

FACULTATEA CALCULATOARE, INFORMATICA SI
MICROELECTRONICA

UNIVERSITATEA TEHNICA A MOLDOVEI

MEDII INTERACTIVE DE DEZVOLTARE A
PRODUSELOR SOFT

LUCRAREA DE LABORATOR#1

Version Control Systems si modul de setare a unui server

Autor:
Ernest BITCA

lector asistent:
Irina COJANU
lector superior:
Radu MELNIC

Lucrarea de laborator #1

1 Scopul lucrării de laborator

De a se învăța utilizarea unui Version Control System și modul de setare a unui server.

2 Obiective

Studierea Version Control Systems (git).

3 Mersul lucrării de laborator

3.1 Cerintele

Initializarea unui nou repositoryu.
Configurarea VCS.
Commit, Push pe branch.
Folosirea fisierului .gitignore.
Revenire la versiunile anterioare.
Crearea branch-urilor noi.
Commit pe ambele branch-uri.
Merge la 2 branchuri.
Rezolvarea conflictelor.

3.2 Analiza Lucrării de laborator

Linkul la repository <https://github.com/Ernest96/MIDPS>

Sunt mai multe modalitati de a initializa un repository pe github. Putem crea o mapa goala in care vom plasa gitul nostru prin intermediul comenzii **git init**.

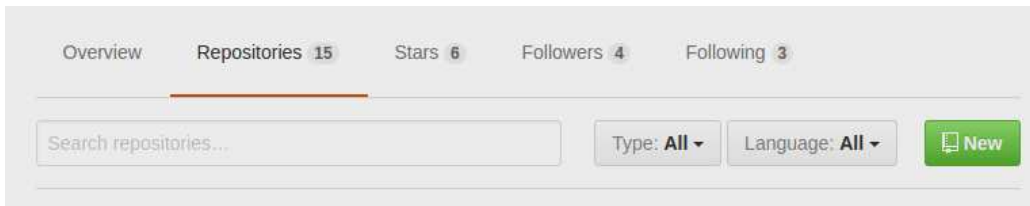
Urmatorul pas este crearea insusi a noului repository pe care il vom crea utilizind urmatoarea comanda **curl -u 'USER' https://api.github.com/user/repos -d '{"name":"NUME"}'**. Unde cuvintele scrise cu CAPS se vor inlocui cu numele utilizatorului si numele repositoryului.

Dupa aceasta este necesar sa unim gitul nostru gol cu repositoryul creat. Vom folosi urmatoarea comanda **git remote add origin "Linkul la repo"**



```
ernest@Lenovo: ~/testam
ernest@Lenovo:~$ cd testam/
ernest@Lenovo:~/testam$ git init
Initialized empty Git repository in /home/ernest/testam/.git/
ernest@Lenovo:~/testam$ curl -u 'Ernest96' https://api.github.com/user/repos -d '{"name":"test"}'
ernest@Lenovo:~/testam$ git remote add origin https://github.com/Ernest96/test.git
ernest@Lenovo:~/testam$
```

O alta metoda de a crea un repository este cea online. Pentru aceasta este nevoie sa deschidem pagina noastra pe github , sa alegem **repositories** si sa apasam butonul **new**.



Configurarea gitului const in mai multe etape. La inceput vom configura numele si emailul. Scriu urmatoarele comenzi:

```
git config --global user.name "NUMELE"
```

```
git config --global user.email EMAIL
```

```
ernest@Lenovo: ~/MIDPS/LAB1
ernest@Lenovo:~/MIDPS/LAB1$ git config --global user.name "Ernest96"
ernest@Lenovo:~/MIDPS/LAB1$ git config --global user.email "bitca@inbox.ru"
ernest@Lenovo:~/MIDPS/LAB1$ git config --list
user.name=Ernest96
user.email=bitca@inbox.ru
core.repositoryformatversion=0
core.filemode=true
core.bare=false
core.logallrefupdates=true
remote.origin.url=https://github.com/Ernest96/MIDPS.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
```

Urmatorul pas consta in generarea la cheia **SSH** (Secure Shell). Scriem in CLI **ssh-keygen**, iar cheia obtinuta o copiem in setarile noastre de pe git.

Este de dorit sa initializam repositoryul nostru cu un fisier **README.md** si un **.gitignore**. In fisierul README.md vom adauga niste informatie pentru cei care se vor folosi de repository iar in fisierul .gitignore vom adauga toate fisierele ce trebuiesc ignorate (adica sa nu fie incarcate).

```
ernest@Lenovo: ~/MIDPS/LAB1
ernest@Lenovo:~/MIDPS/LAB1$ vim README.md
ernest@Lenovo:~/MIDPS/LAB1$ vim .gitignore
ernest@Lenovo:~/MIDPS/LAB1$ cat README.md
# MIDPS
Ernest Bitca TI-153
ernest@Lenovo:~/MIDPS/LAB1$ cat .gitignore
#fisierele ce vor fi ignorate :
ignore.txt
#mapa ce va fi ignorata :
ignore/
ernest@Lenovo:~/MIDPS/LAB1$ |
```