CSD

Agilar

GIT - version control (time: 30')

Next exercise is focused on know the bases of working over GIT source control.

Before start

Please read carefully, go step by step. Verify if you have installed the required software for this exercise.

- GIT (www.git.org)

open terminal (Linux/MacOSX) or command line (windows) and check if GIT is installed executing

```
$git help
$git help -a
```

Initiation to GIT

1. Init the repository
   - Create new folder in your hard disk and navigate on it
   - Create an empty repository `$git init`
   - `$git status` to see the status of your repository

2. Configure the repository
   - `$git config --system core.editor <editor>`
   - `$git config –global –edit` opens the config for global configuration

3. Add some content
   - Create new file called "main.java" with "hello world example" content
   - Create a new folder called "model" inside the folder, and add a module called "A.java" to it.
   - Executes `$git status`
   - To track this files inside STAGE, `$git add .` or `$git add <name_of_file>`
   - To commit, this changes to the local repository `$git commit -m "commit_message"`

4. Track some changes
   - Make some changes on your file… maybe say "Goodbye", and save it.
   - `$git status` to see what has happened on the repo
   - Commit your changes.
   - To view all the commits, `$git log –graph –decorate ($git help log` for help)

5. Branch
   - Create new branch called "develop":
     - `&git branch develop`
     - `$git checkout develop`
     - or `$git checkout -b develop` (which is the same of two commands below)
   - `$git branch –list` to view the list of branches
   - Change your file and commit changes to develop branch
   - Switch branch to 'master' and take a look at your file.
   - Change your file on 'master' and commit again.

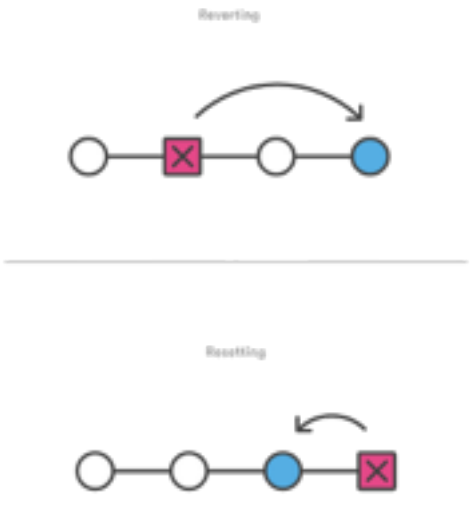
- Try to merge your change. Placed on ‘master’ execute: `$git merge develop`
- End merge commit and take a look a what happens to your file.

6. Reverting commits
   - Take a look at your file content, and remember it.
   - Make some new commit to your file, adding for each commit one new line or changing it.
   - To undo commits, `$git log <option>` to get the commit you want to revert. Don’t take de last one, use one older.
   - `$git revert <commit>`
   - Take a look at the content of your file, the content that corresponds to the selected commit has disappear.

7. Resolving conflicts
   - `$git checkout develop`, using ‘develop’ branch change your file, and remember where the change has been done.
   - Commit this change to ‘develop’
   - `$git checkout master`, and change your file at same position, with different content.
   - Commit this change to ‘master’
   - `$git diff develop` to view what is differing between branches.
   - Then try to merge branches.
   - A conflict appears, to solve them you can use a difftool, `$git config –global –edit` to put your preferred difftool. If you use a difftool, you will be able to choose what changes you keep or not.
   - When, you have resolved the conflict, commit the merge result.

8. Retrieving specific versions
   - To get a concrete commit for of a file, `$git log <option>` to know what commits are available to retrieve
   - Choose one commit, and execute: `$git checkout <commit> <filename>`
   - Open the file, the content has changed with the content of file for the concrete commit

9. Reset
   - First of all… let’s some concepts… Revert creates commit undoing an older commit, and reset deletes all commits after the selected to reset.



   - Choose some older commit, and `$git reset <commit> –hard`
   - There’re different flags to reset command: `–hard, –soft, – mixed, –merge,` … take a look `$git help reset`

10. .gitignore
    - Create a file called Main.class and add some content on it
    - `$git status`, and take a look if this file is unstaged or untracked
    - Create a file named: ".gitignore"
    - Edit this file adding following content:

      ```
      *.class
      ```

    - Save, and `$git add .gitignore`
    - `$git status`, look if your .class file appears?