

MINISTERUL EDUCATIEI AL REPUBLICII MOLDOVA

UNIVERSITATEA TEHNICA A MOLDOVEI

FACULTATEA CALCULATORARE INFORMATICA SI MICROELECTRONICA

RAPORT

Lucrarea de laborator nr.1

Disciplina: Medii Interactive de Dezvoltare a Produselor Soft

Tema: Version Control Systems si modul de setare a unui server

A efectuat
st.gr. TI-154

Popusoi Victor

A controlat
lect., asis.

Cojanu Irina

Chisinau 2017

1 Scopul lucrarii de laborator

Insusirea notiunii de Version Control Systems si a modului de setare a unui server.

2 Obiectivele lucrarii de laborator

Version Control Systems (git - bitbucket - mercurial - svn)

3 Sarcina lucrarii de laborator

Sa se studieze sistemul de control al versiunilor Git. Sa se realizeze un proiect in repozitoriul local. Proiectul sa se incarce in repozitoriul GitHub. Sa se efectueze diferite modificari.

4 Efectuarea lucrarii de laborator

4.1 Sarcinile propuse pentru efectuare lucrarii de laborator

Basic Level (nota 5 - 6) :

- initializeaza un nou repozitoriu
- configureaza-ti VCS
- crearea branch-urilor (creeaza cel putin 2 branches)
- commit pe ambele branch-uri (cel putin 1 commit per branch)

Normal Level (nota 7 - 8):

- seteaza un branch to track a remote origin pe care vei putea sa faci push (ex. Github, Bitbucket or custom server)
- reseteaza un branch la commit-ul anterior salvarea temporara a schimbarilor care nu se vor face commit imediat.
- folosirea fisierului .gitignore

Advanced Level (nota 9 - 10):

- merge 2 branches
- rezolvarea conflictelor a 2 branches
- comenzile git care trebuie cunoscute

4.2 Realizarea lucrarii de laborator

Link-ul de la repozitoriu [here](#)

Basic Level (nota 5 - 6) :

- Primul pas in executarea acestei lucrari de laborator a fost crearea unui repozitoriu, apasind butonul New de pe pagina utilizatorului, tab-ul cu denumirea Repositories.

Dupa setarea numelui pentru repository, s-a apasat create repository. (Screen here 4.1)

- Configurarea VCS. S-a creata un ssh key si s-a copiat in lista de key in account-ul github. Apoi s-a clonat repositoryul local in urma caruia s-a utilizat comenzile:

`git clone git@github.com:PopusoiVictor/MIDPS.git` (Screen here 4.2 and 4.3)

- S-a configurat config-ul la git prin intermediul comenzilor `git config user.name "PopusoiVictor"` si `git config user.email "v.p.130796@gmail.com"` (Screen here 4.4)

- Au fost create 2 branchuri (Screen here 4.5), comenzile care au fost folosite pentru crearea branchurilor: `git checkout -b 'denumire branchului'` si `git checkout -b 'denumire branchului'`. Apoi cu ajutorul comenzii `git push origin 'denumirea branchului'` au fost incarcate pe github. (Screen here 4.6 and 4.7)

- Apoi s-a realizat cate un commit pentru fiecare din branchurile create (Screen here 4.8 and 4.9) folosind comenzile:

`git checkout 'denumirea branchului'` (pentru a schimba ramificarea directorului git.)

`git add .` (adauga fisiere la commit)

`git commit -m "descriere"` (salveaza schimbarile in head)

`git push origin 'denumirea branchului'` (Trimite/publica ramificare curenta.)

Normal Level (nota 7 - 8):

- S-a setat un branch to track pe care s-a facut push (Screen here 4.10) cu comenzile: `git checkout -track -b new origin/master` (seteaza un branch to track)

`git add .` (adauga fisiere la commit)

`git commit -m "add new branch"` (salveaza schimbarile in head)

`git push origin new` (Trimite/publica ramificare curenta.)

- S-a resetat branchul branch two la commitul anterior (Screen here 4.11) cu ajutorul comenzilor: `git log --graph --all --oneline` (pentru a vedea evolutia directorului git.)

