

Guion curso Git Galpon

29/03/2014

1 Revisión rápida de los conceptos del 1º taller

Ver archivo "Git_avanzado_presentación.pdf"

2 BitBucket. Colaborando en un servidor remoto: remote, fetch, push, pull.

BitBucket 101 <https://confluence.atlassian.com/display/BITBUCKET/Bitbucket+101>

- Git instalado en local
 - which git
 - git --version
- Crear cuenta <https://bitbucket.org/account/signup/>

Emprezando un proyecto desde cero.

- Crear un repositorio. Botón "Create". Nombre "prueba_html".
 - Privado.
 - Allow only private forks.
 - Git.
 - Issue tracking.
 - Wiki.
 - Language: HTML/CSS.
- Ejecutar:
 - mkdir ~/prueba_html/
 - cd ~/prueba_html/
 - git init
 - git remote -> salida vacía
 - git remote add origin https://amieiro@bitbucket.org/amieiro/prueba_html.git
 - git remote
 - git remote -v
 - echo "#Mi fichero README" >> README.md
 - git status
 - git add README.md
 - git commit -m "Commit inicial. Añado el archivo README"
 - git push -u origin master
 - Branch master set up to track remote branch master from origin.
 - Ver Bitbucket
 - touch index.html

- touch style.css
- git status
- git push -> Everything up-to-date
- git add .
- git commit -m "Añado el index.html y el style.css"
- git push
- Ver Bitbucket

Sincronizando un proyecto local existente

- Ejecutar:
 - mkdir ~/prueba_html_2/
 - cd ~/prueba_html_2/
 - git init
 - touch index.html
 - touch style.css
 - git status
 - git add .
 - git commit -m "Commit inicial. Añado index.html y style.css"
 - git tag -a v.0.1 -m "Commit inicial"
- Crear un repositorio. Botón "Create". Nombre "prueba_html_2".
 - Privado.
 - Allow only private forks.
 - Git.
 - Issue tracking.
 - Wiki.
 - Language: HTML/CSS.
- Ejecutar:
 - git remote
 - git remote add origin https://amieiro@bitbucket.org/amieiro/prueba_html_2.git
 - git remote -v
 - git push -u origin --all
 - Ver Bitbucket
 - git push -u origin --tags
 - Ver Bitbucket

- echo "#Mi fichero README" >> README.md
- git add README.md
- git commit -m "Añado el archivo README"
- git push -u origin master //Con upstream valdría git push
- echo "#Una segunda línea comentada" >> README.md
- git commit -am "Añado una segunda línea al archivo README.md"
- git push //-----u origin master
- Añado 3 ramas a partir de la master (ejecutar):
 - git branch rama_1
 - git branch rama_2
 - git branch rama_3
 - git checkout rama_1
 - echo "Creo el archivo_1.txt en la rama_1" >> archivo_1.txt
 - git add .
 - git commit -m "Creo el archivo_1.txt en la rama 1"
 - git branch
 - git log --oneline
 - git push //---- fallo. The current branch rama_1 has no upstream branch
 - git push -u origin rama_1
 - git checkout rama_2
 - echo "Creo el archivo_2.txt en la rama_2" >> archivo_2.txt
 - git add .
 - git commit -m "Creo el archivo_2.txt en la rama 2"
 - git branch
 - git log --oneline
 - git push -u origin rama_2:mi_rama_2
 - -
 - more ~/.gitconfig
 - git config --global push.default upstream
 - more ~/.gitconfig
 - -
 - echo "Añado una línea al archivo_2.txt" >> archivo_2.txt
 - git commit -am "Añado una línea al archivo_2.txt en la rama 2"

- Ver Bitbucket
- git push
- Ver Bitbucket

Añado un segundo usuario:

- En Bitbucket ir a Share e introducir el Usuario. Dar permisos de administración.

En el panel de administración del segundo usuario ir a Inbox.

- You have been given admin access to amieiro/prueba_html_2

En una shell con la configuración del segundo usuario

- cd ~
- git clone https://JesusAmieiro@bitbucket.org/amieiro/prueba_html_2.git
prueba_html_2
- cd ~/prueba_html_2
- git log --oneline
- git branch
- git branch -a
- cat .git/config
- git branch rama_1
- git branch -a
- cat .git/config
- git branch -d rama_1
- git branch -a
- git branch rama_1 origin/rama_1 -> Branch rama_1 set up to track remote branch rama_1 from origin.
- git branch -a -v
- cat .git/config
- git checkout master
- git branch -a -v
- git checkout -b rama_2 origin/mi_rama_2 -> Branch rama_2 set up to track remote branch mi_rama_2 from origin. Switched to a new branch 'rama_2'
- git branch
- git branch -a -v
- echo "Añado una tercera línea al archivo_2.txt" >> archivo_2.txt
- git add .
- git commit -m "Añado una tercera línea al archivo_2.txt"

