**COMANDOS DE DESARROLLO CON GIT**

**Instalación y configuración en cliente y servidor**

[Manual](https://docs.google.com/file/d/0BzEEY_WNVMUYaXJPUXlrcUdzNXc/edit)

**Configuración local**

* git config --global user.name "Nombre del Desarrollador
  + Especifica el nombre del commiter que aparecerá en los commits
* git config --global user.email "email@dominio"
  + Especifica el email del commiter que aparecerá en los commits
* git config --global color.ui true
  + Activa los colores
* git config --system core.longpaths true
  + Habilita las rutas de hasta 4096 caracteres
* git config --global credential.helper wincred
  + Habilitar el guardado de credenciales en caché

**Clonar proyecto**

* git clone usuario@servidor:/ruta/a/repositorio/repo.git
  + Clona un proyecto
* git clone ssh://servidor/ruta/a/repositorio/repo.git NuevoNombreProjecto (se necesita tener agregadas las llaves públicas en el servidor)

**Estado de GIT**

* git status
  + Muestra información tal como: rama actual, cambios listos para commitir y cambios no versionados
* git --version
  + Muestra la versión de git instalada

**Preparar cambios**

* git add .
  + Agrega al área de preparación todos los cambios que hayamos hecho hasta ese momento para que estén listos para commitirse
* git add ruta/archivo/a/agregar/al/area/de/preparacion
  + Agrega al área de preparación el archivo especificado para que esté listo para commitirse

**Obtener y Enviar cambios de otro repositorio**

* git pull
  + Si estamos en la rama “master”, actualiza todo nuestras ramas locales con las existentes en el servidor
  + Si estamos en una rama configurada para hacer pull automáticamente, actualiza solo esa rama con la existente en el servidor
  + Internamente se realiza un merge. Si el merge falla, debido a que Git requiere nuestra decisión para fusionar, deberemos resolver conflictos, agregar los archivos al área de preparación y hacer nosotros un commit de los cambios
* git pull origin nombreRama
  + Actualiza nuestra rama actual con la rama especificada existente en el servidor. Internamente se realiza un merge. Si el merge falla, debido a que Git requiere nuestra decisión para fusionar, deberemos resolver conflictos, agregar los archivos al área de preparación y hacer nosotros un commit de los cambios
* git push
  + Si estamos en una rama configurada para hacer push automáticamente, actualiza la rama existente en el servidor con los cambios locales, siempre y cuando nuestra rama local sea más reciente que la del servidor
* git push origin nombreRama
  + Actualiza la rama especificada en el servidor con los cambios locales, siempre y cuando nuestra rama local sea más reciente que la del servidor

**Commit**

* git commit -a
  + Crea un commit de los cambios que están especificados en el área de preparación. Nos abre un editor de texto para que especifiquemos la descripción del commit
* git commit -m “descripción del commit”
  + Crea un commit de los cambios que están especificados en el área de preparación. Permite especificar entre comillas la descripción del commit
* [git reset --hard HEAD~1](http://www.juandefrias.com/posts/c-mo-borrar-el-ltimo-commit-en-git-11)
  + Borra un commit del que todavía no se ha hecho push
* [git reset](http://aprendegit.com/como-deshacer-el-ultimo-commit-en-git/)  [HEAD~1](http://www.juandefrias.com/posts/c-mo-borrar-el-ltimo-commit-en-git-11)
  + Borra un commit del que no se ha hecho push pero dejando los cambios en físico

[**Ramas**](http://www.genbetadev.com/herramientas/manejo-de-ramas-de-desarrollo-con-git) **y** [**ramas**](https://github.com/progit/progit/blob/master/es/03-git-branching/01-chapter3.markdown)

