**TESTING GIT, VERSION CONTROL, GITHUB:**

**PREPARATION**

**GITHUB:**

- Create a TEST repositorio in Github

URL repositorio: https://github.com/cristinaverdi/repository-test.git

**TERMINAL (local computer):**

- Create a DEV folder in your local computer.

**NOTE:** it is good practice to **always** work from 'dev' folder, not 'Documents' /'Downloads'.

**START**

STEP 1.

- Open the terminal in your USER
- Create a "dev" folder: `**mkdir dev**`**:**

```
adriana.vile@ES-A2S-SC-0011 MINGW64 ~/Desktop/repository-test-master (development)
$ mkdir dev
```

STEP 2.

- Go to dev`**cd dev**`

```
adriana.vile@ES-A2S-SC-0011 MINGW64 ~/Desktop/repository-test-master (development)
$ cd dev
```

STEP 3.

- Clone repository into dev folder
- `**git clone** https://github.com/cristinaverdi/repository-test.git`

**adriana.vile@ES-A2S-SC-0011 MINGW64 ~/Desktop/repository-test-master/dev (development)**
**$ git clone https://github.com/cristinaverdi/repository-test.git**

```
Cloning into 'repository-test'...
remote: Counting objects: 42, done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 42 (delta 2), reused 0 (delta 0), pack-reused 29
Unpacking objects: 100% (42/42), done.
```

Now you have created a **dev** folder, with the repository name **'repository test'**
This is a **copy on your local computer** of the remote repository in Github.

**Troubleshoot**

- Print the folders , use `**$ ls**`
- to see **list of files inside your folder** (on the local computer)

```
adriana.vile@ES-A2S-SC-0011 MINGW64 ~/Desktop/repository-test-master (development)
$ ls
file1.txt   file2.txt  'Practice Git.md'
```

- To go to **user folder,**, use `**$ cd**`**:**

```
adriana.vile@ES-A2S-SC-0011 MINGW64 /c/Users
$ cd Adriana.vile
```

- Clone remote repository into local dev folder: `**git clone URL**`
- = creat a copy of the project in your computer

- To go to **dev folder in User,**, use `**$ mkdir dev` & $ cd dev:**

adriana.vile@ES-A2S-SC-0011 MINGW64 ~ (development)
$ mkdir dev
$ cd dev

- To clone **remote repository into dev folder on local computer**,
- use `**$ git clone URL`:**

**adriana.vile@ES-A2S-SC-0011 MINGW64 ~ (development)**
**$ git clone https://github.com/cristinaverdi/repository-test.git**

Cloning into 'repository-test'...
remote: Counting objects: 42, done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 42 (delta 2), reused 0 (delta 0), pack-reused 29
Unpacking objects: 100% (42/42), done.

- To go to the **root folder** (=user), use `**$ cd ..`:**

adriana.vile@ES-A2S-SC-0011 MINGW64 ~ (development)
$ cd ..

adriana.vile@ES-A2S-SC-0011 MINGW64 /c/Users

- To **choose correct user**, use `**$ cd username`:**

cd Adriana-vile

- In user folder, create **dev folder.**

**NOTE:** Whenever you need to create a new project or clone a repo, you can do it in dev. there's no master folder:

- To **move to repository**, use `**$ cd repository-test`:**

cd repository-test

**IN EDITOR:**

- **Open your Editor and see `user/dev/repository test`**
- in **Visual Studio Code** you can also see the terminal inside:

1) Go to **'View'** menu in top bar.
2) Choose **Integrated Terminal**

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
PS C:\Users\Adriana.vile\dev\repository-test>
```

- **Now do `git branch` inside Visual Studio Code terminal:**

```
PS C:\Users\Adriana.vile\dev\repository-test> git branch
* master
```

**NOTE: `*` =** 'current branch' **By default,** after cloning a remote repo, the current branch = **master**

STEP 4.

- Create a new branch in repository,when you are IN master.
- Use **`git checkout -b <BRANCH_NAME>`**
- This is a 'child branch' of master
- Call it **feature/info-about-the-repository**

```
PS C:\Users\Adriana.vile\dev\repository-test> git checkout -b feature/info-about-the-repository
Switched to a new branch 'feature/info-about-the-repository'
```

**To check which branch you are add: `git branch`**

