Universidad Politécnica de Durango

Institución Pública de la más alta calidad

SOFTWARE MAINTENDANCE

Student name:

Delgado Espinoza Alexis Osvaldo - Leader

Ponce Barraza Javier Eulogio

Pérez Calzada Tania Jaqueline

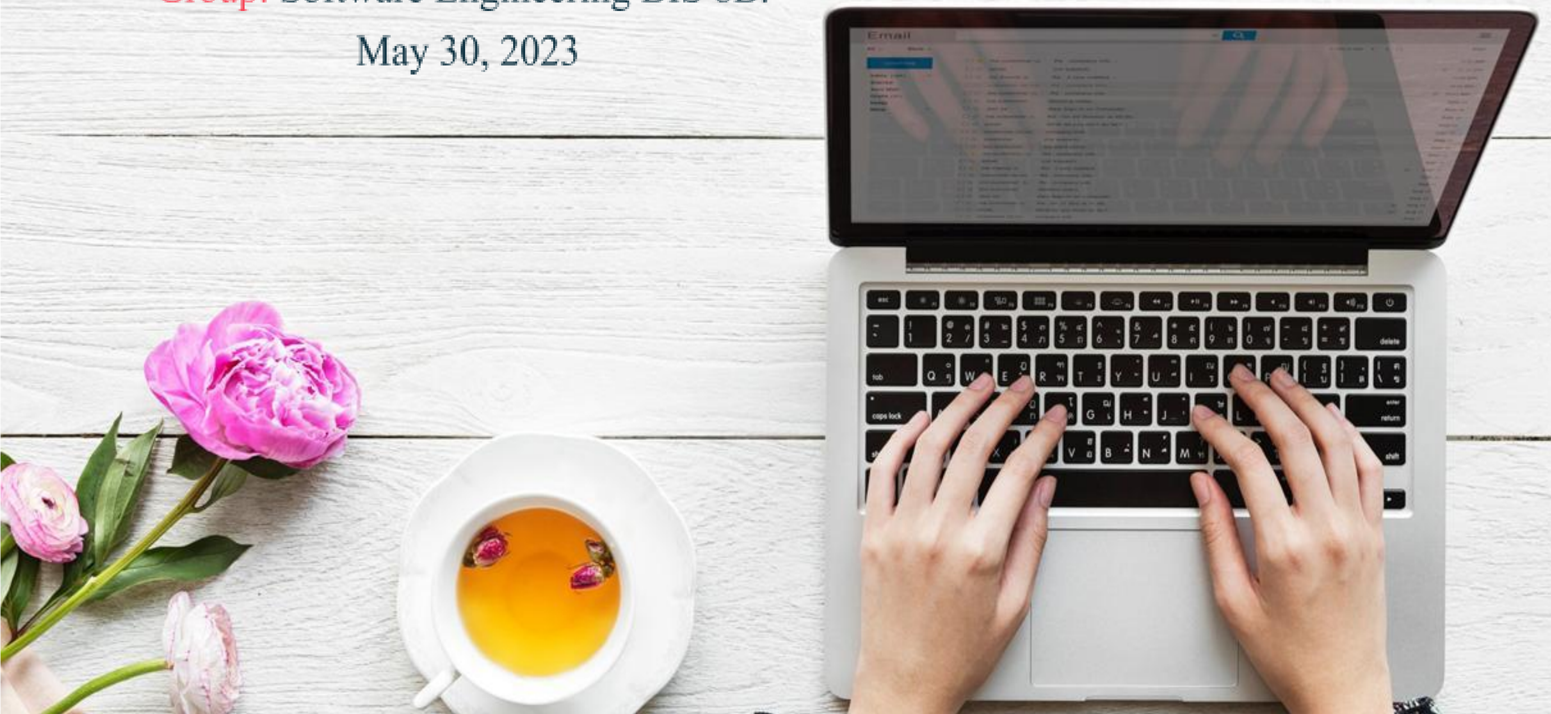
Sanchez Barraza Ahastari Yamaressi

Hernandez Avila Angel Guadalupe

Teacher: ALONDRA YARELI RAMOS FLORES

Group: Software Engineering BIS 8B.

May 30, 2023



INTRODUCTION

GitHub is a potential tool for many different reasons like:

Version Control: GitHub provides a robust version control functionality through Git. This allows developers to track changes, manage multiple versions of the code, and collaborate seamlessly with other team members. With Git, developers can easily revert to previous versions, identify specific changes, and merge code changes efficiently.

