

MANUAL GIT

SENA

CEET

ADSI

FICHA 1566614

Contenido

Introducción:	3
Repositorio:	3
Commit:	3
Rama	3
Merge:	3

Introducción:

GIT es una herramienta que nos permite llevar un control de todos los cambios que realicemos en nuestros proyectos, mediante versiones, estos cambios se van almacenando de distintos archivos que el mismo GIT administra para que podamos acceder a estos de una manera mucho más eficaz. Además de llevar un control de nuestro proyecto GIT nos permite compartir dicho proyecto con todos los colaboradores y desarrolladores del mismo, logrando así un control efectivo de todas las versiones que se lleven a cabo.

Vamos a agregar unos cuantos términos importantes que se deberían saber antes de empezar esta guía sobre el control de versiones.

Repositorio: Este es el espacio donde se guardaran todos nuestros archivos y los nuevos que se irán generando a medida que creemos commits.

Commit: este es un punto en la línea del tiempo de nuestro proyecto, en donde queremos resaltar un cambio que consideremos importante, se puede crear tantos commits como se crean necesarios. Estos van junto con una pequeña descripción que explica que cambio se realizó.

Rama: Es una copia del código original, al trabajar en esta copia podemos trabajar en base a esta y decidir si la queremos implementar o no como una nueva versión.

Merge: Fusión de dos ramas para crear una sola que contenga los cambios hechos en ambas, esto puede generar conflictos.

Conflicto: cuando dos ramas se fusionan (merge) y existen cambios en las mismas líneas, en ambas ramas, se genera un conflicto. Es aquí donde un moderador debe intervenir para decidir cuál de los cambios se guardan y cual se descarta.

Instalación:

Se instala como un programa normal, descargamos el instalador desde la página oficial de Git

<https://git-scm.com/downloads>

Normalmente como en todo lo que realizamos le damos todo a siguiente sin importar lo que estemos aceptando y continuamos con la instalación (si eres alguien más experimentado puede ajustar las opciones conforme usted vea necesario)

Ya al final nos saldrá esta pantalla y podemos empezar a administrar nuestros proyectos y todos los cambios que puedan tener.

Completing the Git Setup Wizard

Setup has finished installing Git on your computer. The application may be launched by selecting the installed shortcuts.



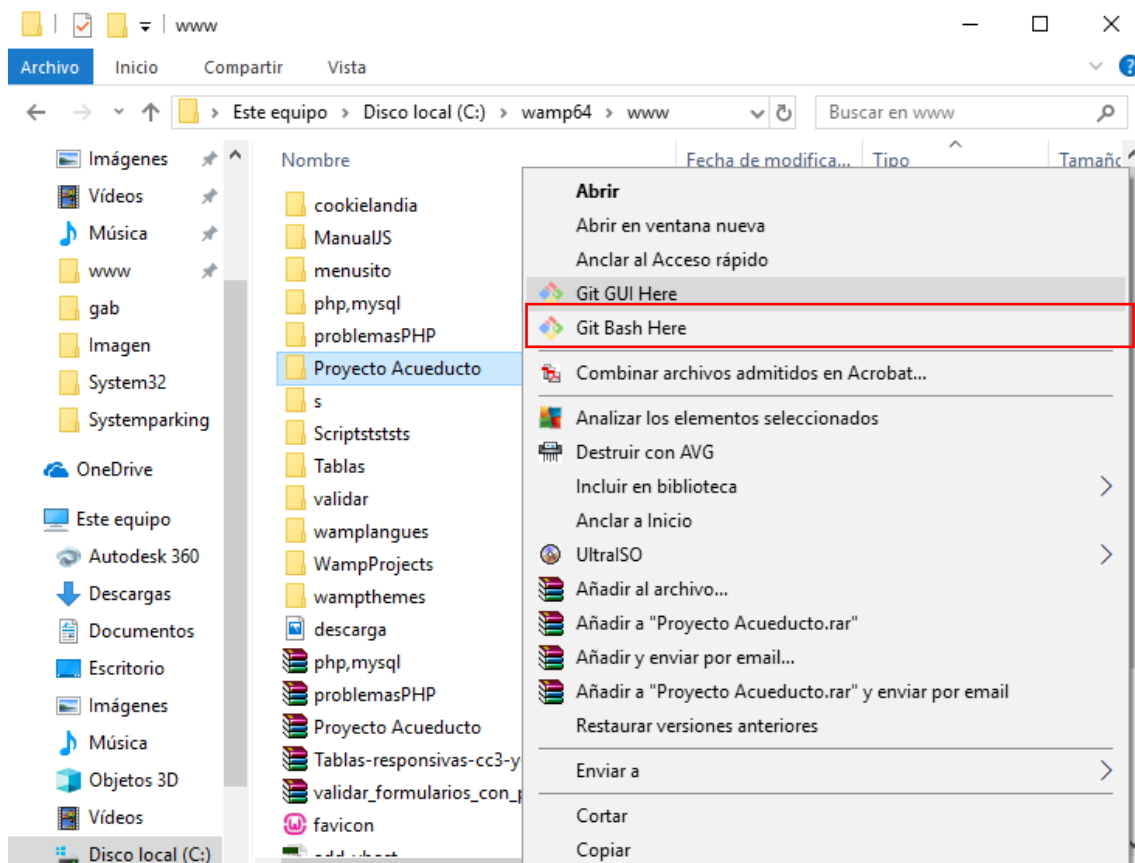
Click Finish to exit Setup.

