**Instalación y uso de git**

En el presente documento se detalla como instalar y versionar proyectos usando git.

Instalar GIT

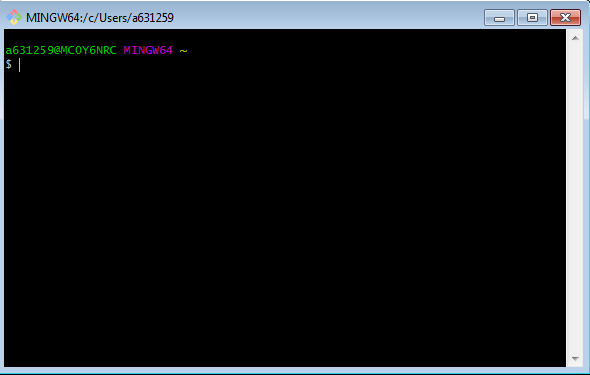
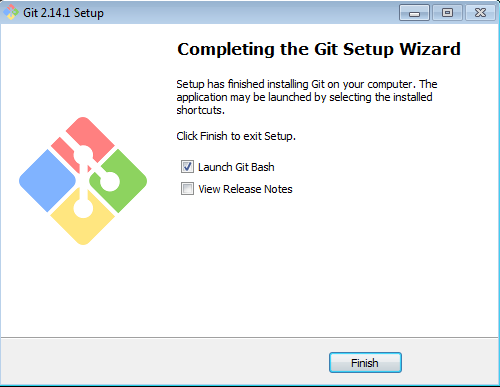
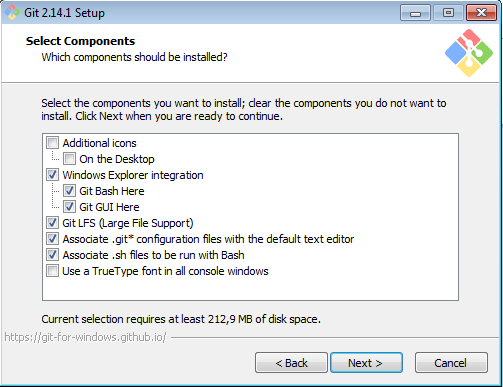
<https://git-scm.com/downloads>

