

Ministerul Educației al Republicii Moldova

Universitatea Tehnică a Moldovei

Departamentul Inginerie software și automatică

RAPORT

Lucrare de laborator Nr.1
la PR

Tema: Versionarea codului sursă utilizind GIT

A elaborat :

st.gr. TI-142: Donca Diana

A verificat :

Donos Eugenia

Chișinău 2017

Tema Versionarea codului sursă utilizind GIT

Scop și Obiective

Lucrarea de laborator are ca scop studiul și înțelegerea principiilor de funcționare și utilizare a sistemului distribuit de control al versiunilor numit GIT.

Obiectiv: Crearea unui repozitoriu distant, localizat de serviciul gitlab.ati.utm.md, și sincronizarea tuturor modificărilor efectuate asupra repozitoriului local.

Noțiuni Teoretice

Sistemele de versionare (VCS, Version Control Systems - eng.) servesc la gestionarea versiunilor multiple ale fișierelor incluse într-un proiect colaborativ. Fiecare modificare efectuată asupra elementului de proiect se memorizează împreună cu autorului schimbării. Important de menționat că în orice moment de timp se poate reveni la o versiune anterioară a entității.

Motivatia principală constă în posibilitatea ca diferiți membri ai echipei, aflați eventual în spații geografice îndepărtate, să poată lucra simultan la proiect, urmând ca, la final, modificările lor să fie reunite în noi versiuni ale proiectului. De asemenea, există și alte avantaje. Când se observă un bug, se poate reveni la o versiune anterioară, în vederea determinării momentului introducerii acestuia în program. În același timp, se poate urma o dezvoltare pe ramuri (branches), în care se lucrează, în paralel, la multiple versiuni ale proiectului - de exemplu, una în care se dorește înlăturarea bug-urilor, iar cealaltă, în care se urmărește adăugarea de noi funcționalități, înaintea slefuirii celor existente.

Există două modele de VCS-uri:

- centralizat (ex: SVN): codul sursă este situat pe un server central, de unde clienții pot obține variante de lucru pe mașina locală (working copy). După efectuarea locală a modificărilor, dezvoltatorul solicită actualizarea variantei de pe server.
- distribuit (ex: Git): nu există un server central, procesul de sincronizare desfășurându-se la nivel peer-to-peer.

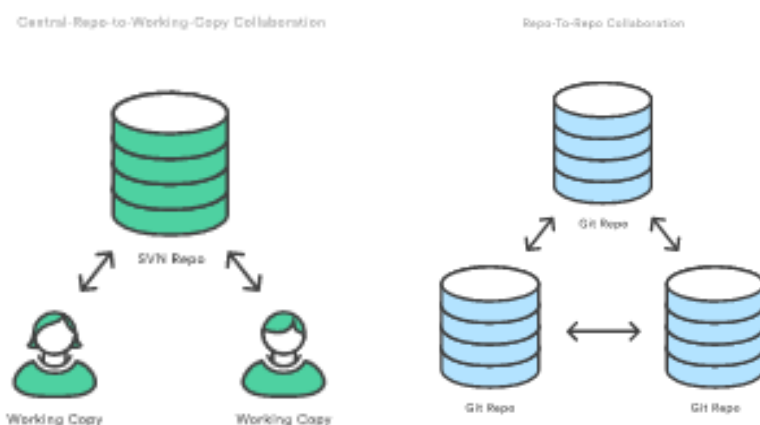


Figura 1 - Modul de interacțiune cu dipozitorile

Terminologie:

repository - pe server, conține ierarhia de fișiere și informațiile de versiune;

working copy - varianta locală, obținută de la server, pe care se fac modificările;

revision - o versiune a unui document. (v1, v2, v3...).

checkout - aducerea pe masina locala a versiunii de pe server, sub forma unei *working copy*

update/pul - actualizarea repozitoriului local în funcție de modificările survenite, între timp, pe server. Se aduc doar fișierele modificate;

commit - înregistrează o nouă versiune a fișierului (fișierelor) modificat în repozitoriu.

commit message - un mesaj asociat unei acțiuni *commit* care descrie schimbările făcute în noua versiune.

changelog - o listă a versiunilor (commit-urilor) unui fișier/proiect de obicei însoțită de mesajele asociate fiecărui *commit*.

diff: Afișează diferențele dintre două versiuni a unui fișier sau dintre fișierul modificat local (pe *working copy*) și o versiune de pe repository.

revert - renunțarea la ultimele modificări (locale) făcute într-un fișier din *working copy*, și revenirea la ultima versiune aflată în repozitoriu sau la o versiune la alegere.

branch - creează o “copie” a unui fișier/proiect pentru modificări „în paralel” fără a afecta starea actuală a unui proiect.

merge - aplică ultimele modificări dintr-o versiune a unui fișier peste alt fișier;

conflict - situația în care un *merge* nu se poate executa automat și modificările locale sunt în conflict cu modificările din repozitoriu.

resolve: rezolvarea (de obicei manuală) a conflictelor apărute într-un fișier după un *merge*.