- ☒ Launch Git Bash
- ☐ View Release Notes

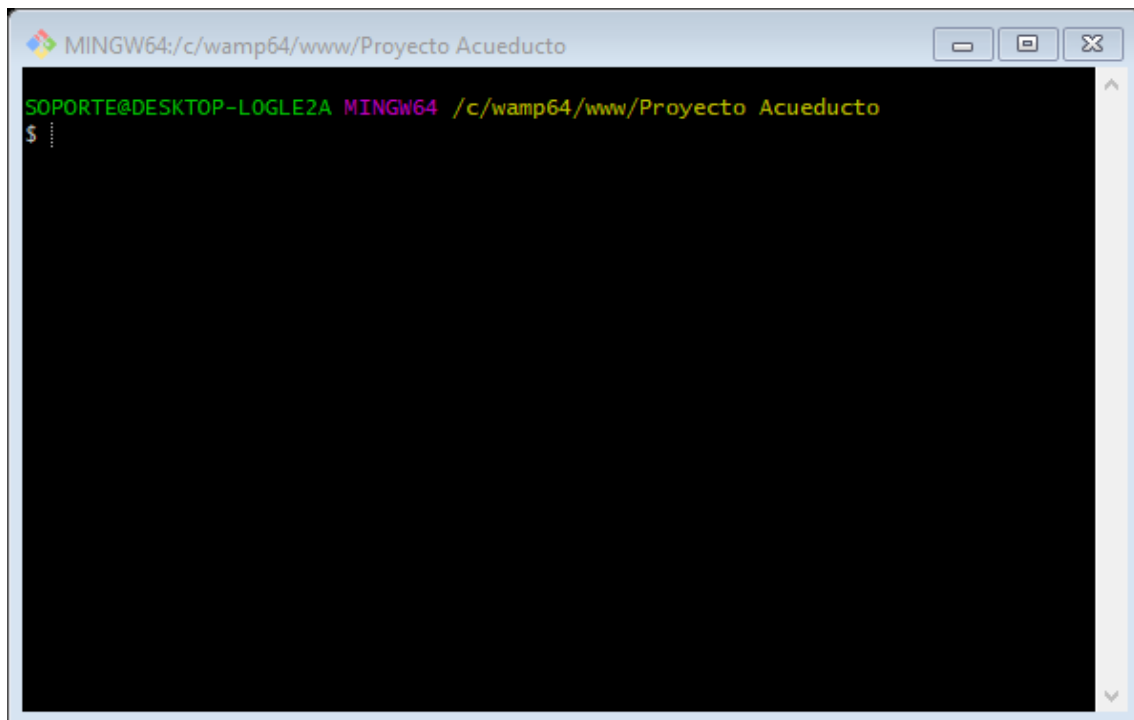
Finish

Configuración del GIT:

Primero debes saber como utilizar Git en un proyecto, primero irás a la carpeta donde esta almacenado todos tus archivos, le dará click derecho y dará en la opción que dice 'Git Bash here', así de esta manera establecemos donde queremos realizar el control.