Collaboration and Teamwork: GitHub offers a collaborative environment for developers to work together on a project. Multiple team members can contribute to the codebase simultaneously by creating branches, making changes, and proposing them for review.

Issue Tracking: GitHub provides an inherent issue tracking system, enabling project members to report bugs, suggest improvements, or discuss new features. The issue tracking system helps streamline the software maintenance process and store all discussions and updates in one place.

Benefits of using Miro:

- Using Miro has justifications that depend on the context and needs of the work team.
- Miro enables efficient and productive collaboration, facilitating communication and problem solving.
- The tool offers visual tools to organize and represent information in a clear way, which helps to understand, identify patterns and establish connections.
- Miro is flexible and adaptable, allowing you to customize dashboards and use predefined templates for different purposes.
- Miro's intuitive interface and accessibility from different devices and operating systems make it easy to use and collaborate anytime, anywhere.
- Miro integrates with other popular tools, making it easy to import, export and synchronize content across platforms.
- Using Miro can save time and resources by centralizing information and avoiding face-to-face meetings and lengthy emails.

Disadvantages of Miro:

- Miro has an initial learning curve, which may require time to become familiar with all of its functionality.
- Depending on the features needed, Miro may have a cost associated with it, which can be a disadvantage for those with limited budgets.
- The tool requires a stable Internet connection, which may make it difficult to use in areas with limited access or connectivity issues.
- The free version of Miro has limitations in terms of number of dashboards, collaborators, storage and advanced features, which may affect larger teams or those with specific needs.
- In more traditional work environments, there may be resistance to change and a preference for conventional methods and tools, making Miro adoption difficult and requiring adaptation by the team.
- There are concerns about data security and privacy when using a cloud platform such as Miro, even if security measures are implemented, there is always an inherent risk in sharing information online.

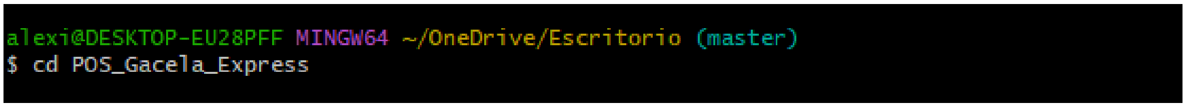
Some differences with other tools are as follows:

- Miro offers a wider range of collaboration functionality and features compared to Microsoft Whiteboard.
- Miro focuses on visual collaboration and idea organization, while Trello focuses on project management and task tracking.
- Miro offers a wider range of visual functionalities compared to Lucidchart, which specializes in technical diagramming.
- Miro is a more comprehensive and versatile collaboration platform compared to Google Jamboard, which is designed specifically for interactive whiteboards.

STEPS TO UPLOAD THE PROJECT IN GITHUB

Cd POS_Gacela_Express

When the command "cd POS_Gacela_Express" is executed, the current working directory in the command line or terminal is changed to the "POS_Gacela_Express" directory. From that point on, any command or action performed will apply within that specific directory.

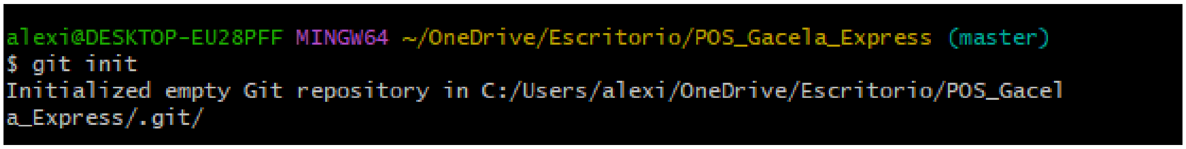


```
alexi@DESKTOP-EU28PFF MINGW64 ~/OneDrive/Escritorio (master)
$ cd POS_Gacela_Express
```

$git init

The command is used to initialize a new repository. When executed, it sets up a new version control system in the current directory, creating a hidden subfolder called .git. This subfolder is where all the related version control information will be stored.

