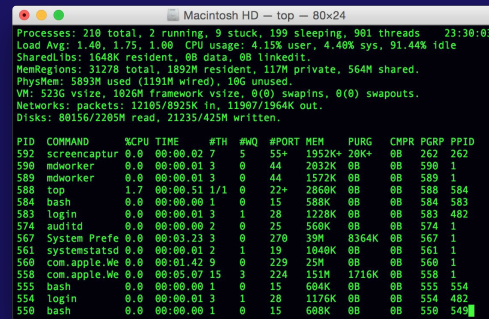




Terminal, Git & Github

TERMINAL

Funcția unui terminal este de obicei limitată la transcrierea și introducerea datelor. Un PC poate rula un emulator de terminal care reproduce funcțiile unui terminal din lumea reală, permițând uneori utilizarea concomitentă a programelor locale și accesul la un sistem gazdă terminal îndepărtat.



```
Macintosh HD — top — 80x24
Processes: 210 total, 2 running, 9 stuck, 199 sleeping, 901 threads 23:30:03
Load Avg: 1.40, 1.75, 1.00 CPU usage: 4.15% user, 4.40% sys, 91.44% idle
Shared libs: 1640K resident, 0B data, 0B linkedit.
MemRegions: 31278 total, 1892M resident, 117M private, 564M shared.
PhysMem: 5893M used (1191M wired), 10G unused.
VM: 523G vszize, 1026M framework vszize, 0(0) swapins, 0(0) swapouts.
Networks: packets: 12105/6923K in, 11907/1964K out.
Disks: 80156/2205M read, 21235/425M written.
```

PID	COMMAND	%CPU	TIME	#TH	#MQ	#PORT	MEM	PURG	CMPR	PRGR	PPID
592	screencaptur	0.0	00:00:02	7	5	55+	1952K+	20K+	0B	262	262
590	mdworker	0.0	00:00:01	3	0	44	2032K	0B	0B	590	1
589	mdworker	0.0	00:00:01	3	0	44	1572K	0B	0B	589	1
588	top	1.7	00:00:51	1/1	0	22+	2860K	0B	0B	588	584
584	bash	0.0	00:00:00	1	0	15	509K	0B	0B	584	583
583	login	0.0	00:00:01	3	1	20	1220K	0B	0B	583	402
574	auditd	0.0	00:00:00	2	0	25	560K	0B	0B	574	1
567	System Prefe	0.0	00:03:23	3	0	270	39M	8364K	0B	567	1
561	systemstatsd	0.0	00:00:01	2	1	19	1040K	0B	0B	561	1
560	com.apple.Me	0.0	00:01:42	9	0	220	25M	0B	0B	560	1
558	com.apple.Me	0.0	00:05:07	15	3	224	151M	1716K	0B	558	1
555	bash	0.0	00:00:00	1	0	15	604K	0B	0B	555	554
554	login	0.0	00:00:01	3	1	20	1176K	0B	0B	554	482
550	bash	0.0	00:00:00	1	0	15	600K	0B	0B	550	549