```
PS C:\Users\Adriana.vile\dev\repository-test> git branch
* feature/info-about-the-repository
  master
```

This is a list of the local branches you have in your computer
Now you see the **`*` in `feature ...`** because it is the local branch

**IN BROWSER-GITHUB:**

STEP 5. **Vinculate the new branch** in local computer to remote repository:

1) Open the **browser**
2) Visit this URL of remote repository
   **https://github.com/cristinaverdi/repository-test**
3) Check the branches with the **'Branch' menu**
4) **On Remote repository,** there is still no other branch then **Master branch.**

**IN TERMINAL-EDITOR:**

- To **add dev branch** to remote repository:

```
git push --set-upstream origin https://github.com/cristinaverdi/repository-test
git remote set-url origin https://github.com/cristinaverdi/repository-test.git
```

**IN BROWSER-GITHUB:**

- Check branches on remote repository in browser

git push --set-upstream origin feature/info-about-the-repository

`Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/cristinaverdi/repository-test.git
 * [new branch]      feature/info-about-the-repository -> feature/info-about-the-repository
Branch 'feature/info-about-the-repository' set up to track remote branch
'feature/info-about-the-repository' from 'origin'.
PS C:\Users\Adriana.vile\dev\repository-test>`

**IN TERMINAL-EDITOR:**

PS C:\Users\Adriana.vile\dev\repository-test> **git add .**

PS C:\Users\Adriana.vile\dev\repository-test> **git status**
On branch feature/info-about-the-repository
Your branch is up to date with 'origin/feature/info-about-the-repository'.

**IN EDITOR:**

STEP 6

- **Create a README.md file** and save it.

**IN TERMINAL:**

- Add a file with **`git add .`**

PS C:\Users\Adriana.vile\dev\repository-test> git add README.md

- **Commit file changes to remote repository with**
  `git commit -m "<APPROPRIATE_COMMIT_MESSAGE>"`

PS C:\Users\Adriana.vile\dev\repository-test> git commit -m "Add README.md file"
On branch feature/info-about-the-repository
Your branch is ahead of 'origin/feature/info-about-the-repository' by 1 commit.
  (use "git push" to publish your local commits)

- To check file changes in terminal: **`git status`**

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   README.md

**IN TERMINAL:**

STEP 7

- To add changes from local to remote repository **`git push .`**

PS C:\Users\Adriana.vile\dev\repository-test> git push

Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.

Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 329 bytes | 329.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/cristinaverdi/repository-test.git
   b76cd09..644da20  feature/info-about-the-repository -> feature/info-about-the-repository
PS C:\Users\Adriana.vile\dev\repository-test>

PS C:\Users\Adriana.vile\dev\repository-test> git status
On branch feature/info-about-the-repository
Your branch is up to date with 'origin/feature/info-about-the-repository'.

**IN TERMINAL:**

STEP 8 **Upload the changes in local** to remote repository, **`git push`**

- See changes: **git status**

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)
        new file:   README.md

- **Git add** README.md

PS C:\Users\Adriana.vile\dev\repository-test> **git add README.md**

- **Git commit** to commit changes to remote.

PS C:\Users\Adriana.vile\dev\repository-test> **git commit -m "Format  README.md file"**
[feature/info-about-the-repository 6e2cdb0] Format  README.md file
 1 file changed, 32 insertions(+)

- **Git push** to send changes to remote repository.

PS C:\Users\Adriana.vile\dev\repository-test> **git push**

Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.19 KiB | 407.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/cristinaverdi/repository-test.git
   644da20..6e2cdb0  feature/info-about-the-repository -> feature/info-about-the-repository
PS C:\Users\Adriana.vile\dev\repository-test>

To create a **snapshot: git add**

- **`git add .`**=‘add all the files in this project’ NOTE: the `.`= ‘all files’
- **git add  README.md** = to add only README.md file
- **‘Unstage’ =** ‘don’t prepare to upload those files’
- ``git commit -m "Add README.md file"``= create snapshot of state of project
  The commit messages need to explain what have been done.

To upload the changes: **`git push`**
in remote: `feature/info-about-the-repository (less than a minute ago)`
you have just contributed to a remote repository

**IN GITHUB:**
**STEP 9: Creating a pull request**
Open a pull request to merge changes into master

1. On **GitHub,** navigate to the main page of the repository.
2. In the "**Branch"** menu, choose the branch that contains your commits.
3. To the right of the Branch menu, click **New pull request**.
4. Use the *base* branch dropdown menu to **select the branch you'd like to merge** your changes into, then use the *compare* branch drop-down menu to choose the topic branch you made your changes in.
5. Type a **title and description** for your pull request.
6. Click **Create pull request**.

Now the README file is in the Master branch, merged and added with a pull request.
**NOTE:** A **pull request =** a double check. It ensures that the incoming changes are ok and can be included in the main branch. The developer who opens the pulls request, has to wait until the pull request is reviewed and accepted.