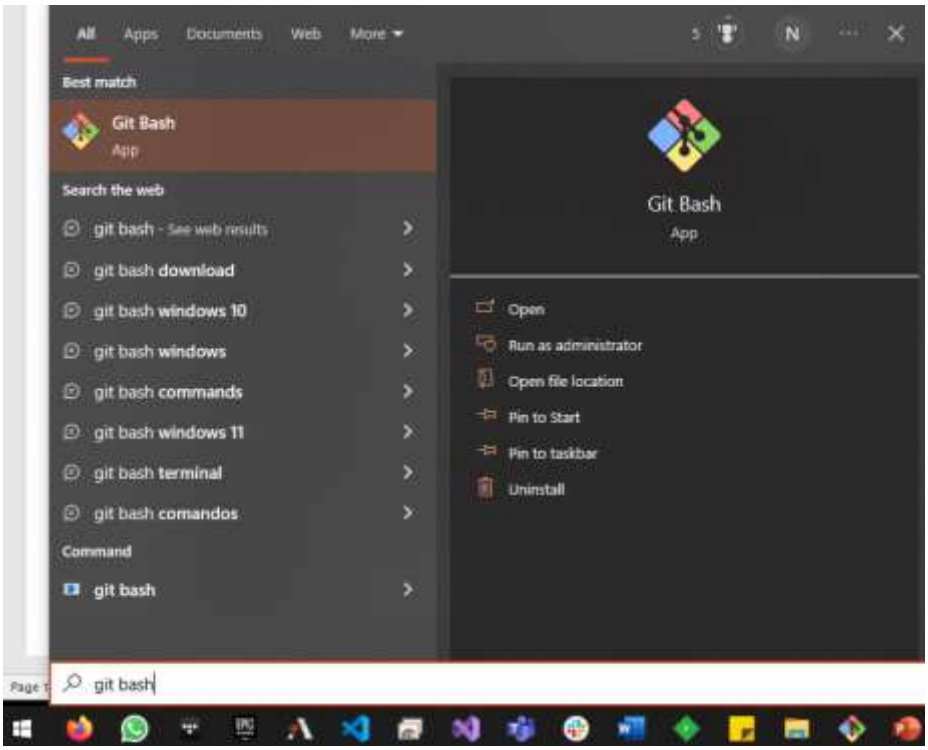
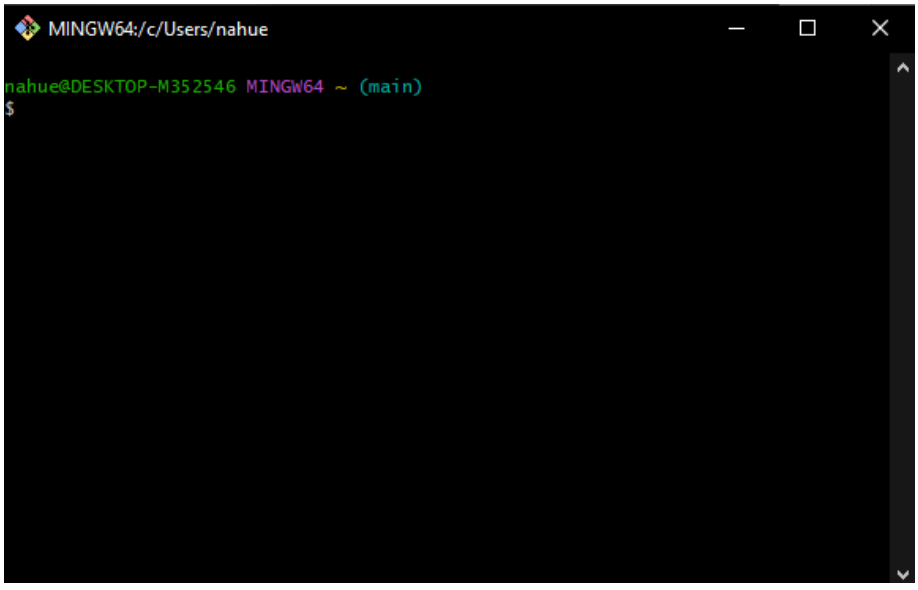
Team International AQA Bootcamp Task 1: Git

Student: *Nahuel Rodríguez Maury*

Part 1: Repository initialization and first commit

1. Open Git Bash from Start:





2. Move to desktop:

```
MINGW64:/c/Users/nahue/desktop

nahue@DESKTOP-M352546 MINGW64 ~ (main)
$ cd desktop

nahue@DESKTOP-M352546 MINGW64 ~/desktop (main)
$
```

3. Create a working directory:

```
MINGW64:/c/Users/nahue/desktop

nahue@DESKTOP-M352546 MINGW64 ~/desktop (main)
$ mkdir "Team International AQA Bootcamp - Task Git"
```

4. Enter into the folder:

```
MINGW64:/c/Users/nahue/desktop/Team International AQA Bootcamp - Task Git

nahue@DESKTOP-M352546 MINGW64 ~/desktop (main)
$ cd "Team International AQA Bootcamp - Task Git"

nahue@DESKTOP-M352546 MINGW64 ~/desktop/Team International AQA Bootcamp - Task Git (main)
$
```

5. Initialize the repository:

```
MINGW64:/c/Users/nahue/desktop/Team International AQA Bootcamp - Task Git

nahue@DESKTOP-M352546 MINGW64 ~/desktop/Team International AQA Bootcamp - Task Git (main)
$ git init
Initialized empty Git repository in C:/Users/nahue/Desktop/Team International AQA Bootcamp - Task Git/.git/
```

6. Customize the config file:

```
MINGW64:/c/Users/nahue/desktop/Team International AQA Bootcamp - Task Git

nahue@DESKTOP-M352546 MINGW64 ~/desktop/Team International AQA Bootcamp - Task Git (main)
$ git config --global user.email "nahuelrmaury@gmail.com"

nahue@DESKTOP-M352546 MINGW64 ~/desktop/Team International AQA Bootcamp - Task Git (main)
$ git config --global user.name "Nahuel Rodriguez"
```