<https://git-scm.com/download/win>

Ejecutar el instalador como administrador

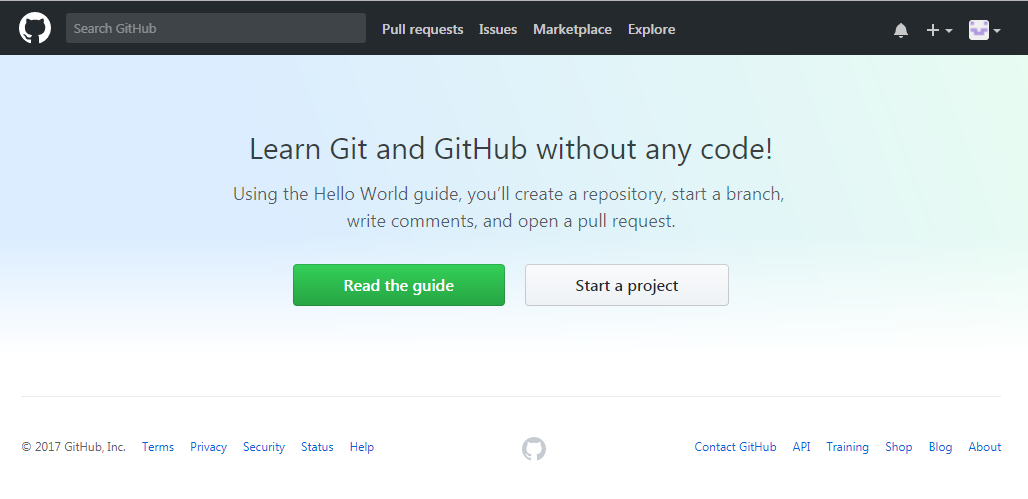
Git-2.14.1-64-bit.exe

Dejar las valores por defecto



Crear cuenta en github

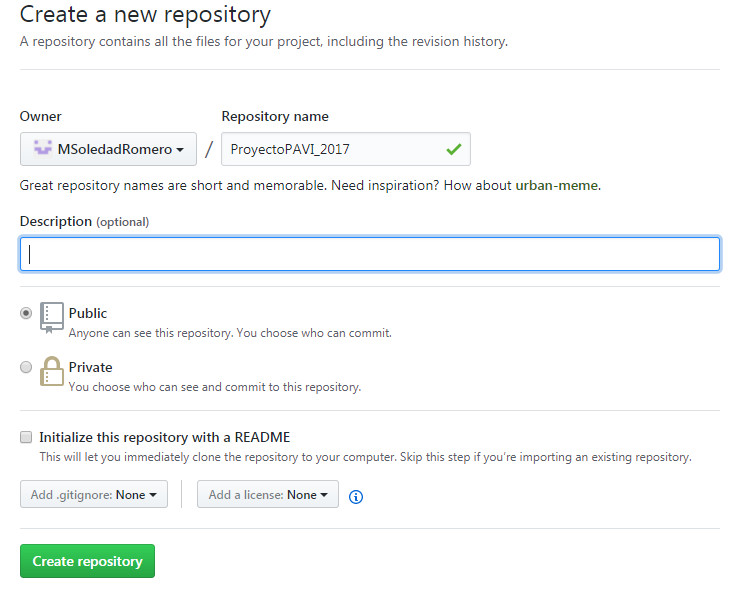
Loguearse



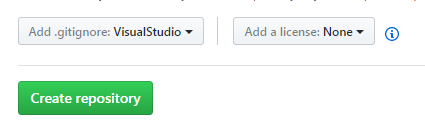
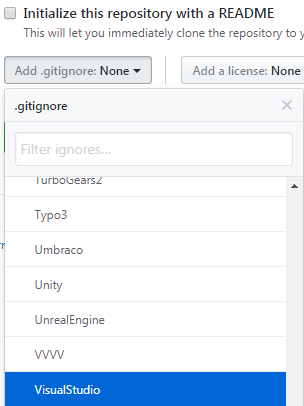
Seleccionar: Start a Project / Comenzar un proyecto

Crear un repositorio

ProyectoPAVI\_2017

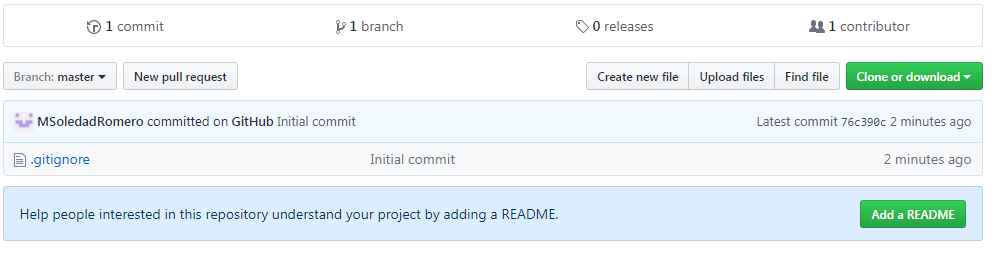


Seleccionar Add.gitignore (se incluyen los archivos que no queremos compartir, los ejecutables, etc.)