`git reset --hard` (Reinitializeaza indexul si directorul de lucru la starea ultimului commit)

- S-a adaugat un fisier in .gitignore ce a facut comenzile git sa nu il considere ca un fisier din proiect. (Screen here 4.12). Dupa cum se observa in screen ca dupa ce am creat fisierul Ignore.lib si apoi am selectat comanda 'git status' a aparut mesajul ca nu a fost facut nici o modificare in repository, ceea ce inseamna ca fisierul a fost ignorat. Comenzile folosite:

`echo "Ignore" Ignore.lib` (Creaza un fisier sau inscrie ceva in fisier)

`git status` (arata starea fisierelor)

Advanced Level (nota 9 - 10 :

- S-a facut merge la 2 branchuri cu ajutorul comenzii `git merge` si denumirea la branch-ul cu care s-a dorit sa se faca merge. (Screen here 4.13)

- S-a creat un conflict intre 2 branchuri, schimbând acelasi fisier apoi am facut 'git merge master' ce a generat un conflict. (Screen here 4.14) . Fisierul unde sa produs conflictul (Screen here 4.15). Rezolvarea conflictelor a 2 branchuri s-a facut prin selectarea manuala a informatiei care trebuie sa ramana, si stergerea celei inutile sau care nu mai este necesara.(Screen here 4.16) Dupa care s-a facut commit si conflictul a fost inlaturat. (Screen here 4.17)
- S-a facut cunostinta cu majoritatea comenzilor Git care au fost oferite de catre profesor.
<https://www.siteground.com/tutorials/git/commands.htm>
- S-a creat un tag nou (Screen here 4.18)

4.3 Imagini

9 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

File/Folder	Commit Message	Time Ago
PopusoiVictor README modified	Latest commit 82dfcc4	an hour ago
Laborator 1	README modified	an hour ago
Laborator 2	Add folders	11 days ago
Laborator 3	Add folders	11 days ago
Laborator 4	Add folders	11 days ago
Laborator 5	Add folders	11 days ago
.gitignore	Create .gitignore	12 days ago
README.md	README modified	an hour ago
hello	first commit	12 days ago

README.md

MIDPS

Lucrari de laborator la disciplina Medii Interactive de Dezvoltare a Produselor Soft

Studentul grupei TI-154 Popusoi Victor

Lucrarea de laborator nr.1


Pentru a merge la continutul lucrarii de laborator nr.1 tastati aici -> [here](#)

Figure 4.1– Repozitoriul

SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

 **GitHub Desktop - Admin-PC**
Fingerprint: 68:11:74:f7:07:62:c3:1f:c4:e1:bc:36:23:4d:fa:67
Added on Feb 21, 2017 — Never used Delete


 **Acer Aspire E5-511**
Fingerprint: 42:73:13:32:8c:70:2a:ab:74:0e:2c:37:2b:ed:92:0d
Added on Feb 24, 2017 — Last used within the last day Delete