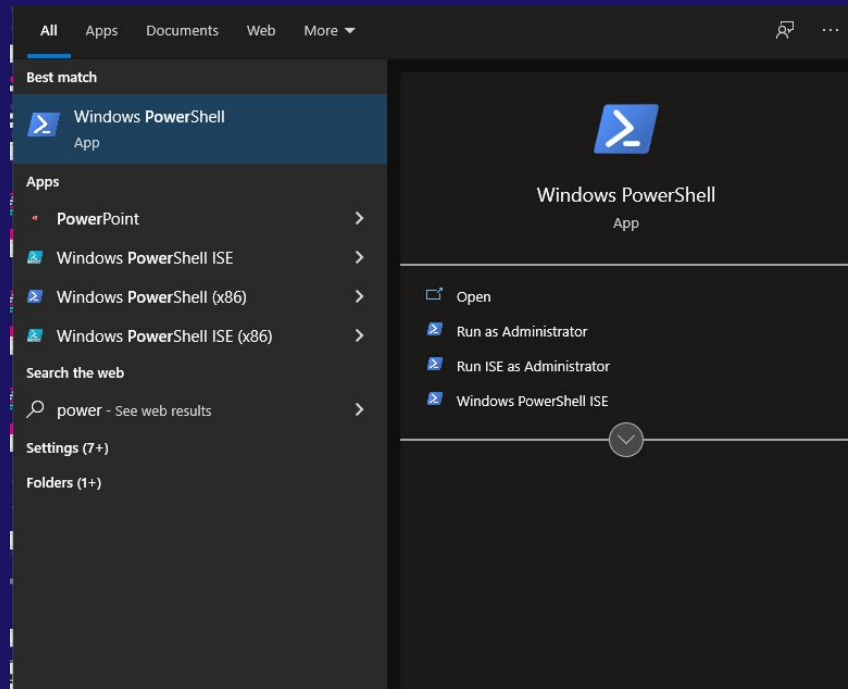


“CLI – Command line interface”

CLI Tools i.e ‘virtualenv’ ‘tree’

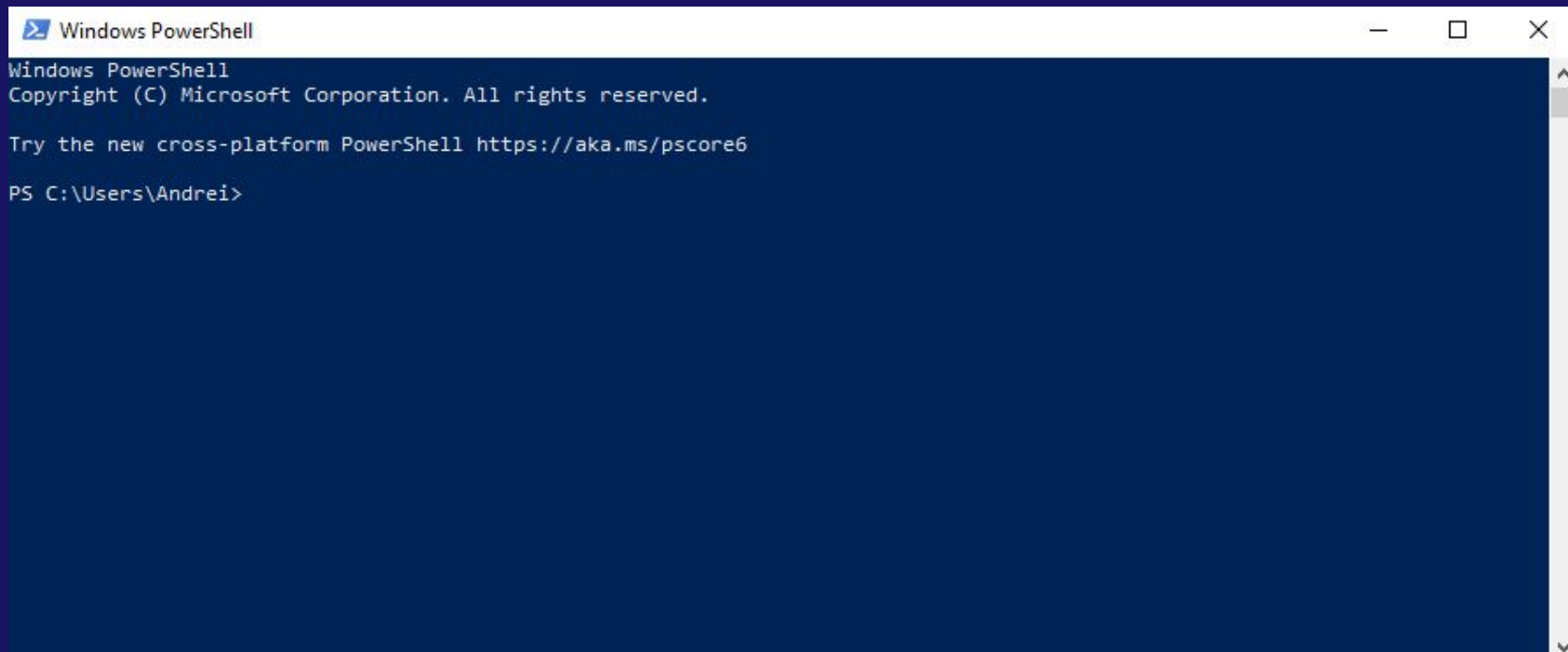
CUM FOLOSIM TERMINALUL?

