

Xóchitl Analí Cabañas Mota

NAO ID: 3319

September 12th, 2025

In-Mexico Program Backend
Developer Certification

GitHub and Digital Repository
Management:

Develop – Sprint 3

Table of contents

Branch A.....	3
Branch B.....	4
Conflict	6
Revert.....	7

This report shows the process of branch management, conflict resolution, and change reversion in GitHub. Each section is briefly documented along a screenshot to have a graphic description.

Branch A

This screenshot shows the creation of branch **A** from the Master branch using Git commands. This step ensures that work on branch A can proceed independently without affecting the main codebase.

```
Xoch@DESKTOP-J714T83 MINGW64 ~/Documents/handson-ml (master)
$ git checkout -b A
Switched to a new branch 'A'
```

Here, the docs.txt file is created in branch A, specifically for this branch.

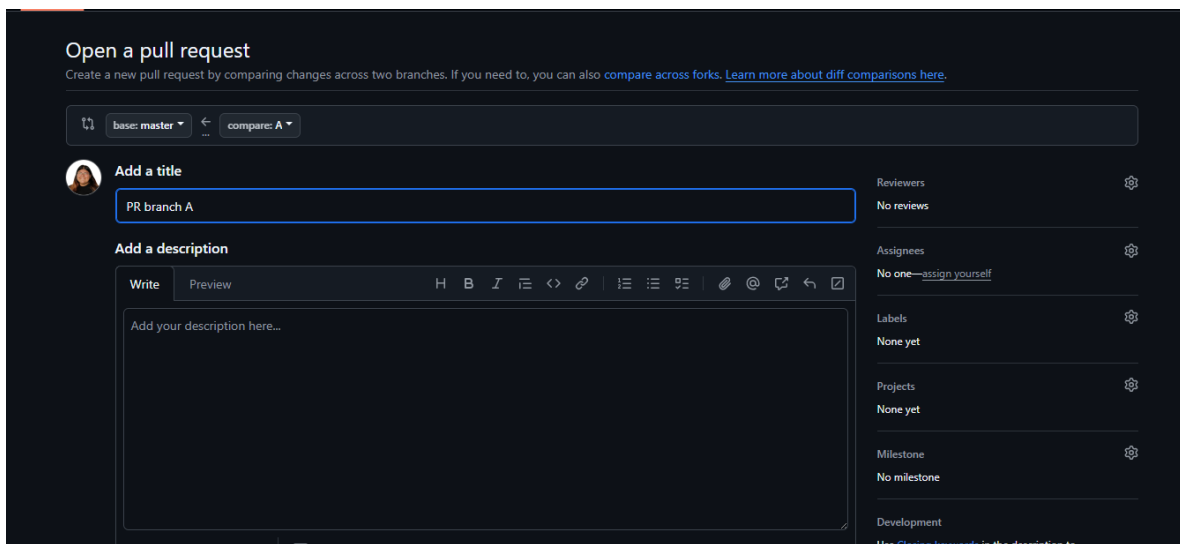
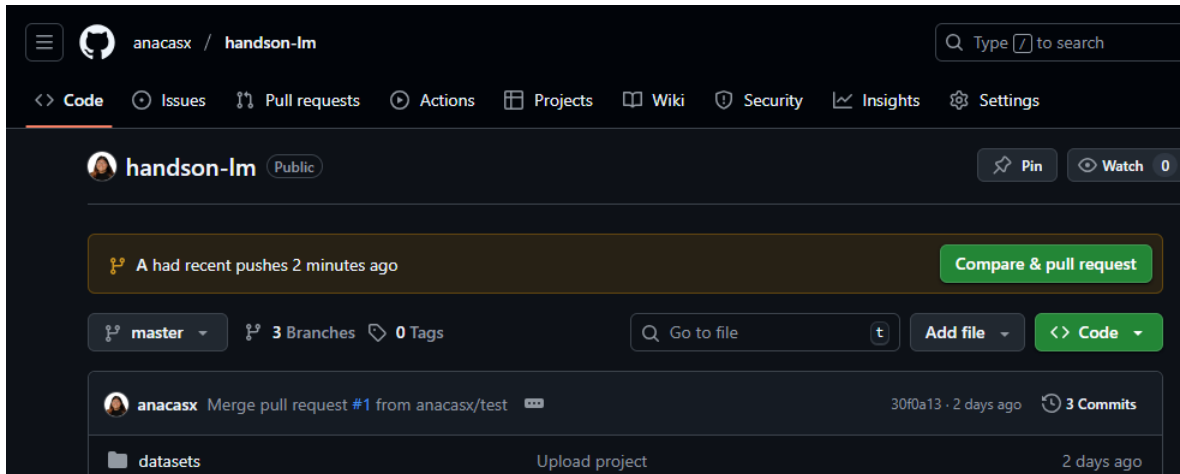
```
Xoch@DESKTOP-J714T83 MINGW64 ~/Documents/handson-ml (A)
$ echo "Hello World" > docs.txt

Xoch@DESKTOP-J714T83 MINGW64 ~/Documents/handson-ml (A)
$ git add docs.txt
warning: in the working copy of 'docs.txt', LF will be replaced by CRLF the
time Git touches it

Xoch@DESKTOP-J714T83 MINGW64 ~/Documents/handson-ml (A)
$ git commit -m "Add docs.txt"
[A cd20467] Add docs.txt
1 file changed, 1 insertion(+)
create mode 100644 docs.txt

Xoch@DESKTOP-J714T83 MINGW64 ~/Documents/handson-ml (A)
$ git push origin A
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 277 bytes | 277.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'A' on GitHub by visiting:
remote:   https://github.com/anacasx/handson-1m/pull/new/A
remote:
To https://github.com/anacasx/handson-1m.git
 * [new branch]      A -> A
```

These images capture the process of opening a Pull Request (PR) from branch A to Master on GitHub.



Branch B

The screenshot shows the creation of branch **B** from Master. This ensures that development in branch B is independent and starts from the latest version of the main codebase.

```
Xoch@DESKTOP-J714T83 MINGW64 ~/Documents/handson-m1 (master)
$ git checkout -b B
Switched to a new branch 'B'
```

A different docs.txt file is created in branch B. The screenshot confirms that both branches A and B contain distinct versions of the same file

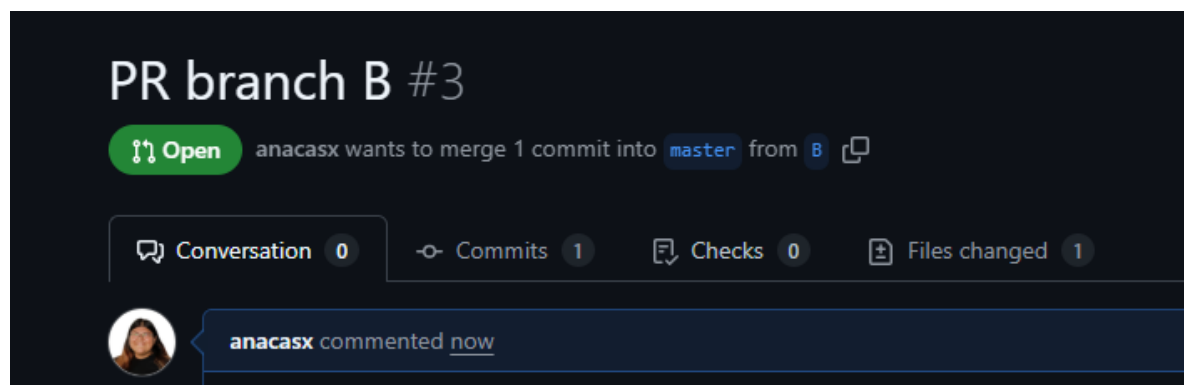
```
Xoch@DESKTOP-J714T83 MINGW64 ~/Documents/handson-m1 (B)
$ echo "Hello Kitty" > docs.txt

Xoch@DESKTOP-J714T83 MINGW64 ~/Documents/handson-m1 (B)
$ git add docs.txt
warning: in the working copy of 'docs.txt', LF will be replaced by CRLF
time Git touches it

Xoch@DESKTOP-J714T83 MINGW64 ~/Documents/handson-m1 (B)
$ git commit -m "Add docs.txt"
[B 1bc8120] Add docs.txt
1 file changed, 1 insertion(+)
create mode 100644 docs.txt

Xoch@DESKTOP-J714T83 MINGW64 ~/Documents/handson-m1 (B)
$ git push origin B
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 276 bytes | 276.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'B' on GitHub by visiting:
remote:   https://github.com/anacasx/handson-1m/pull/new/B
remote:
To https://github.com/anacasx/handson-1m.git
 * [new branch]      B -> B
```

Here, a Pull Request is created from branch B to Master.



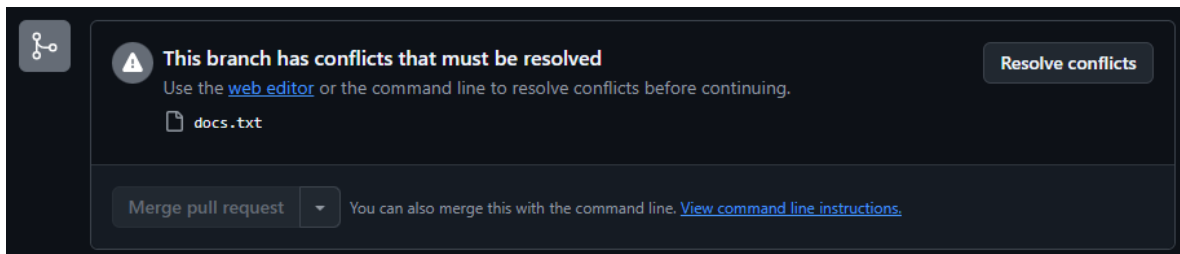
Conflict

Trying to merge branch A.



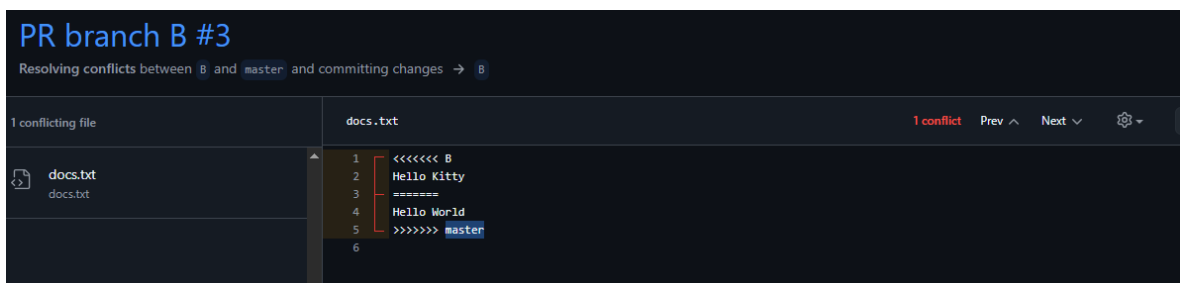
A GitHub merge confirmation dialog. On the left is a green icon with a white branching diagram. The main area has a 'Commit message' section with a text box containing 'Merge pull request #2 from anacasx/A'. Below it is an 'Extended description' section with a text box containing 'PR branch A'. At the bottom, it says 'This commit will be authored by 156331754+anacasx@users.noreply.github.com.' and has two buttons: 'Confirm merge' (green) and 'Cancel' (grey).

GitHub lets us know there is a conflict.



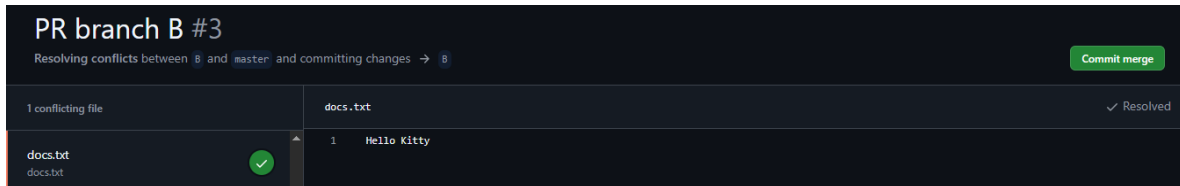
A GitHub warning box for conflicts. It has a grey icon with a white branching diagram on the left. The main area has a warning icon and the text 'This branch has conflicts that must be resolved'. Below this, it says 'Use the [web editor](#) or the command line to resolve conflicts before continuing.' and shows a file icon for 'docs.txt'. On the right is a 'Resolve conflicts' button. At the bottom, there is a 'Merge pull request' button with a dropdown arrow, followed by the text 'You can also merge this with the command line. [View command line instructions.](#)'

It highlights the conflict between the two versions of docs.txt. The GitHub interface displays both conflicting versions, and the resolution process involves editing and committing a unified version of the file.