Figure 4.2– ssh-Key

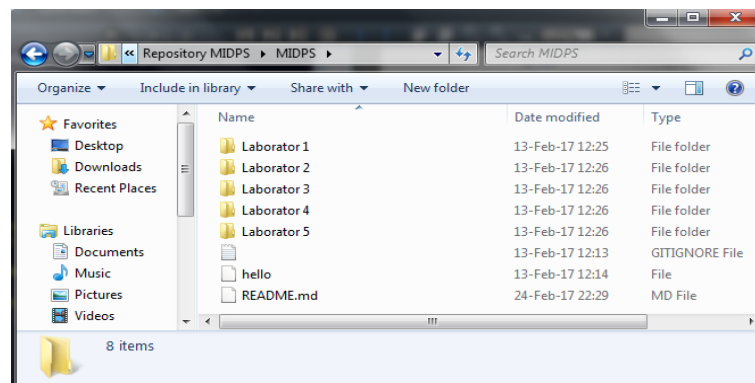


Figure 4.3– Repozitoriul local

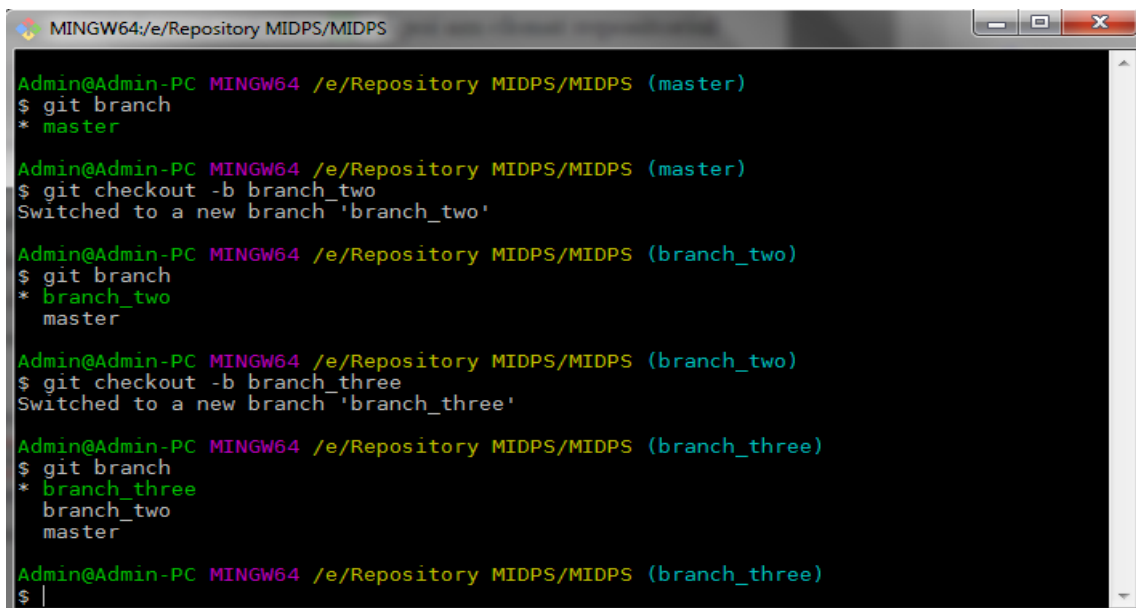
```
Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (master)
$ git config user.name "PopusoiVictor"

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (master)
$ git config user.email "v.p.130796@gmail.com"

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (master)
$ git config --list
core.symlinks=false
core.autocrlf=true
core.fscache=true
color.diff=auto
color.status=auto
color.branch=auto
color.interactive=true
help.format=html
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
diff.astextplain.textconv=astextplain
rebase.autosquash=true
credential.helper=manager
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
user.name=PopusoiVictor
user.email=v.p.130796@gmail.com
core.repositoryformatversion=0
core.filemode=false
core.bare=false
core.logallrefupdates=true
core.symlinks=false
core.ignorecase=true
remote.origin.url=git@github.com:PopusoiVictor/MIDPS.git
remote.origin.fetch=+refs/heads/*:refs/remotes/origin/*
branch.master.remote=origin
branch.master.merge=refs/heads/master
user.email=v.p.130796@gmail.com
user.name=PopusoiVictor

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (master)
```

Figure 4.4– Config-ul



```
Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (master)
$ git branch
* master

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (master)
$ git checkout -b branch_two
Switched to a new branch 'branch_two'

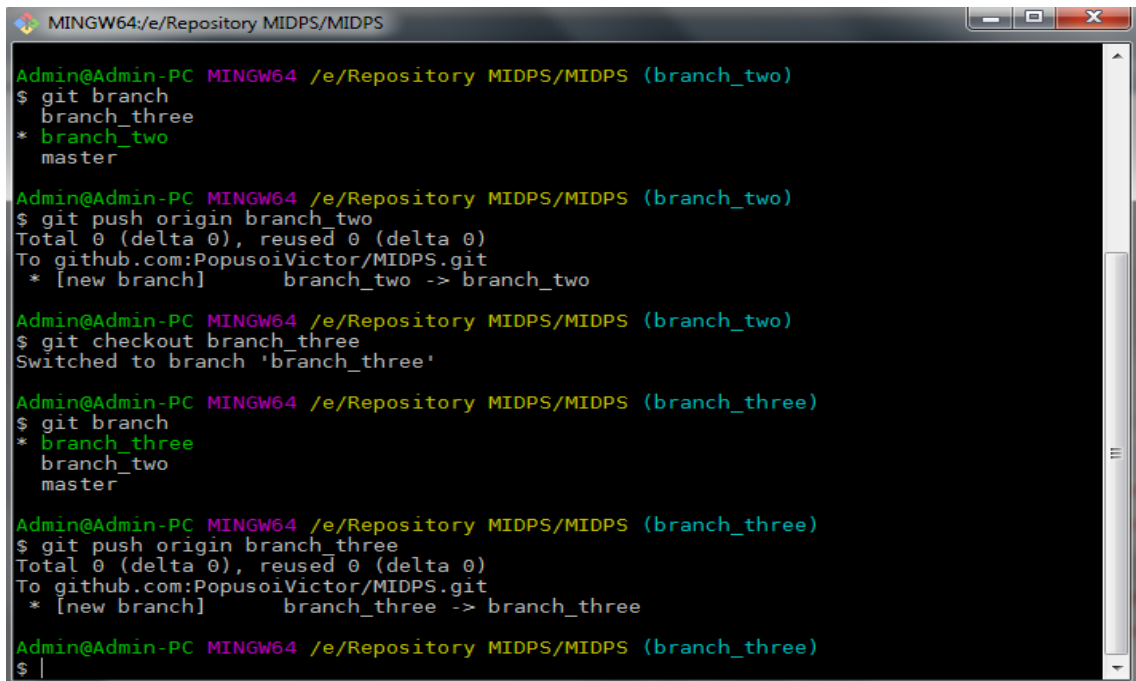
Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_two)
$ git branch
* branch_two
  master

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_two)
$ git checkout -b branch_three
Switched to a new branch 'branch_three'

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_three)
$ git branch
* branch_three
  branch_two
  master

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_three)
$ |
```

Figure 4.5– Crearea a 2 branch-uri



```
Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_two)
$ git branch
  branch_three
* branch_two
  master

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_two)
$ git push origin branch_two
Total 0 (delta 0), reused 0 (delta 0)
To github.com:PopusoiVictor/MIDPS.git
 * [new branch]      branch_two -> branch_two

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_two)
$ git checkout branch_three
Switched to branch 'branch_three'

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_three)
$ git branch
* branch_three
  branch_two
  master

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_three)
$ git push origin branch_three
Total 0 (delta 0), reused 0 (delta 0)
To github.com:PopusoiVictor/MIDPS.git
 * [new branch]      branch_three -> branch_three

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_three)
$ |
```

Figure 4.6– Upload branch

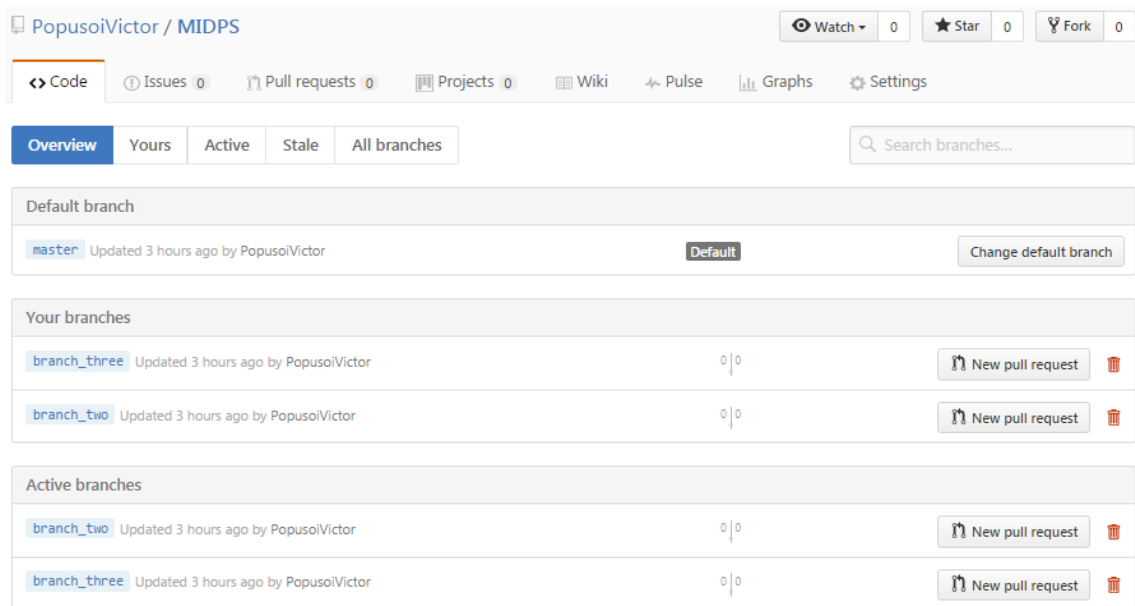


Figure 4.7– Branches

```

MINGW64/e/Repository MIDPS/MIDPS
Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_two)
$ git add .

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_two)
$ git status
On branch branch_two
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   branch_two.txt

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_two)
$ git commit -m "created new branch"
[branch_two 8ddd410] created new branch
1 file changed, 1 insertion(+)
create mode 100644 branch_two.txt

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_two)
$ git push origin branch_two
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 294 bytes | 0 bytes/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local objects.
To github.com:PopusoiVictor/MIDPS.git
  82dfcc4..8ddd410  branch_two -> branch_two

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_two)
$

```

Figure 4.8– Commit on branch two

```
MINGW64:/e/Repository MIDPS/MIDPS
Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_two)
$ git branch
  branch_three
* branch_two
  master

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_two)
$ git checkout branch_three
Switched to branch 'branch_three'

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_three)
$ git status
On branch branch_three
Untracked files:
  (use "git add <file>..." to include in what will be committed)

        branch_three.txt

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

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_three)
$ git add .

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_three)
$ git commit -m "commit on branch_three"
[branch_three 312544e] commit on branch_three
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 branch_three.txt

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_three)
$ git push origin branch_three
Counting objects: 2, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), 280 bytes | 0 bytes/s, done.
Total 2 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local objects.
To github.com:PopusoiVictor/MIDPS.git
   82dfcc4..312544e  branch_three -> branch_three

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_three)
$ |
```

Figure 4.9 – Commit on branch three

```
MINGW64:/e/Repository MIDPS/MIDPS
Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (branch_three)
$ git checkout --track -b new origin/master
Branch new set up to track remote branch master from origin.
Switched to a new branch 'new'

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (new)
$ git add .

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (new)
$ git commit -m "add new branch"
On branch new
Your branch is up-to-date with 'origin/master'.
nothing to commit, working tree clean

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (new)
$ git push origin new
Total 0 (delta 0), reused 0 (delta 0)
To github.com:PopusoiVictor/MIDPS.git
 * [new branch]      new -> new

Admin@Admin-PC MINGW64 /e/Repository MIDPS/MIDPS (new)
$ |
```

Figure 4.10 – Set branch to track


```

MINGW64:/e/MIDPS/MIDPS

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (master)
$ git stash
Saved working directory and index state WIP on master: 6fb7c00 reset
HEAD is now at 6fb7c00 reset

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (master)
$ git add .

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (master)
$ git commit -m "reset"
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
nothing to commit, working tree clean

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (master)
$ git log --graph --all --oneline
* ae28c61 WIP on master: 6fb7c00 reset
| \
|  * ff55aa1 index on master: 6fb7c00 reset
| /
|
* 6fb7c00 reset
* ecd4354 Modified .gitignore
* e3300e2 Ignored file
* 9c04a73 ignore file
* 4fd3dd6 File modified
* 222e640 gitignore modified
| * 312544e commit on branch_three
| /
| * 8ddd410 created new branch
| /
* 82dfcc4 README modified
* 1dda085 README modified
* e0a9030 README modified
* 7d8153b README modify
* b323bac added modified readme file, bugfix: #48
* 5ee3878 Add folders
* 36ccb7a first commit
* eeffea73 Create .gitignore
* d663f28 Initial commit

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (master)
$ git reset --hard HEAD
HEAD is now at 6fb7c00 reset

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)
nothing to commit, working tree clean

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (master)
$ git reset HEAD~

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (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)

        reset.txt

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

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (master)
$ |

```

Figure 4.11 – Reset branch to last commit

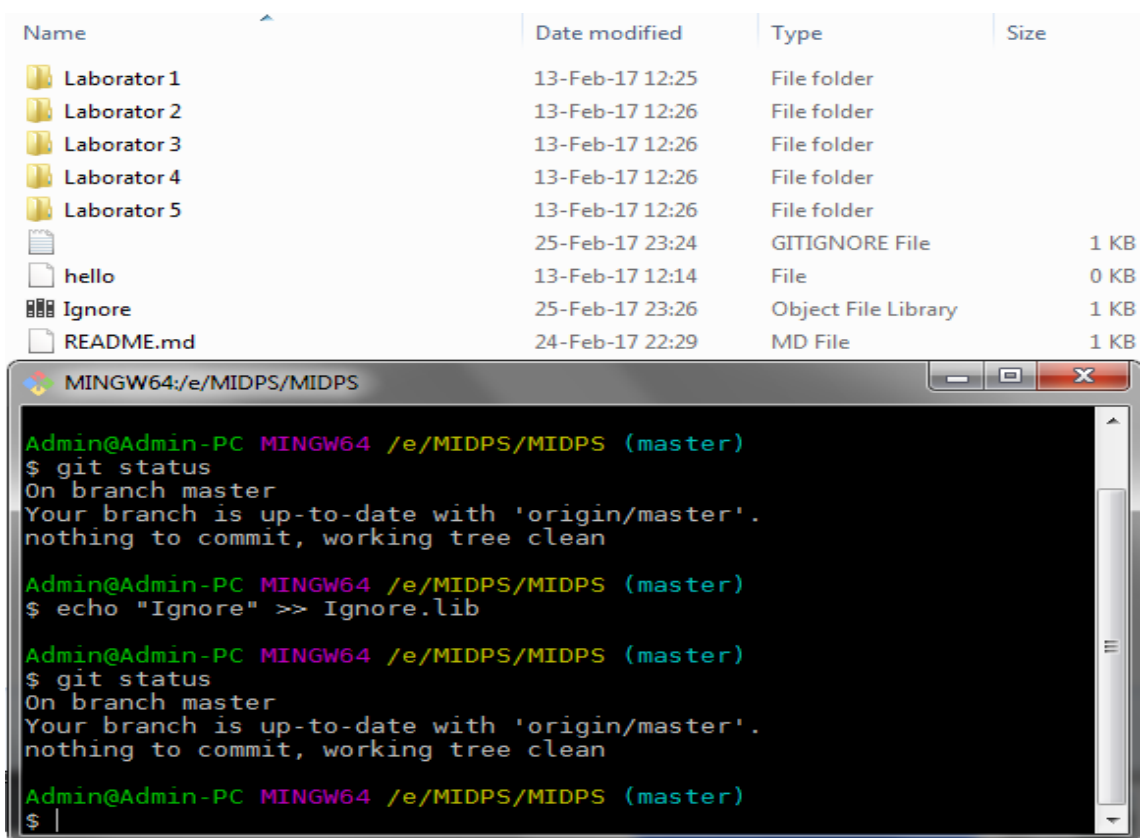


Figure 4.12– Ignore file

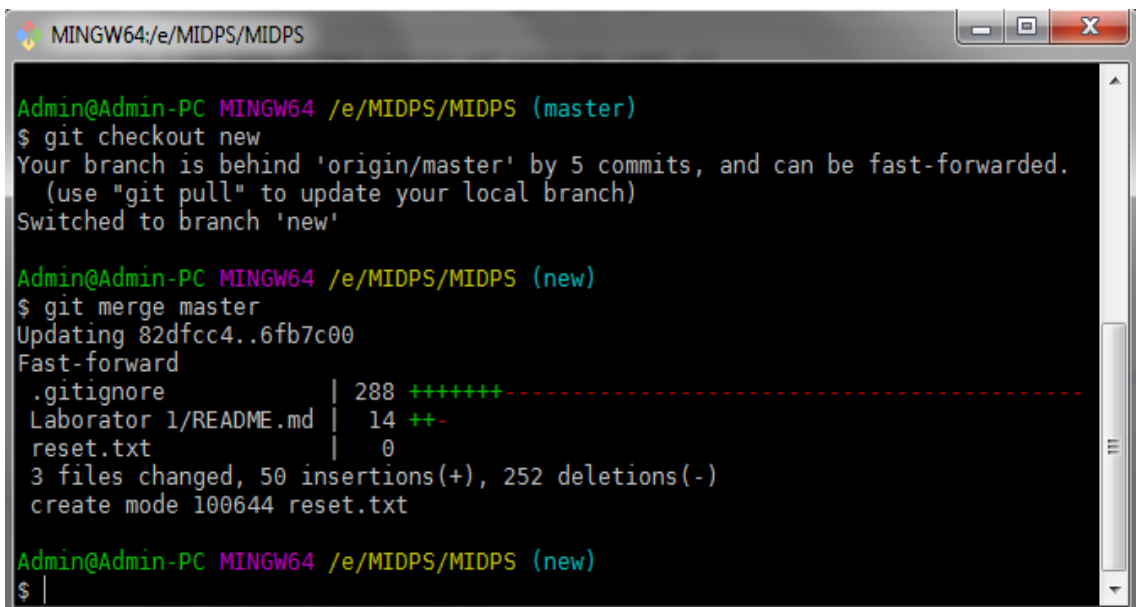


Figure 4.13– Merge 2 branches

```

MINGW64:/e/MIDPS/MIDPS
Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (master)
$ git add .

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (master)
$ git commit -m "README modified"
[master e18f9fe] README modified
1 file changed, 2 insertions(+), 2 deletions(-)

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (master)
$ git checkout branch_three
Switched to branch 'branch_three'

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (branch_three)
$ git status
On branch branch_three
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (branch_three)
$ git add .

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (branch_three)
$ git commit -m "README modified"
[branch_three 47abbe9] README modified
1 file changed, 1 insertion(+), 1 deletion(-)

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (branch_three)
$ git checkout new
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)
Switched to branch 'new'

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (new)
$ git status
On branch new
Your branch is ahead of 'origin/master' by 2 commits.
  (use "git push" to publish your local commits)
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (new)
$ git add .

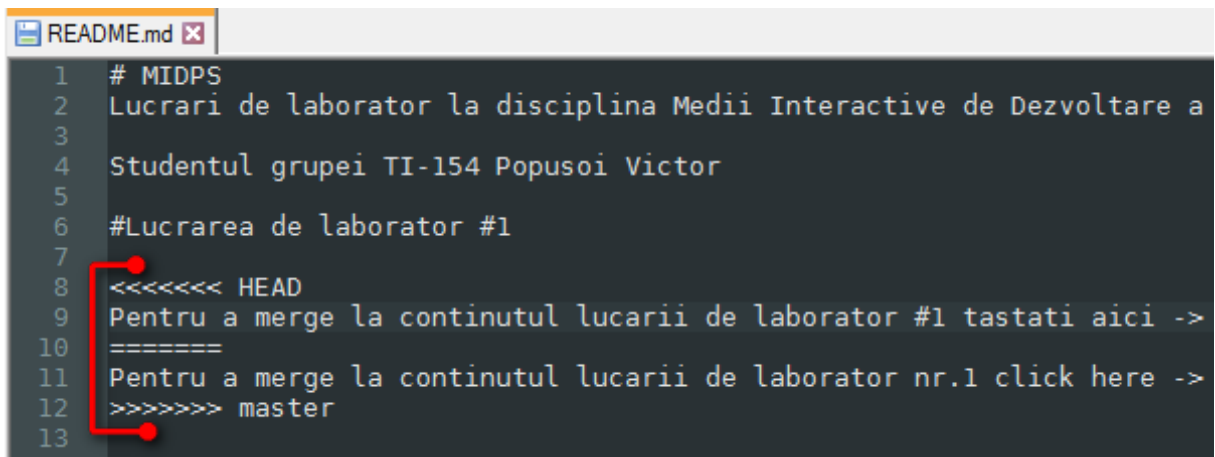
Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (new)
$ git commit -m "README modified"
[new 84028bb] README modified
1 file changed, 2 insertions(+), 2 deletions(-)

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (new)
$ git merge master
Auto-merging README.md
CONFLICT (content): Merge conflict in README.md
Automatic merge failed; fix conflicts and then commit the result.

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (new|MERGING)
$ |

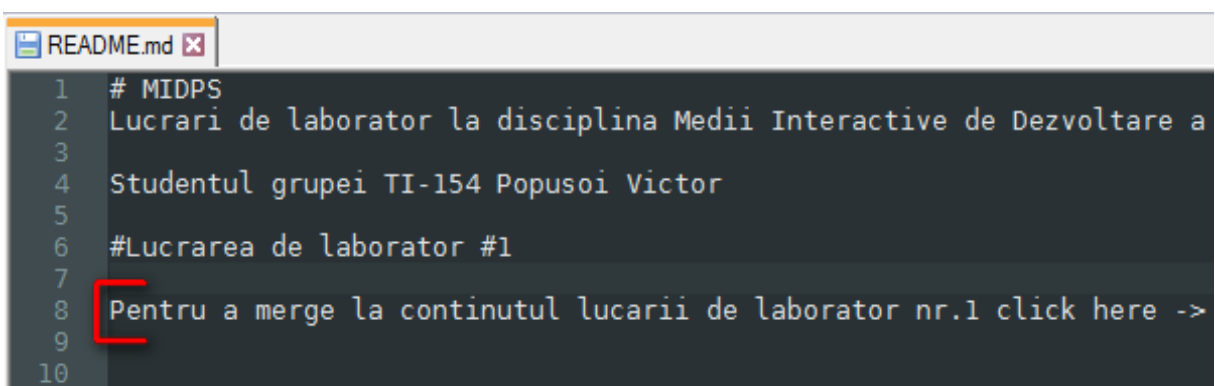
```

Figure 4.14– S-a creat un conflict



```
1 # MIDPS
2 Lucrari de laborator la disciplina Medii Interactive de Dezvoltare a
3
4 Studentul grupei TI-154 Popusoi Victor
5
6 #Lucrarea de laborator #1
7
8 <<<<<< HEAD
9 Pentru a merge la continutul lucarii de laborator #1 tastati aici ->
10 =====
11 Pentru a merge la continutul lucarii de laborator nr.1 click here ->
12 >>>>>> master
13
```

Figure 4.15– Conflictul in fisierul README



```
1 # MIDPS
2 Lucrari de laborator la disciplina Medii Interactive de Dezvoltare a
3
4 Studentul grupei TI-154 Popusoi Victor
5
6 #Lucrarea de laborator #1
7
8 Pentru a merge la continutul lucarii de laborator nr.1 click here ->
9
10
```

Figure 4.16– Conflictul a fost inlaturat

```
Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (new|MERGING)
$ git status
On branch new
Your branch is ahead of 'origin/master' by 3 commits.
  (use "git push" to publish your local commits)
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)

        both modified:   README.md

no changes added to commit (use "git add" and/or "git commit -a")

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (new|MERGING)
$ git add .

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (new|MERGING)
$ git commit -m "Conflict fixed"
[new 523dc44] Conflict fixed

Admin@Admin-PC MINGW64 /e/MIDPS/MIDPS (new)
$ |
```

Figure 4.17– Dupa commit conflictul a fost rezolvat

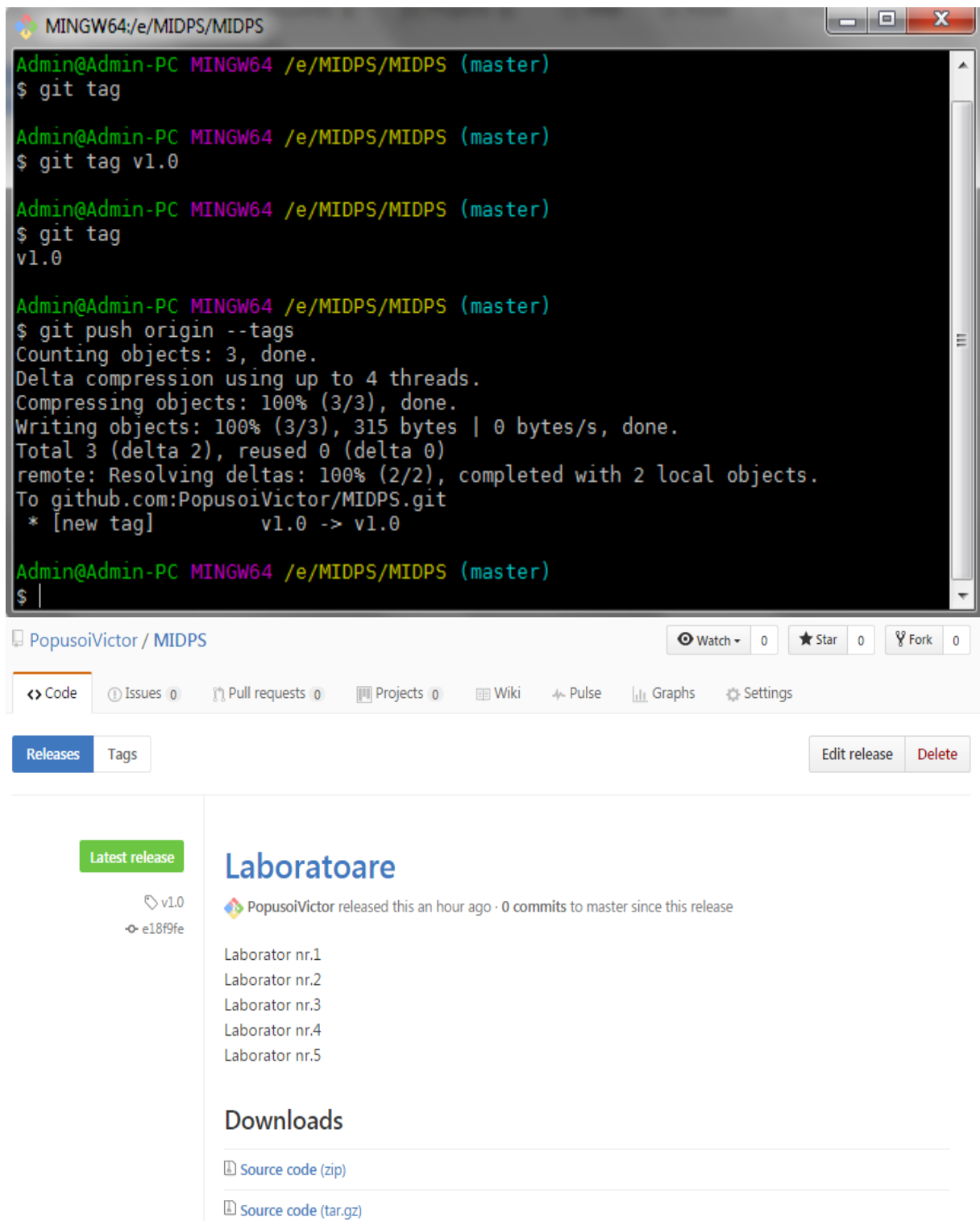


Figure 4.18– S-a creat un tag

Concluzie

În urma efectuării lucrării de laborator nr.1, a fost studiat sistemul de control al versiunilor, Git, particularitățile acestuia, comenzile bash generale, s-a creat noi branch-uri prin terminalul git bash.

S-a încărcat în branch-uri fișiere cu ajutorul instrucțiunii git commit pe repositoryul creat MIDPS, în același timp a fost studiată noțiunea de version control system (este o categorie de instrumente software pentru a ajuta grupurile care se ocupă de software să poată dirija cu schimbările în codurile sale în orice timp la dorința acestora).

Au fost create și modificate repositorye locale și publice. S-a concluzionat faptul că Git-ul este un sistem necesar oricărui programator și developer, întrucât permite dezvoltarea și structurarea pas cu pas a proiectelor și versionarea fișierelor. Un posibil neajuns al sistemului este logica complicată a modului de structurare al informației (dezvoltarea pe branch-uri). Au fost obținute noi cunoștințe în domeniul mediilor interactive de dezvoltare a produselor soft.

Bibliografie

1. Repozitoriul public GitHub:

<https://github.com/PopusoiVictor/MIDPS>

2. Tutorial Git:

<http://all.webng.md/traduceri/everyday-ro.html>