GIT - MANUAL

CONTROL DE VERSIONES

Sonia Celis
Project Leader & Big Data Analytics.

GIT – OPEN SOURCE

- Control de versiones
- Equipos de trabajo
- Mantenimiento de código
- Volver a cambios anteriores
- Repositorios Locales / Remotos
- GIT - GIT-SCM.COM - VCS - VERSION-CONTROL-SOFTWARE - SISTEMA DISTRIBUIDO - CONTROL DE VERSIONES :: LINUS TORVALDS ::

SOME CONCEPTS

- WorkingDirectory -> archivos de trabajos antes de crear la versión - trabajo inmediato que se pasará al staging area.
- Staging_Area -> area de preparación de ficheros (listado de los ficheros a guardar)
- Repositorio –> versiones de los archivos, cambios definitivos,
- Git bash -> línea de comandos de git
- snapshot -> Imágenes de código / “fotos” en un instante
- ramas locales - Master, Develop
- ramas remotas - En la nube - origin - pueden apuntar a GitHub, Bitbucket, Gitlab
- Branch - fork - proyecto diferente que se crea a partir de otro proyecto
-

BASIC COMMANDS

- git help tutorial - show commands -> para ver todos los comandos undo y reset
- git init -> para empezar a trabajar con git, crea la carpeta .git/ permite administras los cambios del código, esta carpeta está oculta
- git status -> para saber que archivos estamos trabajando:
  - On branch master
  - Your branch is up to date with 'origin/master'.
  - nothing to commit, working tree clean
  - Untracked files: –> archivos sin agregar al staging area.
- git add <name file> -> pasar ficheros del working al staging area.
- git add README.md -> add ficheros uno a uno..
- git add <name file> -> realiza el Staged a los archivos, es decir, las modificaciones que se van haciendo en local se van añadiendo al staging area (area de trabajo)
- git add -> add todos los ficheros a la vez.
- git commit -> del staging al repositorio, primer snapshot.
- git commit -m "first commit” - primera foto - snapshot
- git commit -> realiza el commit a los archivos del staging area al repositorio - checksum individual
- git commit y con vi insertamos un comentario del commit(I insert - wq guardar y salir)

BASIC COMMANDS

- git config -> global user.email “soniacelis@gmail.com"
- git config -> global user.name “Sonia”
- git clone https://github.com/SCelisV/NameRepository.git --> clonar un repositorio existente desde el servidor al sitio que tu quieras en el pc.
- git push -> subir al repositorio remoto, actualiza el repository ó feature remoto, para que otros desarrolladores lo puedan ver
- git push -u origin master --> Actualiza el repositorio remoto.
- git pull -> git log -> para ver lo que hemos hecho, vemos la historia de los commit (en todos los branch), de forma descendente… con sus correspondientes checksum
- git checkout - <name file> --> para revertir los cambios hechos en el fichero <name file>
- git checkout <branch name> --> me sirve para moverme entre branch
- git diff <name file> --> para ver las diferencias entre los ficheros antes de hacer el commit
- git diff --cached --> spx.txt --> revisar los cambios antes de realizar el push
- git diff --staged --> cambios preparados vs último push
- git remote - git push - git pull
- git remote add origin https://github.com/SCelisV/NameRepository.git --> Origen en donde vamos a copiar el código
- git pull - recuperar los cambios de otros desarrolladores - https://github.com/SCelisV/NameRepository.git

BASIC COMMANDS

- git branch --> muestra los branch - “ramas-versiones” que existen!!
- git branch <name> --> crea un nuevo branch con un nombre
- git rm -d <name branch> –> borrar una rama
- * git remote - git push - git pull
- git remote add origin https://github.com/SCelisV/NameRepository.git --> Origen en donde vamos a copiar el código
- git push -u origin master
- git pull https://github.com/SCelisV/NameRepository.git

Sonia Celis
Project Leader & Big Data Analytics.

BASIC COMMANDS

- Ignorar código:-
- .gitignore --> se incluyen los ficheros y ó tipos de ficheros que no queremos registrar!!

Sonia Celis
Project Leader & Big Data Analytics.