7. Create a working directory in it:

```
MINGW64:/c/Users/nahue/desktop/Team International AQA Bootcamp - Task Git

nahue@DESKTOP-M352546 MINGW64 ~/desktop/Team International AQA Bootcamp - Task Git (main)
$ mkdir student_Rodriguez
```

8. Enter into the new folder and create 3 project files:

```
MINGW64:/c/Users/nahue/desktop/Team International AQA Bootcamp - Task Git/student_Rodriguez

nahue@DESKTOP-M352546 MINGW64 ~/desktop/Team International AQA Bootcamp - Task Git (main)
$ cd student_Rodriguez

nahue@DESKTOP-M352546 MINGW64 ~/desktop/Team International AQA Bootcamp - Task Git/student_Rodriguez (main)
$ touch hello_Rodriguez.rb hey_Rodriguez.rb hi_Rodriguez.rb

nahue@DESKTOP-M352546 MINGW64 ~/desktop/Team International AQA Bootcamp - Task Git/student_Rodriguez (main)
$ ls
hello_Rodriguez.rb  hey_Rodriguez.rb  hi_Rodriguez.rb
```

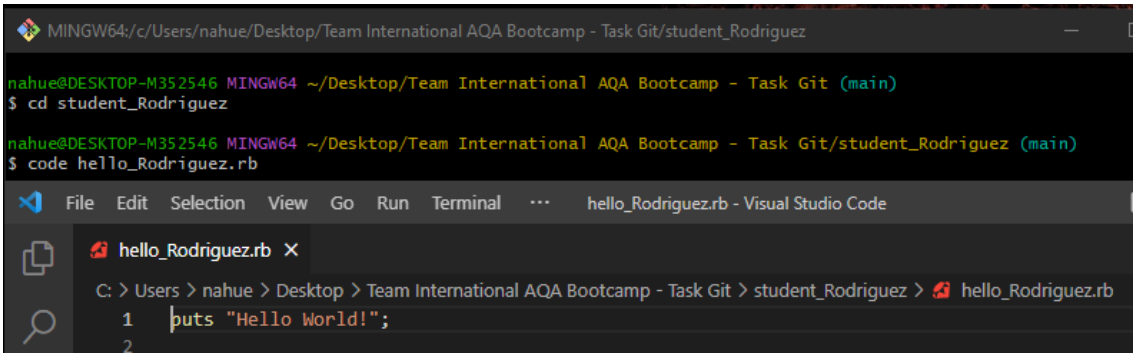
9. Move back and check repository status:

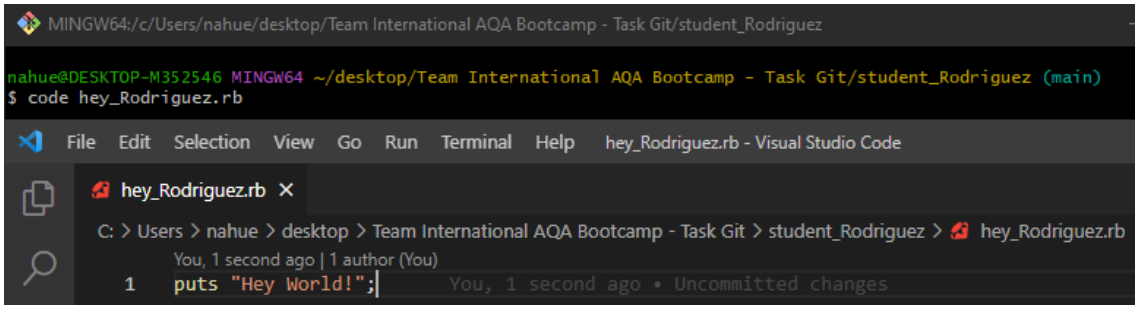
```
MINGW64:/c/Users/nahue/desktop/Team International AQA Bootcamp - Task Git

nahue@DESKTOP-M352546 MINGW64 ~/desktop/Team International AQA Bootcamp - Task Git/student_Rodriguez (main)
$ cd ..

nahue@DESKTOP-M352546 MINGW64 ~/desktop/Team International AQA Bootcamp - Task Git (main)
$ git status
On branch main

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        student_Rodriguez/

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

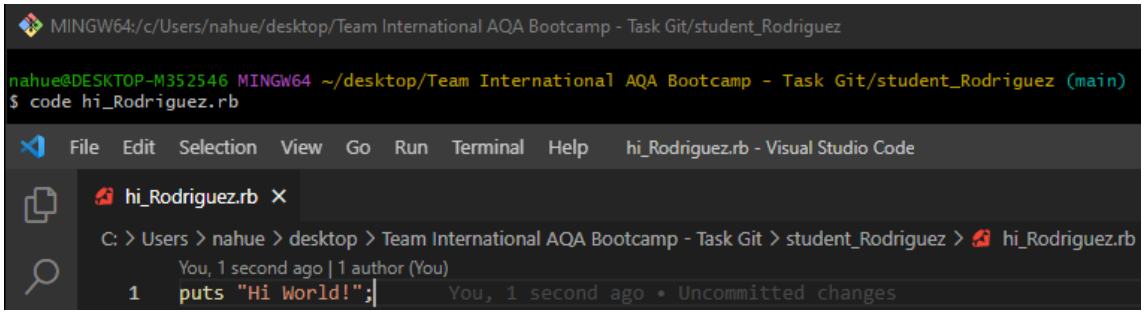
10. Index the files and make the first commit:

```
nahue@DESKTOP-M352546 MINGW64 ~/Desktop/Team International AQA Bootcamp - Task Git (main)
$ git add .