Mersul Lucrări

1. Instalăm sistemul de control al versiunilor
 - GitBash - Downloads, <https://git-scm.com/downloads>
2. Creăm cont <https://github.com/>
 - Vă adresați profesorului pentru acest punct
 - Adăugați cheia publică generată de ssh-keygen (în git-bash terminal) în profilul utilizatorului. (Atenție! Cheia publică se regăsește în conținutul fișierului generat cu extensia **.pub**; cheile fiind localizate în directorul ascuns **.ssh** al utilizatorului de sistem: **./../Users/NumeUtilizator/.ssh/id_rsa**) reprezentare figura 1,2.

```
user@hp MINGW64 /d/programarea in retea/lab1 (master)
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
```

Figura 2 - Generare parolă

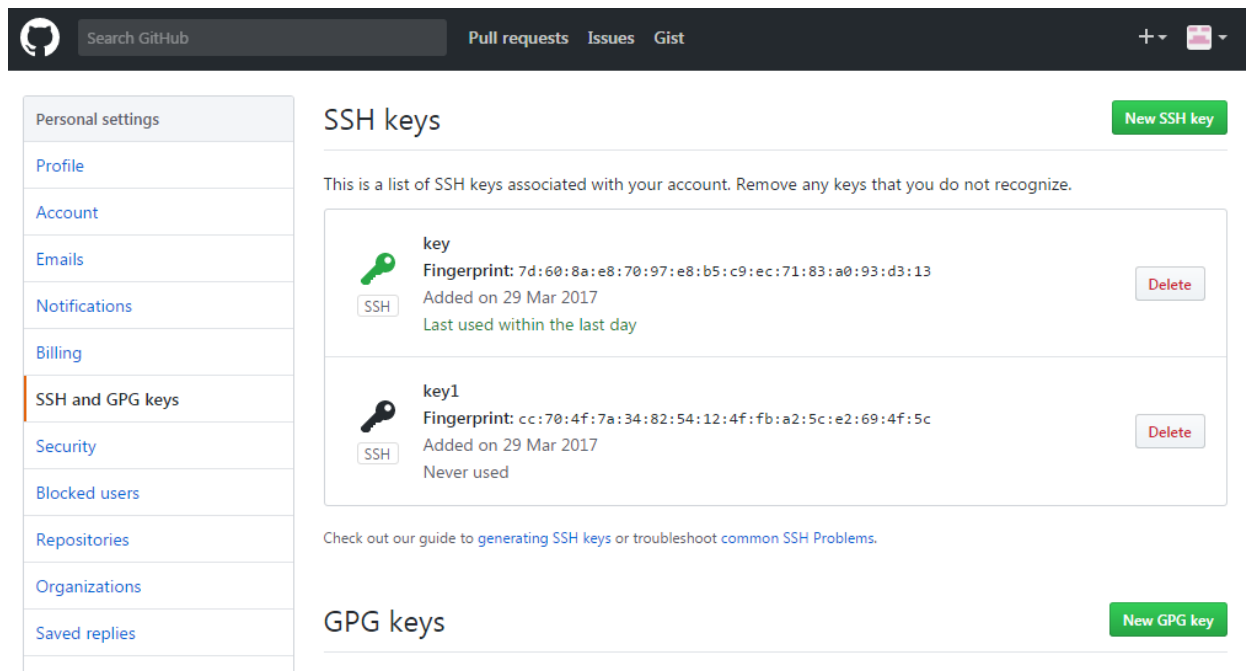


Figura 3 - Adăugarea parolei pe Github

3. Creăm un director local de lucru în care se vor afla fișierele de versionat
 - Puteți utiliza comanda **mkdir** pentru crearea directorului în linia de comandă a terminalului, iar navigarea o puteți realiza prin: **cd NumeDirector**.
4. Inițializați repozițorul GIT în acest director figura 4.
 - **git init**

```
user@hp MINGW64 /d/programarea in retea/lab1 (master)
$ git init
Reinitialized existing Git repository in D:/programarea in retea/lab1/.git/
```

Figura 4 - Inițializarea unui depozitoriu local

5. Modificați conținutul directorului figura 5.
 - adăugăm 1-2 fișiere.

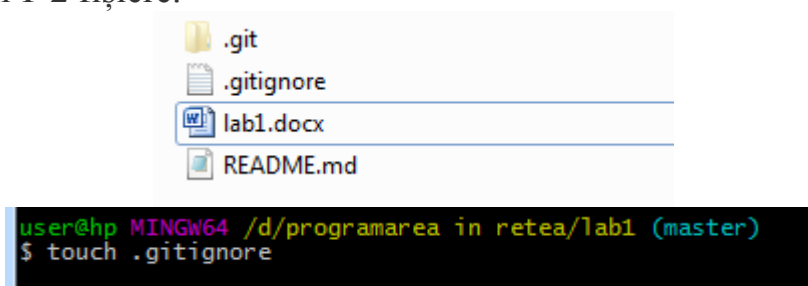


Figura 5 - Inițializarea unui fișier gitignore

6. Modificăm repozițoriul local, ori de câte ori se modifică directorul de lucru:

- STAGE - *git add* . (figura 6)
- HEAD - *git commit -m "First commit"* (figura 7)

```
user@hp MINGW64 /d/programarea in retea/lab1 (master)
$ git add .
```

