USUARIO@MBP MINGW64 ~/Desktop/TALLER Git-GitHub/git-intro (master)
$
```

## PASO 8: MODIFICAR EL CONTENIDO DE DEVASC.txt EN LA RAMA PRINCIPAL

Utilice el comando **sed** para cambiar la palabra "Cisco" a "DevNet" en el archivo **DEVASC.txt**

```
sed -i 's/Cisco/DevNet/' DEVASC.txt
```



A terminal window titled 'MINGW64:/c/Users/USUARIO/Desktop/TALLER\_Git-GitHub/git-intro'. The prompt is 'USUARIO@MBP MINGW64 ~/Desktop/TALLER\_Git-GitHub/git-intro (master)'. The command '\$ sed -i 's/Cisco/DevNet/' DEVASC.txt' is entered and executed. A red arrow points to the command. The output shows the file content after the change: 'Estoy empezando a entender Git!', 'Este texto se agregó originalmente en la rama de feature', and 'Estoy en camino para pasar el examen DEVASC de DevNet'.

## PASO 9: COMPRUEBE EL CONTENIDO DEL ARCHIVO DEVASC.txt MODIFICADO EN LA RAMA PRINCIPAL.

Verificar el cambio en el archivo: **cat DEVASC.txt**

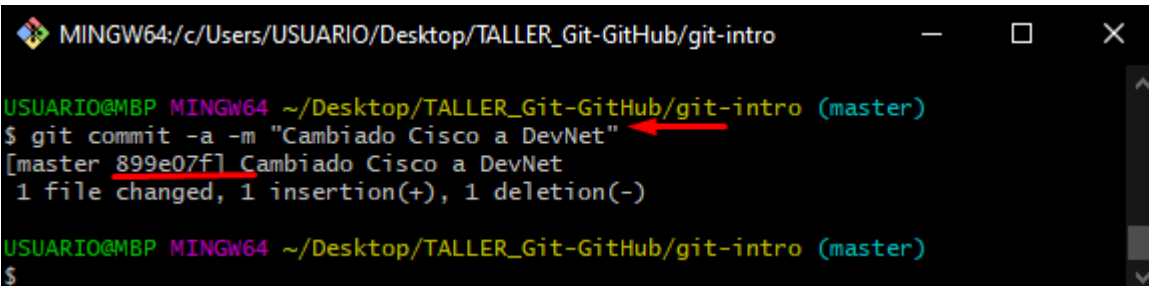


A terminal window titled 'MINGW64:/c/Users/USUARIO/Desktop/TALLER\_Git-GitHub/git-intro'. The prompt is 'USUARIO@MBP MINGW64 ~/Desktop/TALLER\_Git-GitHub/git-intro (master)'. The command '\$ cat DEVASC.txt' is entered and executed. A red arrow points to the command. The output shows the file content: 'Estoy en camino para pasar el examen DEVASC de DevNet', 'Estoy empezando a entender Git!', and 'Este texto se agregó originalmente en la rama de feature'.

## PASO 10: PONGA EN EJECUCIÓN Y CONFIRME LA RAMA DE MAESTRA.

Ponga en ejecución y confirme el archivo con un solo comando **git commit -a**.

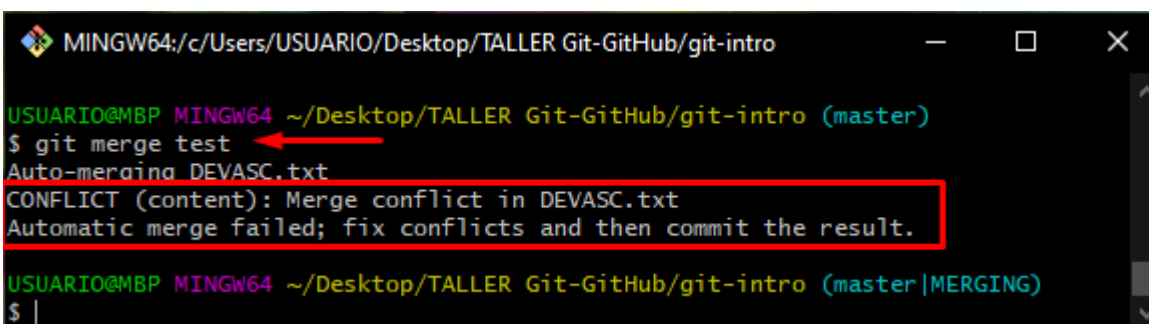
```
git add DEVASC.txt
git commit -a -m "Cambiado Cisco a DevNet"
```



A terminal window titled 'MINGW64:/c/Users/USUARIO/Desktop/TALLER\_Git-GitHub/git-intro'. The prompt is 'USUARIO@MBP MINGW64 ~/Desktop/TALLER\_Git-GitHub/git-intro (master)'. The command '\$ git commit -a -m "Cambiado Cisco a DevNet"' is entered and executed. A red arrow points to the command. The output shows the commit message: '[master 899e07f] Cambiado Cisco a DevNet' and '1 file changed, 1 insertion(+), 1 deletion(-)'.

## PASO 11: INTENTAR FUSIONAR LA RAMA test EN LA RAMA maestra

Intentar combinar el historial de rama test en la rama maestra: **git merge test**



A terminal window titled 'MINGW64:/c/Users/USUARIO/Desktop/TALLER\_Git-GitHub/git-intro'. The prompt is 'USUARIO@MBP MINGW64 ~/Desktop/TALLER\_Git-GitHub/git-intro (master)'. The command '\$ git merge test' is entered and executed. A red arrow points to the command. The output shows a conflict: 'CONFLICT (content): Merge conflict in DEVASC.txt' and 'Automatic merge failed; fix conflicts and then commit the result.' The prompt changes to '(master|MERGING)'.



## PASO 12: ENCUENTRA EL CONFLICTO.

- A. Utilice el comando **git log** para ver las confirmaciones. Observe que la versión HEAD es la rama maestra. Esto será útil en el siguiente paso.

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master|MERGING)
$ git log
commit 899e07f211afeb4b047889f635affbb9388d5ca5 (HEAD -> master)
Author: MBP54321 <manuel.baquero2@misena.edu.co>
Date: Wed Oct 27 17:16:09 2021 -0500

 Cambiado Cisco a DevNet
```