Pentru acest exercitiu, vom apasa tasta de Windows iar apoi vom scrie PowerShell



CUM FOLOSIM TERMINALUL?

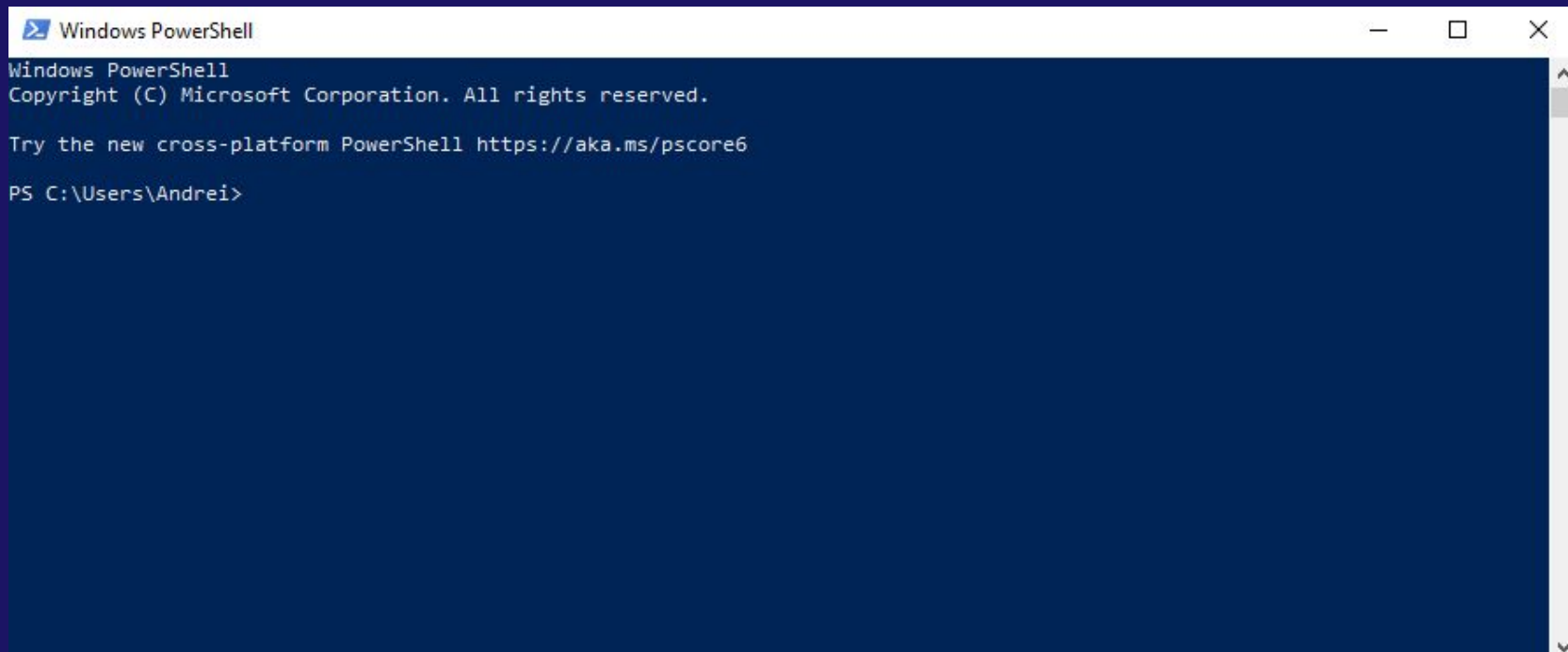
Odata deschis, vom avea in fata urmatoarea fereastra

A screenshot of a Windows PowerShell terminal window. The window has a title bar with the text "Windows PowerShell" and standard Windows window controls (minimize, maximize, close). The terminal content is as follows:

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
  
PS C:\Users\Andrei>
```

CUM FOLOSIM TERMINALUL?

Introducem comanda **ls** si observam ce se intampla acum

A screenshot of a Windows PowerShell terminal window. The window has a title bar with the text "Windows PowerShell" and standard minimize, maximize, and close buttons. The terminal content is as follows:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Andrei>
```

CUM FOLOSIM TERMINALUL?