Figura 6 - Adăugarea în depozitoriu.

```
user@hp MINGW64 /d/programarea in retea/lab1 (master)
$ git commit -m "init file"
[master e0c04e4] init file
Committer: Diana <Diana>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

2 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 .gitignore
create mode 100644 lab1.docx
```

Figura 7 - Commitul inițial

7. Adăugăm repozitoriul distant figura 8.

- *git remote add origin* [git@github.com:ddonca/programarea-in-retea.git](https://github.com/ddonca/programarea-in-retea.git)

```
user@hp MINGW64 /d/programarea in retea/lab1 (master)
$ git remote add origin git@github.com:ddonca/programarea-in-retea.git
```

Figura 8 - Inițializarea noilor configurări

8. Reînnomim repozitoriul distant *origin*, ramura *master* figura 9

- *git push -u origin master*

```
user@hp MINGW64 /d/programarea in retea/lab1 (master)
$ git push -u origin master
The authenticity of host 'github.com (192.30.253.113)' can't be established.
RSA key fingerprint is SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.253.113' (RSA) to the list of known hosts.
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 272 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:ddonca/programarea-in-retea.git
a214578..e0c04e4 master -> master
Branch master set up to track remote branch master from origin.
```

Figura 9 - Adăugarea la depozitoriul extern

9. Utilizăm frecvent comanda *git status* pentru a vedea starea directorului de lucru și repozitoriului local figura 10.

```
user@hp MINGW64 /d/programarea in retea/lab1 (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    hwCPP.cpp

nothing added to commit but untracked files present (use "git add" to track)
```

Figura 10 - Starea fișierelor

10. Creăm o ramură nouă de dezvoltare *Diana*, *numebranch* figura 11.

- *git branch Diana*

```
user@hp MINGW64 /d/programarea in retea/lab1 (master)
$ git branch Diana
```

Figura 11 - Crearea unui nou branch

11. Trecem în ramura nou creată *Diana*

- *git checkout Diana*
- ultimele două operații de creare și migrare la branch pot fi combinate prin: *git checkout -b Diana*

```
user@hp MINGW64 /d/programarea in retea/lab1 (master)
$ git checkout Diana
Switched to branch 'Diana'

user@hp MINGW64 /d/programarea in retea/lab1 (Diana)
$ |
```

Figura 12 - Migrarea la noua ramură

12. Modificăm directorul de lucru și repositoryul local figura 13

- Adăugați un nou fișier pe lângă cele existente
- Reînnoim repositoryul local (similar p. 6)

```
user@hp MINGW64 /d/programarea in retea/lab1 (Diana)
$ git commit -m "Adaugam un file cpp"
[Diana cd166ce] Adaugam un file cpp
Committer: Diana <Diana>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 10 insertions(+)
create mode 100644 hwCPP.cpp

user@hp MINGW64 /d/programarea in retea/lab1 (Diana)
$
```

Figura 13 - Adăugarea în depozitorul local

13. Verificăm dacă sunt modificări în repositoryul distant figura 14

- *git pull*
- studiem care alte comenzi *git* cuprinde această comandă!

```
user@hp MINGW64 /d/programarea in retea/lab1 (Diana)
$ git pull
Warning: Permanently added the RSA host key for IP address '192.30.253.112'
to the list of known hosts.
From github.com:ddonca/programarea-in-retea
* [new branch]      Diana      -> origin/Diana
There is no tracking information for the current branch.
Please specify which branch you want to merge with.
See git-pull(1) for details.

    git pull <remote> <branch>

If you wish to set tracking information for this branch you can do so with:

    git branch --set-upstream-to=origin/<branch> Diana
```

Figura 14 - Monitorizarea schimbărilor la nivel de repository

14. Reînnoim repozitoriul distant, aflîndu-vă pe ramura *Diana* figura 15

- *git push -u origin Diana*

```
user@hp MINGW64 /d/programarea în retea/lab1 (Diana)
$ git push -u origin Diana
Counting objects: 3, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 407 bytes | 0 bytes/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To git@github.com:ddonca/programarea-in-retea.git
   e0c04e4..cd166ce Diana -> Diana
Branch Diana set up to track remote branch Diana from origin.
user@hp MINGW64 /d/programarea în retea/lab1 (Diana)
$
```