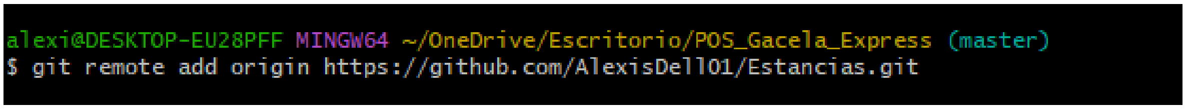
By using git init, the local repository is established as an independent Git repository. From this point, you can start tracking changes in files, creating commits, and utilizing other Git functionalities to manage the project's history.



```
alexi@DESKTOP-EU28PFF MINGW64 ~/OneDrive/Escritorio/POS_Gacela_Express (master)
$ git init
Initialized empty Git repository in C:/Users/alexi/OneDrive/Escritorio/POS_Gacel
a_Express/.git/
```

$git remote add origin https://github.com7AlexisDell01/Estancias.git

By executing this command, a connection is established between the local repository and the remote repository. The name origin refers to the connection, allowing future commands such as git push or git pull to reference that remote repository.

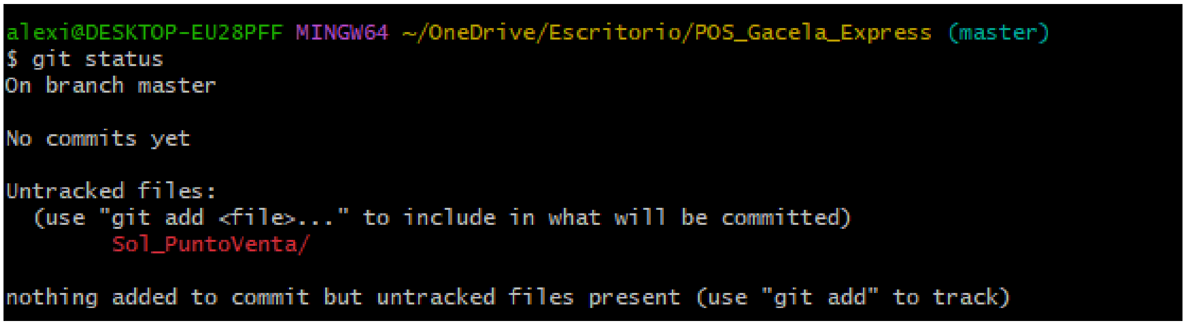


```
alexi@DESKTOP-EU28PFF MINGW64 ~/OneDrive/Escritorio/POS_Gacela_Express (master)
$ git remote add origin https://github.com/AlexisDell01/Estancias.git
```

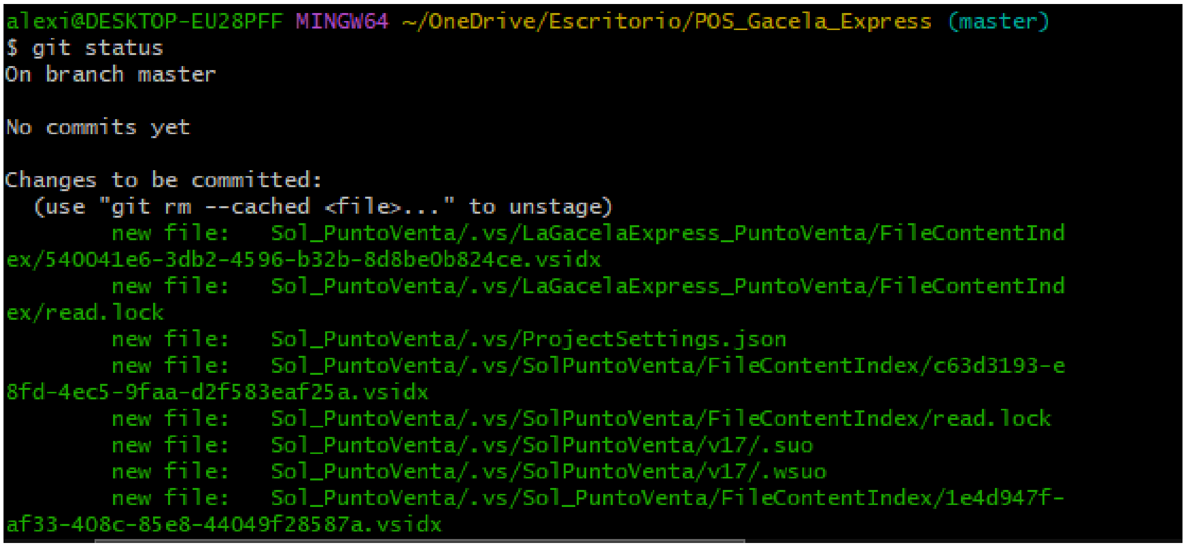
$git status

provides you with a description of the current state of the repository, allowing informed decisions about which changes to commit, which files to add for tracking, or what actions to take regarding branches.