Vom observa ca ni s-a facut afisarea tuturor fisierelor si folderelor aflate in locatia de baza

```
Windows PowerShell
#Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Andrei> ls

Directory: C:\Users\Andrei

Mode                LastWriteTime         Length Name
----                -
d-----        23-Jan-21   3:27 PM             .Albion Online Launcher
d-----        28-Jan-21   1:04 PM             .android
d-----        14-Nov-21   6:40 PM             .installanywhere
d-----        15-Jul-21   7:36 PM             .jdk8
d-----        14-Feb-21   5:47 PM             .junique
d-----        09-Feb-21   3:02 PM             .m2
d-----        06-Feb-21  10:29 PM             .oracle_jre_usage
d-----        12-Mar-21   4:35 PM             .pylint.d
d-----        23-Jan-21   3:27 PM             .QtWebEngineProcess
d-----        09-Nov-21   9:56 PM             .VirtualBox
d-----        16-Jan-21   5:42 PM             .vscode
d-----        16-Jan-21   5:16 PM             3D Objects
d-----        31-Jul-21   4:06 PM             ansel
d-----        16-Jan-21   5:16 PM             Contacts
d-----        01-Dec-21  11:59 AM             Desktop
d-----        29-Nov-21   9:55 PM             Documents
d-----        02-Dec-21   2:44 PM             Downloads
d-----        16-Jan-21   5:16 PM             Favorites
d-----        28-Jan-21   1:05 PM             IdeaProjects
d-----        16-Jan-21   5:16 PM             Links
d-----        30-Oct-21  10:07 PM             Music
d-----        14-Aug-21  12:35 PM             OneDrive
d-----        23-Nov-21   8:54 PM             Pictures
d-----        13-Nov-21  10:36 AM             Postman
d-----        25-Nov-21  10:42 PM             Saved Games
d-----        16-Jan-21   5:18 PM             Searches
d-----        03-Dec-21   9:48 AM             Videos
d-----        06-Nov-21   3:05 PM             VirtualBox VMs
d-----        21-Nov-21   8:19 PM             870 .bash_history
d-----        26-Feb-21   5:05 PM             108 .gitconfig
d-----        30-Oct-21  11:30 AM             0 index.html
d-----        06-Nov-21  11:07 AM             499 main.java
d-----        13-Nov-21  11:59 AM             340 main.json
d-----        12-Nov-21   9:29 PM             124 main.py
d-----        01-Jul-21   2:45 PM             3188 Quiz.json

PS C:\Users\Andrei>
```

CUM FOLOSIM TERMINALUL?

Pentru a naviga, putem folosi comanda de **cd** care vine de la 'change directory' urmata de folderul in care vrem sa mergem din lista de mai sus. In cazul de fata, am ales folderul de Desktop

```
PS C:\Users\Andrei> cd Desktop
```


CUM FOLOSIM TERMINALUL?

Odata ajunsi in folderul de Desktop, putem executa inca o data comanda de **ls** si vom observa ca lista noastra arata diferit.

```
PS C:\Users\Andrei\Desktop> ls

Directory: C:\Users\Andrei\Desktop

Mode                LastWriteTime         Length Name
----                -
d-----          13-Nov-21   12:23 PM             ClassesPractice
d-----          01-Oct-21    6:35 PM             Demo-Azimut
d-----          08-Nov-21    5:45 PM             Materiale_3
d-----          03-Dec-21    2:10 PM             New folder
d-----          24-Nov-21    8:12 PM             Outer World - Config files backup
d-----          23-Apr-21    8:24 PM             Personal
d-----          13-Nov-21   12:24 PM             Snippets
-a-----          22-Sep-21   10:23 PM           2236 Discord.lnk
-a-----          21-Apr-21    5:17 PM           1427 Dofus.lnk
-a-----          16-Jan-21    8:06 PM            223 Hades.url
-a-----          22-Jul-21   11:56 PM          2260 Notion.lnk
-a-----          13-Nov-21   10:39 AM          2190 Postman.lnk
-a-----          17-Oct-21    9:36 PM          1491 Riot Client.lnk
-a-----          17-Jan-21    8:00 PM          1035 Telegram.lnk
-a-----          16-Jan-21    8:18 PM            222 Terraria.url
-a-----          30-Jul-21    6:39 PM          1427 Wakfu.lnk
-a-----          09-May-21    6:55 PM          1934 Zoom.lnk

PS C:\Users\Andrei\Desktop> 
```

CUM FOLOSIM TERMINALUL?

Pentru a ne intoarce la folderul anterior vom executa comanda **cd ..**

```
PS C:\Users\Andrei\Desktop> cd ..
```

CUM FOLOSIM TERMINALUL?

Putem observa ca ne-am întors la folderul anterior

```
PS C:\Users\Andrei\Desktop> cd ..  
PS C:\Users\Andrei>
```



01

GIT



CE ESTE GIT?

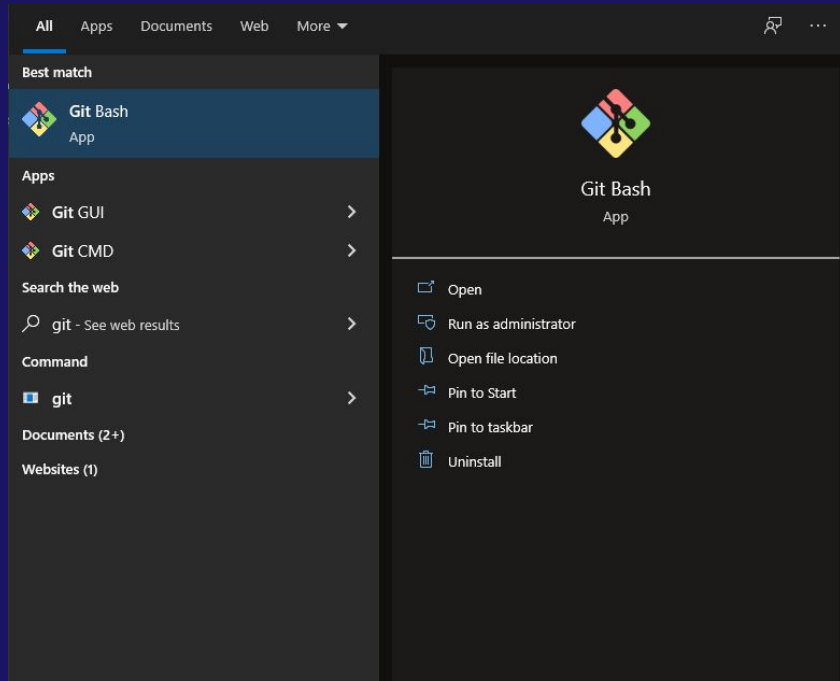
Un sistem de control al versiunilor, sau VCS, urmărește istoricul schimbărilor pe măsură ce oamenii și echipele colaborează la proiecte împreună. Pe măsură ce dezvoltatorii fac modificări în proiect, orice versiune anterioară a proiectului poate fi recuperată în orice moment.



CE PUTEM URMARI CU GIT?

- Ce modificări au fost făcute?
- Cine a făcut modificările?
- Când au fost făcute modificările?
- De ce au fost necesare schimbări?

GIT BASH



Vom apasa din nou tasta
Windows si de aceasta
data vom scrie 'Git Bash'

GIT BASH

```
Andrei@DESKTOP-CNOMI37 MINGW64 ~  
$
```

Dupa deschiderea
terminalului, vom vedea
urmatoarea fereastra

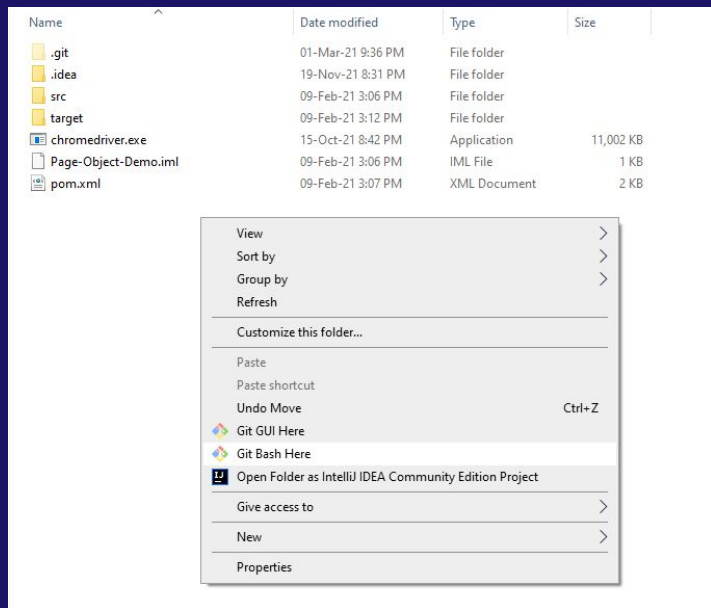
GIT BASH

Acum, folosind comenzile de **ls** si comenzile de **cd** navigati la un folder unde aveti un proiect de IntelliJ

```
Andrei@DESKTOP-CNOMI37 MINGW64 ~  
$ cd Desktop/Demo-Azimut/Page-Object-Demo
```

GIT BASH

Alternativ, puteti da click-dreapta in folderul cu proiectul
si apoi 'Git Bash Here'



GIT BASH

git init - inițializează un nou depozit Git și începe să urmărească un director existent. Acesta adaugă un subfolder ascuns în directorul existent care găzduiește structura internă de date necesară pentru controlul versiunilor.

```
Andrei@DESKTOP-CNOMI37 MINGW64 ~/Desktop/Demo-Azimet/Page-Object-Demo
$ git init
Initialized empty Git repository in C:/Users/Andrei/Desktop/Demo-Azimet/Page-Object-Demo/.git/
```

GIT BASH

git init - inițializează un nou depozit Git și începe să urmărească un director existent. Acesta adaugă un subfolder ascuns în directorul existent care găzduiește structura internă de date necesară pentru controlul versiunilor.

21			
 .git	03-Dec-21 4:00 PM	File folder	
 .idea	19-Nov-21 8:31 PM	File folder	
 src	09-Feb-21 3:06 PM	File folder	
 target	09-Feb-21 3:12 PM	File folder	
 chromedriver.exe	15-Oct-21 8:42 PM	Application	11,002 KB
 Page-Object-Demo.iml	09-Feb-21 3:06 PM	IML File	1 KB
 pom.xml	09-Feb-21 3:07 PM	XML Document	2 KB

GIT BASH

git status - arată starea modificărilor ca neurmărite, modificate sau în staging.

```
Andrei@DESKTOP-CNOMI37 MINGW64 ~/Desktop/Demo-Azimut/Page-Object-Demo (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .idea/
        Page-Object-Demo.iml
        chromedriver.exe
        pom.xml
        src/
        target/

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

GIT BASH

git add - pune o schimbare in staging.

In cazul nostru vom scrie **git add .**

```
Andrei@DESKTOP-CNOMI37 MINGW64 ~/Desktop/Demo-Azimut/Page-Object-Demo (master)
$ git add .
warning: LF will be replaced by CRLF in pom.xml.
The file will have its original line endings in your working directory
```

GIT BASH

git commit - salvează în istoricul proiectului și finalizează procesul de urmărire a modificărilor. Pe scurt, un commit funcționează ca a face o fotografie. Orice lucru care a fost pus în staging cu git add va deveni parte a istoricului cu git commit.

```
Andrei@DESKTOP-CNOMI37 MINGW64 ~/Desktop/Demo-Azimut/Page-Object-Demo (master)
$ git commit -m "Initial commit"
```