Figura 15 - Adăugarea în repozitoriul extern

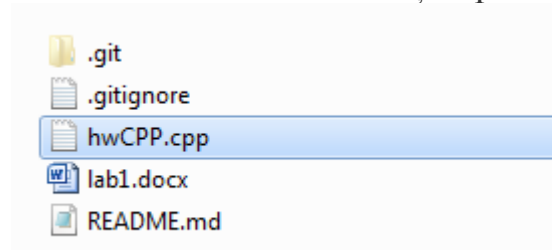
15. Verificam dacă sunt modificări în repozitoriul distant *origin* (figura 16)

```
user@hp MINGW64 /d/programarea în retea/lab1 (Diana)
$ git pull origin
Already up-to-date.
```

Figura 16 - Monitorizarea schimbărilor

16. Trecem pe ramura principală de dezvoltare *master* figura 17.

- *git ckeckout master*
- Observați conținutul directorului de lucru.
- Opțional puteți modifica directorul de lucru și repozitoriul pe ramura *master*.



```
user@hp MINGW64 /d/programarea în retea/lab1 (Diana)
$ git checkout master
Switched to branch 'master'
Your branch is up-to-date with 'origin/master'.
user@hp MINGW64 /d/programarea în retea/lab1 (master)
$ |
```

Figura 17 - Mergerea la nouă ramură

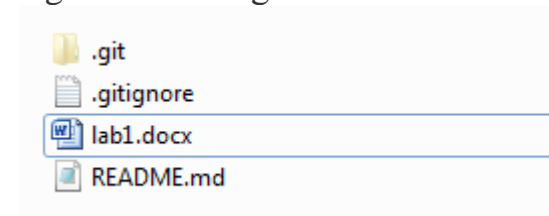


Figura 18. Schimbările apărute în repozitoriu

17. Unificam ramurile *master* și *Diana* figura 19

- *git merge Diana*

```

user@hp MINGW64 /d/programarea in retea/lab1 (master)
$ git merge Diana
Updating e0c04e4..cd166ce
Fast-forward
 hwCPP.cpp | 10 ++++++++
 1 file changed, 10 insertions(+)
 create mode 100644 hwCPP.cpp

```

```

.git
.gitignore
hwCPP.cpp
lab1.docx
README.md

```

Figura 19 - Unificarea ramurilor

18. Reînnoim repozitoriul distant **origin**, ramura **master**

Your recently pushed branches:

🔗 Diana (7 minutes ago) [Compare & pull request](#)

Branch: master ▾ [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download ▾](#)

Diana Init file		Latest commit 256670a 40 minutes ago
.gitignore	Init file	40 minutes ago
README.md	first commit	an hour ago
lab1.docx	Init file	40 minutes ago

Figura 20 - Schimbările in ramura master

```

user@hp MINGW64 /d/programarea in retea/lab1 (master)
$ git push -u origin master
Total 0 (delta 0), reused 0 (delta 0)
To git@github.com:ddonca/programarea-in-retea.git
 e0c04e4..cd166ce master -> master
Branch master set up to track remote branch master from origin.

```

Figura 21 - Adăugăm fișiere în depozitoriul extern

🔗 3 commits 🔗 2 branches 🔗 0 releases 👤 0 contributors

Branch: master ▾ [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download ▾](#)

Diana Adaugam un file cpp		Latest commit fa13f80 11 minutes ago
.gitignore	Init file	41 minutes ago
README.md	first commit	an hour ago
hwC++.cpp	Adaugam un file cpp	11 minutes ago
lab1.docx	Init file	41 minutes ago

