**Creamos directorio**

mkdir prueba1

**Cambiamos al directorio creado**

cd prueba1/

**Iniciamos un repo local**

git init

**Comprobamos estado repo local**

git **status**

**Creamos archivo hola mundo en el directorio local/no en el repo**

vim "holamundo.txt"

**dentro de vim ( es un editor de texto seguir los siguientes pasos: )**  
apretar tecla **“i”** para insertar texto  
escribir **"hola mundo"**  
apretar tecla escape <ESC> para salir de modo edición  
escribir **“:”** para escribir comando  
escribir **“wq”** ( que es write and quit, para guardar y salir)  
**agregamos el archivo al repo local —>> tracked**

git add holamundo.txt

git **status**

**hacemos nuestro primer commit, el archivo ahora esta guardado en el repo local**

git commit

se abre editor de texto  
**dentro de vim ( es un editor de texto seguir los siguientes pasos: )**  
apretar tecla **“i”** para insertar texto  
escribir **"nuestro primer commit hola mundo"**  
apretar tecla escape <ESC> para salir de modo edición  
escribir **“:”** para escribir comando  
escribir **“wq”** ( que es write and quit, para guardar y salir)  
**Vemos el registro de nuestros commits**

git log

**Vemos hacia donde apunta el HEAD (la última versión)**

git **show**

**listamos las ramas creadas, no hay ninguna todavía**

git **branch**

**Creamos la rama prueba a partir de la rama master**

git **branch** prueba

**Vemos nuestra nueva rama en el listado de ramas**

git **branch**

**Cambiamos a la rama prueba para trabajar sobre ella**

git checkout prueba

**Comprobamos que efectivamente hemos cambiado de rama (prueba)**

git **status**

**Vemos a donde apunta la HEAD (ultima versión)**

git **show**

**ls**

vim "holamundo.txt"

**dentro de vim ( es un editor de texto seguir los siguientes pasos: )**  
apretar tecla **“i”** para insertar texto  
escribir **"hola mundo en branch prueba"**  
apretar tecla escape <ESC> para salir de modo edición  
escribir **“:”** para escribir comando  
escribir **“wq”** ( que es write and quit, para guardar y salir)

**Trackeamos los últimos cambios y hacemos commit a la vez en la branch prueba**

git commit -am "hola mundo para branch prueba"

**Cambiamos a rama master**

git checkout **master**

git **status**

**Vemos que en master estamos en la versión anterior**  
**porque los cambios los hemos hecho en “prueba”**

cat holamundo.txt

git log

**\*\***Mezclamos los cambios en la rama master incluyendo los cambios de la rama pruebas**\*\***

git merge prueba

**\*\***Cambiamos a rama prueba**\*\***

git checkout prueba

git merge **master**

**Master y prueba vuelven a estar actualizadas. HEAD apunta a las dos**

git **show**

vim "holamundo.txt"

**Vamos a crear un conflicto cambiando el mismo archivo en prueba y en master**  
**dentro de vim ( es un editor de texto seguir los siguientes pasos: )**  
apretar tecla “i” para insertar texto  
escribir " hola mundo en branch prueba con conflicto"  
apretar tecla escape <ESC> para salir de modo edición  
escribir “:” para escibir comando  
escribir “wq” ( que es write and quit, para guardar y salir)

git commit -am "prueba con conflicto"

git checkout **master**

vim holamundo.txt

**Hacemos el cambio en master para generar el conflicto**  
**dentro de vim ( es un editor de texto seguir los siguientes pasos: )**  
apretar tecla “i” para insertar texto  
escribir " hola mundo en branch prueba con cambio en master para ver conflicto resuelto"  
apretar tecla escape <ESC> para salir de modo edición  
escribir “:” para escibir comando  
escribir “wq” ( que es write and quit, para guardar y salir)

git commit -am "prueba de conflicto"

git merge prueba **\*\***aquí salta el conflicto**\*\***

vim holamundo.txt

<h1>**Resolvemos entrando en el archivo y seleccionando los cambios que queremos y borrando lo demás**  
**dentro de vim ( es un editor de texto seguir los siguientes pasos: )**  
apretar tecla **“i”** para insertar texto  
borrar la linea con conflicto y dejar la linea con conflicto resuelto.  
borrar las lineas <<<<< y =====  
<<<<<<< **HEAD**  
este es un archivo de hola mundo para branch prueba conflicto resuelto</h1>

este es un archivo de hola mundo para branch prueba con conflicto

***prueba*** *apretar tecla escape <ESC> para salir de modo edición  
escribir****“:”****para escibir comando  
escribir****“wq”****( que es write and quit, para guardar y salir)*

git commit -am "archivo corregido" **\*\***hacemos commit de archivo corregido**\*\***

git **status**

git **merge** prueba

cat holamundo.txt

**Hasta aquí esta resuelto el conflicto. Master tiene la versión que queríamos**  
**Prueba tiene aún su propia versión. Si queremos dejarlo todo igual debemos hacer**  
**un merge desde master a prueba**  
**Vamonos a prueba para hacerlo**

git checkout prueba

cat holamundo.txt

**Traigamos el merge hacia prueba**

git merge **master**

**master y prueba vuelven a estar actualizadas y HEAD apunta a las dos**

cat holamundo.txt

git **show**

**Final**