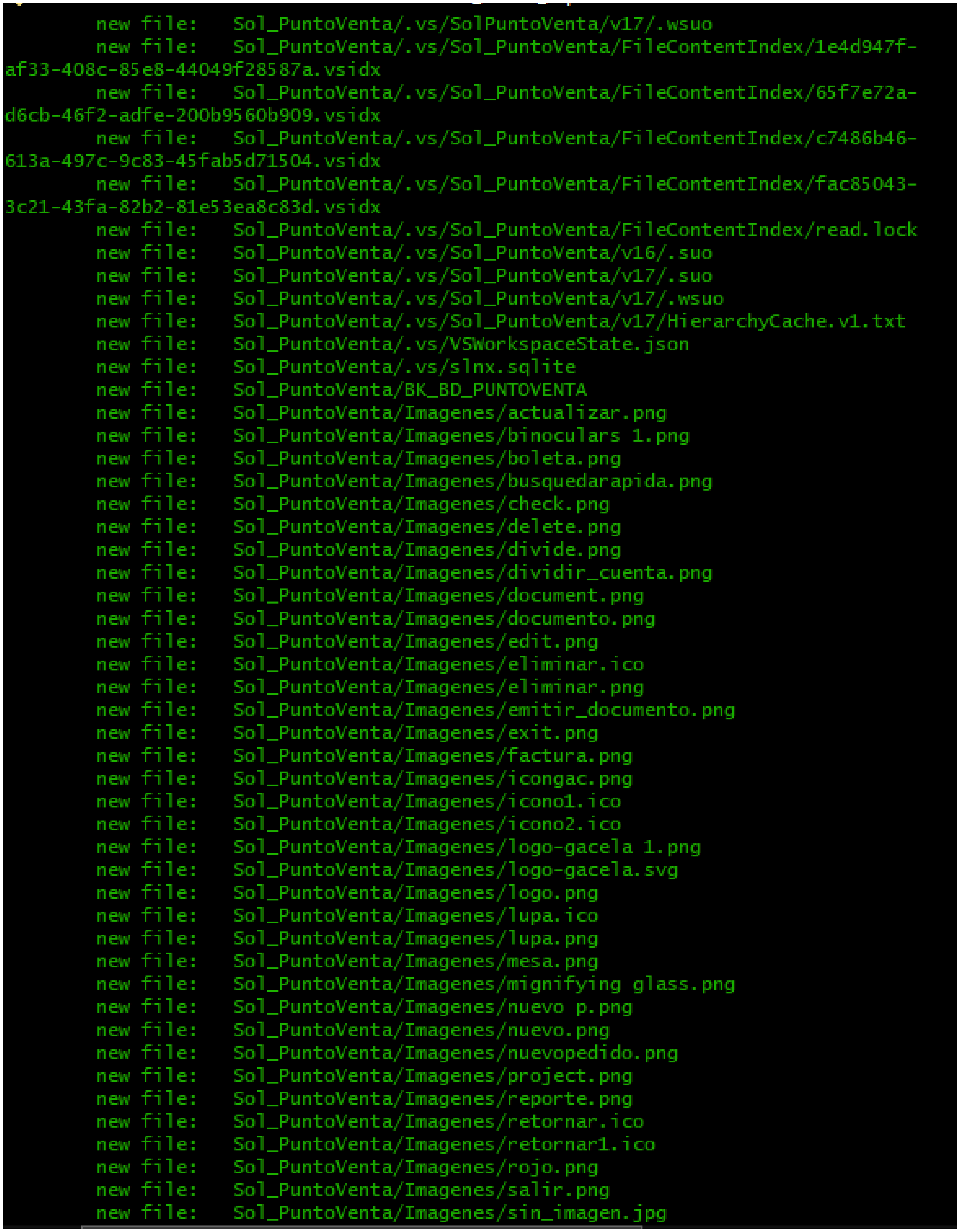
Additionally, git status can provide additional information based on the configuration and specific status of the repository you are working in.



```
alexi@DESKTOP-EU28PFF MINGW64 ~/OneDrive/Escritorio/POS_Gacela_Express (master)
$ git status
On branch master

No commits yet

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

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



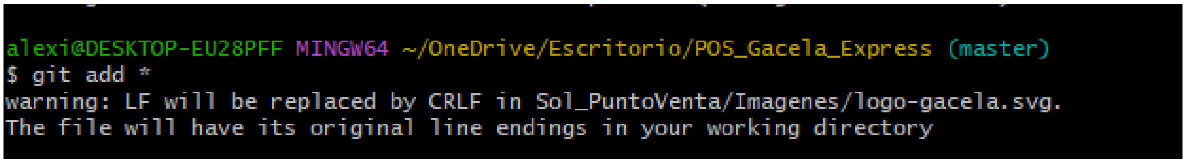
```
alexi@DESKTOP-EU28PFF MINGW64 ~/OneDrive/Escritorio/POS_Gacela_Express (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Sol_PuntoVenta/.vs/LaGacelaExpress_PuntoVenta/FileContentInd
ex/540041e6-3db2-4596-b32b-8d8be0b824ce.vsidx
        new file:   Sol_PuntoVenta/.vs/LaGacelaExpress_PuntoVenta/FileContentInd
ex/read.lock
        new file:   Sol_PuntoVenta/.vs/ProjectSettings.json
        new file:   Sol_PuntoVenta/.vs/SolPuntoVenta/FileContentIndex/c63d3193-e
8fd-4ec5-9faa-d2f583eaf25a.vsidx
        new file:   Sol_PuntoVenta/.vs/SolPuntoVenta/FileContentIndex/read.lock
        new file:   Sol_PuntoVenta/.vs/SolPuntoVenta/v17/.suo
        new file:   Sol_PuntoVenta/.vs/SolPuntoVenta/v17/.wsuo
        new file:   Sol_PuntoVenta/.vs/Sol_PuntoVenta/FileContentIndex/1e4d947f-
af33-408c-85e8-44049f28587a.vsidx
```