- Ver commits en BitBucket
- git push -->>> rama_2 -> mi_rama_2
- Ver commits en BitBucket

Me cambio al usuario 1

- git branch -a -v
- git checkout rama_1
- echo "Añado una segunda línea al archivo_1.txt de la rama_1" >> archivo_1.txt
- git add .
- git commit -m "Añado una segunda línea al archivo_1.txt"
- git push -->>> OK
- Ver commits en BitBucket
- git checkout rama_2
- more archivo_2.txt
- echo "Yo también añado una tercera línea al archivo_2.txt" >> archivo_2.txt
- git add .
- git commit -m "Yo también añado una tercera línea al archivo_2.txt"
- git push -> Error
- git branch -a -v -->>>> [ahead 1]
- git fetch -> From https://bitbucket.org/amieiro/prueba_html_2 mi_rama_2 -> origin/mi_rama_2
- git branch -a -v -->>>>> [ahead 1, behind 1]
- git status -> On branch rama_2 Your branch and 'origin/mi_rama_2' have diverged, and have 1 and 1 different commit each, respectively. (use "git pull" to merge the remote branch into yours)
- git pull -> Auto-merging archivo_2.txt CONFLICT (content): Merge conflict in archivo_2.txt Automatic merge failed; fix conflicts and then commit the result.
- more archivo_2.txt
- git mergetool -->>>> Salgo
- vim archivo_2.txt Arreglo el problema, quedando la línea afectada con el texto "Este es el contenido definitivo de la tercera línea"
- git status
- git commit -am "Arreglo el conflicto en la rama 2"
- git push
- Ver Bitbucket

Me cambio al segundo usuario

- git fetch
 - mi_rama_2 -> origin/mi_rama_2
 - rama_1 -> origin/rama_1
- git branch -a -v
 - rama_1 [behind 1] Creo el archivo_1.txt en la rama 1
 - rama_2 [behind 2] Añado una tercera línea al archivo_2.txt
- git checkout rama_2
 - Your branch is behind 'origin/mi_rama_2' by 2 commits, and can be fast-forwarded. (use "git pull" to update your local branch).
- more archivo_2.txt -> contenido sin actualizar
- git checkout rama_1 -> Your branch is behind 'origin/rama_1' by 1 commit, and can be fast-forwarded. (use "git pull" to update your local branch).
- git merge origin/rama_1
- git branch -a -v
- git checkout rama_2
- git pull
- git branch -a -v
- git pull --all
 - Fetching origin
 - Already up-to-date.

Cambio al usuario 1

- git checkout rama_1
- echo "Añado una línea más al archivo_1" >> archivo_1.txt
- git add .
- git commit -m "Añado una línea más al archivo_1.txt"
- git push
- git checkout rama_2
- echo "Añado otra línea más después de arreglar el conflicto" >> archivo_2.txt
- git add .
- git commit -m "Añado una línea más al archivo_2.txt"
- git push

Cambio al usuario 2

- `git checkout rama_2`
- `git pull`
- `git branch -a -v`
 - `rama_1 [behind 1]` Añado una segunda línea al `archivo_1.txt`
- `git checkout rama_1`
- `git pull`
- `git branch -a -v`
- `--`
- `git pull --all`
 - Fetching origin
 - Already up-to-date.

3 Ejercicio por parejas

- Ejercicio por parejas. Cada uno de los dos integrantes del equipo va a tener un perfil distinto: usr_1 y usr_2.
- Crear una cuenta en Bitbucket.
- El usr_1 va a ser el dueño del repositorio.