* git branch
  + Lista las ramas locales existentes
* git branch -a
  + Lista todas las ramas, tanto locales como remotas
* [git branch -r](http://es.gitready.com/intermediate/2009/02/13/list-remote-branches.html)
  + Lista las ramas remotas que conoce nuestro repositorio
* git branch nombreRama
  + Crea una rama nueva con el nombre especificado. No cambia automáticamente a la rama creada
* git branch -d nombreRama
  + Borra de manera normal la rama especificada
* git branch -D nombreRama
  + Borra de manera forzosa la rama especificada, por ejemplo, sin hacer commit
* git branch -m nombreRamaActual nombreRamaNuevo
  + Renombra una rama siempre y cuando no exista otra rama con el mismo nombre
* git checkout nombreRama
  + Cambia a la rama git especificada, siempre y cuando no tengamos cambios pendientes en la rama actual
* git merge nombreRamaLocalMerge
  + Fusiona o combina la rama especificada en nuestra rama actual. Crea un commit para guardar los cambios nuevos del merge
  + Si el merge falla debido a que Git requiere nuestra decisión para fusionar, deberemos resolver conflictos, agregar los archivos al área de preparación y hacer nosotros un commit de los cambios
* [git checkout --track -b nombreRamaLocal servidor/nombreRamaServidor](http://es.gitready.com/intermediate/2009/01/09/checkout-remote-tracked-branch.html)or
  + Crea una copia de la rama del servidor especificada con un nombre específico local
* git checkout -b nombreRamaNuevo origin/nombreRamaBase
  + checkout de una rama especifica definiendo el nombre de la rama a crear
* git push servidor nombreRamaLocal
  + Envía la rama local especificada al servidor
* git push origin :nombreRamaServidorABorrar
  + Borra una rama del servidor/b

[**Repositorios**](http://www.kernel.org/pub/software/scm/git/docs/git-remote.html) **y rutas**

* git remote
  + Muestra los nombres de los repositorios remotos a los que tenemos agregados
* git remote -v
  + Muestra un listado de las rutas que apuntan a los repositorios remotos que tenemos agregados
* git remote add nombreRepositorio usuario@servidor:/ruta/a/repositorio/repo.git
  + Agrega un nombre y ruta a un repositorio remoto
* git remote rm nombreRepositorio
  + Borra la ruta a un repositorio remoto
* git remote set-url nombreRepositorio usuario@servidor:/nueva/ruta/a/repositorio/repo.git
  + Modifica la ruta a un repositorio remoto