```
        new file:   Sol_PuntoVenta/.vs/SolPuntoVenta/v17/.wsuo
        new file:   Sol_PuntoVenta/.vs/Sol_PuntoVenta/FileContentIndex/1e4d947f-
af33-408c-85e8-44049f28587a.vsidx
        new file:   Sol_PuntoVenta/.vs/Sol_PuntoVenta/FileContentIndex/65f7e72a-
d6cb-46f2-adfe-200b9560b909.vsidx
        new file:   Sol_PuntoVenta/.vs/Sol_PuntoVenta/FileContentIndex/c7486b46-
613a-497c-9c83-45fab5d71504.vsidx
        new file:   Sol_PuntoVenta/.vs/Sol_PuntoVenta/FileContentIndex/fac85043-
3c21-43fa-82b2-81e53ea8c83d.vsidx
        new file:   Sol_PuntoVenta/.vs/Sol_PuntoVenta/FileContentIndex/read.lock
        new file:   Sol_PuntoVenta/.vs/Sol_PuntoVenta/v16/.suo
        new file:   Sol_PuntoVenta/.vs/Sol_PuntoVenta/v17/.suo
        new file:   Sol_PuntoVenta/.vs/Sol_PuntoVenta/v17/.wsuo
        new file:   Sol_PuntoVenta/.vs/Sol_PuntoVenta/v17/HierarchyCache.v1.txt
        new file:   Sol_PuntoVenta/.vs/VSWorkspaceState.json
        new file:   Sol_PuntoVenta/.vs/slnx.sqlite
        new file:   Sol_PuntoVenta/BK_BD_PUNTOVENTA
        new file:   Sol_PuntoVenta/Imagenes/actualizar.png
        new file:   Sol_PuntoVenta/Imagenes/binoculars 1.png
        new file:   Sol_PuntoVenta/Imagenes/boleta.png
        new file:   Sol_PuntoVenta/Imagenes/busquedarapida.png
        new file:   Sol_PuntoVenta/Imagenes/check.png
        new file:   Sol_PuntoVenta/Imagenes/delete.png
        new file:   Sol_PuntoVenta/Imagenes/divide.png
        new file:   Sol_PuntoVenta/Imagenes/dividir_cuenta.png
        new file:   Sol_PuntoVenta/Imagenes/document.png
        new file:   Sol_PuntoVenta/Imagenes/documento.png