nahue@DESKTOP-M352546 MINGW64 ~/Desktop/Team International AQA Bootcamp - Task Git (main)
$ git commit -m "first commit Rodriguez"
[main (root-commit) 10bc86e] first commit Rodriguez
 3 files changed, 7 insertions(+)
 create mode 100644 student_Rodriguez/hello_Rodriguez.rb
 create mode 100644 student_Rodriguez/hey_Rodriguez.rb
 create mode 100644 student_Rodriguez/hi_Rodriguez.rb
```

11. Make changes to files:

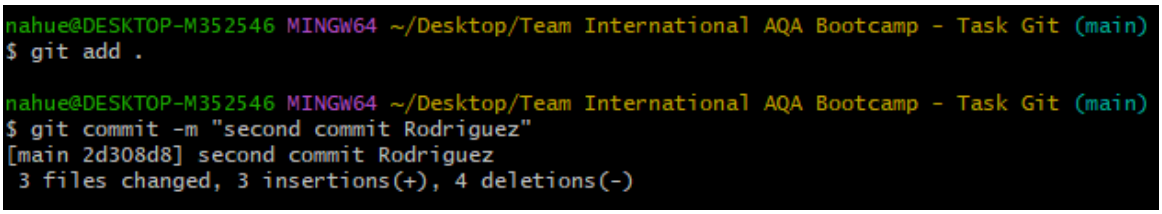






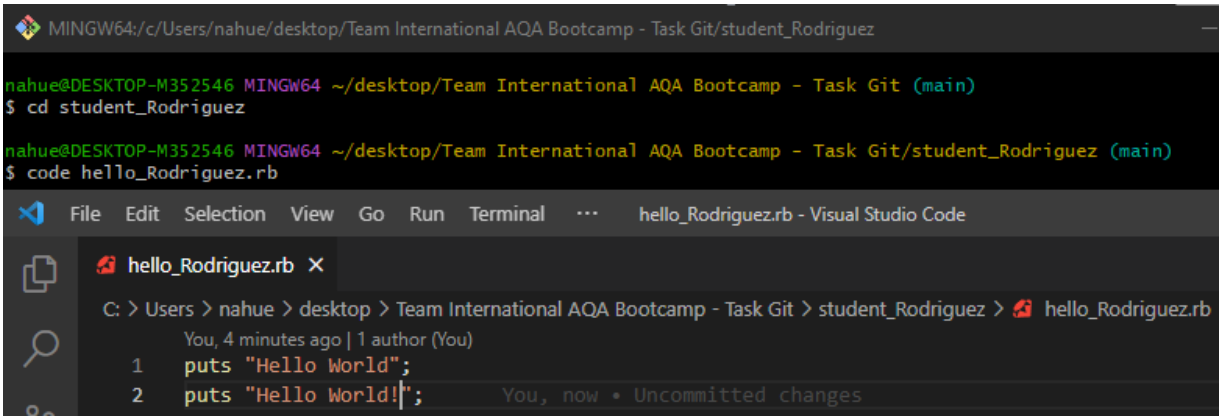
12. Index the files and make a commit of the changes:

(I did not was sure about this step, because the task just says to make a commit, but I index the files before that so It can commit it correctly)

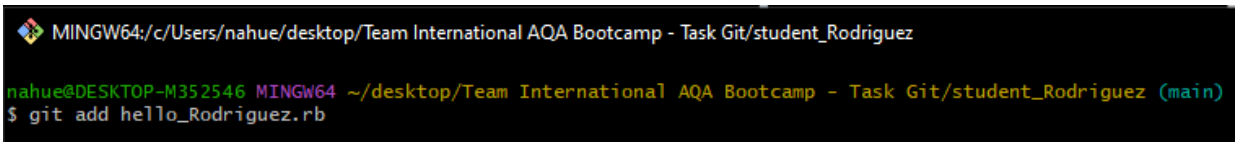