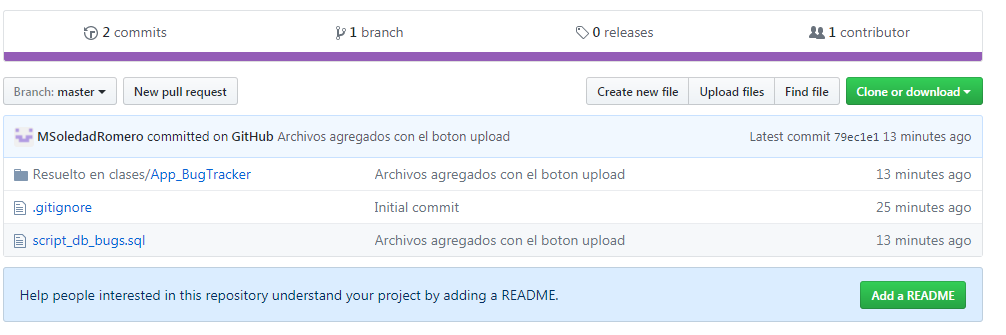
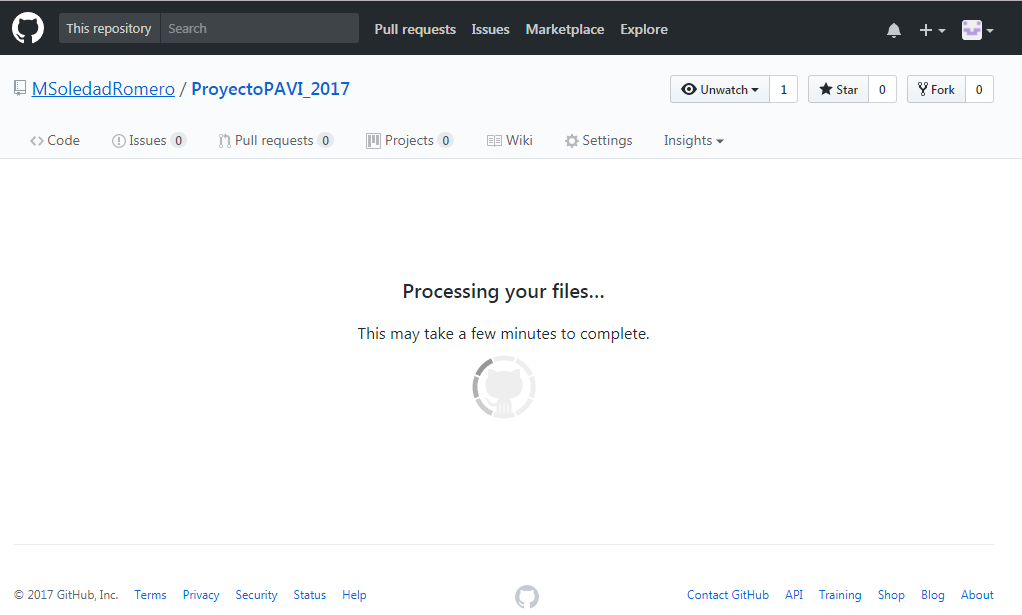
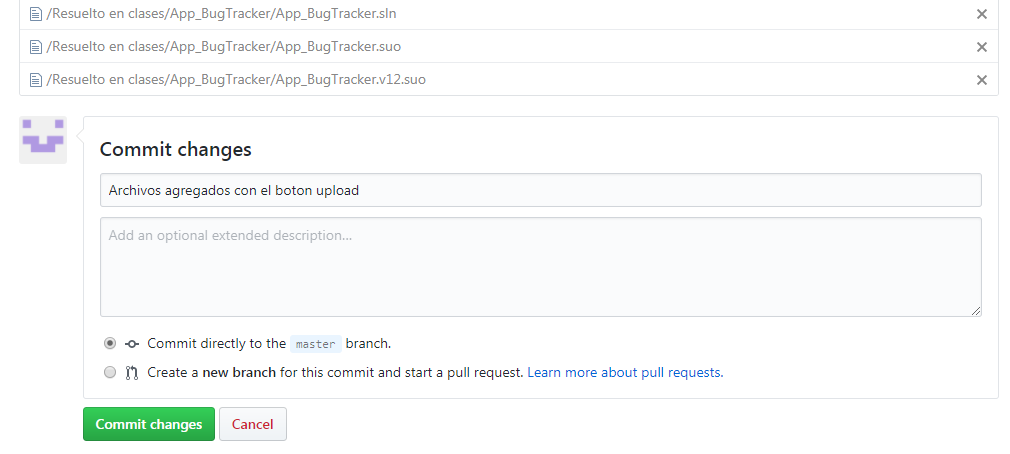
Presionar “Create Repository”

Seleccionar “Upload file” para subir por primera vez el proyecto a git



Arrastrar los archivos:

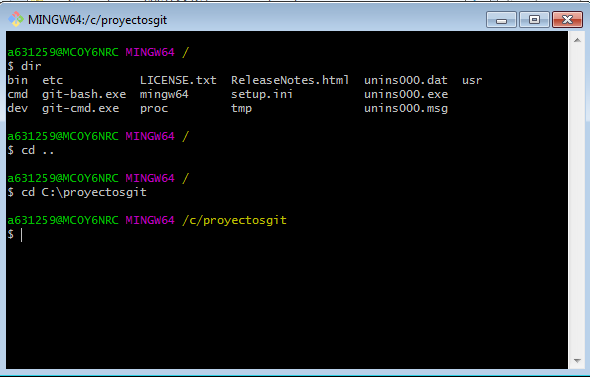
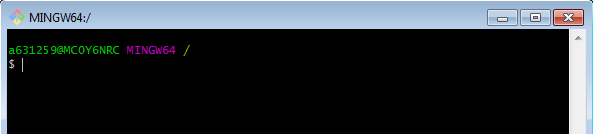
Ingresar el mensaje para el commit “Archivos agregados con el boton upload”



Nota: la subida de archivos también puede hacerse mediante el command git add seguido del comando git commit.