        new file:   Sol_PuntoVenta/Imagenes/edit.png
        new file:   Sol_PuntoVenta/Imagenes/eliminar.ico
        new file:   Sol_PuntoVenta/Imagenes/eliminar.png
        new file:   Sol_PuntoVenta/Imagenes/emitir_documento.png
        new file:   Sol_PuntoVenta/Imagenes/exit.png
        new file:   Sol_PuntoVenta/Imagenes/factura.png
        new file:   Sol_PuntoVenta/Imagenes/icongac.png
        new file:   Sol_PuntoVenta/Imagenes/icono1.ico
        new file:   Sol_PuntoVenta/Imagenes/icono2.ico
        new file:   Sol_PuntoVenta/Imagenes/logo-gacela 1.png
        new file:   Sol_PuntoVenta/Imagenes/logo-gacela.svg
        new file:   Sol_PuntoVenta/Imagenes/logo.png
        new file:   Sol_PuntoVenta/Imagenes/lupa.ico
        new file:   Sol_PuntoVenta/Imagenes/lupa.png
        new file:   Sol_PuntoVenta/Imagenes/mesa.png
        new file:   Sol_PuntoVenta/Imagenes/mignifying glass.png
        new file:   Sol_PuntoVenta/Imagenes/nuevo p.png
        new file:   Sol_PuntoVenta/Imagenes/nuevo.png
        new file:   Sol_PuntoVenta/Imagenes/nuevopedido.png
        new file:   Sol_PuntoVenta/Imagenes/project.png
        new file:   Sol_PuntoVenta/Imagenes/reporte.png
        new file:   Sol_PuntoVenta/Imagenes/retornar.ico
        new file:   Sol_PuntoVenta/Imagenes/retornar1.ico
        new file:   Sol_PuntoVenta/Imagenes/rojo.png
        new file:   Sol_PuntoVenta/Imagenes/salir.png
        new file:   Sol_PuntoVenta/Imagenes/sin_imagen.jpg
```

$git add *

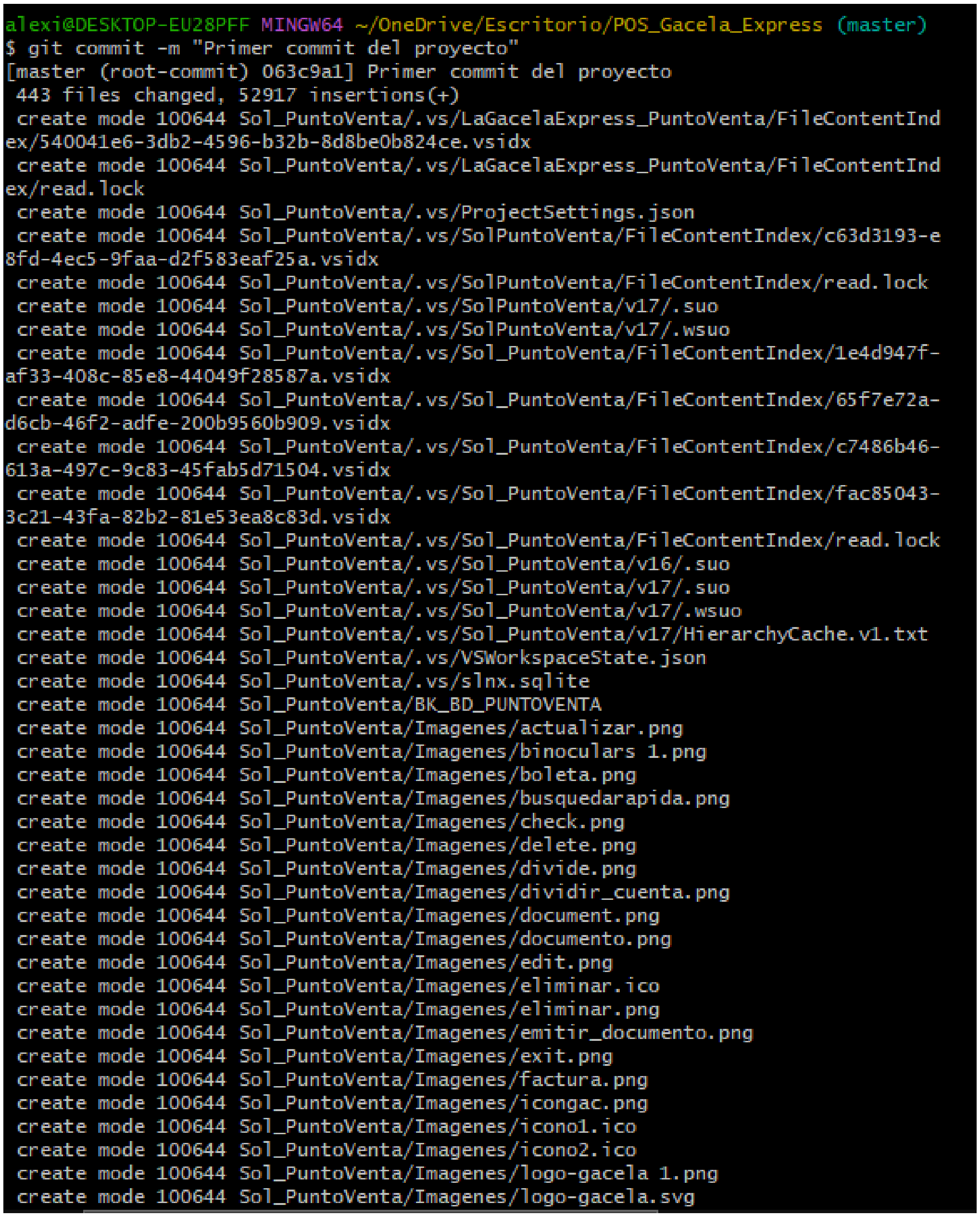
It is used to add all changes made to the files in the current working directory to the staging area.