Luego abrirá la consola donde inicia el Git.



```
MINGW64:/c/wamp64/www/Proyecto Acueducto
SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto
$
```

Para realizar la configuración de los datos para administrar pondremos los siguientes códigos

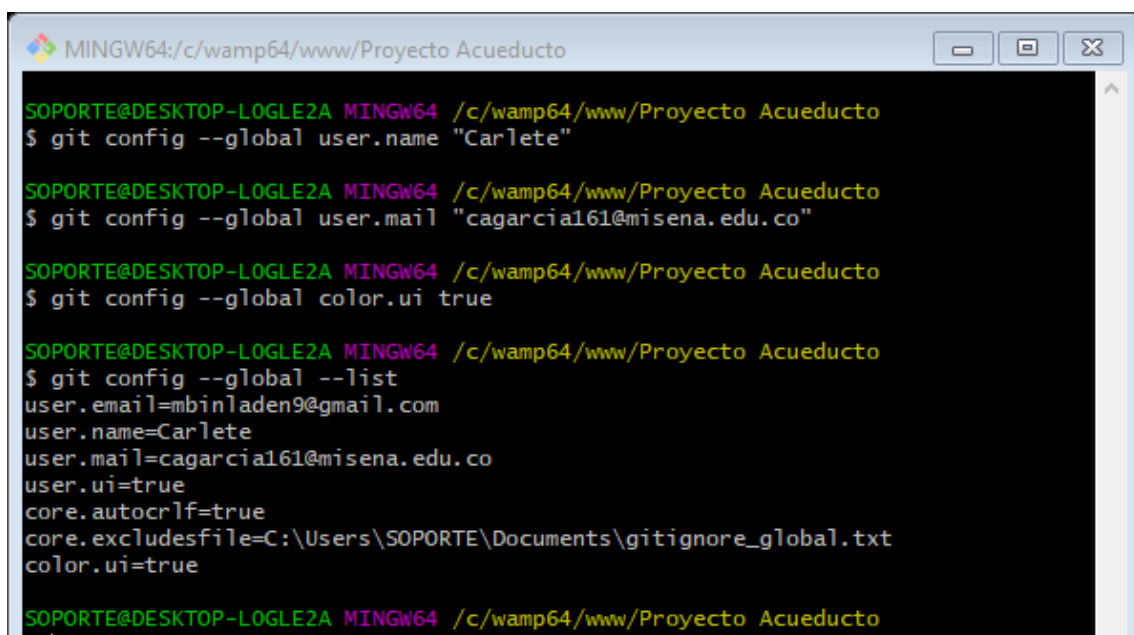
Git config --global user.name "El nombre que desee" (Para establecer el nombre del admin)

Cabe resaltar que no utilizamos un final de sentencia como lo es el punto y coma, solo debemos escribir la sentencia y dar enter.

Git config --global user.mail "Su e-mail" (Para establecer mail del admin)

Git config --global color.ui true (colorea todo lo impreso en pantalla)

Git config --global --list (Muestra todos los datos que fueron almacenados)



```
MINGW64:/c/wamp64/www/Proyecto Acueducto
SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto
$ git config --global user.name "Carlete"

SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto
$ git config --global user.mail "cagarcia161@misena.edu.co"

SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto
$ git config --global color.ui true

SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto
$ git config --global --list
user.email=mbinladen9@gmail.com
user.name=Carlete
user.mail=cagarcia161@misena.edu.co
user.ui=true
core.autocrlf=true
core.excludesfile=C:\Users\SOPORTE\Documents\gitignore_global.txt
color.ui=true

SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto
$
```

Existen más comandos para configurar nuestro Git, pero estos son los más esenciales para iniciar.

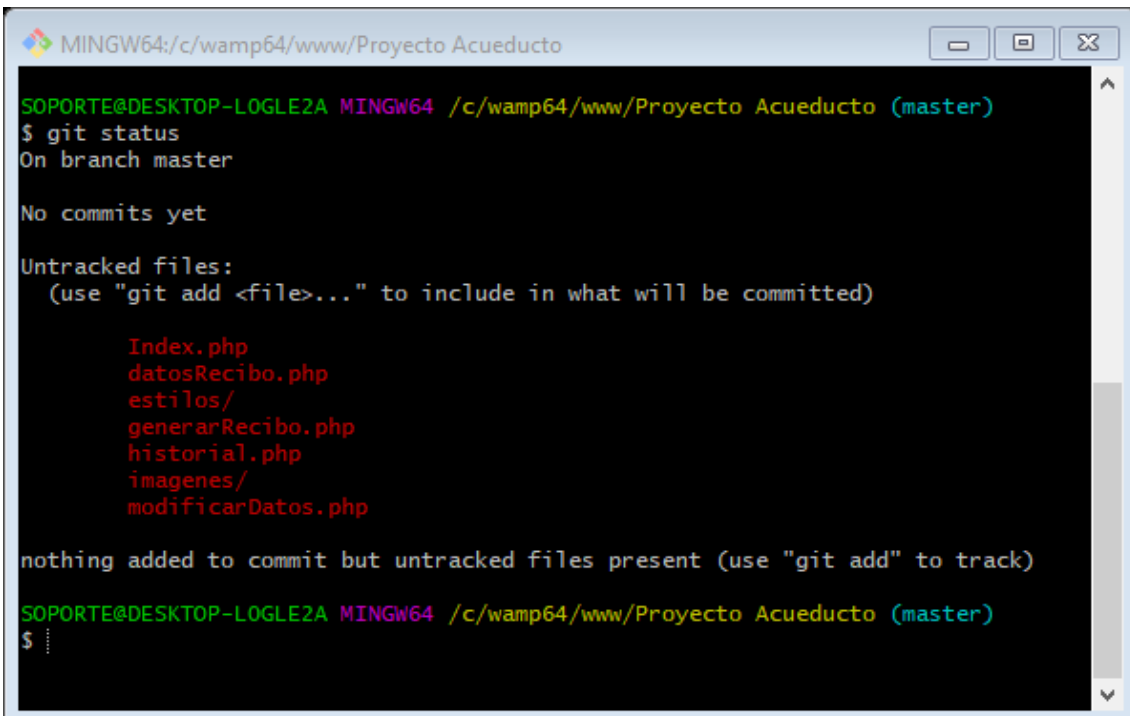
Una vez configurado nuestro Git por primera vez podemos proceder a iniciar nuestro primer proyecto utilizando el siguiente comando.

Git init

```
SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto
$ git init
Initialized empty Git repository in C:/wamp64/www/Proyecto Acueducto/.git/
```

Para mostrar todos los archivos que estén agregados a nuestro Git usamos la siguiente sentencia

Git status



```
MINGW64:/c/wamp64/www/Proyecto Acueducto
SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        Index.php
        datosRecibo.php
        estilos/
        generarRecibo.php
        historial.php
        imagenes/
        modificarDatos.php

nothing added to commit but untracked files present (use "git add" to track)
SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto (master)
$ .....
```

Como podemos apreciar todo esta en un color rojo, lo que significa que aun no hemos realizado la carga de los datos, para cargar todos los datos al Git, utilizamos la siguiente sentencia

Git add -A

Luego volvemos a escribir la sentencia para revisar nuestros archivos

Git status

```
MINGW64:/c:/wamp64/www/Proyecto Acueducto
SOPORTE@DESKTOP-LOGLE2A MINGW64 /c:/wamp64/www/Proyecto Acueducto (master)
$ git add -A

SOPORTE@DESKTOP-LOGLE2A MINGW64 /c:/wamp64/www/Proyecto Acueducto (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   Index.php
        new file:   datosRecibo.php
        new file:   estilos/datosActual.css
        new file:   estilos/esModificarDatos.css
        new file:   estilos/estilo.css
        new file:   estilos/generarest.css
        new file:   estilos/historialEst.css
        new file:   estilos/modificarDatos.css
        new file:   generarRecibo.php
        new file:   historial.php
        new file:   imagenes/fondo.jpg
        new file:   imagenes/logo_acueducto.png
        new file:   modificarDatos.php

SOPORTE@DESKTOP-LOGLE2A MINGW64 /c:/wamp64/www/Proyecto Acueducto (master)
$
```

Como vemos todos los archivos de nuestro proyecto ya están montados en nuestro Git y ya podemos empezar a trabajar.

Nota: Siempre se deberá documentar los progresos con un mensaje para identificar más fácilmente los cambios realizados, esto se realiza con la siguiente sentencia

Git commit -m "el mensaje que desee"

```
MINGW64:/c:/wamp64/www/Proyecto Acueducto
new file:   estilos/historialEst.css
new file:   estilos/modificarDatos.css
new file:   generarRecibo.php
new file:   historial.php
new file:   imagenes/fondo.jpg
new file:   imagenes/logo_acueducto.png
new file:   modificarDatos.php

SOPORTE@DESKTOP-LOGLE2A MINGW64 /c:/wamp64/www/Proyecto Acueducto (master)
$ git commit -m "Se comenzo esta mierda"
[master (root-commit) 46b6ff9] Se comenzo esta mierda
13 files changed, 617 insertions(+)
create mode 100644 Index.php
create mode 100644 datosRecibo.php
create mode 100644 estilos/datosActual.css
create mode 100644 estilos/esModificarDatos.css
create mode 100644 estilos/estilo.css
create mode 100644 estilos/generarest.css
create mode 100644 estilos/historialEst.css
create mode 100644 estilos/modificarDatos.css
create mode 100644 generarRecibo.php
create mode 100644 historial.php
create mode 100644 imagenes/fondo.jpg
create mode 100644 imagenes/logo_acueducto.png
create mode 100644 modificarDatos.php

SOPORTE@DESKTOP-LOGLE2A MINGW64 /c:/wamp64/www/Proyecto Acueducto (master)
$
```

Ramas: Como explicamos anteriormente una rama es una copia de todos los archivos de nuestro proyecto, esta copia la someteremos a cambios y modificaciones del código. Si el resultado final lo deseamos añadir al proyecto real realizaremos una fusión.

Para crear una rama se utiliza el comando

Git branch ejemploDeRama

```
SOPORTE@DESKTOP-LOGLE2A MINGW64 /c:/wamp64/www/Proyecto Acueducto (master)
$ git branch ejemploDeRama
```

Para ver las ramas que tiene nuestro proyecto usamos el comando

Git branch

```
SOPORTE@DESKTOP-LOGLE2A MINGW64 /c:/wamp64/www/Proyecto Acueducto (master)
$ git branch
ejemploDeRama
* master
```

Nota: ya que hemos activado el color.ui nos indicara de un color diferente la rama en la que nos encontramos ubicados actualmente, esto también se ve en la parte superior al final de la dirección donde se encuentra nuestro proyecto

Para cambiar de rama usamos el comando

Git checkout ejemploDeRama


```

SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto (master)
$ git checkout ejemploDeRama
Switched to branch 'ejemploDeRama'

SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto (ejemploDeRama)
$ git branch
* ejemploDeRama
  master

```

Verificamos con el comando

Git branch

Trabajamos sobre nuestro proyecto y guardamos todo el progreso que llevemos.

Una vez realizados todos los cambios usamos el comando

Git status

Y nos muestra los documentos que han sido alterados

```

SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto (ejemploDeRama)
$ git status
On branch ejemploDeRama
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   Index.php

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

```

Como vemos nos muestra que el documento index.php ha sido modificado.

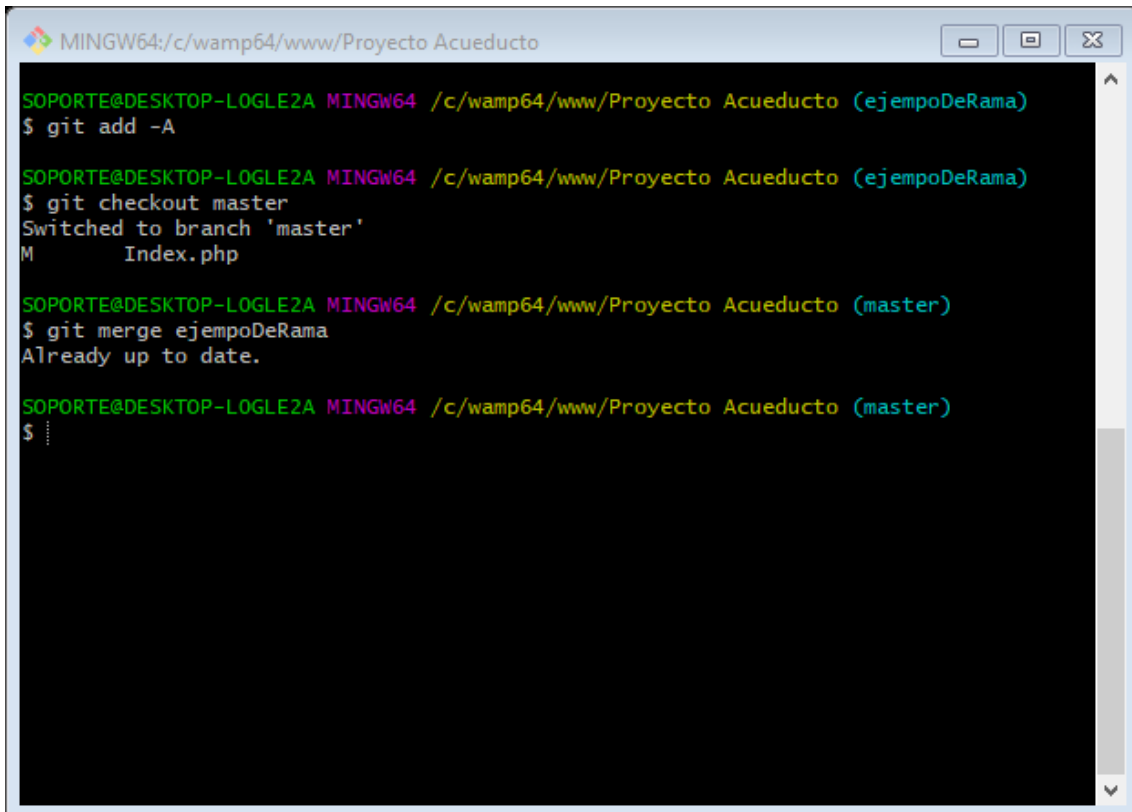
Para agregar los cambios utilizamos el comando que ya vimos anteriormente

Git add -A

Ahora, para combinar los cambios que hicimos en la rama ejemploDeRama a master, primero nos situamos en la rama master y escribimos la siguiente sentencia

Git merge ejemploDeRama

Nota: No olvidar realizar los comentarios necesarios con la sentencia del commit, así de esta manera no habrá ningún conflicto o confusión en las versiones que corran.



A screenshot of a Windows command prompt window. The title bar shows the path "MINGW64:/c/wamp64/www/Proyecto Acueducto". The window contains the following text:

```
SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto (ejemploDeRama)
$ git add -A

SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto (ejemploDeRama)
$ git checkout master
Switched to branch 'master'
M       Index.php

SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto (master)
$ git merge ejemploDeRama
Already up to date.

SOPORTE@DESKTOP-LOGLE2A MINGW64 /c/wamp64/www/Proyecto Acueducto (master)
$ .....
```

The window has a standard Windows interface with minimize, maximize, and close buttons in the top right corner. A vertical scrollbar is visible on the right side of the command prompt area.