Part 2: Working with changes

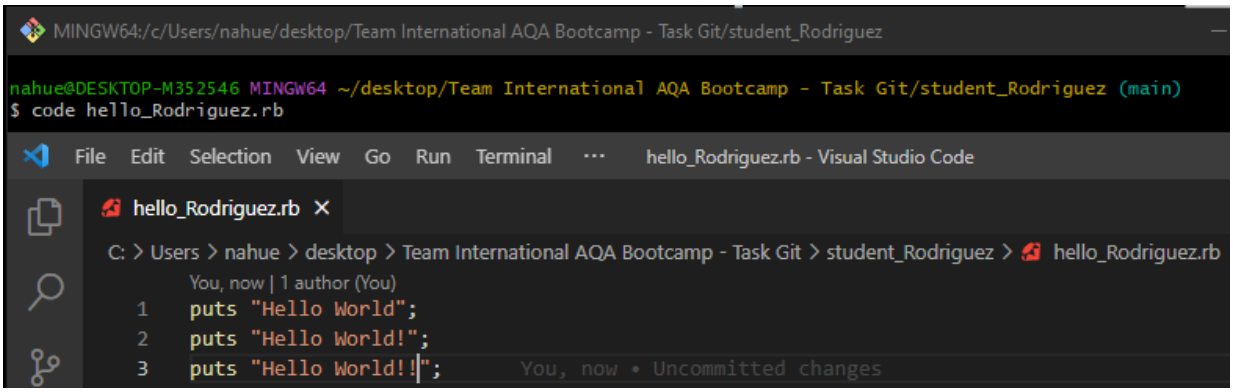
1. Make changes to the first file:



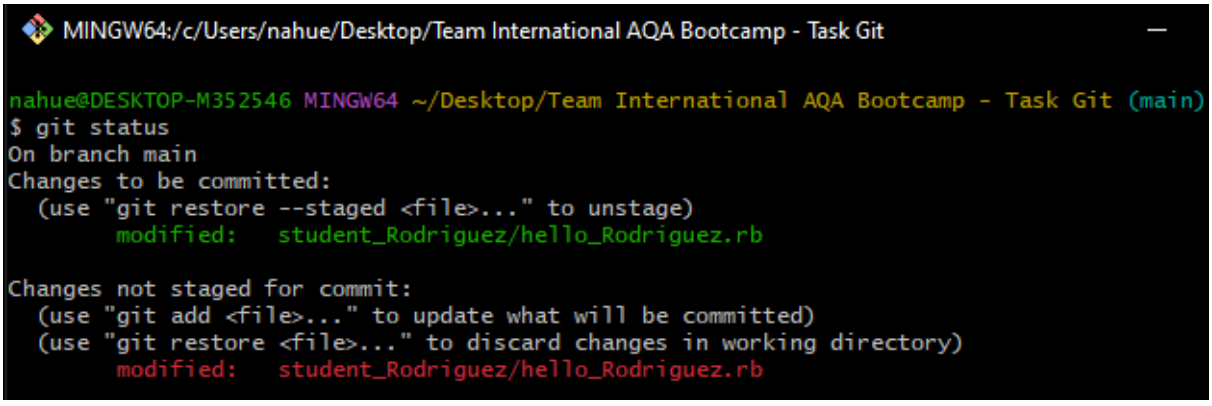
2. Index the file:



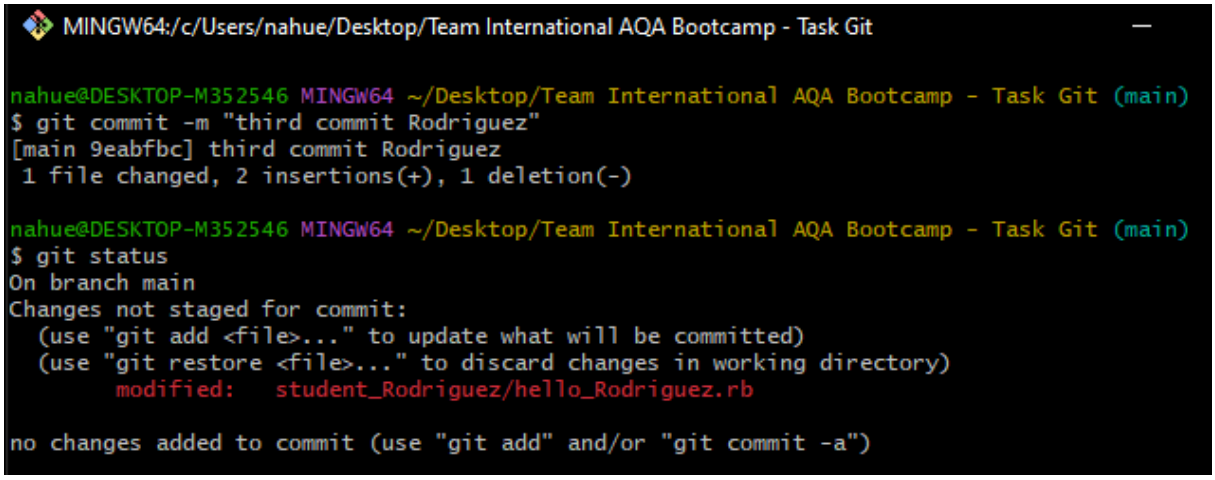
3. Make new changes to the first file:



4. Check status:



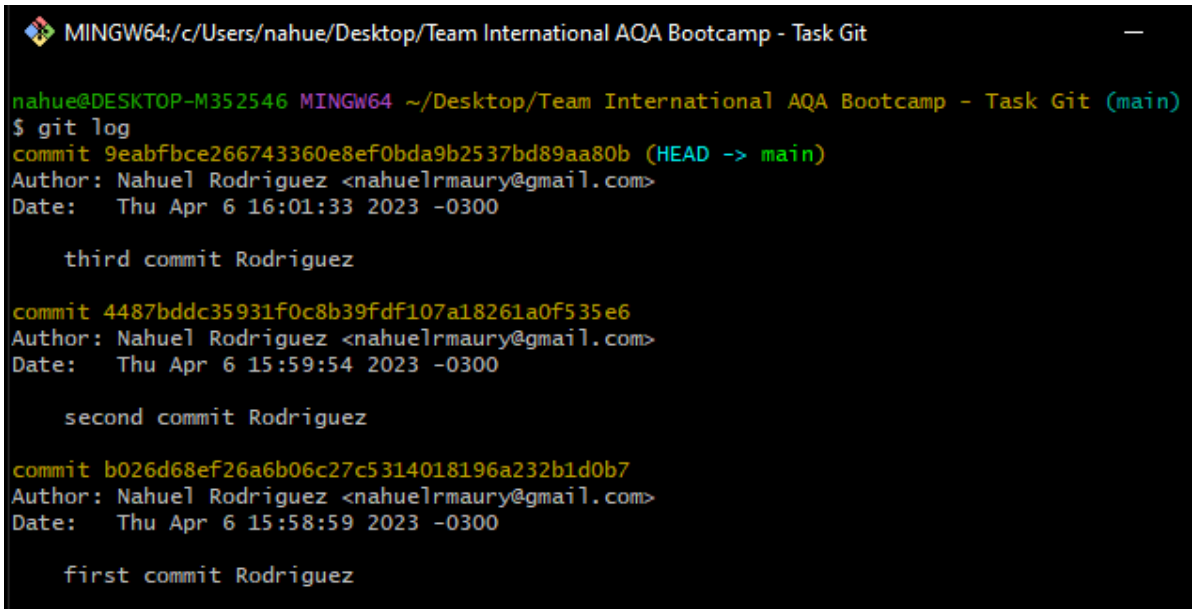
5. Make commit and check status:



6. Explain the result:

I first made changes to the first file and then tracked it with “git add hello_Rodriguez.rb”, this way I move it to staging area. After, I made more changes to the first file. Then, I committed, and the result was that the first changes were committed and passed to the local repository and the second changes stayed in the working directory.