- B. Usar el comando **cat** para ver los contenidos del archivo **DEVASC.txt**. El archivo ahora contiene información para ayudarlo a encontrar el conflicto. La versión **HEAD (rama maestra)** que contiene la palabra "**DevNet**" está en conflicto con la versión de la rama **test** y la palabra "**NetACad**"

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master|MERGING)
$ cat DEVASC.txt
<<<<<<< HEAD
Estoy en camino para pasar el examen DEVASC de DevNet
=====
Estoy en camino para pasar el examen DEVASC de NetACad
>>>>>>> test
Estoy empezando a entender Git!
Este texto se agregó originalmente en la rama de feature

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master|MERGING)
$ |
```

## PASO 13: EDITAR MANUALMENTE EL ARCHIVO DEVASC.txt PARA ELIMINAR EL TEXTO EN CONFLICTO.

- A. Utilice el comando **vim** para editar el archivo: **vim DEVASC.txt**

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
<<<<<<< HEAD
Estoy en camino para pasar el examen DEVASC de DevNet
=====
Estoy en camino para pasar el examen DEVASC de NetACad
>>>>>>> test
Estoy empezando a entender Git!
Este texto se agregó originalmente en la rama de feature
~
~
~
DEVASC.txt [dos] (01:05 27/10/2021) 1,1 A11
"DEVASC.txt" [dos] 7L, 240B
```

- B. Utilice las flechas arriba y abajo para seleccionar la línea de texto adecuada. Presione **dd (delete)** en las siguientes líneas que se resaltan. **dd** eliminará la línea en la que está el cursor.

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
===== HEAD
Estoy en camino para pasar el examen DEVASC de DevNet
=====
Estoy en camino para pasar el examen DEVASC de NetACad
>>>>>>> test
Estoy empezando a entender Git!
Este texto se agregó originalmente en la rama de feature
~
~
~
DEVASC.txt [dos] (17:17 27/10/2021) 2,1 A11
-- INSERT --
```

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
Estoy en camino para pasar el examen DEVASC de DevNet
Estoy empezando a entender Git!
Este texto se agregó originalmente en la rama de feature
~
~
~
~
DEVASC.txt[+] [dos] (17:37 27/10/2021) 2,1 A11
:wq
```

#### PASO 14: COMPROBAR LAS EDICIONES DE DEVASC.txt EN LA RAMA PRINCIPAL.

Comprobar los cambios mediante el comando **cat**.

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master|MERGING)
$ cat DEVASC.txt
Estoy en camino para pasar el examen DEVASC de DevNet
Estoy empezando a entender Git!
Este texto se agregó originalmente en la rama de feature
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master|MERGING)
$
```

#### PASO 15: PONGA EN EJECUCIÓN Y CONFIRME LA RAMA DE MAESTRA

Ponga en ejecución y confirme el archivo con comando **git add** y **git commit -a**.

```
git add DEVASC.txt
```

```
git commit -a -m "Fusionada manualmente desde la rama test"
```

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master|MERGING)
$ git add DEVASC.txt
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master|MERGING)
$ git commit -a -m "Fusionada manualmente desde la rama test"
[master dacf457] Fusionada manualmente desde la rama test
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$
```

#### PASO 16: VERIFICAR LA CONFIRMACIÓN.

Usar el comando **git log** para mirar las confirmaciones. Si es necesario, puede usar **q** para salir de la pantalla del registro de git.

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro (master)
$ git log
commit dacf45750dbc716bdb4868a0b44258d81d92dd40 (HEAD -> master)
Merge: 899e07f 349d744
Author: MBP54321 <manuel.baquero2@misena.edu.co>
Date: Wed Oct 27 17:39:41 2021 -0500