```
alexi@DESKTOP-EU28PFF MINGW64 ~/OneDrive/Escritorio/POS_Gacela_Express (master)
$ git add *
warning: LF will be replaced by CRLF in Sol_PuntoVenta/Imagenes/logo-gacela.svg.
The file will have its original line endings in your working directory
```

$git commit -m "First commit of the Project"

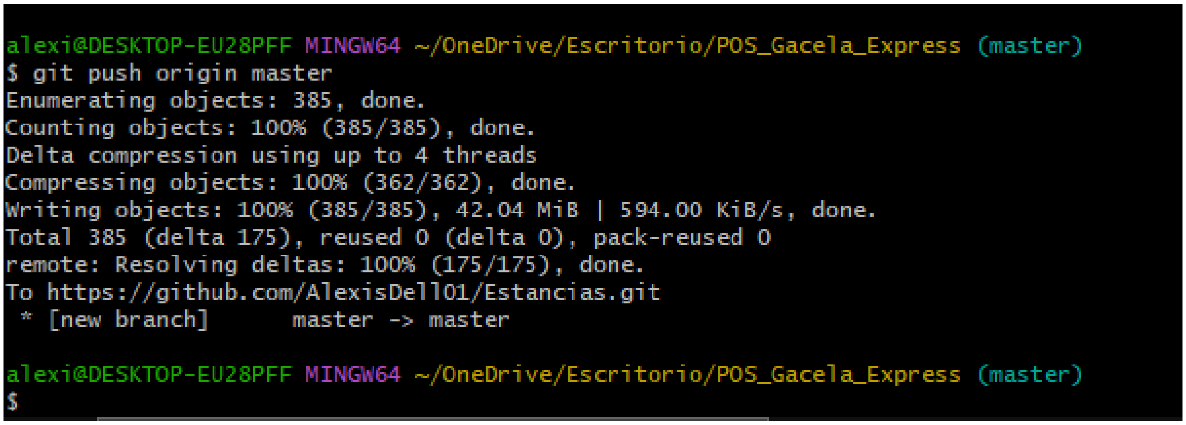
When this command is executed, it takes all the files that have been added or modified in the staging area and creates a new commit with those changes. The commit is saved in the repository's history and is assigned a unique identifier (hash) that uniquely identifies it.