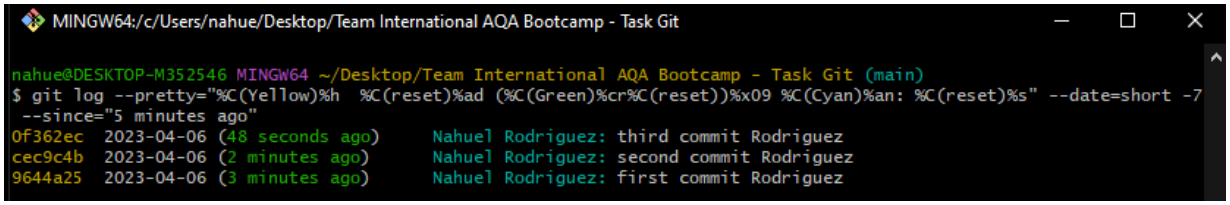
7. Display the history of the project:

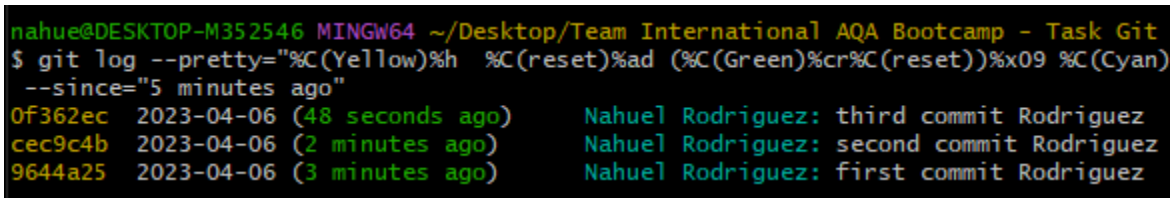


8. Use the command git log –pretty=oneline, adjusting relevant options to display the latest commits for 5 minutes:

If I had used –pretty=oneline, it would not show relevant options. So, I had to give it some format.

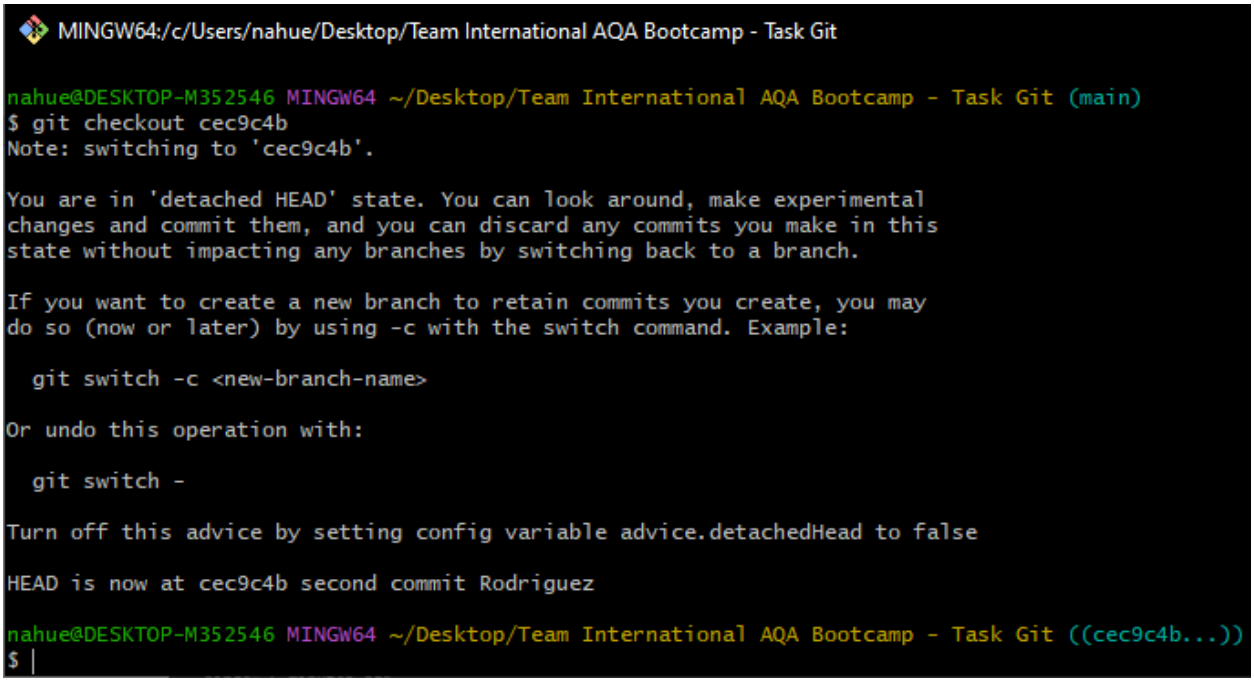
Command used: git log --pretty="%C(Yellow)%h %C(reset)%ad (%C(Green)%cr%C(reset))%x09 %C(Cyan)%an: %C(reset)%s" --date=short -7 --since="5 minutes ago"



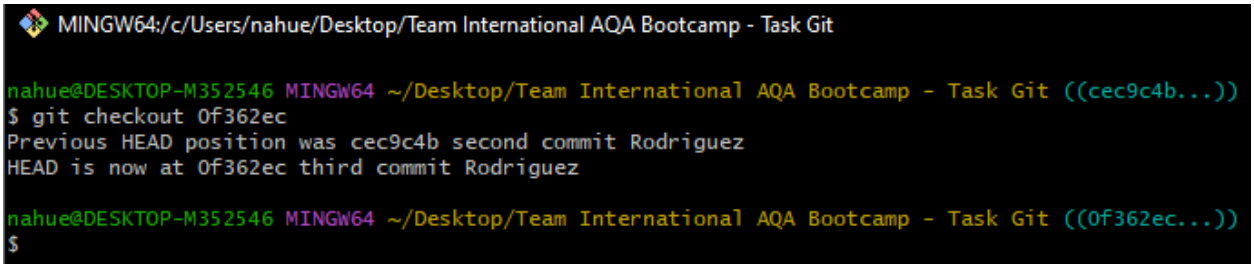


Part 3: Working with the previous states of the repository

1. Revert the state of the files:

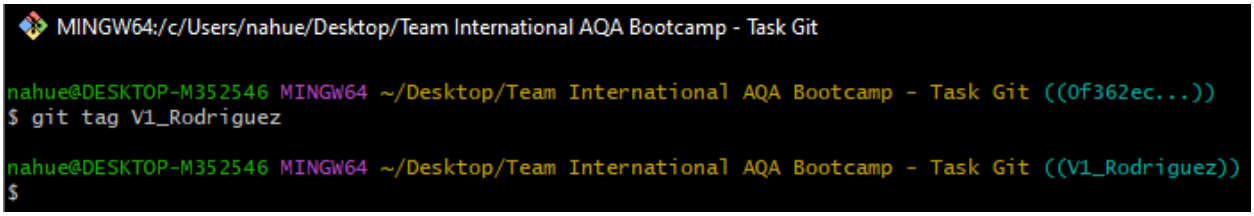


2. Revert the state of the files to the latest commit:

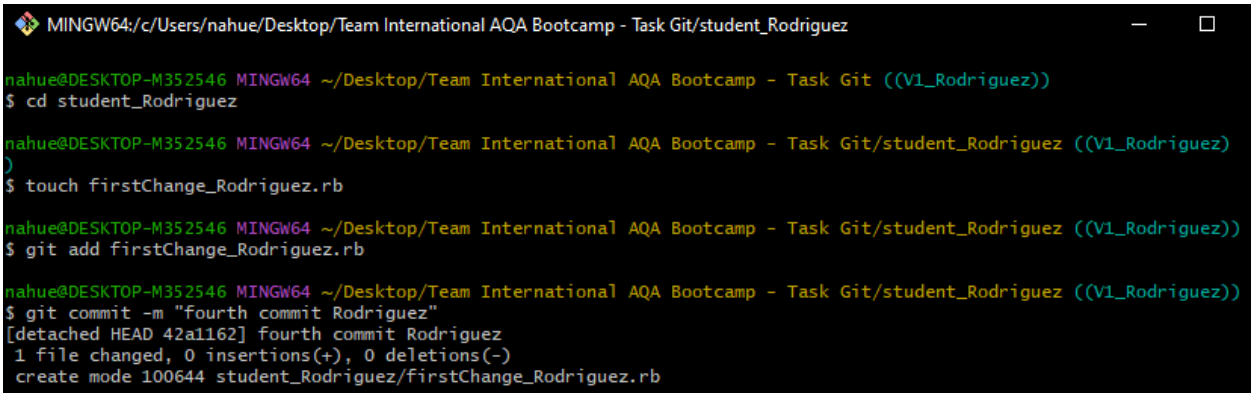


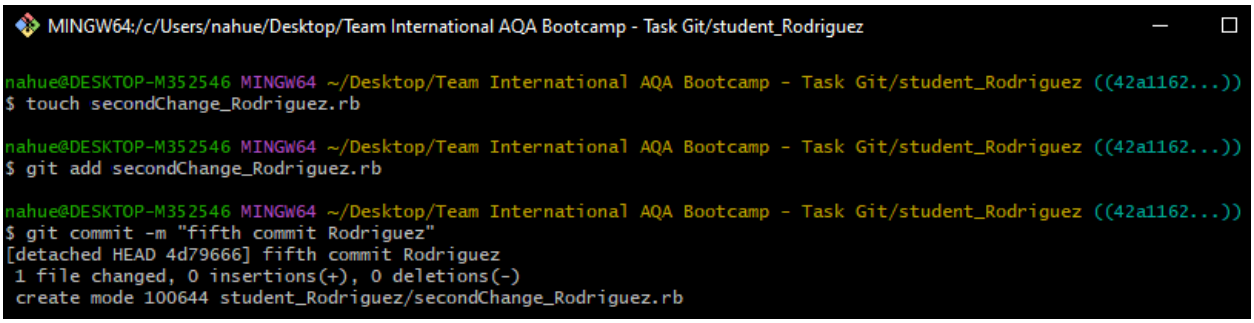
Part 4: Working with tags

1. Create a tag in the current repository snapshot:

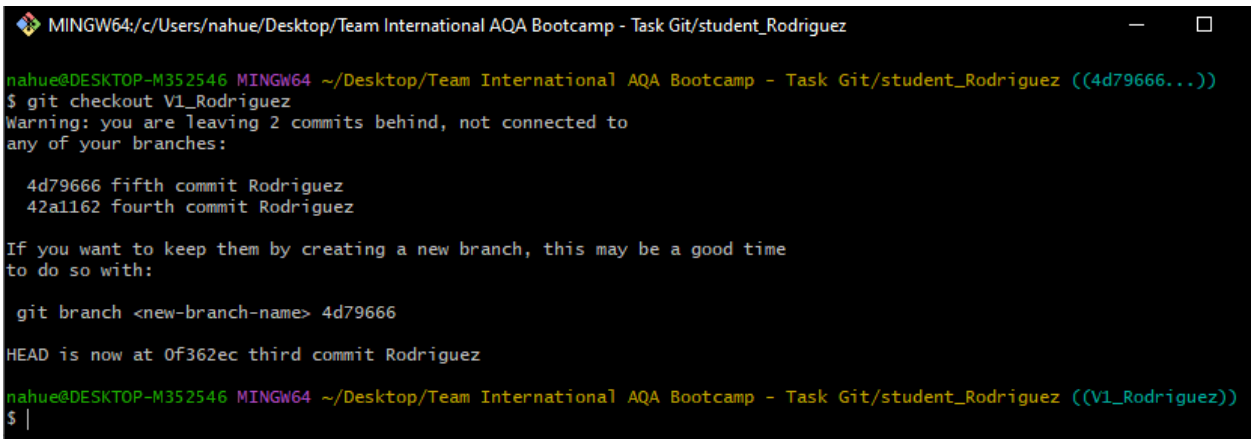


2. Make changes and at least two commits in the current directory:

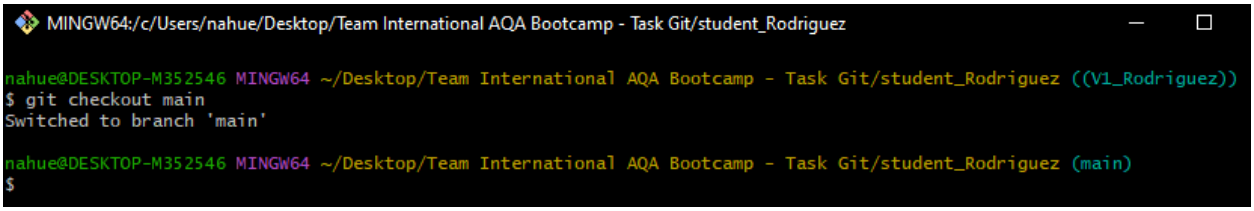




3. Demonstrate reverting to a tagged repository state:

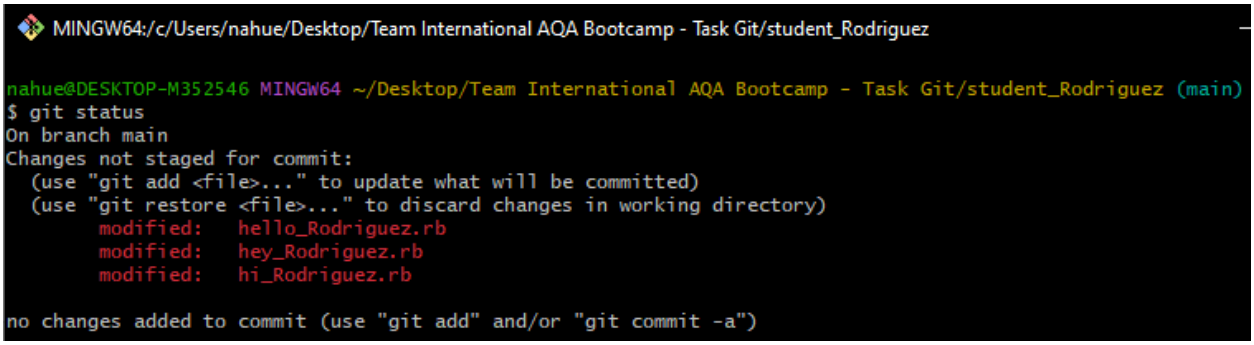


4. Revert to the last master state:

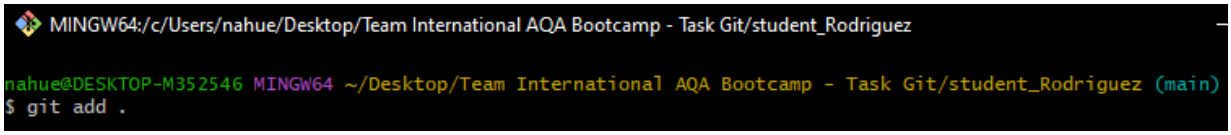


Part 5: Working with the revert function

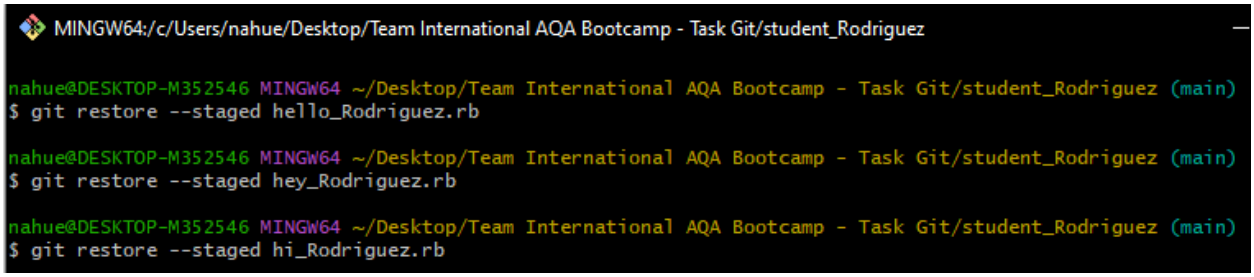
1. Make changes to the current state of the working directory:

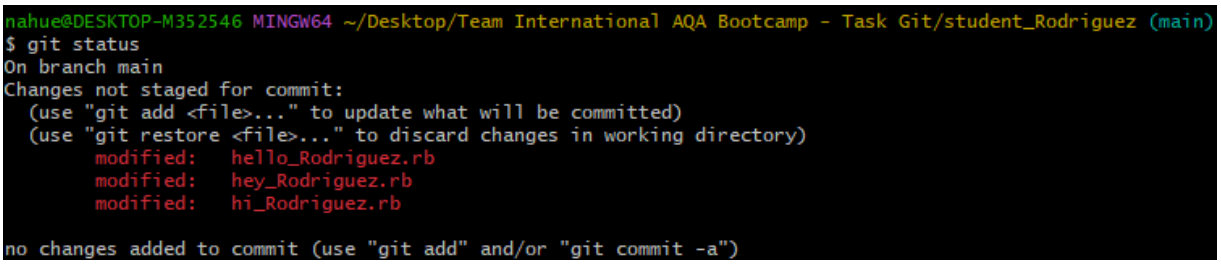


2. Index the changes:

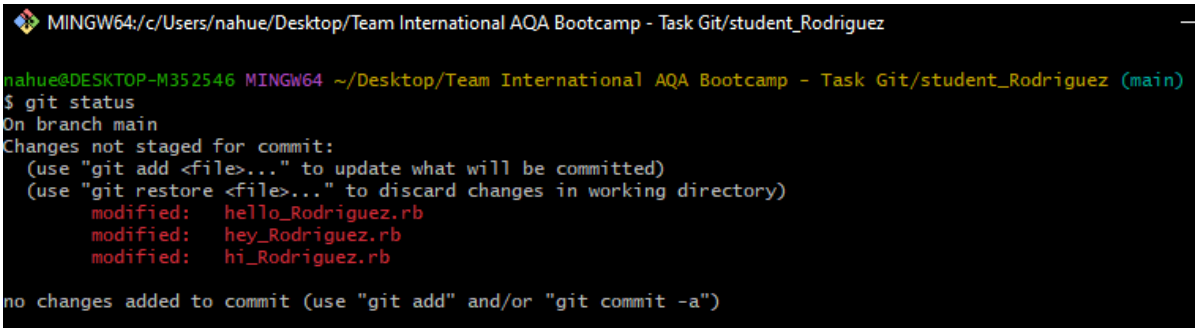


3. Use the restore command to restore the original state of the files:

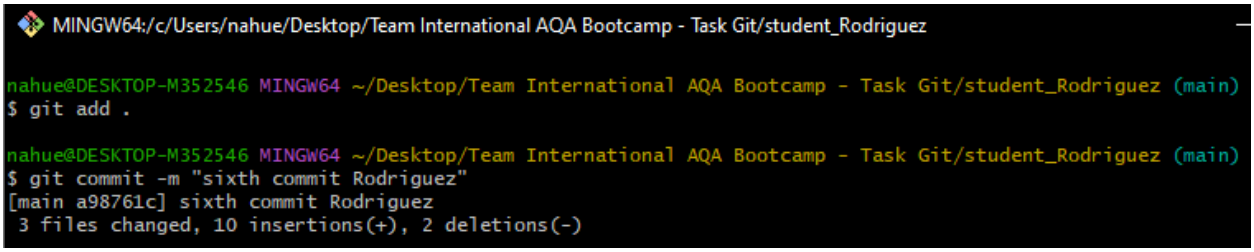




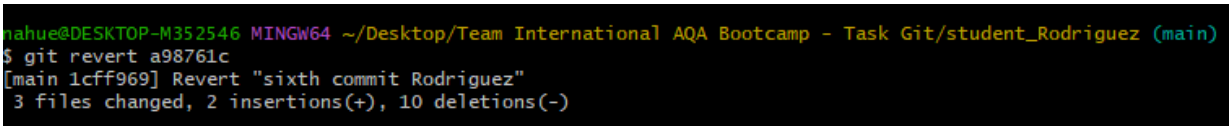
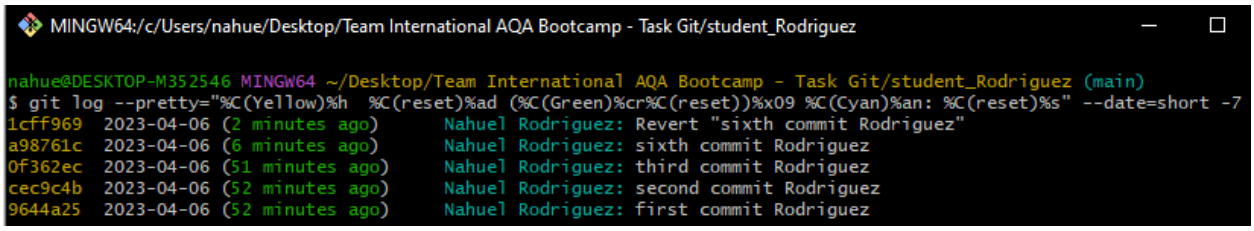
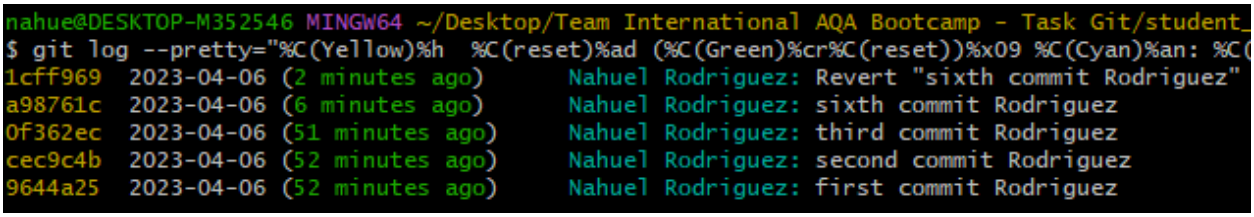
4. Make directory state changes:



5. Index and commit:



6. Revert the last commit:



7. Show log: