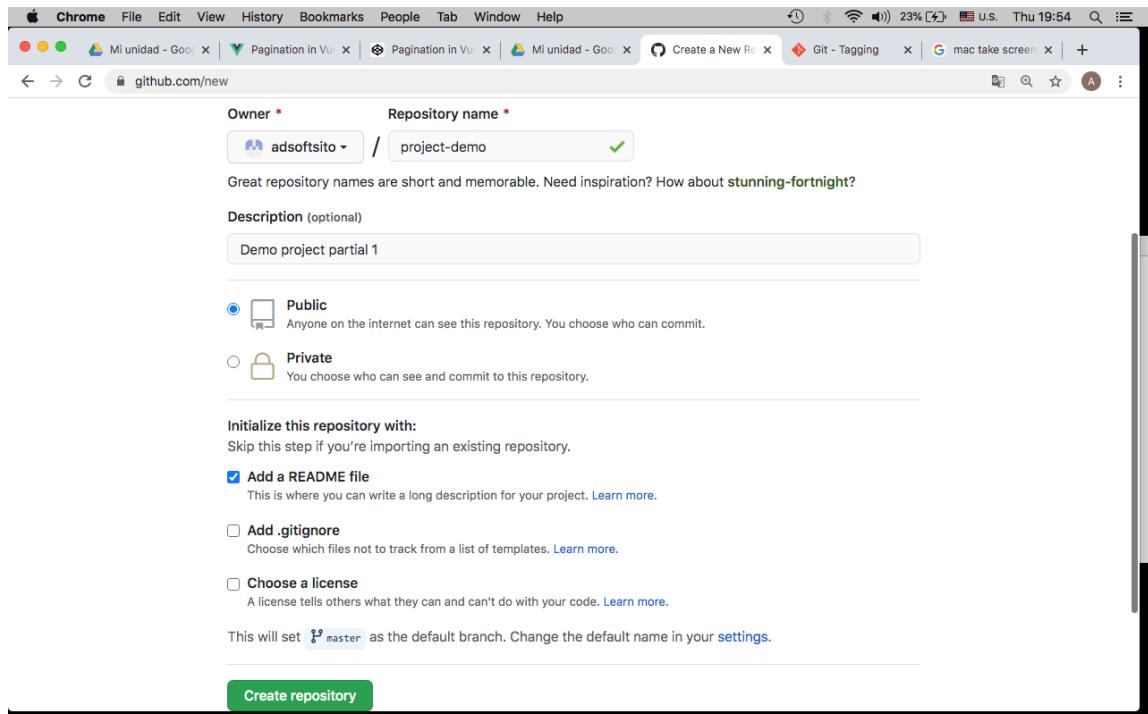


Git branching model & Automated kanban boards

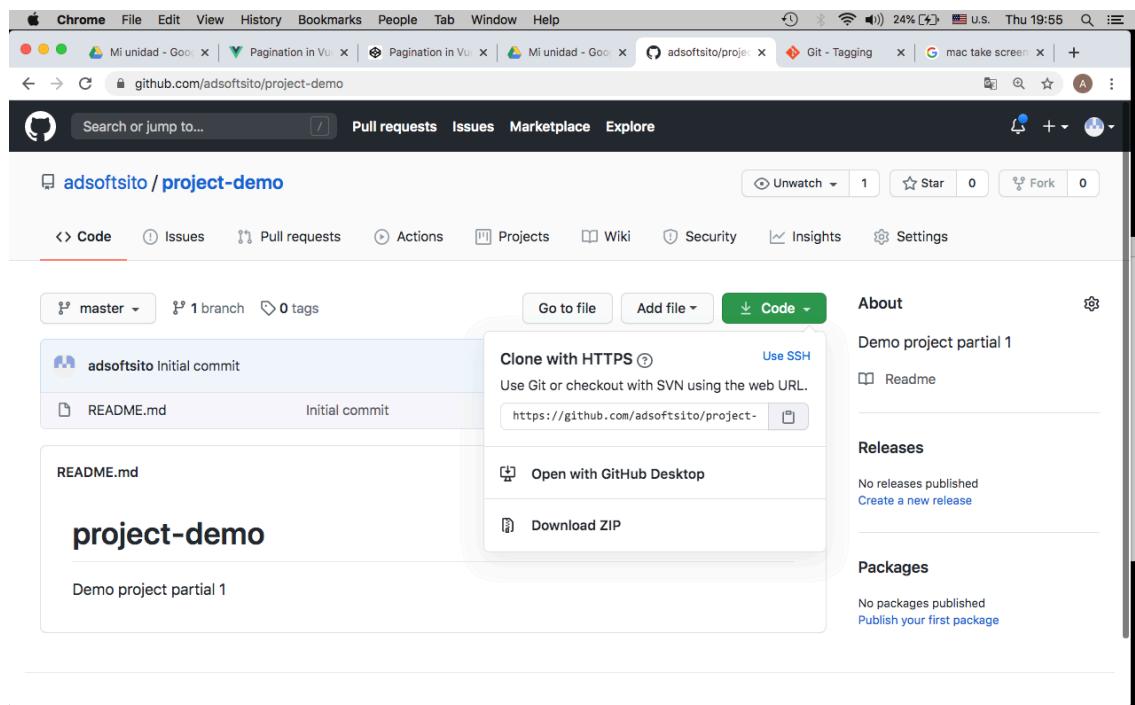
1.- Go to <https://github.com>

2.- Create new repository **project-demo**



The screenshot shows the 'Create a New Repository' page on GitHub. The 'Repository name' field is set to 'project-demo'. The 'Description (optional)' field contains 'Demo project partial 1'. The 'Visibility' section has 'Public' selected, with a note: 'Anyone on the internet can see this repository. You choose who can commit.' The 'Initialize this repository with:' section includes options for 'Add a README file' (selected), 'Add .gitignore', 'Choose a license', and a note about setting the default branch to 'master'. A large green 'Create repository' button is at the bottom.

3.- Copy url repository



The screenshot shows the GitHub repository page for 'adsoftsito/project-demo'. The repository has 1 branch and 0 tags. The main content area shows files like 'README.md' and 'adsoftsito Initial commit'. On the right, there's an 'About' section with 'Demo project partial 1', a 'Clone with HTTPS' button, and links for 'Readme', 'Releases', and 'Packages'. The 'Clone with HTTPS' button has the URL <https://github.com/adsoftsito/project-demo>.

4.- Clone repository in workstation

```
$ git clone https://github.com/<gituser>/project-demo.git
```

5.- Change to repo directory

```
$ cd project-demo
```

6.- Check url repository

```
$ pwd  
$ git remote -v
```

```
origin  https://github.com/adsoftsito/project-demo.git (fetch)  
origin  https://github.com/adsoftsito/project-demo.git (push)
```

7.- Edit README.md, add # v0.1 tag

```
$ vi README.md
```



8.- Commit, push changes

```
$ git add .  
$ git commit -m "update README with v0.1 tag"  
$ git push -u origin master
```

9.- Tag base code with v0.1

```
$ git tag -a v0.1 -m "my version 0.1"  
$ git tag  
$ git show v0.1
```

9.- Publish tag

```
$ git push origin v0.1
```

10.- create branch hotfixes, release and develop

```
$ git checkout master
$ git branch hotfixes
$ git checkout hotfixes

$ vi HOTFIXES.md
# hotfixes branch here
:wq

$ git add .
$ git commit -m "init hotfixes branch"
$ git push -u origin hotfixes
#####
$ git checkout master
$ git branch release
$ git checkout release

$ vi RELEASE.md
# release branch here
:wq

$ git add .
$ git commit -a -m "init release branch"
$ git push -u origin release
#####

$ git checkout master
$ git branch develop
$ git checkout develop
$ vi DEVELOP.md
# develop branch here
:wq

$ git add .

$ git commit -a -m "init develop branch"
$ git push -u origin develop
```

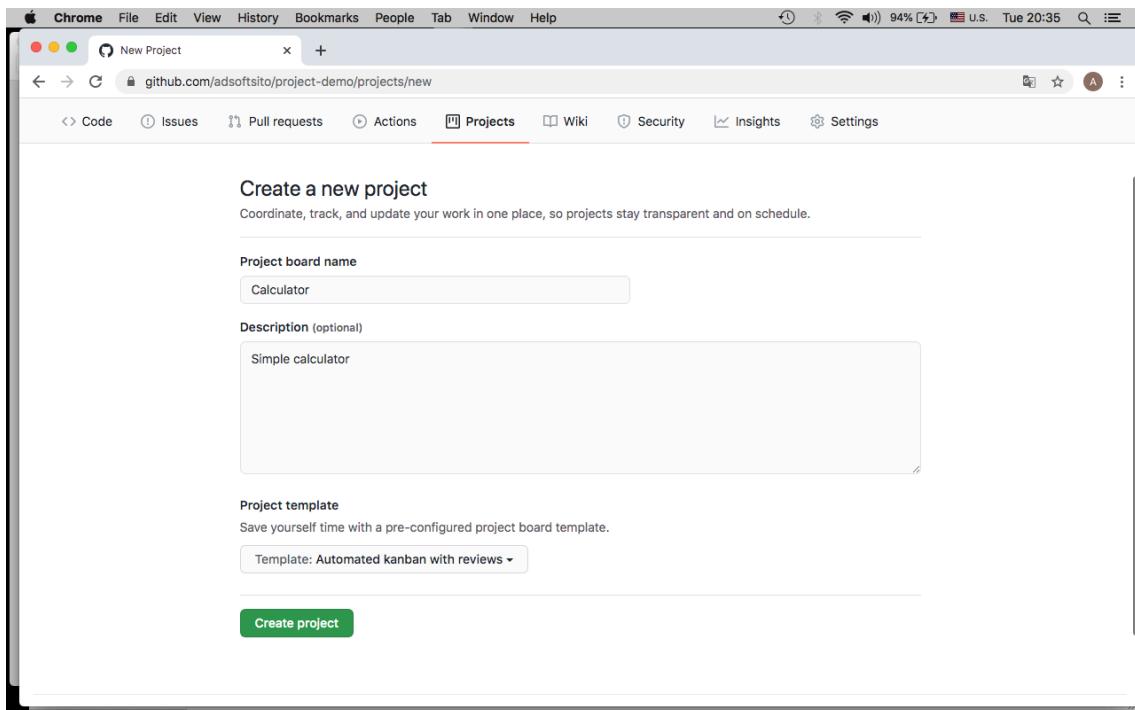
11.- Visualize Network graph in Insights menu

A screenshot of a Chrome browser window displaying the GitHub Network Graph for the repository 'adsoftsito/project-demo'. The URL in the address bar is <https://github.com/adsoftsito/project-demo/network>. The page shows a 'Network graph' section with a timeline of commits. On the left, there is a sidebar with options like Pulse, Contributors, Community, Traffic, Commits, Code frequency, Dependency graph, Network (which is selected), and Forks. The main area shows a commit history for the 'adsoftsito' owner. A single commit from 'adsoftsito' on 'Sep 1' is shown, with arrows pointing upwards to 'master', 'release', 'hotfix', and 'develop'. The commit itself is labeled 'master'.

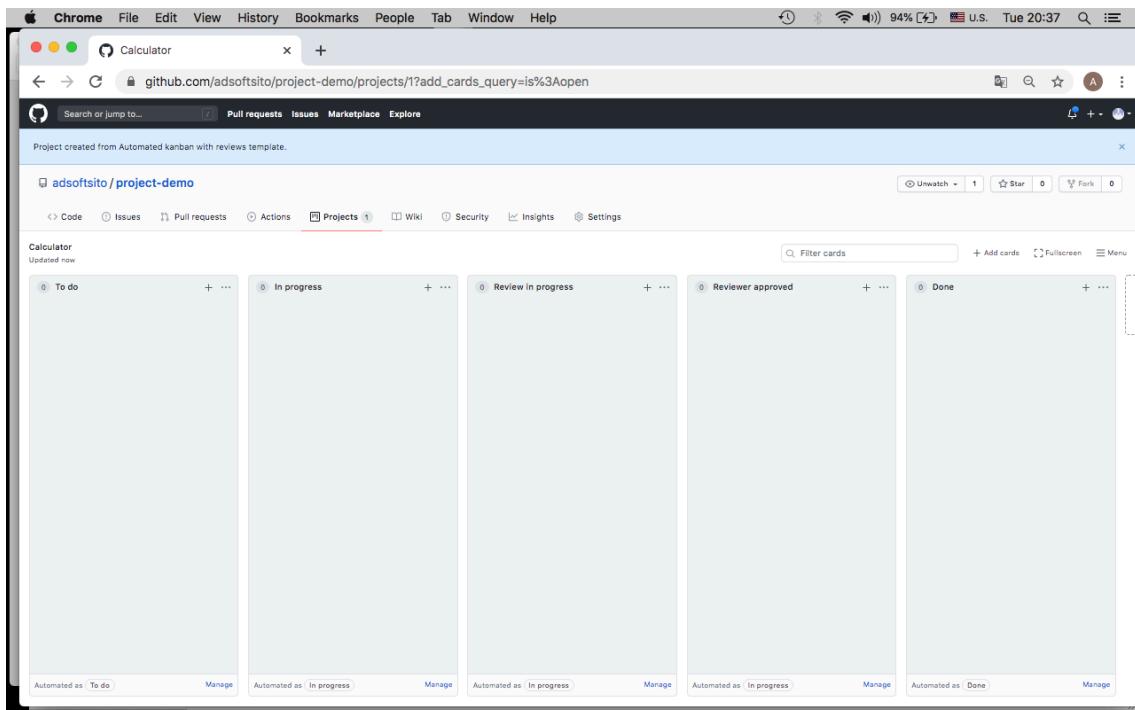
12.- Add team members to repository

A screenshot of a Chrome browser window displaying the GitHub Settings page for the repository 'adsoftsito/project-demo'. The URL in the address bar is <https://github.com/adsoftsito/project-demo/settings/access>. The page shows the 'Manage access' tab selected. A modal dialog titled 'Invite a collaborator to project-demo' is open, showing a search input field with 'kubee' typed in, and a list of results. The top result is 'kubee Invite collaborator' with a blue background, and below it is another result 'Kubeeus Invite collaborator'. Below the modal, a message says 'You haven't invited any collaborators yet' with a 'Invite a collaborator' button.

13.- Create a Project in github.com, select Automated Kanban with Reviews template.



14.- Delete default cards in “To do” Column



15.- create “simple calculator” milestone

New milestone

Create a new milestone to help organize your issues and pull requests. Learn more about [milestones and issues](#).

Title
simple calculator

Due date (optional)
08/30/2020

Description
simple operations:
addition
subtraction
multiplication
division|

Create milestone

16.- créer “scientific calculator” milestone

New milestone

Create a new milestone to help organize your issues and pull requests. Learn more about [milestones and issues](#).

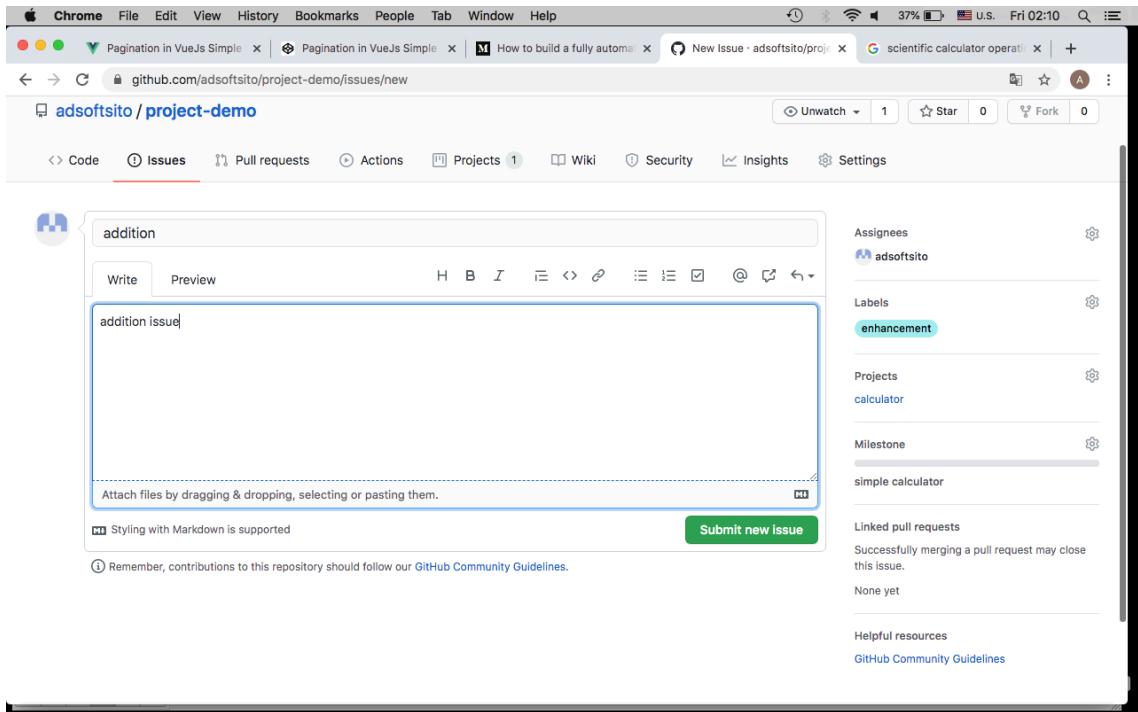
Title
scientific calculator

Due date (optional)
09/01/2020

Description
operations :
- exponents
- log
- natural log (\ln)
- sin
- cos
- tan|

Create milestone

17.- Add addition issue's to calculator Project



18.- complete issue's according the next table

issue name	write	assignees	labels	Project	milestone
addition	add ..	developer1	enhancement	calculator	simple calculator
substraction	sub ...	developer2	enhancement	calculator	simple calculator
multiplication	mul ...	developer3	enhancement	calculator	simple calculator
division	div ...	developer4	enhancement	calculator	simple calculator
exponents	exp ...	developer1	enhancement	calculator	scientific calculator
log	log ...	developer2	enhancement	calculator	scientific calculator
natural log	In ...	developer3	enhancement	calculator	scientific calculator
sin	sin ...	developer4	enhancement	calculator	scientific calculator
cos	cos ...	developern	enhancement	calculator	scientific calculator
tan	In ...	developern	enhancement	calculator	scientific calculator

19.- issue's should looks like :

The screenshot shows a list of ten open issues on a GitHub page. All issues are labeled 'enhancement' and are categorized under the 'calculator' project. The issues are listed in descending order of creation time:

- #10 tan enhancement
- #9 cos enhancement
- #8 sin enhancement
- #7 natural log enhancement
- #6 log enhancement
- #5 exponents enhancement
- #4 division enhancement
- #3 multiplication enhancement
- #2 subtraction enhancement
- #1 addition enhancement

Each issue includes a link to its details and a small profile picture of the author.

21.- Calculator project board looks like :

The screenshot shows a GitHub project board titled 'Calculator'. The board has four columns: 'To do', 'In progress', 'Review in progress', and 'Reviewer approved'. Each column contains cards representing the issues listed above. The 'To do' column has four cards: 'addition', 'subtraction', 'multiplication', and 'division'. The 'In progress' column has one card: 'calculator'. The 'Review in progress' and 'Reviewer approved' columns are currently empty. A search bar at the top right allows filtering of cards.

22.- Create a branch for each issue assigned to you

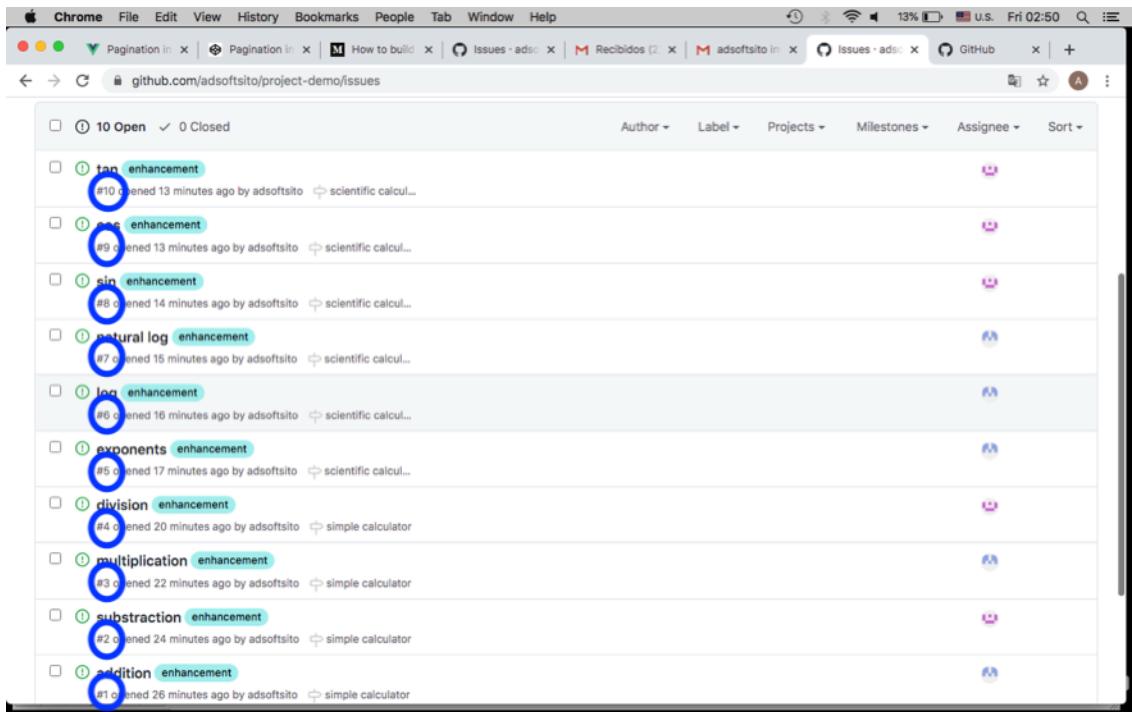
```
$ git checkout develop
```

```
$ git branch <issue>
# example : git branch addition
```

```
$ git checkout <issue>
# example: git checkout addition

# show current branch
$ git branch
```

23.- Add some commits with issue reference #issuenumber, check issue number before



```
$ vi <issue>.java
# example: vi addition.java
```

```
/*
insert code here...
*/
```

```
:wq
```

```
$ git add <issue>.java
# example: git add addition.java
```

```
$ git commit -m "add <issue> base code #<issuenumber>"
$ example: git commit -m "add addition base code #1"
```

```
# check local status vs remote status
$ git status
```

```
$ git push -u origin <issue>
$ example: git push -u origin addition
```

24.- Check issue log

The screenshot shows a GitHub issue log for issue #1. The commit history is as follows:

- adsoftsito commented 4 days ago: addition issue
- adsoftsito added the enhancement label 4 days ago
- adsoftsito added this to the simple calculator milestone 4 days ago
- adsoftsito self-assigned this 4 days ago
- adsoftsito added this to To do in calculator via automation 4 days ago
- adsoftsito added a commit that referenced this issue 3 minutes ago (circled in blue): add addition base code #1

The right sidebar shows the issue details: Assignees (adsoftsito), Labels (enhancement), Projects (calculator), Milestone (simple calculator), and Linked pull requests (None yet). Notifications are customized, and the URL is https://github.com/adsoftsito/project-demo/issues/1#event-3702713565.

25.- Add more commits with #<issuenumber> and verify issue status

```
$ vi <issue>.java
# example: vi addition.java

/*
Author: adsoft
Date: Sep 2, 2020
*/

/*
insert code here...
*/
:wq

$ git commit -a -m "add author and date #<issuenumber>"
$ example: git commit -a -m "add autor an date #1"

$ git status # check local status vs remote status
#####
$ vi <issue>.java
# example: vi addition.java

/*
Author: adsoft
Date: Sep 2, 2020
*/

/*
insert code here...
```

```

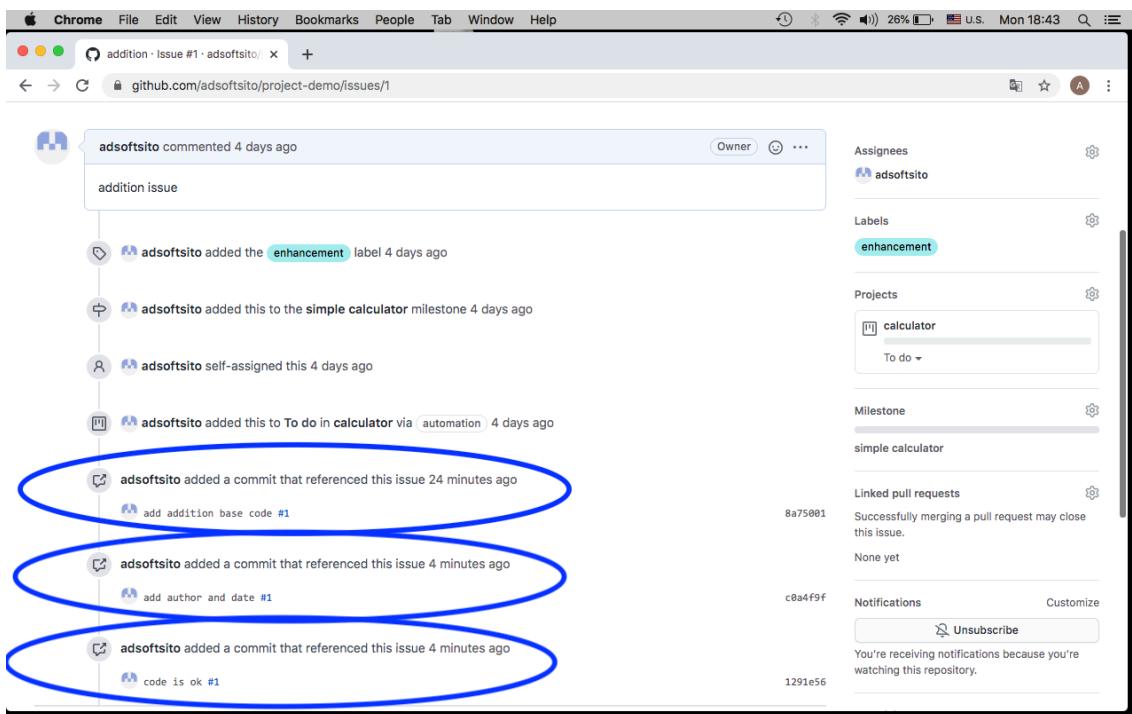
*/
/*
code is ready !
*/
:wq

$ git commit -a -m "code is ok #<issuenumber>"
$ example: git commit -a -m "code is ok #1"

$ git status # check local status vs remote status

$ git push -u origin <issue>
$ example: git push -u origin addition

```



26.- Finally when code is ready, create a Pull Request from this <issue> to branch develop

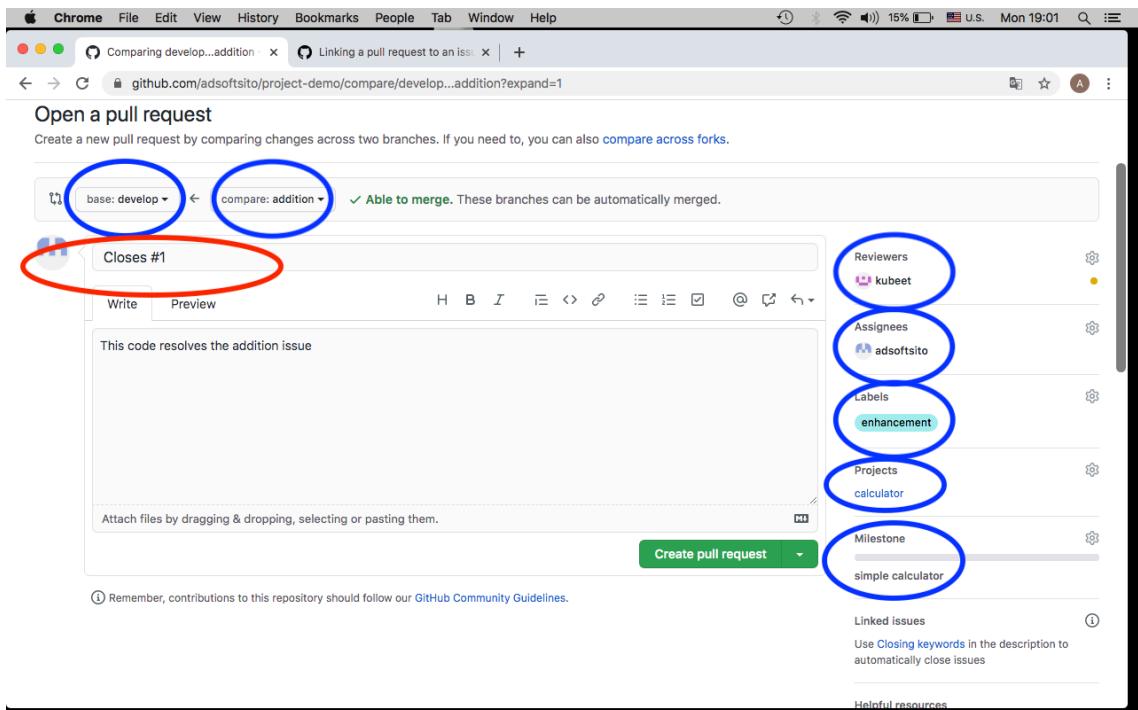
The screenshot shows a GitHub repository page for 'adsoftsito / project-demo'. At the top, there's a yellow banner indicating 'addition had recent pushes 12 minutes ago'. Below it, there's a green 'Compare & pull request' button, which is also circled in blue. On the left, there's a dropdown menu set to 'addition'. The main area shows a list of files: 'README.md' (updated 13 minutes ago), 'addition.java' (updated 13 minutes ago). To the right, there are sections for 'About', 'Releases', and 'Packages'.

NOTE: close issue with some supported keywords

The screenshot shows a GitHub documentation page titled 'Linking a pull request to an issue using a keyword'. The sidebar has a 'MANAGING YOUR WORK ON GITHUB' section with links like 'About issues', 'Creating an issue', etc. The main content lists supported keywords for closing issues: 'close', 'closes', 'closed', 'fix', 'fixes', 'fixed', 'resolve', 'resolves', and 'resolved'. It then explains the syntax for closing keywords based on whether the issue is in the same repository or not. A table provides examples:

Linked issue	Syntax	Example
Issue in the same repository	<code>KEYWORD #ISSUE-NUMBER</code>	<code>Closes #10</code>
Issue in a different repository	<code>KEYWORD OWNER/REPOSITORY#ISSUE-NUMBER</code>	<code>Fixes octo-org/octo-repo#100</code>
Multiple issues	Use full syntax for each issue	<code>Resolves #10, resolves #123, resolves octo-org/octo-repo#100</code>

Verify PR contains supported keywords, **reviewers**, assignees, labels, projects and milestone



27.- Verify new PR is “In Progress” Column

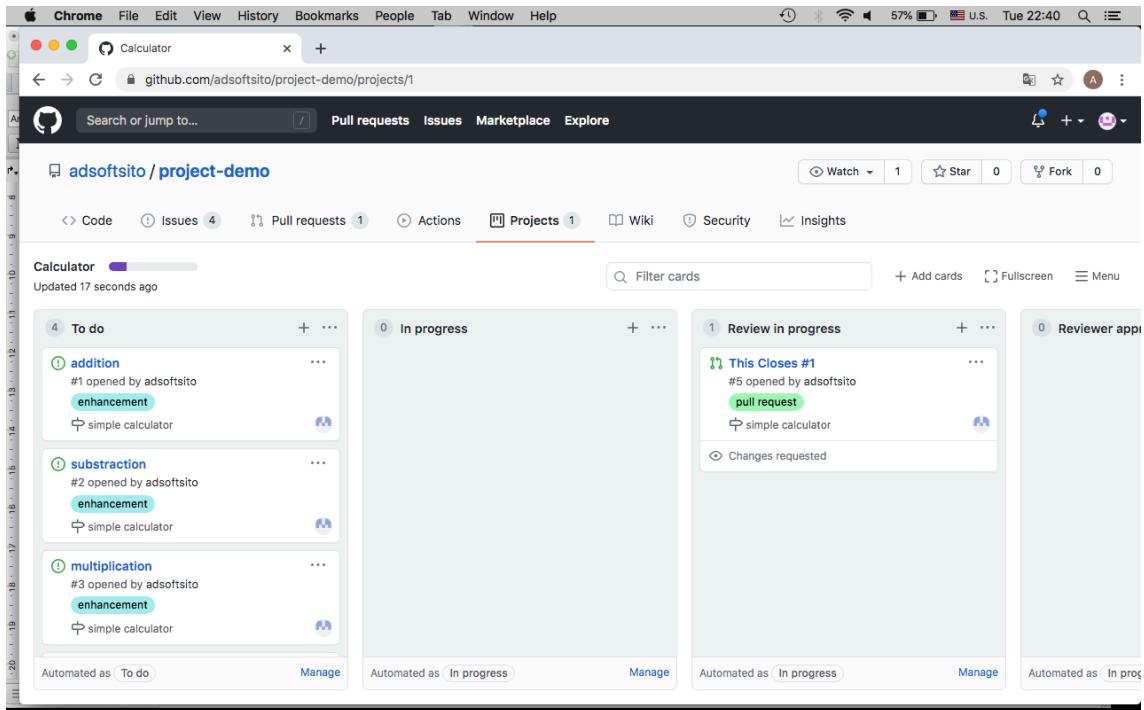
28.- Reviewer should add Review for this issue

A screenshot of a GitHub pull request page. At the top, there's a message: "adsoftsito requested your review on this pull request." Below it, the title "This Closes #1 #5" is shown. A green button labeled "Add your review" is circled in red. The pull request details show 3 commits from "adsoftsito" and 1 check from "kubee". The right sidebar shows the assignee "adsoftsito" and the label "pull request".

29.- Reviewer might “Request Changes”

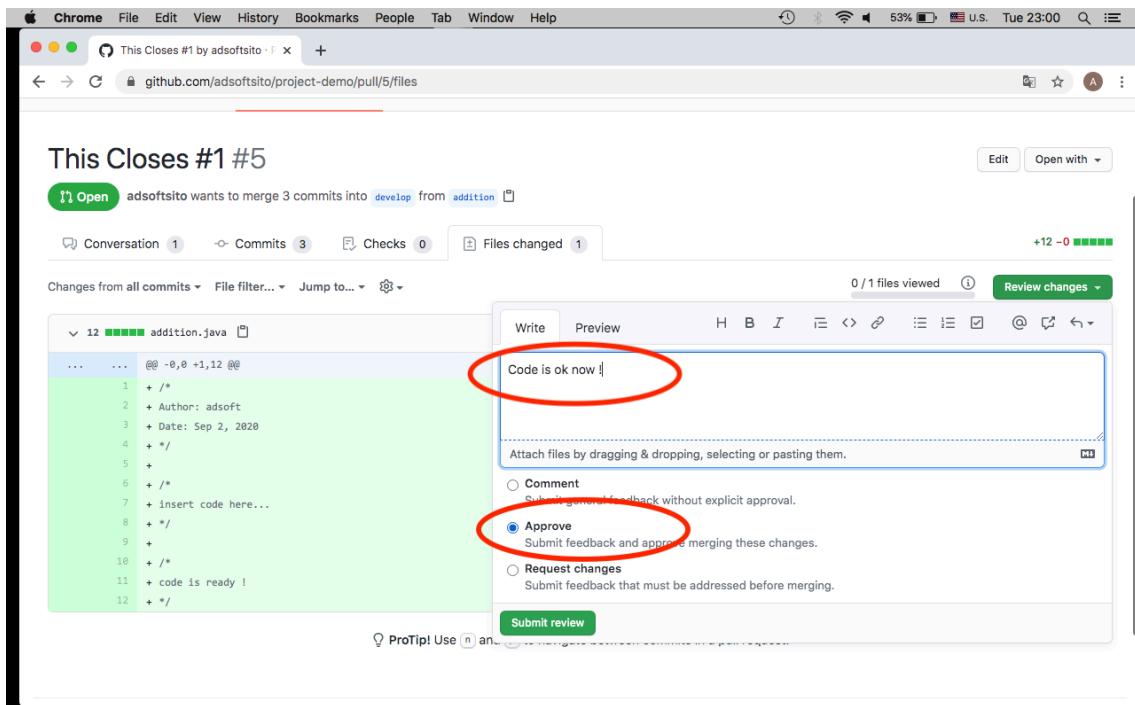
A screenshot of a GitHub pull request review interface. A comment box contains the text "Code must be refactoring!" which is circled in red. Below the comment box, there are three radio button options: "Comment", "Approve", and "Request changes". The "Request changes" option is selected and highlighted with a red circle. A green "Submit review" button is at the bottom.

30.- Verify PR moves to “Review in Progress”

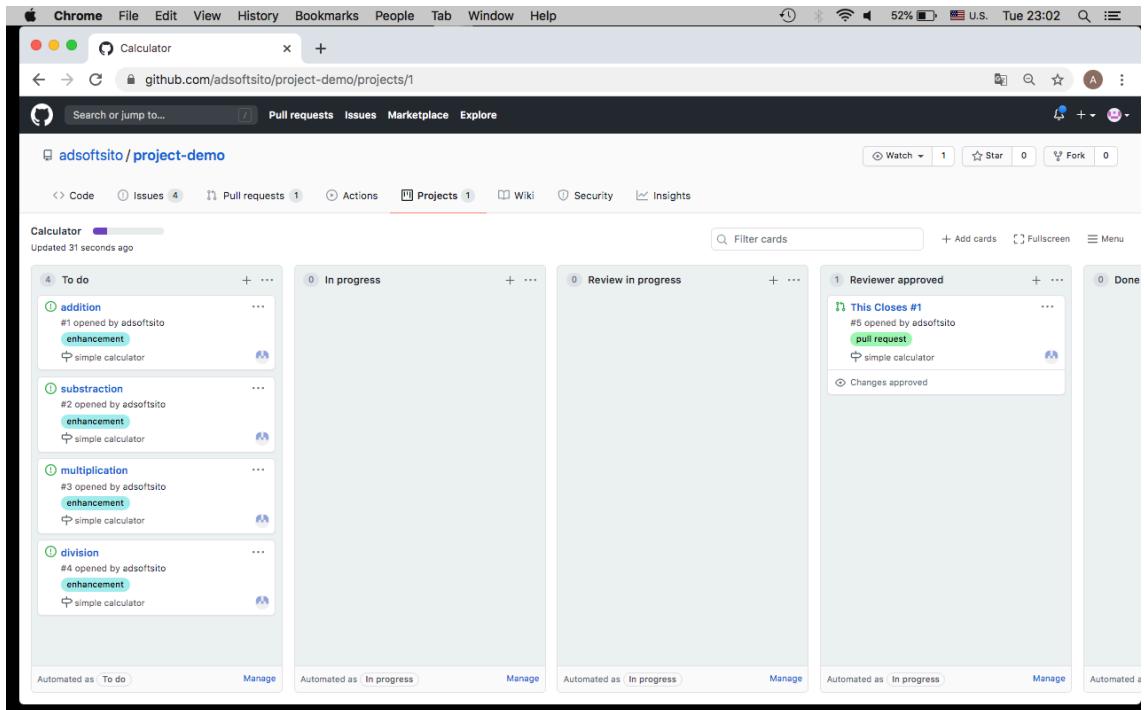


31.- Add more commits to resolve this “Request Changes” (optionally)

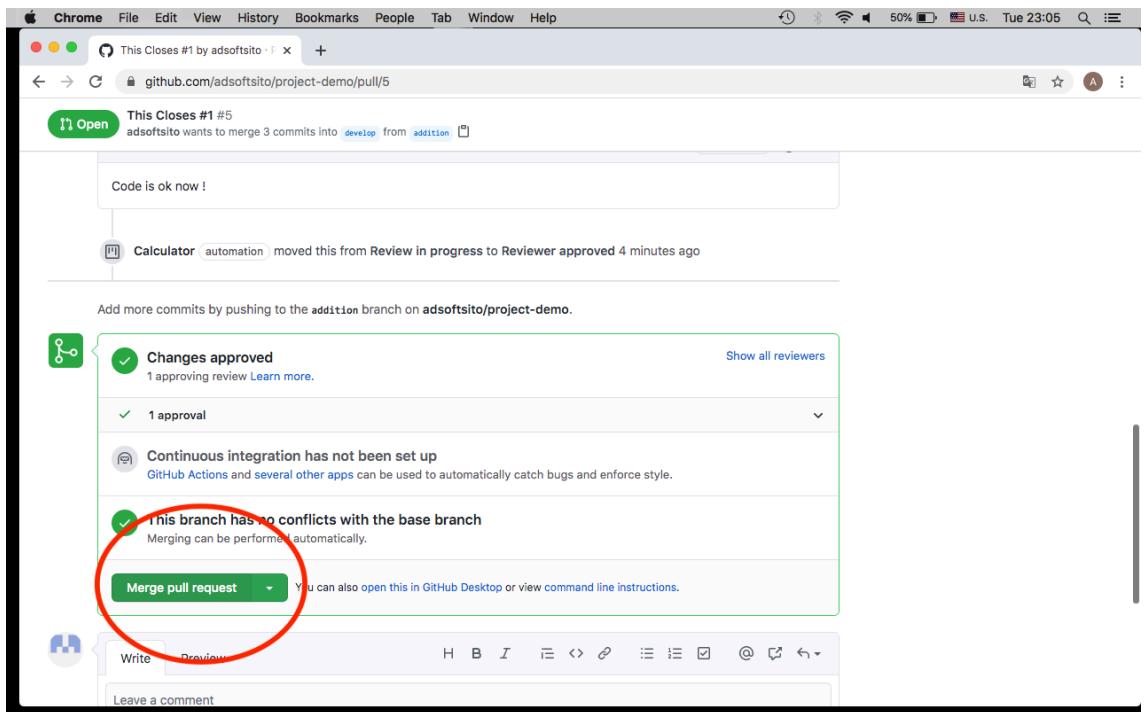
32.- Now, Reviewer should approve issue.



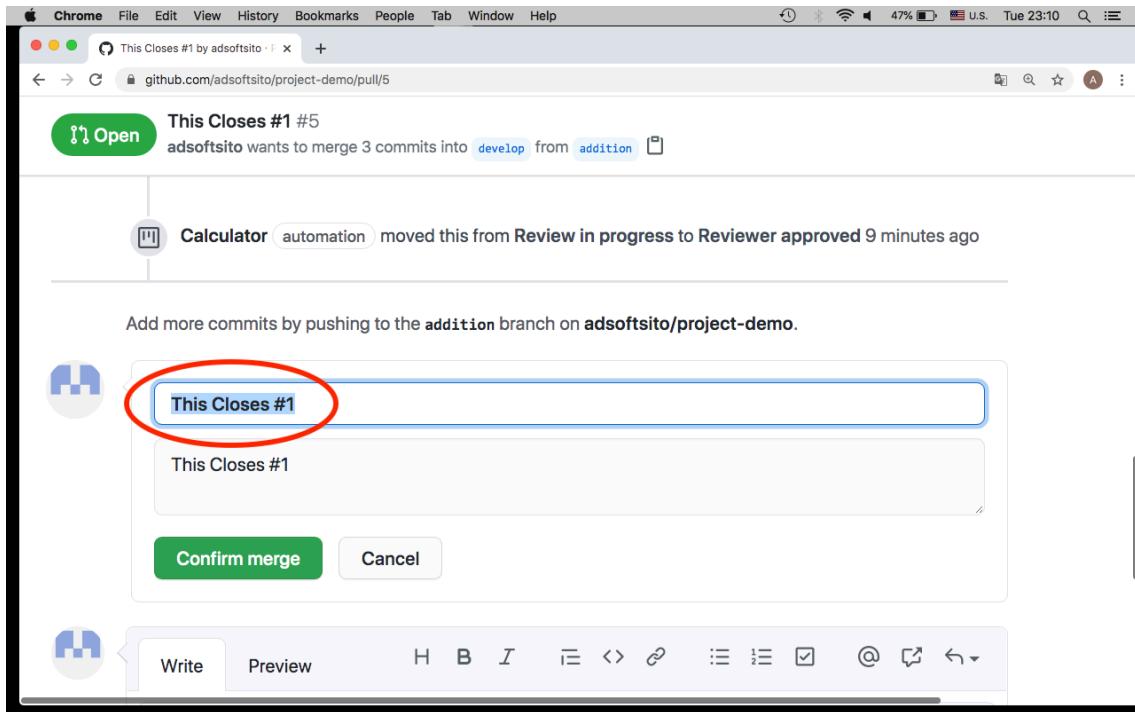
33.- With PR approved, PR moves to “Reviewer Approved”



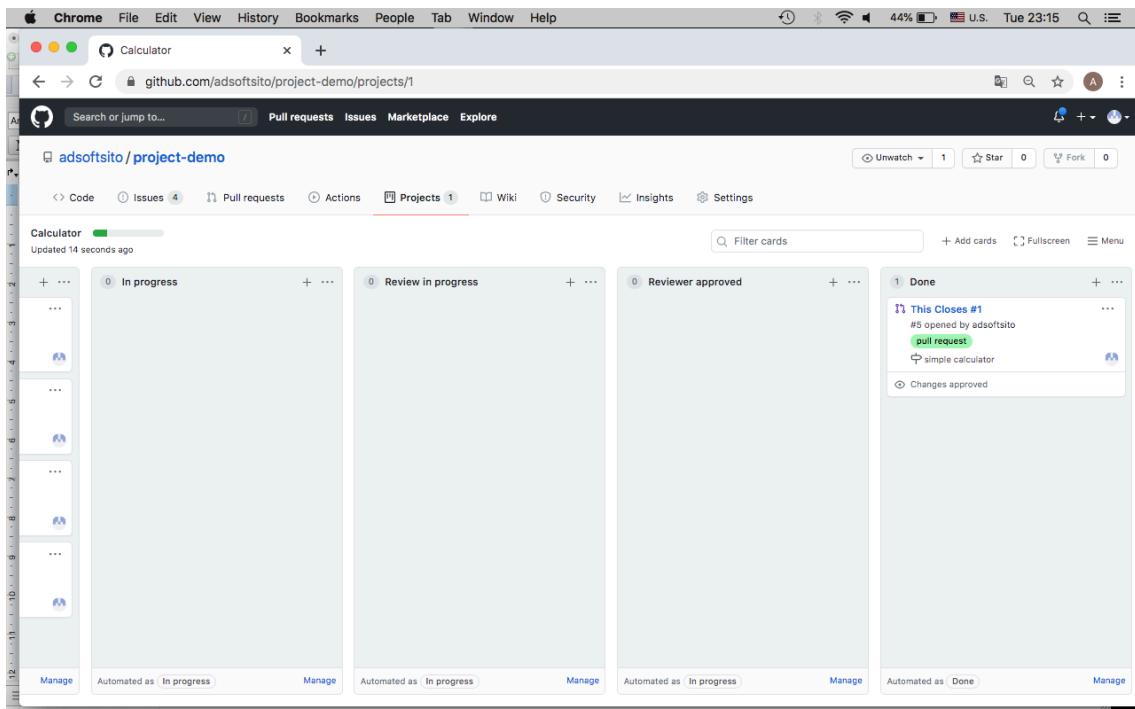
34.- Finally issue owner, must “Merge Pull Request”



35.- Make sure commit message contain keywords : Closes #<n>, Resolves #<n>, Fixes#<n>

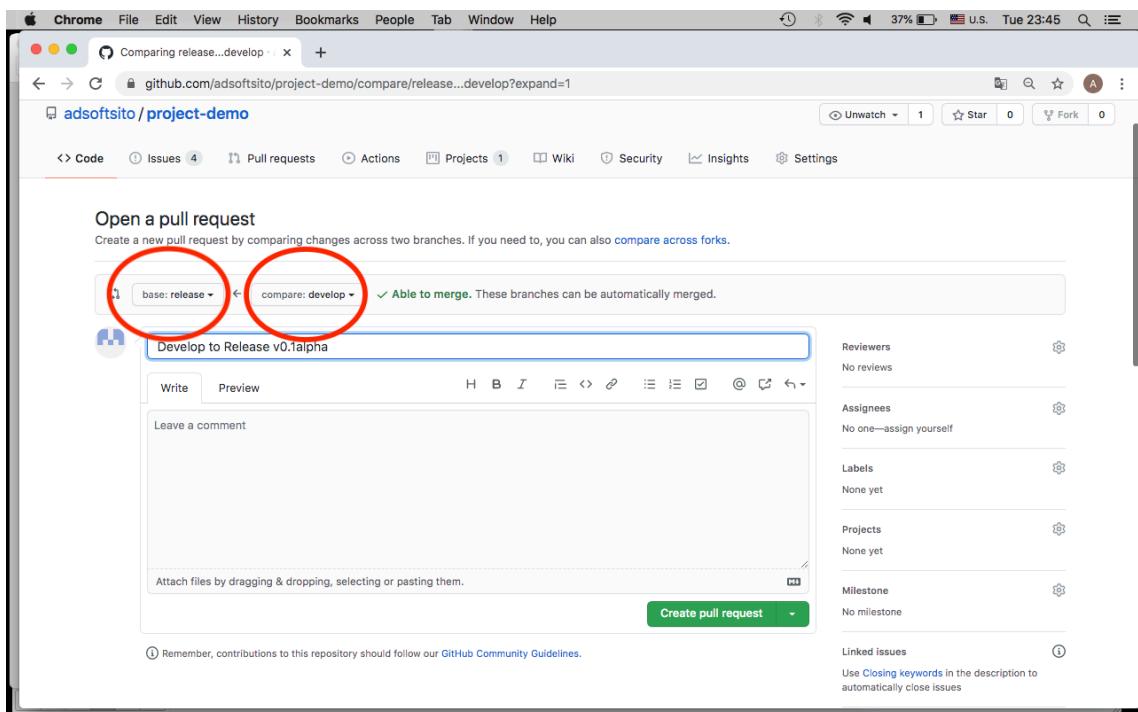


36.- Now, PR moves to “Done” column.

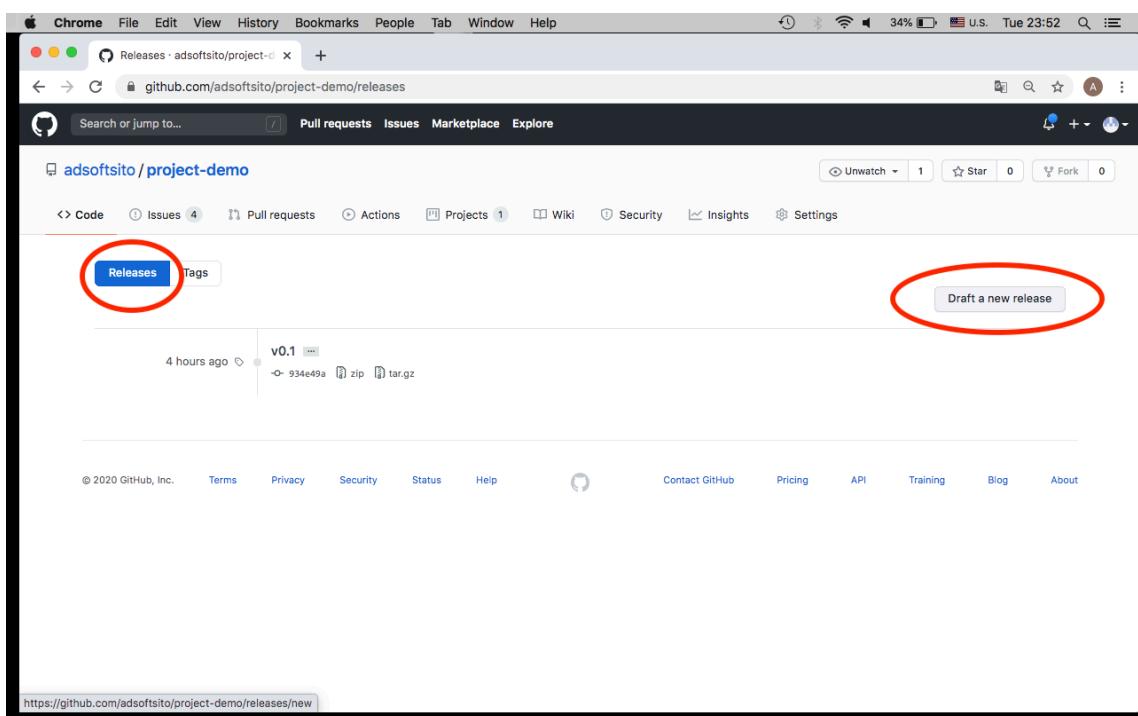


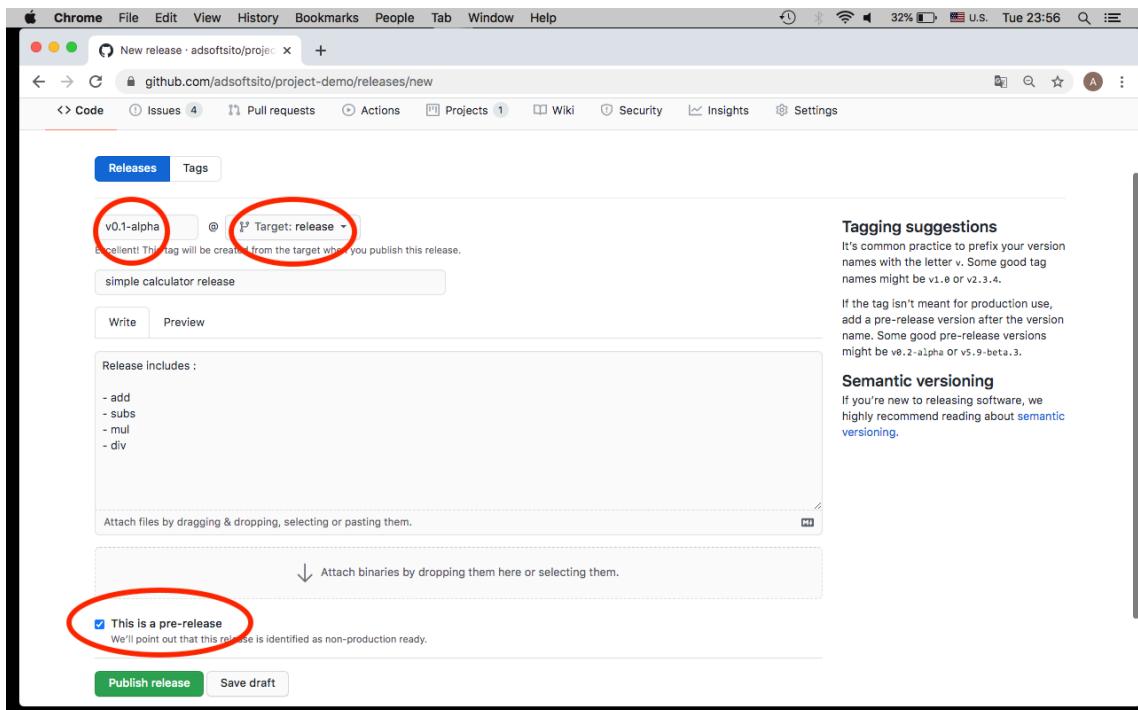
37.- Repeat this process for each issue

38.- Create a new PR from **develop** to **release** branch and confirm Merge.

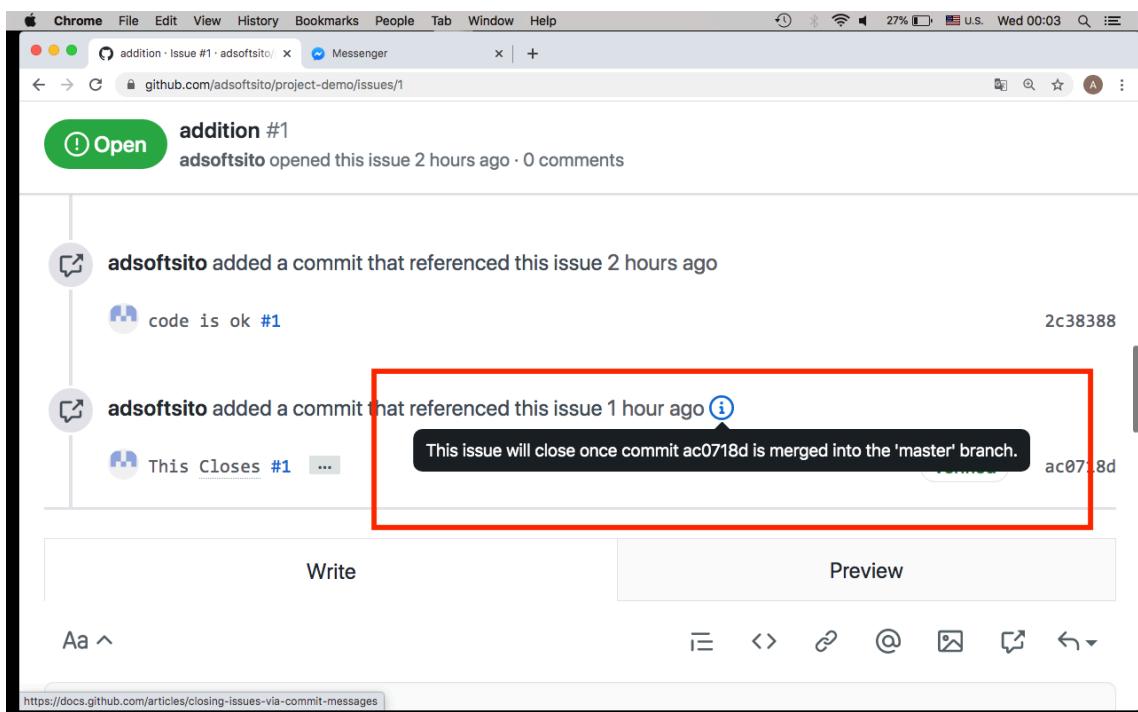


39.- Create a Release

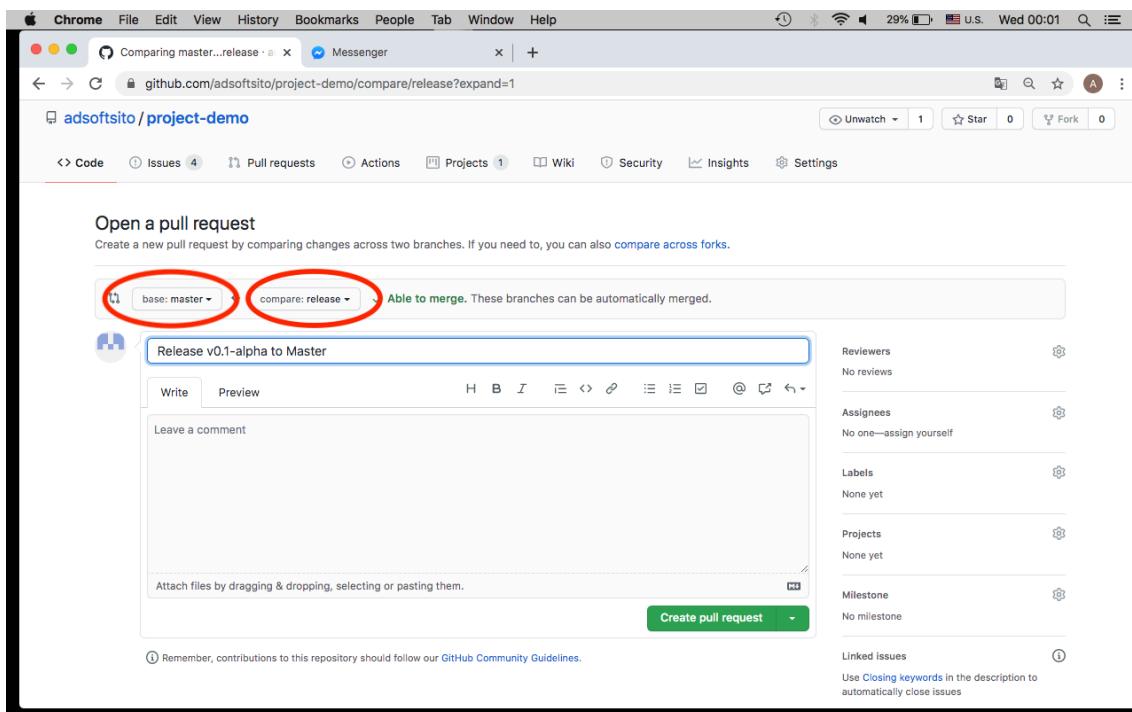




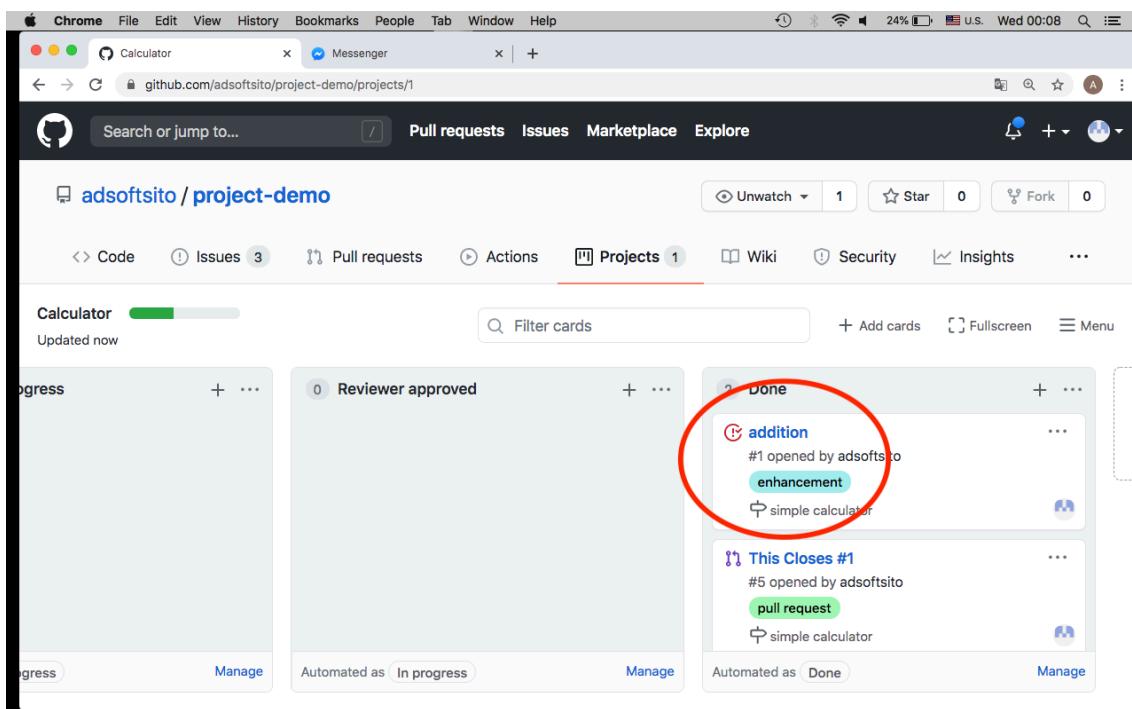
40.- Issues are “closed” but continues in “To do” column, check issue log.



41.- Publish to Production, create PR from **release** to **master** branch



42.- Finally, issues are closed when code is merged in master branch



43.- Now, Check network graph