Figura 22 - Schimbările survenite in repozitoriu

19. Observați modificările realizate în proiectul localizat în <https://github.com/ddonca/programarea-in-retea.git>

20. Compilăm un fișier cpp in git bash figura 23.

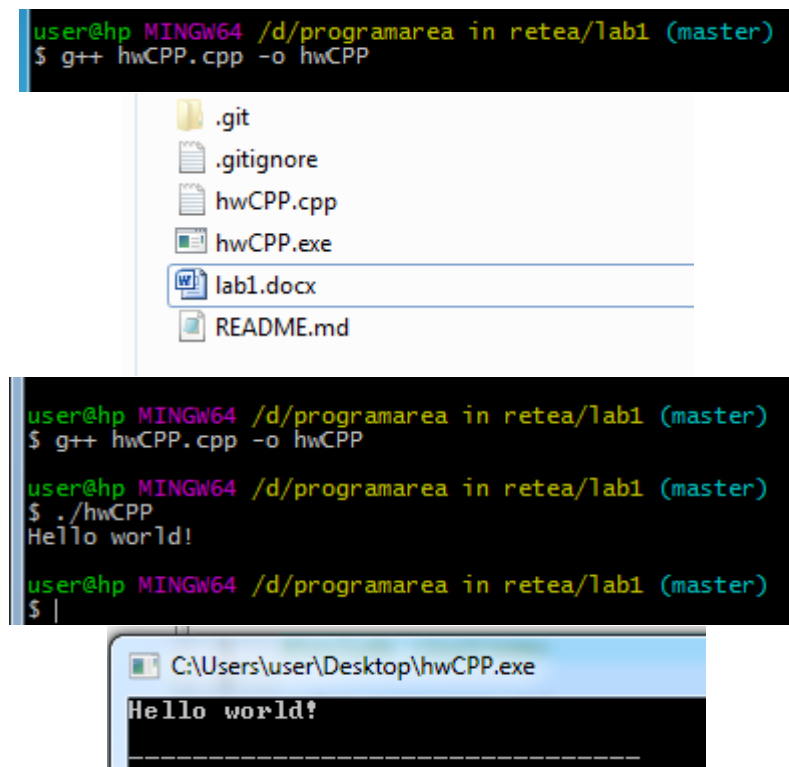


Figura 23 - Compilarea, lansarea în execuție

Construirea proiectelor Java utilizând Apache Maven

Obiectivele lucrării: înțelegerea procesului de construire automată a unui proiect Java, cunoașterea fazelor esențiale Maven; obiectivul specific constând în setarea unui proiect Maven dependent de librării externe localizate pe servere centrale și analiza avantajelor oferite de acest

Crem un depozitoriu visual studio InitProj ca în figura 24 instrument.

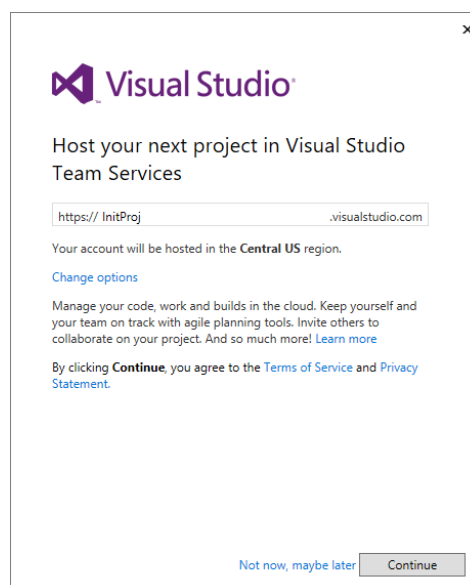


Figura 24 - Crearea unui depozitoriu VS

Creare un proiect comun unde mai mulți v-a avea acces pentru modificări și redactare (figura 25).

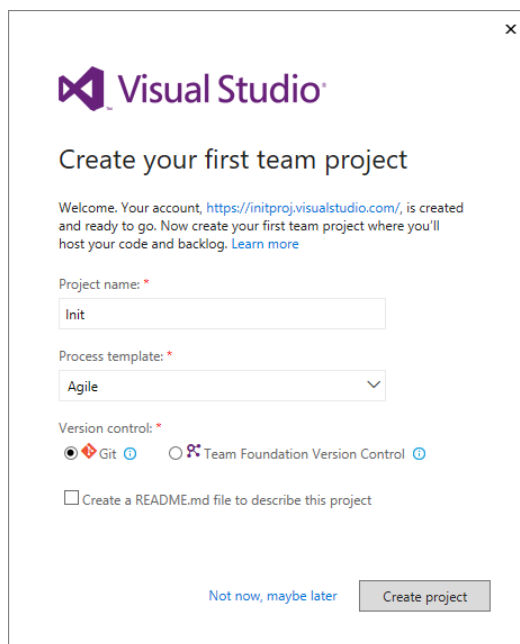
The image shows a dialog box titled 'Visual Studio' with the subtitle 'Create your first team project'. It contains a welcome message and a 'Project name' field with the text 'Init'. Below this is a 'Process template' dropdown menu set to 'Agile'. There are two radio buttons for 'Version control': 'Git' (selected) and 'Team Foundation Version Control'. A checkbox for 'Create a README.md file to describe this project' is unchecked. At the bottom, there are two buttons: 'Not now, maybe later' and 'Create project'.

Figura 25 - Crearea unui proiect

Asemănător putem inițializa un host project in <https://app.vsaex.visualstudio.com> reprezentare în figura 26.