**Varios**

* git show hashCommit:ruta/al/archivo > ruta/al/archivo.bak
  + Vacía un archivo específico de un commit específico a un archivo de respaldo.bak para su posterior visualización
  + [Otras formas de recuperar versiones anti](http://www.google.com/url?q=http%3A%2F%2Fwww.alvaroremesal.net%2Fblog-alvaroremesal%2Frecuperar-archivos-antigos-con-git&sa=D&sntz=1&usg=AFQjCNHi3PPhAYVXAWexsRHdvLpV2sxITg)[guas de archivos](http://www.alvaroremesal.net/blog-alvaroremesal/recuperar-archivos-antigos-con-git)
* [git update-index](http://www.kernel.org/pub/software/scm/git/docs/git-update-index.html) --asume-unchanged ruta/al/archivo
  + [Dejar de seguir un archivo](http://hernanaguilera.com.ve/951/untrack-files-in-git/)
* git checkout <branch\_name> -- <paths>
  + Merge de un archivo específico en otra rama
  + Ejemplo: git checkout catalogoVehiculo -- ./ServiciosGenerales/web/vistas/sgl/oficina/cambiarOficina.xhtml
  + <http://nicolasgallagher.com/git-checkout-specific-files-from-another-branch/>
  + <http://jasonrudolph.com/blog/2009/02/25/git-tip-how-to-merge-specific-files-from-another-branch/>
  + <http://stackoverflow.com/questions/2364147/how-to-get-just-one-file-from-another-branch>
  + [http://ruturaj.net/git-merge-specific-files-from-another-branch-or-revision/](http://www.google.com/url?q=http%3A%2F%2Fruturaj.net%2Fgit-merge-specific-files-from-another-branch-or-revision%2F&sa=D&sntz=1&usg=AFQjCNG9lmoSvCs3Zg0Yz_qoIUc1eUjjaQ)
* [git checkout --patch <branch\_name> <path>](http://code.medula.cl/article_Apuntes-de-comandos-en-GIT.html)
  + Merge de un archivo específico de otra rama en la rama actual
* [git log -p /ruta/al/archivo/](http://cambio.name/personal/content/comandos-git)
  + Muestra el historial de un archivo en consola
* [gitk /ruta/al/archivo/](http://es.softuses.com/33190)
  + Muestra el historial de un archivo en una interfaz gráfica
* [git diff servidor/nombreRama -- ruta/local/archivo](http://stackoverflow.com/questions/21101572/git-diff-between-file-in-local-repo-and-origin)
  + Muestra las diferencias entre un archivo local y su versión en el servidor
* [git checkout -- nombrearchivo.extension](https://git-scm.com/book/es/v1/Fundamentos-de-Git-Deshaciendo-cosas)
  + Deshace los cambios en un archivo
* [git checkout .](http://sheldonrules.blogspot.mx/2012/04/deshacer-los-cambios-locales-en-git.html)
  + Deshace todos los cambios realizados en local
* [git checkout NUMCOMMIT /ruta/local/archivo](http://stackoverflow.com/questions/215718/reset-or-revert-a-specific-file-to-a-specific-revision-using-git)
  + Revierte un archivo a su estado en un commit específico
* [git push servidor nombreRamaLocal:nombreRamaRemota](https://git-scm.com/book/es/v1/Ramificaciones-en-Git-Ramas-Remotas)
  + Toma los cambios de la rama local y los pone en la rama remota indicada
* [git merge --abort](http://stackoverflow.com/questions/5741407/how-to-undo-a-git-merge-with-conflicts)
  + Al hacer un merge y este tener conflictos, esta instrucción devuelve el proyecto al estado anterior a intentar hacer el merge
* git merge --no-ff
  + Al hacer un merge integra la rama pero como una rama separada, no como una rama plana, haciendo que sea fácil ver la incorporación de ramas al histórico.
* [git rm --cached <file>](http://es.gitready.com/beginner/2009/01/19/ignoring-files.html)
  + Quitar un archivo del controlador
* [git rm --cached -r <dir>](http://es.gitready.com/beginner/2009/01/19/ignoring-files.html)
  + Quitar un directorio del controlador y todos los archivos que contiene
* git rm -r --cached <dir>/\*
  + Quitar un directorio del controlador y todos los archivos que contiene
* [git checkout HASHCOMMMIT /ruta/directorio](http://es.gitready.com/intermediate/2009/03/18/restoring-a-directory-from-history.html)
  + Recupera todo un directorio completo desde el commit especificado
* [Clonado](https://kiberion.repositoryhosting.com/projects/2/help/git_access)

**Comparar archivos**

* [git diff COMMITXXX ruta/archivo](http://es.stackoverflow.com/questions/40518/c%C3%B3mo-ver-las-diferencias-de-un-archivo-con-un-commit-espec%C3%ADfico)
  + Compara la versión actual de un archivo con la de un commit específico

**Enlaces de interés**

* [GIT - Guía rápida](http://www.edy.es/dev/docs/git-guia-rapida/)
* [Guía de Referencia Visual de GIT](http://www.google.com/url?q=http%3A%2F%2Fmarklodato.github.com%2Fvisual-git-guide%2Findex-en.html&sa=D&sntz=1&usg=AFQjCNEEiXg5JL5BalC4-D7VprADUiU9XQ)
* [Fundamentos de Git - Consejos y Trucos (autocompletado y alias)](http://www.google.com/url?q=http%3A%2F%2Fgit-scm.com%2Fbook%2Fes%2FFundamentos-de-Git-Consejos-y-trucos&sa=D&sntz=1&usg=AFQjCNHtUpPb5nVn2rNzfrOWg9yoUpcmzg)
* [Pro Git - Scott Chacon (2009)](https://docs.google.com/file/d/0BzEEY_WNVMUYWm80QlZ0S2Q5Z3M/edit)
* [Comparación GIT vs Suversion vs Mercurial](https://docs.google.com/file/d/0BzEEY_WNVMUYaUZwX1Jhb09hajg/edit)
* [Comparativa GIT vs SVN](https://docs.google.com/file/d/0BzEEY_WNVMUYOXdjZV9VTE9jaFE/edit)
* [Bajar branch remoto](http://7sabores.com/blog/descargar-un-branch-remoto-cuando-usamos-git)
* [Chuleta de comandos](https://elbauldelprogramador.com/mini-tutorial-y-chuleta-de-comandos-git/)
* [Git ignore](http://aprendegit.com/ignorando-ficheros-en-git-parte-i-formas-de-ignorar-ficheros/)
* [Git Exclude](http://aprendegit.com/ignorando-ficheros-en-git-parte-ii-prioridades/)

git remote show origin - Muestra las ramas que hay en el servidor

git fetch origin - Actualiza listado de ramas remotas conocidas.

[git remote prune origin](http://www.kernel.org/pub/software/scm/git/docs/git-remote.html) - Quita todas las ramas refs (ramas que en algún momento existieron pero que ya no existen en el servidor)

[Tags](http://www.genbetadev.com/sistemas-de-control-de-versiones/tags-con-git)

[Repositorio remoto](http://misapuntesinformatica.es/category/linux/utilidades/)

[Configurar .gitignore](http://aprendegit.com/ignorando-ficheros-en-git-parte-ii-prioridades/)

[Chuleta de comandos elbauldelprogramador.com](http://elbauldelprogramador.com/mini-tutorial-y-chuleta-de-comandos-git/)

[Git snippets](http://universo.emergya.es/snippets)

[Buscar archivo eliminado](http://www.7sabores.com/blog/recuperar-archivos-borrados-git)

[Eliminar archivo completamente del historial](http://es.gitready.com/beginner/2009/03/06/ignoring-doesnt-remove-a-file.html)

[Quitar archivo del versionado](http://git-scm.com/book/es/Fundamentos-de-Git-Guardando-cambios-en-el-repositorio)

[Quitar archivo del versionado 2](http://git-scm.com/book/es-ni/Git-Basics-Recording-Changes-to-the-Repository)

**SVN to GIT**

[Git y SVN - Libro oficial](http://librosweb.es/pro_git/capitulo_8/git_y_subversion.html)

[git-svn](https://www.kernel.org/pub/software/scm/git/docs/git-svn.html)

[GIT+SVN](http://javisantana.github.io/slides/git-svn/#1)

[De SVN a GIT - 1](http://idiginbpel.wordpress.com/2010/04/19/de-svn-a-git/)

[De GIT a SVN - 2](http://idiginbpel.wordpress.com/2010/04/19/trabajando-efectivamente-con-svn-y-git/)

Otros

[Crear repositorio git remoto desde proyecto local existente](http://www.danielnavarroymas.com/crear-repositorio-git-con-proyecto-existente/)

[Comandos interesantes](http://code.medula.cl/article_Apuntes-de-comandos-en-GIT.html) (posible solución)

[Comandos Interesantes 2](https://gist.github.com/jelcaf/1404619)

Workflows

<https://www.atlassian.com/es/git/workflows#!workflow-feature-branch>

[Escenarios de trabajo](http://rooteando.com/escenarios-de-trabajo-en-git)

[Merge desde otro repositorio](http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=git)

[Aprendegit.com](http://aprendegit.com/)

[Git inmmersion](http://gitimmersion.mx/index.html)

[Diff visual](http://blog.openalfa.com/como-comparar-visualmente-dos-versiones-de-un-fichero-en-git)

Ejemplo buscar archivo eliminado y restaurarlo

MinideFcturable:FBComprobantesFiscales facturable$ git log --oneline --name-status --diff-filter=D -- \*\*/ComprobanteSimpleVO.java

error: cannot run textedit: No such file or directory

750ead6 clases que se movieron a FExternos

D src/main/java/com/oss/facturable/vo/ComprobanteSimpleVO.java

MinideFcturable:FBComprobantesFiscales facturable$ git checkout 750ead6^ src/main/java/com/oss/facturable/vo/ComprobanteSimpleVO.java

**Github**

* [git pull origin master --allow-unrelated-histories](https://stackoverflow.com/questions/37937984/git-refusing-to-merge-unrelated-histories-on-rebase) - Soluciona el error git pull fatal: refusing to merge unrelated histories