Ejercicio

- Usr_1. Crear un repositorio local en el directorio "~/git_galpon".
- Usr_1. Añadir un archivo index.html.
- Usr_1. Añadir el texto "Simulación del archivo index.html" al archivo index.html.
- Usr_1. Añadir un archivo style.css.
- Usr_1. Añadir el texto "Simulación del archivo style.css" al archivo style.css.
- Usr_1. Crear un commit con el texto "Commit inicial. Añado el index.html y el style.css".
- Usr_1. Añadir una etiqueta con el nombre "v0.0.1" y con el texto "Commit inicial".
- Usr_1. Crear una nueva rama de desarrollo, llamada "dev".
- Usr_1. En la rama "dev" añadir un archivo "todo.txt".
- Usr_1. En la rama "dev" añadir el archivo .gitignore y excluir al archivo todo.txt.
- Usr_1. En la rama "dev" crear un commit con el texto "Añado el archivo .gitignore".
- Usr_1. Crear un nuevo repositorio en BitBucket con el nombre "git_galpon".
 - Privado.
 - Allow only private forks.
 - Git.
 - Con Issue tracking.
 - Con Wiki.
 - Language: HTML/CSS.
- Usr_1. Añadir la referencia al repositorio remoto que acabamos de crear.
- Usr_1. Subir todas las ramas al repositorio remoto.
- Usr_1. Subir todas las etiquetas al repositorio remoto. git push -u origin --tags.
- Usr_1. Dar acceso de administración al usr_2 al proyecto creado.
- Usr_2. Descargar a local el proyecto con todas las ramas y todas las etiquetas, en el directorio raíz del usuario.
- Usr_2. En la rama "dev" añadir la carpeta "img" y los archivos "img/logo.png" y /img/logo.xcf.

- Usr_2. En la rama "dev" añadir al archivo .gitignore los archivos con la extensión .png de la carpeta /img/.
- Usr_2. En la rama "dev" crear un commit con el texto "Añado el logo y .gitignore".
- Usr_2. Llevar los cambios a la rama principal.
- Usr_2. Llevar los cambios al servidor de Bitbucket.
- Usr_1. Descargar a local los cambios hechos por usr_2.
- Usr_1. En la rama "dev" añadir la carpeta "js".
- Usr_1. En la rama "dev" añadir al archivo "js/jquery-2.1.0.min.js".
- Usr_1. En la rama "dev" crear un commit con el texto "Añado el archivo jquery-2.1.0.min.js"
- Usr_1. Llevar los cambios a la rama principal.
- Usr_1. Llevar los cambios al servidor de Bitbucket.

Solución al ejercicio

Usr_1

- mkdir ~/git_galpon
- cd ~/git_galpon/
- git init
- touch index.html
- echo "Simulación del archivo index.html" >> index.html
- touch style.css
- echo "Simulación del archivo style.css" >> style.css
- git add .
- git commit -m "Commit inicial. Añado el index.html y el style.css"
- git tag -a v0.0.1 -m "Commit inicial"
- git checkout -b dev
- touch todo.txt
- touch .gitignore
- echo "todo.txt" >> .gitignore
- git add .gitignore
- git commit -m "Añado el archivo .gitignore"
- git remote add origin https://amieiro@bitbucket.org/amieiro/git_galpon.git (o también git remote add origin git@bitbucket.org:amieiro/git_galpon.git).
- git push -u origin master
- git push -u origin dev
- git push --tags

Usr_2

- cd ~
- git clone https://JesusAmieiro@bitbucket.org/amieiro/git_galpon.git
- cd git_galpon/
- git branch -a -v
- git checkout -b dev origin/dev
- mkdir img
- cd img/

- touch logo.png
- touch logo.xcf
- cd ..
- echo "img/*.png" >> .gitignore
- git status
- git add .gitignore
- git add img/logo.xcf
- git status
- git commit -m "Añado el logo y modifíco el .gitignore"
- git checkout master
- git merge dev
- git push --all

Usr_1

- git checkout master
- git pull --all # Explicar
- git branch -a -v ->>>>>> dev [behind 1]
- git checkout dev
- git pull
- git branch -a -v
- mkdir js/
- cd js/
- touch jquery-2.1.0.min.js
- cd ..
- git add .
- git commit -m "Añado el archivo jquery-2.1.0.min.js"
- git checkout master
- git merge dev
- git push --all

4 Flujos de trabajo en Git

Colaborar en un repositorio público o privado.

Ver archivo "Git_avanzado_presentación.pdf"

5 Alias de órdenes

Ver archivo "alias_git.sh"

Para cargarlo ejecutar "source alias_git.sh"

Ver archivo "alias_git_bash.sh"

Para cargarlo ejecutar "source alias_git_bash.sh"

6 GUI

En <http://www.git-scm.com/downloads/guis> hay interfaces de usuario para los tres principales tipos de sistemas operativos.

7 Depurando con Git

git blame

Ejemplo con bisect

git clone <https://github.com/amieiro/error500.git>

Buscamos un fallo introducido en el commit

- 2860baf - Add the responsive menu for mobiles

Pero desconocemos en que commit se ha introducido.

Sabemos que en el actual está mal.

Sabemos que el commit estaba bien

- 3fd1424 - Change the CSS in the main menu

A partir de ahí vamos a buscar dónde se ha introducido el problema.