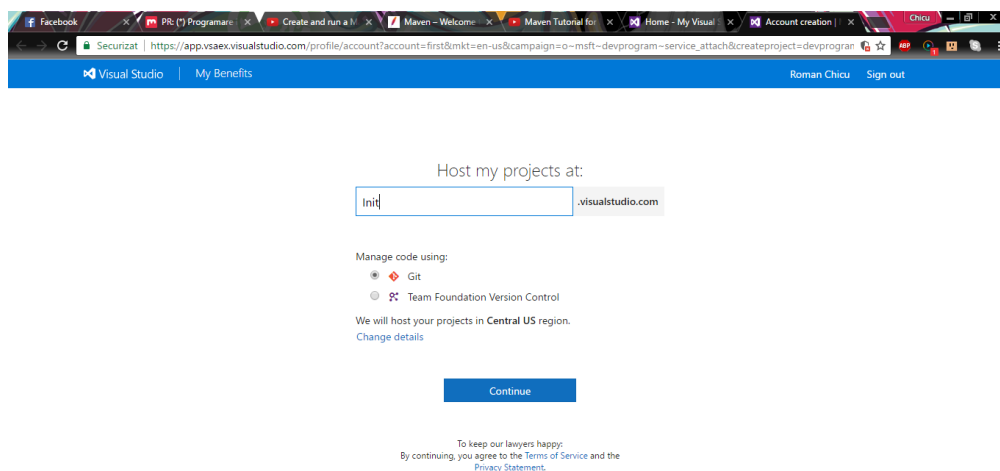
The image shows a web browser window with the URL 'https://app.vsaex.visualstudio.com/profile/account?account=first&mkt=en-us&campaign=o-msft-devprogram-service_attach/createproject=devprogram'. The page has a blue header with 'Visual Studio' and 'My Benefits' links. The main content area has a form titled 'Host my projects at:' with a text input field containing 'Init' and a dropdown menu showing '.visualstudio.com'. Below this, there are two radio buttons for 'Manage code using': 'Git' (selected) and 'Team Foundation Version Control'. A message states 'We will host your projects in Central US region.' with a 'Change details' link. At the bottom, there is a 'Continue' button and a small disclaimer: 'To keep our lawyers happy: By continuing, you agree to the Terms of Service and the Privacy Statement.'

Figura 26 - Crearea unui nume de host project

După inițializare a unui proiect creem un fișier Maven figura 4. Cu reprezentarea repozitoriului și metoda de conlucrare a acestuia figura 27.

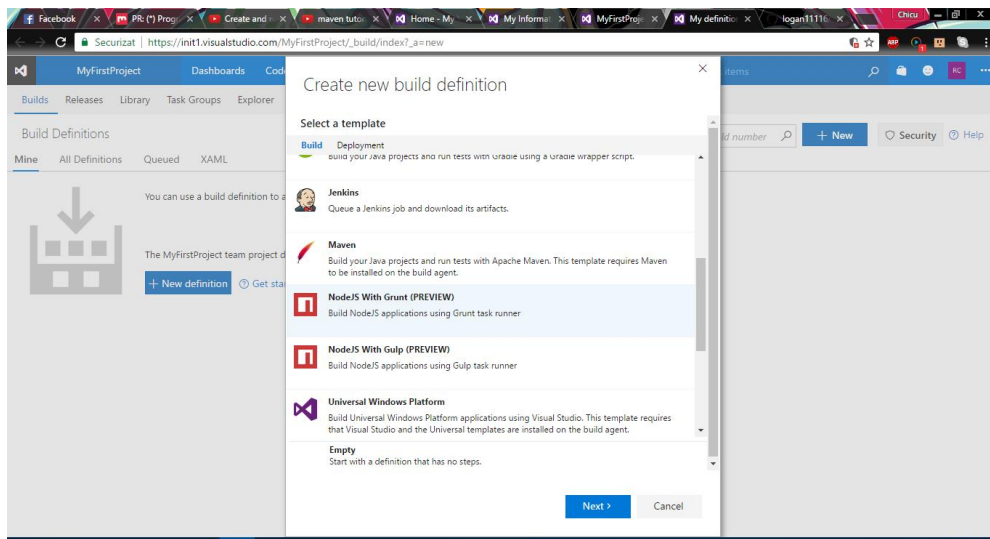


Figura 27 - Creare unui fișier Maven

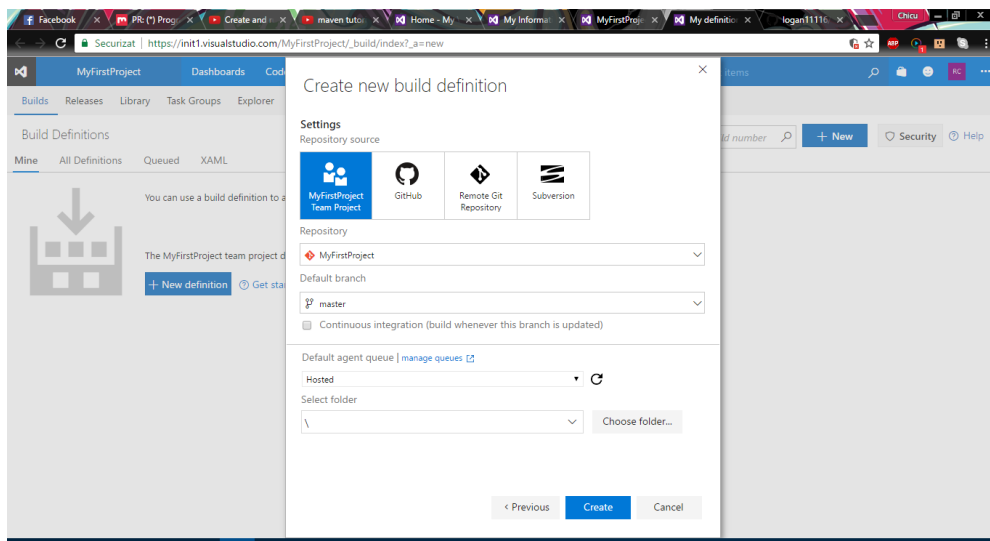


Figura 28 - Init project repository

Configurare unui project Maven se va face în modul reprezentat în figura 29,30 .

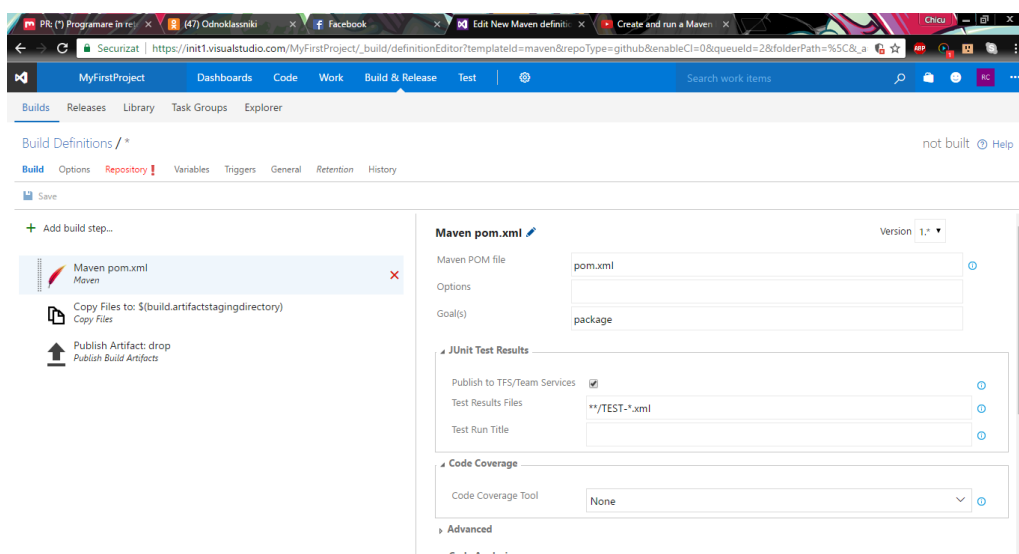


Figura 29 - Configurarea unui Pub. Artifact și Maven file

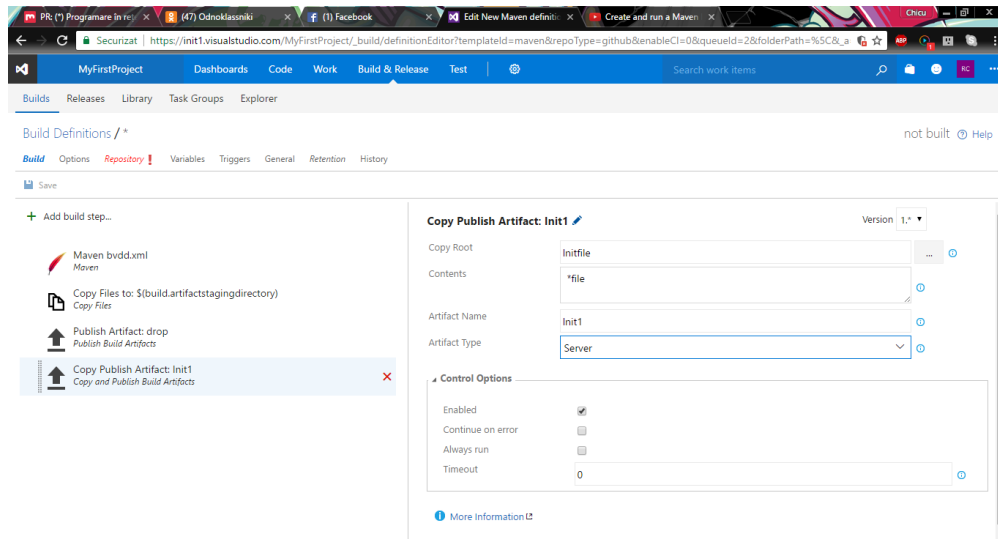


Figura 30 - Crearea unui task

Facem build la project asemănător ca în figura 31.

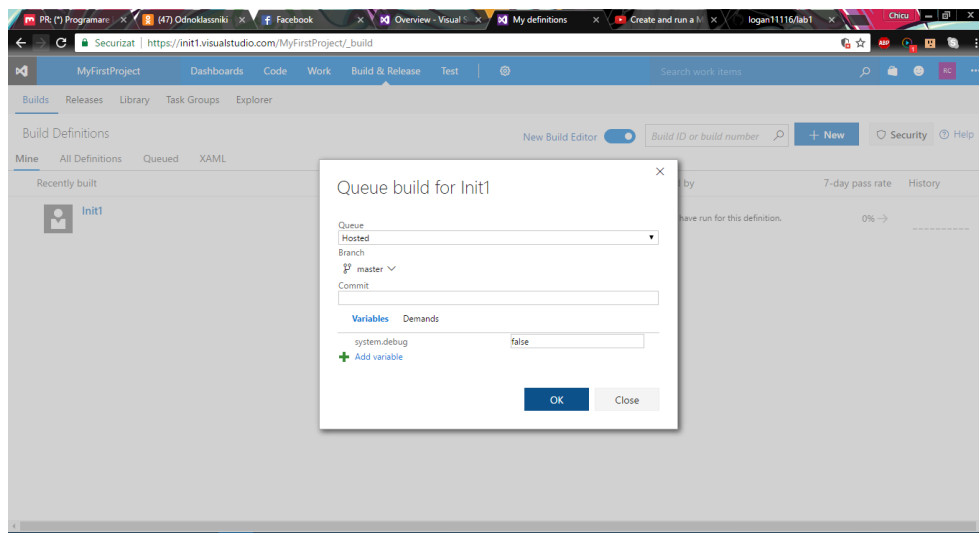


Figura 31 - Build project

În următoarea figură este reprezentată rezultatul buildului proiectului Maven.

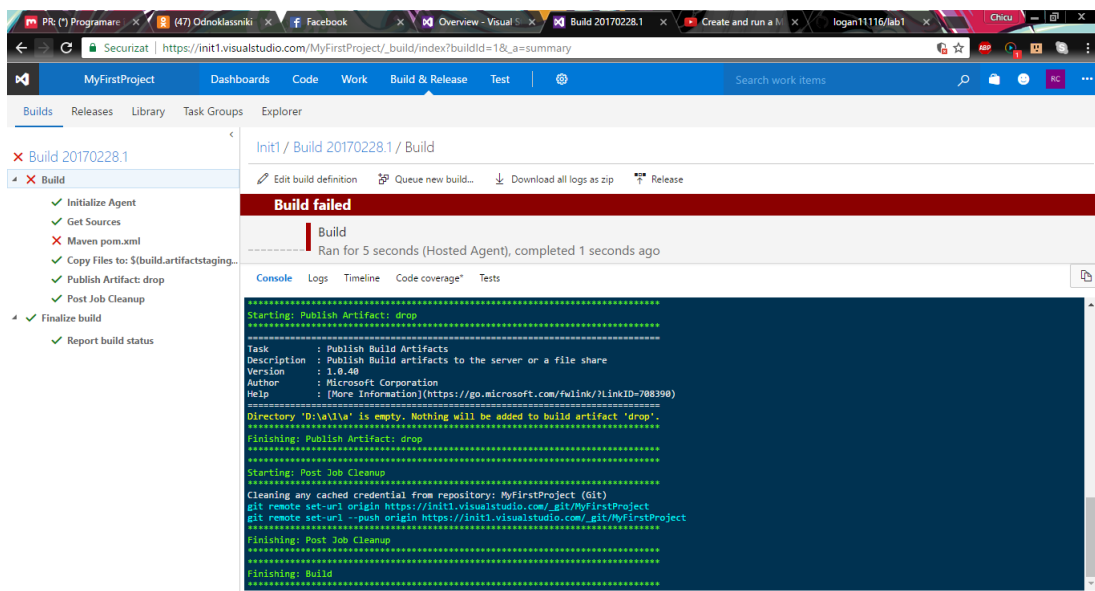


Figura 32 - Rezultatul buildului

Concluzie:

În urma acestei lucrări de laborator au fost obținute abilitați de lucru cu GIT-ul. Au fost obținute abilitați în versionarea codului în GIT. Au fost studiate și înțelese principiile de funcționare și utilizare a sistemului distribuit de control al versiunilor numite GIT.

În aceasta lucrare de laborator a fost creat un repozitor distant, care a fost localizat de serviciul github.com și visualstudio.com. Au fost sincronizate toate modificările efectuate asupra repozitoriului local.

A fost făcut cunoștință su serviciile GIT existente și crearea unui proiect Maven folosind serviciul online Visual Studio.

Bibliografie:

1. Scott Chacon, Pro Git, July 29, 2009 <http://git-scm.com/book>
2. Lars Vogel, Git - Tutorial, actualizat -
14.12.2014, <http://www.vogella.com/tutorials/Git/article.html>
3. Git How To, <http://githowto.com/>
4. Atlassian, Git Tutorials, <https://www.atlassian.com/git/tutorial>
5. Vincent Driessen, A successful Git branching model, January 05, 2010, <http://nvie.com/posts/a-successful-git-branching-model/>
6. [Linux.conf.au 2013] - Git For Ages 4 And Up, Youtube, <https://www.youtube.com/watch?v=1ffBJ4sVUb4>
7. Visualizing Git Concepts with D3, <http://onlywei.github.io/explain-git-with-d3/>
8. tryGit, <https://try.github.io>
9. Learn Git Branching, <http://pcottle.github.io/learnGitBranching/>
10. Code School, Git Real, Free preview, <https://www.codeschool.com/courses/git-real>
11. Code School, Git Real 2, Free preview, <https://www.codeschool.com/courses/git-real-2>