

# Laboratorul 1

## TIDPP

### Interacțiunea cu sistemul de control al versiunilor Git. Utilizarea hostingului de repozitorii Github.

Să se inițializeze un repozitoriu local TIDPP în care să se creeze un folder Laboratorul-1. În folder-ul Laborator-1 creați un fișier **index.html** care să aibă drept conținut numele și prenumele studentului, grupa și facultatea delimitate prin virgulă.

Indexați toate cele create anterior și executați un commit. După executarea commitului, să se creeze 10 branch-uri fiecare având drept nume un limbaj de programare la discreția studentului.

Fiind în folder-ul Laborator-1, din fiecare branch să se creeze un fișier cu numele **main** cu extensia limbajului de programare ce corespunde numelui branch-ului. În fișier să se adauge secvența de cod necesară care va afișa “Hello World” pentru limbajul de programare respectiv. Fișierul cu cod nu trebuie compilat, executat etc.

Să se execute din branch-ul master un merge cu fiecare din cele 10 branch-uri.

Să se creeze în branch-ul master un fișier “**sha-1-heads.txt**” în care se va adăuga HEAD-urile tuturor branch-urilor în formatul “Nume branch : HEAD SHA-1 checksum”, fiecare din rând nou.

Să se creeze un repozitoriu pe Github cu numele **TIDPP** și să se execute un push din repozitoriul local. Pe lângă branch-ul master, în repozitoriul din Github trebuie să fie prezente și cele 10 branch-uri.

În repozitoriul din Github să se creeze un fișier “**FINAL.txt**” cu conținutul numelui și prenumelui studentului. Actualizați repozitoriul local cu schimbările care au fost efectuate.

#### Pentru nota 10:

- Să se soluționeze un conflict dintre 2 branch-uri
- Să se reseteze un branch la un commit anterior
- Să se configureze conectarea cu Github prin SSH

**Studentul trebuie să cunoască următoarele comenzi ce țin de CLI ( Command Line Interface ) :**

- pwd
- ls
- cd
- mkdir
- touch
- less
- head
- vim | nano

**Studentul trebuie să cunoască următoarele comenzi ce țin de Git :**

- git init
- git clone
- git status
- git log
- git branch
- git checkout
- git merge
- git add
- git commit
- git reset –hard
- git pull
- git push
- git remote
- git config

**Teorie:**

- Ce este Git ?
- Ce este Github ?
- Diferența dintre CVCS și DVCS
- Când se creează branch-ul master în repositoryul local ?
- Ce este HEAD ?
- Unde pot fi găsite HEAD-uri în formatul SHA-1 pentru fiecare branch ?
- Diferența dintre repository local și remote
- Pentru ce sunt necesare branch-urile în Git ?

Linkuri utile pentru efectuarea laboratorului:

- <https://maker.pro/linux/tutorial/basic-linux-commands-for-beginners>
- <https://dzone.com/articles/top-20-git-commands-with-examples>
- <https://jrebel.com/wp-content/uploads/2016/02/Git-Cheat-Sheet-pdf-v2.png>
- <https://learn.excelwithbusiness.com/blog/post/web-design/say-hello-world-in-28-different-programming-languages>
- <https://git-scm.com/book/ru/v2>

Pentru acest laborator nu este necesar de făcut raport, tot ce ați făcut o să văd prin commiturile efectuate, așa că denumiți commiturile cât mai pe înțeles.