 Fusionada manualmente desde la rama test
```

## PARTE 6: INTEGRACIÓN de Git CON GitHub

Hasta ahora, todos los cambios que ha realizado en su archivo se han almacenado en su máquina local. Git se ejecuta localmente y no requiere ningún servidor de archivos central ni servicio de alojamiento basado en la nube. Git permite a un usuario almacenar y administrar archivos localmente.

Aunque Git es útil para un solo usuario, integrar el repositorio local de Git con un servidor basado en la nube como GitHub es útil cuando se trabaja dentro de un equipo. Cada miembro del equipo mantiene una copia en el repositorio de su máquina local y actualiza el repositorio central basado en la nube para compartir cualquier cambio.

Hay bastantes servicios populares de Git, incluyendo GitHub, Stash de Atlassian y GitLab. Debido a que es fácilmente accesible, usará GitHub en estos ejemplos.

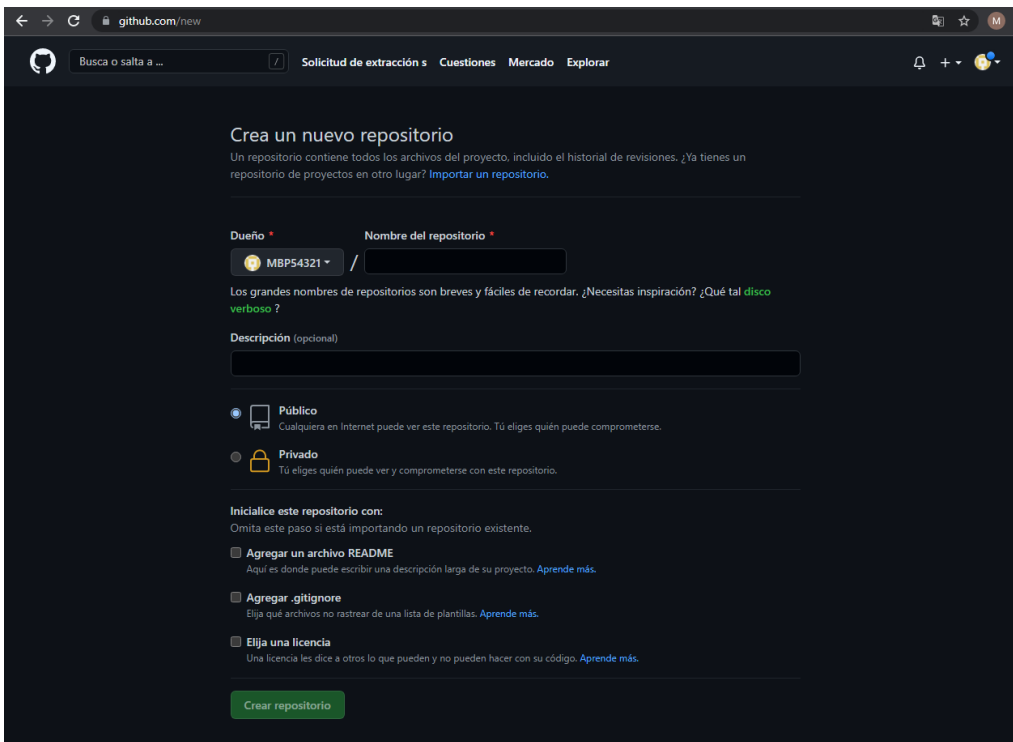
### PASO 1: CREAR UNA CUENTA GitHub.

### PASO 2: INICIE SESIÓN EN SU CUENTA DE GitHub - CREAR UN REPOSITORIO.

Inicie sesión en su cuenta de GitHub.

### PASO 3: CREAR UN REPOSITORIO.

- A. Seleccione el botón **"Nuevo repositorio"** o haga clic en el ícono **"+"** en la esquina superior derecha y seleccione **"Nuevo repositorio"**.



The screenshot shows the GitHub 'Create a new repository' page. At the top, there's a navigation bar with the GitHub logo, a search bar, and links for 'Solicitud de extracción', 'Cuestiones', 'Mercado', and 'Explorar'. Below the navigation bar, the main heading is 'Crea un nuevo repositorio'. A subtext explains that a repository contains all project files and revision history, and offers a link to 'Importar un repositorio' if the user already has one. The form includes fields for 'Dueño' (selected as 'MBP54321') and 'Nombre del repositorio'. A tip suggests using short, memorable names, with a link to 'disco verboso'. There's an optional 'Descripción' field. Under 'Visibility', 'Público' is selected, with a note that anyone on the internet can view it. 'Privado' is also an option, noting that only selected users can view and commit. Below this, a section 'Inicialice este repositorio con:' offers three checkboxes: 'Agregar un archivo README' (with a link to 'Aprende más'), 'Agregar .gitignore' (with a link to 'Aprende más'), and 'Elija una licencia' (with a link to 'Aprende más'). At the bottom, there is a green 'Crear repositorio' button.

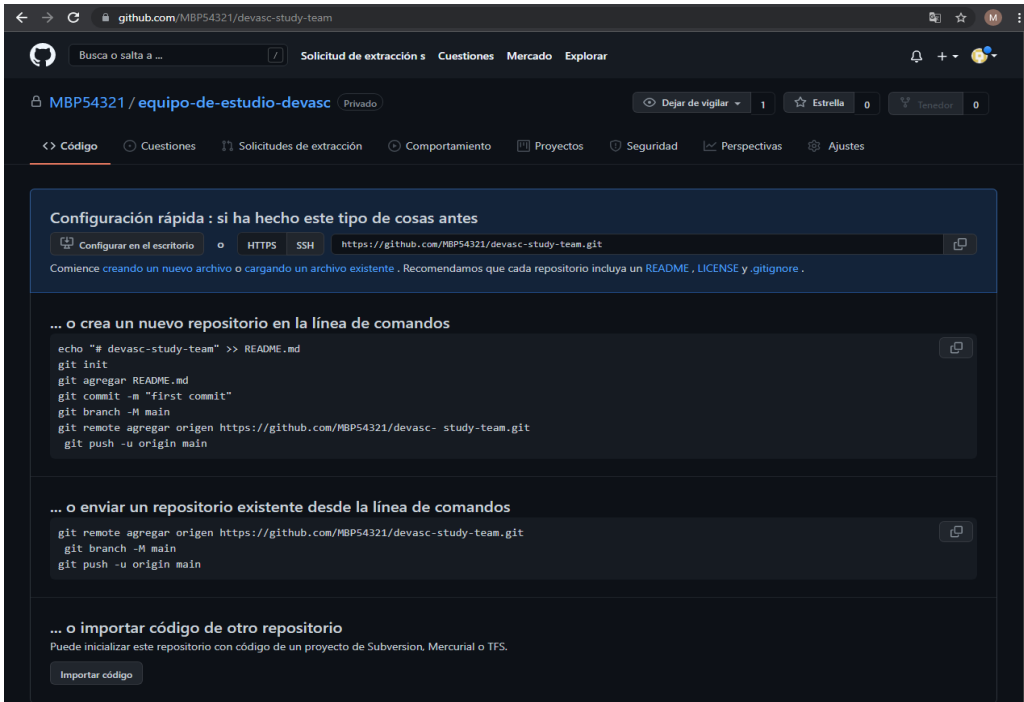
B. Cree un repositorio utilizando la siguiente información:

**Nombre del repositorio:** devasc-study-team

**Descripción:** Trabajando juntos para aprobar el examen DEVASC

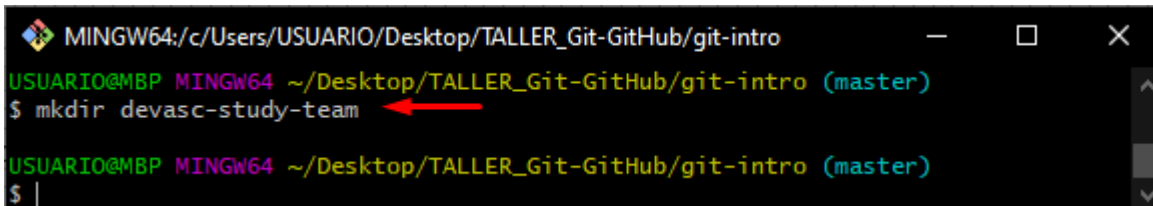
**Público/Privado:** Privado

C. Seleccionar: **Crear repositorio**

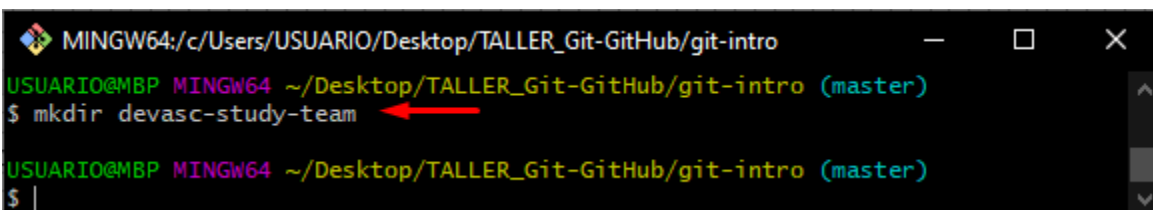


**PASO 4: CREAR UN NUEVO DIRECTORIO LOCAL: devasc-study-team.**

A. Si aún no está en el directorio **git-intro**, cámbielo ahora.

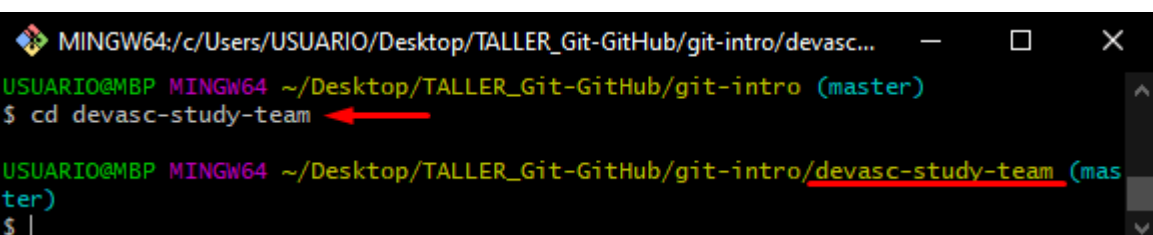


B. Crear un nuevo directorio llamado devasc-study-team. El directorio no tiene que coincidir con el nombre como repositorio.



**PASO 5: CAMBIE EL DIRECTORIO A: devasc-study-team.**

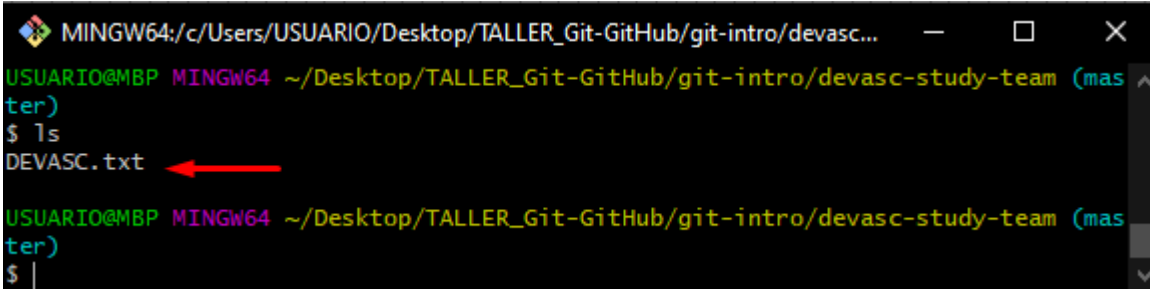
Utilice el comando **cd** para cambiar los directorios a **devasc-study-team**.



## PASO 6: COPIE EL ARCHIVO DEVASC.txt.

- A. Utilice el comando **cp** para copiar el **DEVASC.txt** del directorio principal **git-intro** al subdirectorio **devasc-study-team**. Los **dos puntos y una barra diagonal** anterior al nombre del archivo indican el **directorio principal**. El **espacio y el punto que siguen al nombre del archivo** indica que se debe copiar el archivo en el **directorio actual** con el mismo nombre de archivo.

`cp../DEVASC.txt`



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro/devasc...
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
$ ls
DEVASC.txt
```

- B. Verifique que el archivo se haya copiado con el comando **ls** y el contenido del archivo con el comando **cat**.

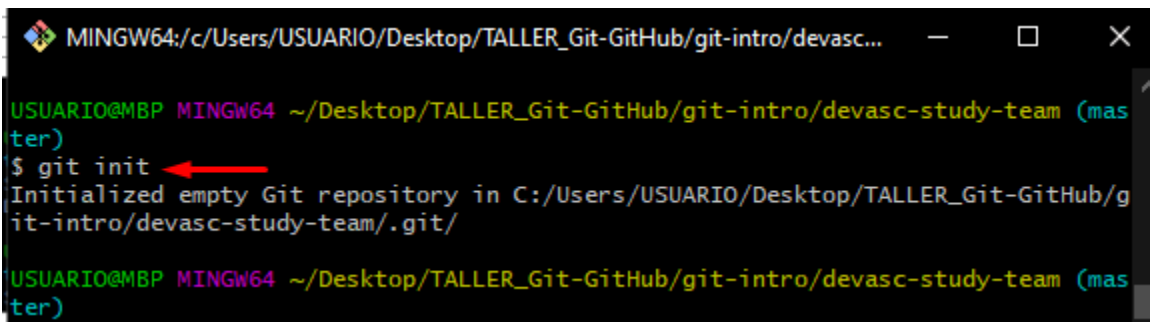


```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro/devasc...
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
$ cat DEVASC.txt
Estoy en camino para pasar el examen DEVASC de DevNet
Estoy empezando a entender Git!
Este texto se agregó originalmente en la rama de feature

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
```

## PASO 7: INICIALIZAR UN NUEVO REPOSITORIO DE Git.

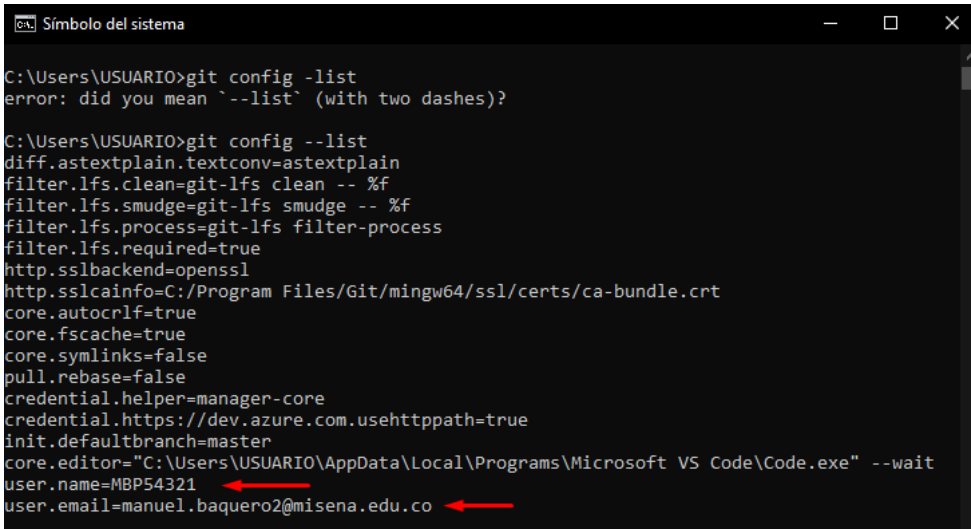
- A. Utilice el comando **git init** para inicializar el directorio actual (**devasc-study-team**) como **repositorio de Git**. El mensaje que se muestra indica que ha creado un repositorio local dentro del proyecto contenido en el directorio oculto **.git**. Aquí es donde se encuentra todo el historial de cambios.



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro/devasc...
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
$ git init
Initialized empty Git repository in C:/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team/.git/

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
```

- B. A continuación, compruebe sus variables globales de git con el comando **git config --list**. Verificar en consola **CMD**



```
C:\Users\USUARIO>git config -list
error: did you mean '--list' (with two dashes)?

C:\Users\USUARIO>git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
core.editor="C:\Users\USUARIO\AppData\Local\Programs\Microsoft VS Code\Code.exe" --wait
user.name=MBP54321
user.email=manuel.baquero2@misena.edu.co
```

- C. Si las variables **user.name** y **user.email** no coinciden con sus credenciales de **GitHub**, cámbielas ahora.

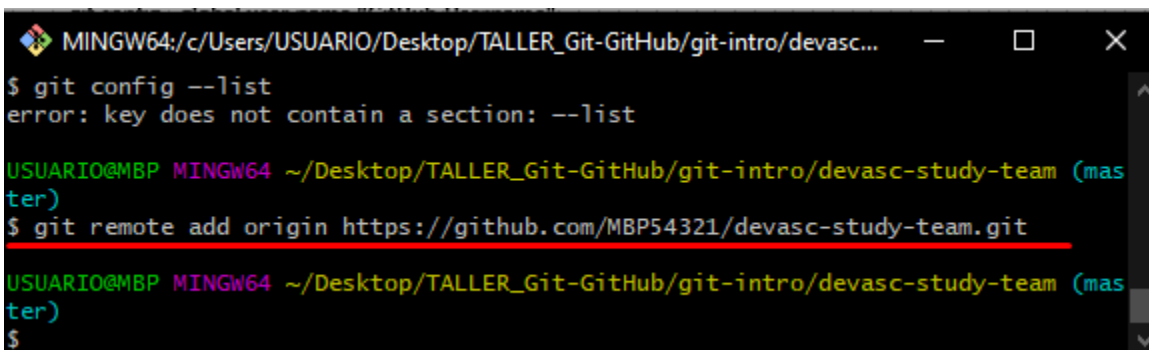
**git config --global user.name "GitHub-Username"**

**git config --global user.email "GitHub-email-address"**

#### PASO 8: APUNTE EL REPOSITORIO DE Git AL REPOSITORIO DE GitHub.

- A. Utilice el comando **git remote add** para agregar una **URL de Git** como un alias remoto. El valor "**origen**" apunta al repositorio recién creado en **GitHub**. Utilice su nombre de usuario de **GitHub** en la ruta **URL** para **github-username**.  
**Nota:** Su nombre de usuario distingue entre mayúsculas y minúsculas

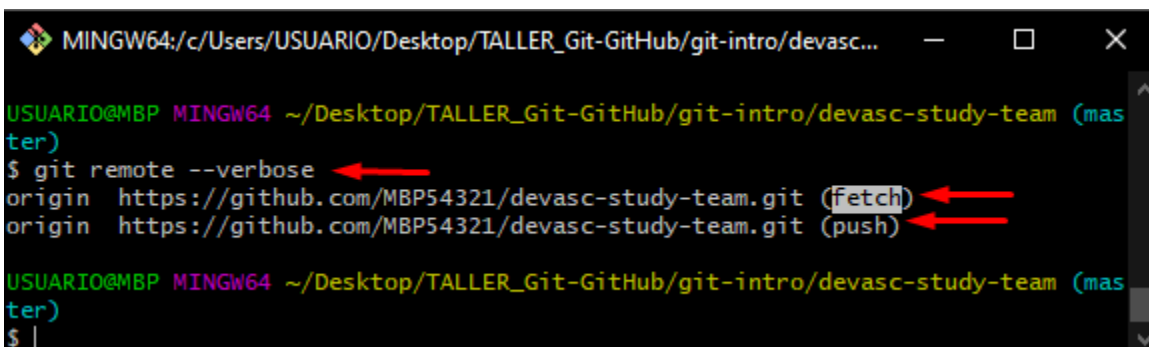
**git remote add origin https://github.com/MBP54321/devasc-study-team.git**



```
$ git config --list
error: key does not contain a section: --list

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
$ git remote add origin https://github.com/MBP54321/devasc-study-team.git
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
$
```

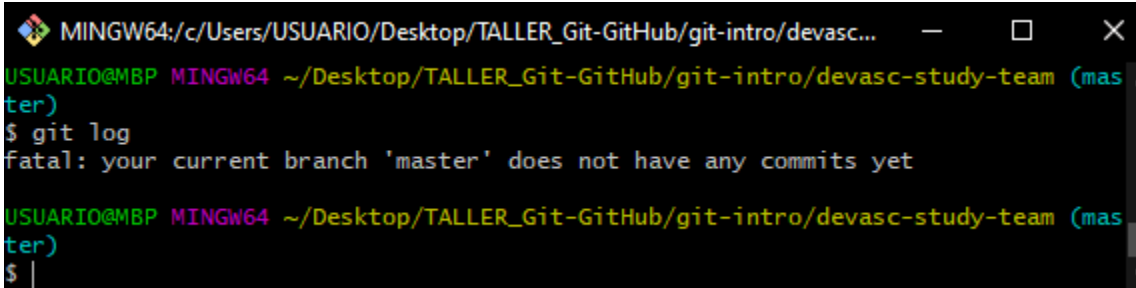
- B. Verifique que el remote se esté ejecutando en **github.com**: **git remote --verbose**



```
$ git remote --verbose
origin https://github.com/MBP54321/devasc-study-team.git (fetch)
origin https://github.com/MBP54321/devasc-study-team.git (push)

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
$ |
```

- C. Ver el registro de **git**. El error indica que no hay confirmaciones: **git log**

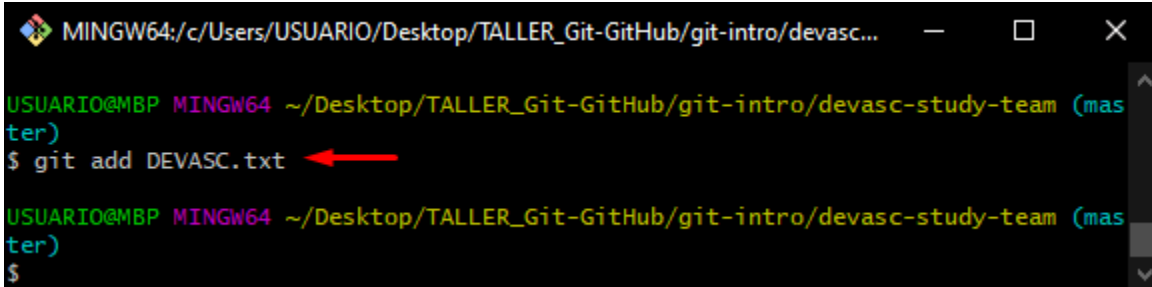


```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro/devasc...
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
$ git log
fatal: your current branch 'master' does not have any commits yet

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
$
```

## PASO 9: PONGA EN ESCENA Y CONFIRME EL ARCHIVO DEVASC.txt.

- A. Utilice el comando **git add** para poner en escena el archivo **DEVASC.txt**



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro/devasc...
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
$ git add DEVASC.txt
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
$
```

- B. Utilice el comando **git commit** para confirmar el archivo **DEVASC.txt**.

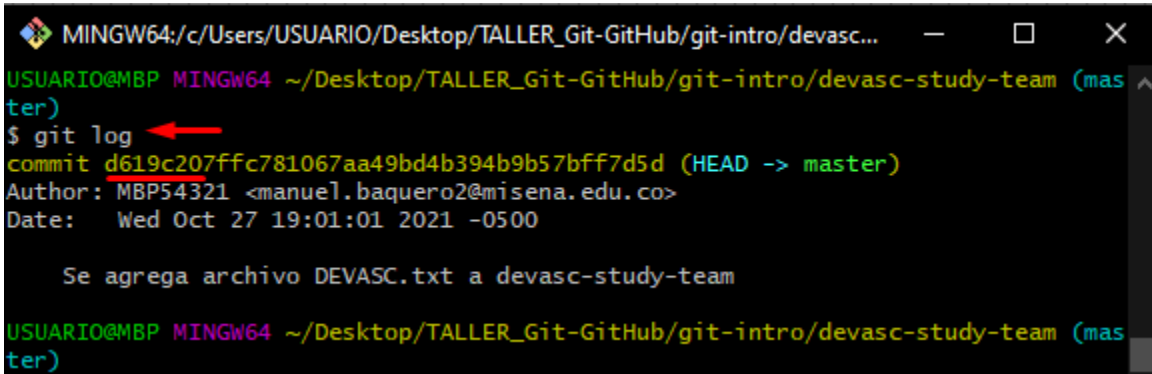
**git commit -m "Se agrega archivo DEVASC.txt a devasc-study-team"**



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro/devasc...
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
$ git commit -m "Se agrega archivo DEVASC.txt a devasc-study-team"
[master (root-commit) d619c20] Se agrega archivo DEVASC.txt a devasc-study-team
1 file changed, 3 insertions(+)
create mode 100644 DEVASC.txt
```

## PASO 10: VERIFICAR LA CONFIRMACIÓN.

- A. Usar el comando **git log** para mirar las confirmaciones.



```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro/devasc...
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
$ git log
commit d619c207ffc781067aa49bd4b394b9b57bff7d5d (HEAD -> master)
Author: MBP54321 <manuel.baquero2@misena.edu.co>
Date: Wed Oct 27 19:01:01 2021 -0500

 Se agrega archivo DEVASC.txt a devasc-study-team

USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
```



- B. Utilice el comando **git status** para ver la información de estado. La frase **"working tree clean"** significa que **Git** ha comparado tu lista de archivos con lo que le has dicho a Git, y **es una pizarra limpia sin nada nuevo que informar**.

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro/devasc...
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
$ git status
On branch master
nothing to commit, working tree clean
```

#### PASO 11: ENVIAR (push) EL ARCHIVO DE Git A GitHub.

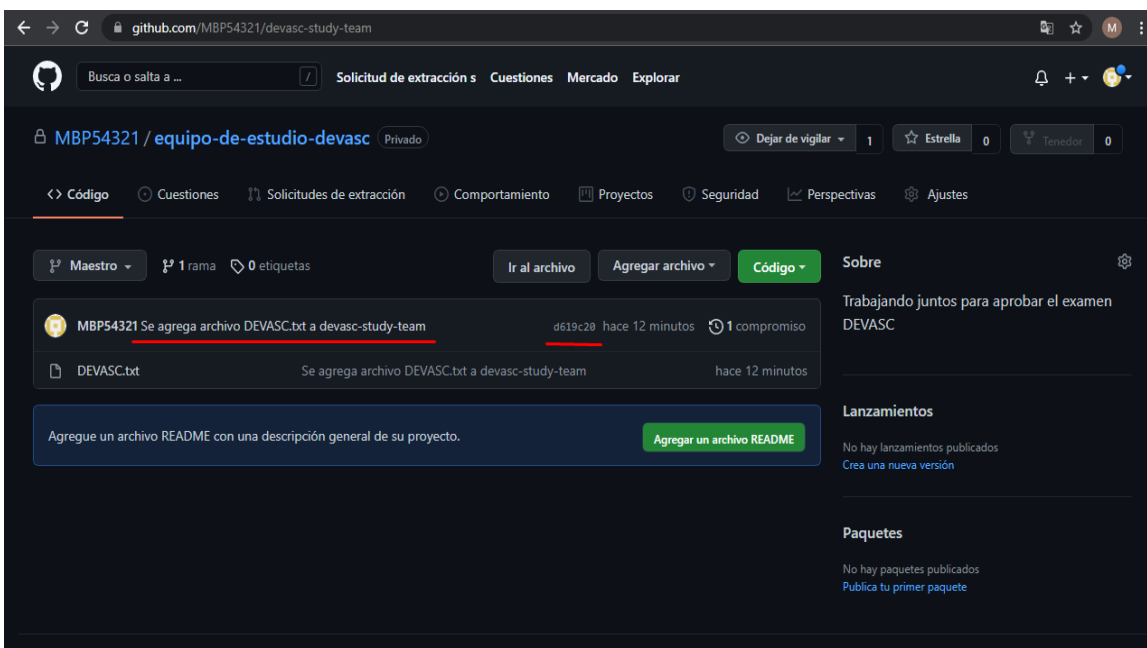
Utilice el comando **git push origin master** para enviar (**push**) el archivo a su repositorio de **GitHub**. Se le pedirá un nombre de usuario y una contraseña, que será la que utilizó para crear su cuenta de **GitHub**.

```
MINGW64:/c/Users/USUARIO/Desktop/TALLER_Git-GitHub/git-intro/devasc...
USUARIO@MBP MINGW64 ~/Desktop/TALLER_Git-GitHub/git-intro/devasc-study-team (master)
$ git push origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 6 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 364 bytes | 182.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/MBP54321/devasc-study-team.git
 * [new branch] master -> master
```

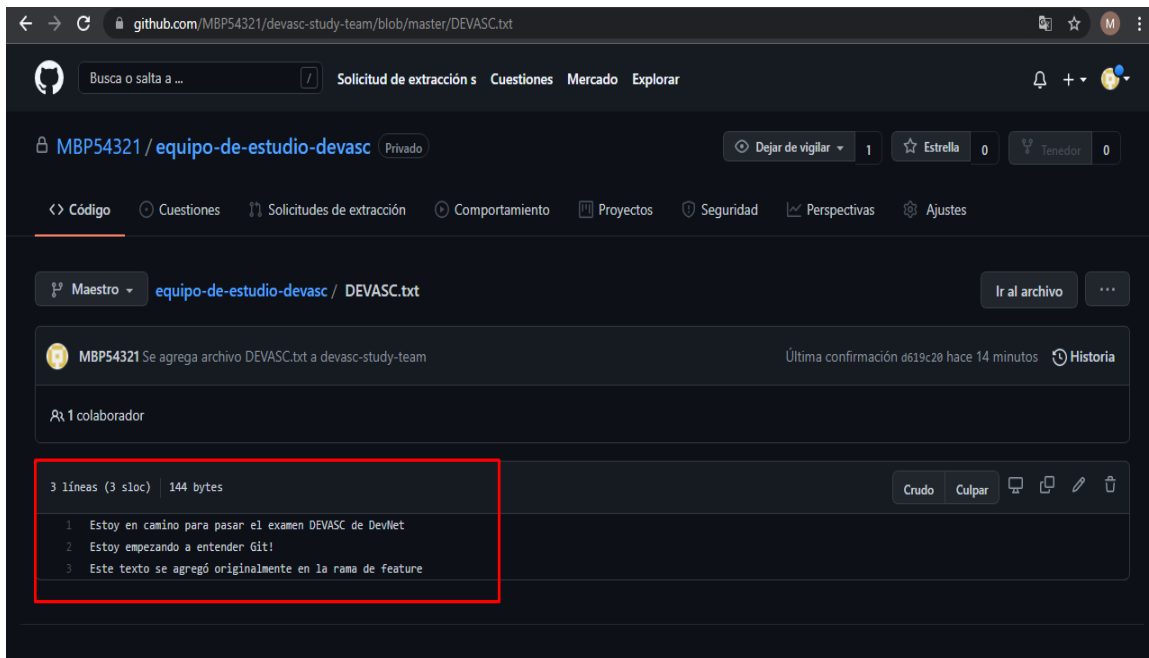
**Nota:** Si, después de introducir su nombre de usuario y contraseña, **obtiene un error fatal que indica que no se encuentra el repositorio, lo más probable es que haya enviado una URL incorrecta**. Deberá revertir su comando **git add** con el comando **git remote rm origin**.

#### PASO 12: VERIFIQUE EL ARCHIVO EN GitHub.

- A. Vaya a su cuenta de **GitHub** y en **"Repositorios"** seleccione nombre de usuario/**devasc-study-team**



- B. Debería ver que el archivo **DEVASC.txt** se ha agregado a este repositorio de **GitHub**. Haga clic en el archivo para ver el contenido.



The screenshot shows a GitHub repository page for 'MBP54321 / equipo-de-estudio-devasc'. The file 'DEVASC.txt' is selected, and its content is displayed in a dark-themed editor. The content consists of three lines of text, which are highlighted by a red rectangular box. The text describes the user's progress in preparing for the DEVASC exam and mentions the 'feature' branch.

3 líneas (3 sloc) | 144 bytes

```
1 Estoy en camino para pasar el examen DEVASC de Devliet
2 Estoy empezando a entender Git!
3 Este texto se agregó originalmente en la rama de feature
```

Buttons: Crudo, Culpas, and icons for file operations.