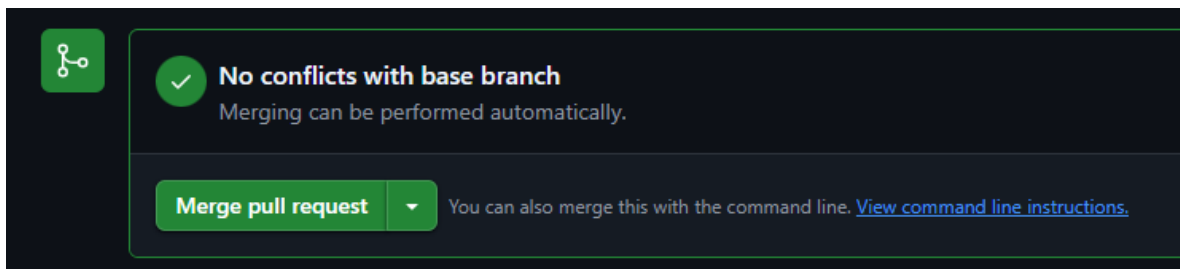


A GitHub interface for resolving conflicts in a pull request. The title is 'PR branch B #3'. Below it, it says 'Resolving conflicts between B and master and committing changes → B'. The main area is split into two panes. The left pane shows '1 conflicting file' and a file icon for 'docs.txt'. The right pane shows the content of 'docs.txt' with line numbers 1 through 6. The content is: 1 <<<<<< B, 2 Hello Kitty, 3 =====, 4 Hello World, 5 >>>>>> master, 6. A red bracket on the right side of the text indicates the conflict between the two versions.

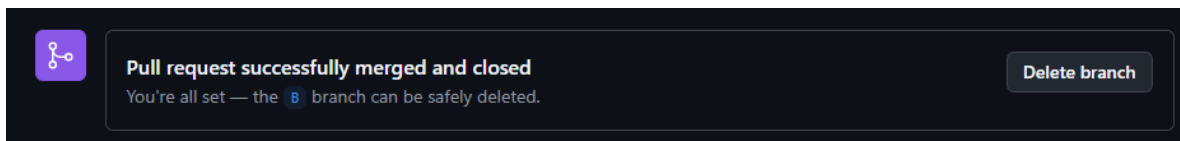
So this, a -new- version must be selected, either is the A, B or C, which would be a mix of last two.



Finally, the approval and merging of both Pull Requests (A and B) into Master can be done.



This confirms that conflicts were successfully resolved and both branches were integrated into the main project.



Revert

This image demonstrates the use of the git revert command in the terminal. It shows how a specific commit can be safely undone by creating a new commit that reverses its changes.

```
Xoch@DESKTOP-J714T83 MINGW64 ~/Documents/handson-m1 (test)
$ git log --oneline
2dba04c (HEAD -> test) Revert "update requiments.txt"
75b3316 (origin/test) update requiments.txt
7495178 (origin/main) Upload project

Xoch@DESKTOP-J714T83 MINGW64 ~/Documents/handson-m1 (test)
$ git revert 75b3316
```

```
Reapply "update requirements.txt"
```

```
This reverts commit 2dba04c2aa5120e916d6f732674ec1ba524959e7.
```

```
# Please enter the commit message for your changes. Lines starting  
# with '#' will be ignored, and an empty message aborts the commit.
```

```
#
```

```
# On branch test
```

```
# Your branch is up to date with 'origin/test'.
```

```
#
```

```
# Changes to be committed:
```

```
#       modified:   requirements.txt
```

```
#
```

```
~
```

```
[test fed30aa] Reapply "update requirements.txt"
```

```
1 file changed, 4 deletions(-)
```