GIT en el equipo local

Ir a Inicio - Git Bash

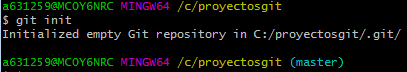


Ejecutar los comandos dir y cd para ver el directorio actual y moverse a un directorio particular.

$cd (para ir a la ubicación de una carpeta particular

Inicializar el repo local

Comando git init, por defecto la rama en local es master.

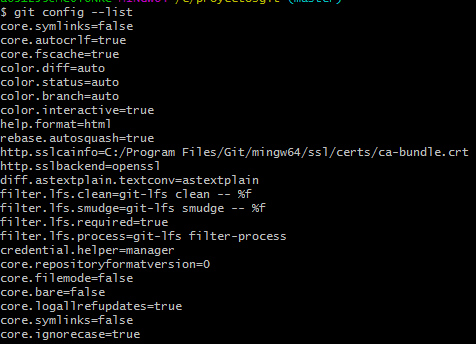


$git init

Para inicializar el repositorio

$git config --list

Configuración de todos los repositorios a nivel local.



$git config --global user.name

git config --global user.name MSoledadRomero

Permite configurar la cuenta creada en github.com

$git config --global email.name

Permite configurar el correo asociado a la cuenta github

|  |
| --- |
| a631259@MC0Y6NRC MINGW64 /c/proyectosgit (master)  $ git config --global user.name MSoledadRomero |

|  |
| --- |
| a631259@MC0Y6NRC MINGW64 /c/proyectosgit (master)  $ git config --global user.email romeroma.soledad@gmail.com |

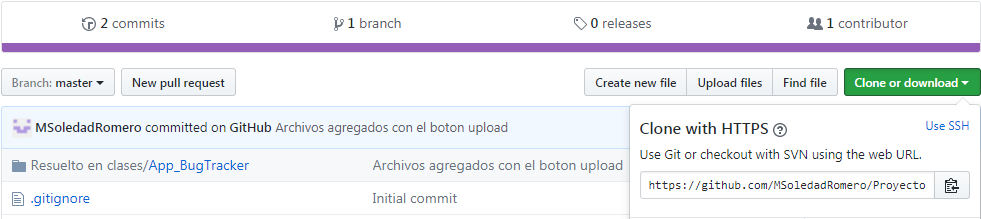
|  |
| --- |
| a631259@MC0Y6NRC MINGW64 /c/proyectosgit (master)  $ git config --list  core.symlinks=false  core.autocrlf=true  core.fscache=true  color.diff=auto  color.status=auto  color.branch=auto  color.interactive=true  help.format=html  rebase.autosquash=true  http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt  http.sslbackend=openssl  diff.astextplain.textconv=astextplain  filter.lfs.clean=git-lfs clean -- %f  filter.lfs.smudge=git-lfs smudge -- %f  filter.lfs.required=true  filter.lfs.process=git-lfs filter-process  credential.helper=manager  user.name=MSoledadRomero  user.email=romeroma.soledad@gmail.com  core.repositoryformatversion=0  core.filemode=false  core.bare=false  core.logallrefupdates=true  core.symlinks=false  core.ignorecase=true |

$git help: para acceder a la ayuda

$git clone

Para clonar el repo

Obtener la dirección del repo:



|  |
| --- |
| a631259@MC0Y6NRC MINGW64 /c/proyectosgit (master)  $ git clone https://github.com/MSoledadRomero/ProyectoPAVI\_2017.git  Cloning into 'ProyectoPAVI\_2017'...  remote: Counting objects: 88, done.  remote: Total 88 (delta 0), reused 0 (delta 0), pack-reused 88  Unpacking objects: 100% (88/88), done. |

$ git pull

Para traerse a local los cambios

“Already up-to-date” Significa que la rama master en el repo y la rama master en local están sincronizadas.

|  |
| --- |
| a631259@MC0Y6NRC MINGW64 /c/proyectosgit/ProyectoPAVI\_2017 (master)  $ git pull  Already up-to-date |

cd "C:\proyectosgit\ProyectoPAVI\_2017\Resuelto en clases\App\_BugTracker\App\_BugTracker\DataLayer\Helper"

$ git status

Por defecto en rojo los archivos modificados sin comitear, en verde los comiteados.

|  |
| --- |
| a631259@MC0Y6NRC MINGW64 /c/proyectosgit (master)  $ git status  On branch master  No commits yet  Untracked files:  (use "git add <file>..." to include in what will be committed)  ProyectoPAVI\_2017/  nothing added to commit but untracked files present (use "git add" to track) |

Realizo un cambio en un archivo BDHelper.vb en local.

|  |
| --- |
| a631259@MC0Y6NRC MINGW64 /c/proyectosgit/ProyectoPAVI\_2017 (master)  $ git status  On branch master  Your branch is up-to-date with 'origin/master'.  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: Resuelto en clases/App\_BugTracker/App\_BugTracker/DataLayer/Helper/BDHelper.vb  modified: Resuelto en clases/App\_BugTracker/App\_BugTracker/obj/x86/Release/App\_BugTracker.vbproj.FileListAbsolute.txt  modified: Resuelto en clases/App\_BugTracker/App\_BugTracker/obj/x86/Release/DesignTimeResolveAssemblyReferencesInput.cache  no changes added to commit (use "git add" and/or "git commit -a") |

a631259@MC0Y6NRC MINGW64 /c/proyectosgit/ProyectoPAVI\_2017 (master)

$ cd "C:\proyectosgit\ProyectoPAVI\_2017\Resuelto en clases\App\_BugTracker\App\_BugTracker\DataLayer\Helper"

$ git add

git add “archivo”

Agrega los archivos modificados.

|  |
| --- |
| a631259@MC0Y6NRC MINGW64 /c/proyectosgit/ProyectoPAVI\_2017/Resuelto en clases/App\_BugTracker/App\_BugTracker/DataLayer/Helper (master)  $ git add BDHelper.vb |

$git commit

Confirmación de los cambios, incluye un comentario.

|  |
| --- |
| a631259@MC0Y6NRC MINGW64 /c/proyectosgit/ProyectoPAVI\_2017/Resuelto en clases/App\_BugTracker/App\_BugTracker/DataLayer/Helper (master)  $ git commit -m "cambio en bdhelper"  [master 4e0163d] cambio en bdhelper  1 file changed, 1 insertion(+), 1 deletion(-) |

|  |
| --- |
| a631259@MC0Y6NRC MINGW64 /c/proyectosgit/ProyectoPAVI\_2017/Resuelto en clases/App\_BugTracker/App\_BugTracker/DataLayer/Helper (master)  $ git status  On branch master  Your branch is ahead of 'origin/master' by 1 commit.  (use "git push" to publish your local commits)  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: ../../../App\_BugTracker.v12.suo  modified: ../../obj/x86/Release/App\_BugTracker.vbproj.FileListAbsolute.txt  modified: ../../obj/x86/Release/DesignTimeResolveAssemblyReferencesInput.cache  no changes added to commit (use "git add" and/or "git commit -a") |

$ git push

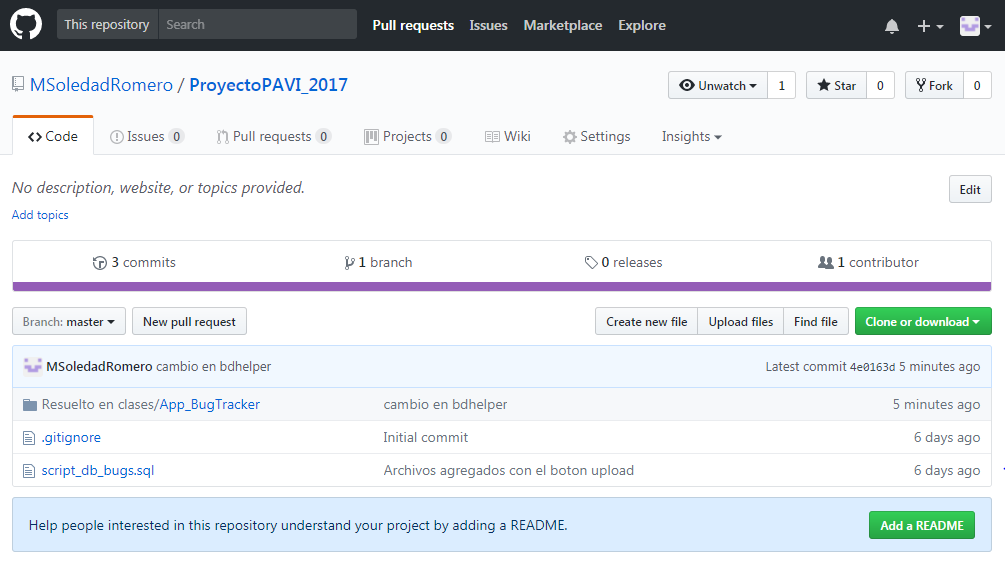
Para enviar los cambios confirmados al repo remoto

|  |
| --- |
| a631259@MC0Y6NRC MINGW64 /c/proyectosgit/ProyectoPAVI\_2017/Resuelto en clases/App\_BugTracker/App\_BugTracker/DataLayer/Helper (master)  $ git push |

Solicitará usuario y contraseña de github.

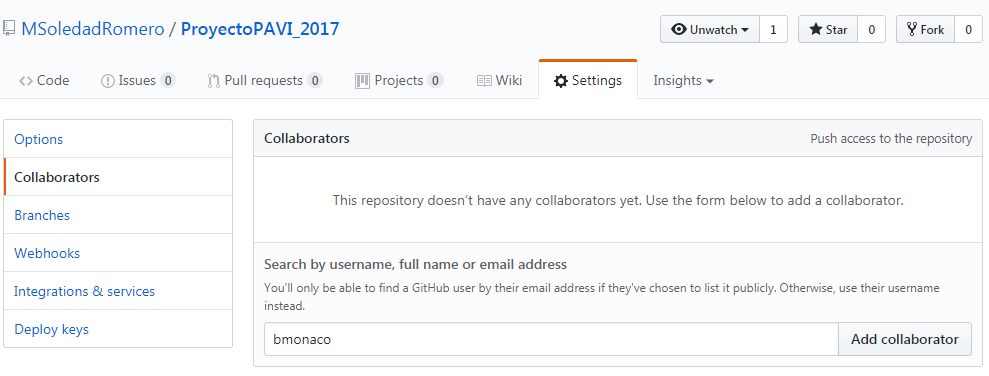


|  |
| --- |
| a631259@MC0Y6NRC MINGW64 /c/proyectosgit/ProyectoPAVI\_2017/Resuelto en clases/App\_BugTracker/App\_BugTracker/DataLayer/Helper (master)  $ git push  Counting objects: 8, done.  Delta compression using up to 4 threads.  Compressing objects: 100% (5/5), done.  Writing objects: 100% (8/8), 849 bytes | 849.00 KiB/s, done.  Total 8 (delta 2), reused 0 (delta 0)  remote: Resolving deltas: 100% (2/2), completed with 2 local objects.  To https://github.com/MSoledadRomero/ProyectoPAVI\_2017.git  79ec1e1..4e0163d master -> master |



Agregado de colaboradores

Seleccionar “Settings” “Collaborators”



Escribir el nombre de usuario github del compañero de grupo.

Presionar Add Collaborator.

Uso de Ramas

Una práctica recomendable es, una vez subido los cambios, crear una rama para trabajar (branch), hacer todos los cambios (add + commit), fusionar con master (checkout - merge) y subir los cambios ()

$ git branch

Crea una rama

git branch rama20170921

git status para ver los archivos modificados

git add "nombre del archivo"  para agregar

git add . para agregar todo un directorio

git commit -m "mensaje del commit" mensaje para los archivos agregados previamente.

$ git checkout

Cambia de rama.

git checkout master (cambia a la rama master)

git pull (se trae el último cambio)

git merge rama20170921

(Si no hay conflictos, git merge une los cambios de la rama nueva con la rama master)

Parados nuevamente en la rama master enviamos los cambios al repo remoto.

git push

Los conflictos son marcados por git en cada archivo de la siguiente manera:

<<<<<<< HEAD

Cambio en el primer repo

=======

Cambio en el segundo repo

>>>>>>> b29196692f5ebfd10d8a9ca1911c8b08127c85f8

Para resolverlo editar el archivo, resolver el conflicto dejando las líneas de código que corresponden y volver al circuito. (git add, git commit, git push)

Referencias

<http://blog.santiagobasulto.com.ar/programacion/2011/11/27/tutorial-de-git-en-espanol.html>

https://git-scm.com/do