```
alexi@DESKTOP-EU28PFF MINGW64 ~/OneDrive/Escritorio/POS_Gacela_Express (master)
$ git commit -m "Primer commit del proyecto"
[master (root-commit) 063c9a1] Primer commit del proyecto
 443 files changed, 52917 insertions(+)
 create mode 100644 Sol_PuntoVenta/.vs/LaGacelaExpress_PuntoVenta/FileContentInd
ex/540041e6-3db2-4596-b32b-8d8be0b824ce.vsidx
 create mode 100644 Sol_PuntoVenta/.vs/LaGacelaExpress_PuntoVenta/FileContentInd
ex/read.lock
 create mode 100644 Sol_PuntoVenta/.vs/ProjectSettings.json
 create mode 100644 Sol_PuntoVenta/.vs/SolPuntoVenta/FileContentIndex/c63d3193-e
8fd-4ec5-9faa-d2f583eaf25a.vsidx
 create mode 100644 Sol_PuntoVenta/.vs/SolPuntoVenta/FileContentIndex/read.lock
 create mode 100644 Sol_PuntoVenta/.vs/SolPuntoVenta/v17/.suo
 create mode 100644 Sol_PuntoVenta/.vs/SolPuntoVenta/v17/.wsuo
 create mode 100644 Sol_PuntoVenta/.vs/Sol_PuntoVenta/FileContentIndex/1e4d947f-
af33-408c-85e8-44049f28587a.vsidx
 create mode 100644 Sol_PuntoVenta/.vs/Sol_PuntoVenta/FileContentIndex/65f7e72a-
d6cb-46f2-adfe-200b9560b909.vsidx
 create mode 100644 Sol_PuntoVenta/.vs/Sol_PuntoVenta/FileContentIndex/c7486b46-
613a-497c-9c83-45fab5d71504.vsidx
 create mode 100644 Sol_PuntoVenta/.vs/Sol_PuntoVenta/FileContentIndex/fac85043-
3c21-43fa-82b2-81e53ea8c83d.vsidx
 create mode 100644 Sol_PuntoVenta/.vs/Sol_PuntoVenta/FileContentIndex/read.lock
 create mode 100644 Sol_PuntoVenta/.vs/Sol_PuntoVenta/v16/.suo
 create mode 100644 Sol_PuntoVenta/.vs/Sol_PuntoVenta/v17/.suo
 create mode 100644 Sol_PuntoVenta/.vs/Sol_PuntoVenta/v17/.wsuo
 create mode 100644 Sol_PuntoVenta/.vs/Sol_PuntoVenta/v17/HierarchyCache.v1.txt
 create mode 100644 Sol_PuntoVenta/.vs/VSWorkspaceState.json
 create mode 100644 Sol_PuntoVenta/.vs/slnx.sqlite
 create mode 100644 Sol_PuntoVenta/BK_BD_PUNTOVENTA
 create mode 100644 Sol_PuntoVenta/Imagenes/actualizar.png
 create mode 100644 Sol_PuntoVenta/Imagenes/binoculars 1.png
 create mode 100644 Sol_PuntoVenta/Imagenes/boleta.png
 create mode 100644 Sol_PuntoVenta/Imagenes/busquedarapida.png
 create mode 100644 Sol_PuntoVenta/Imagenes/check.png
 create mode 100644 Sol_PuntoVenta/Imagenes/delete.png
 create mode 100644 Sol_PuntoVenta/Imagenes/divide.png
 create mode 100644 Sol_PuntoVenta/Imagenes/dividir_cuenta.png
 create mode 100644 Sol_PuntoVenta/Imagenes/document.png
 create mode 100644 Sol_PuntoVenta/Imagenes/documento.png
 create mode 100644 Sol_PuntoVenta/Imagenes/edit.png
 create mode 100644 Sol_PuntoVenta/Imagenes/eliminar.ico
 create mode 100644 Sol_PuntoVenta/Imagenes/eliminar.png
 create mode 100644 Sol_PuntoVenta/Imagenes/emitir_documento.png
 create mode 100644 Sol_PuntoVenta/Imagenes/exit.png
 create mode 100644 Sol_PuntoVenta/Imagenes/factura.png
 create mode 100644 Sol_PuntoVenta/Imagenes/icongac.png
 create mode 100644 Sol_PuntoVenta/Imagenes/icono1.ico
 create mode 100644 Sol_PuntoVenta/Imagenes/icono2.ico
 create mode 100644 Sol_PuntoVenta/Imagenes/logo-gacela 1.png
 create mode 100644 Sol_PuntoVenta/Imagenes/logo-gacela.svg
```

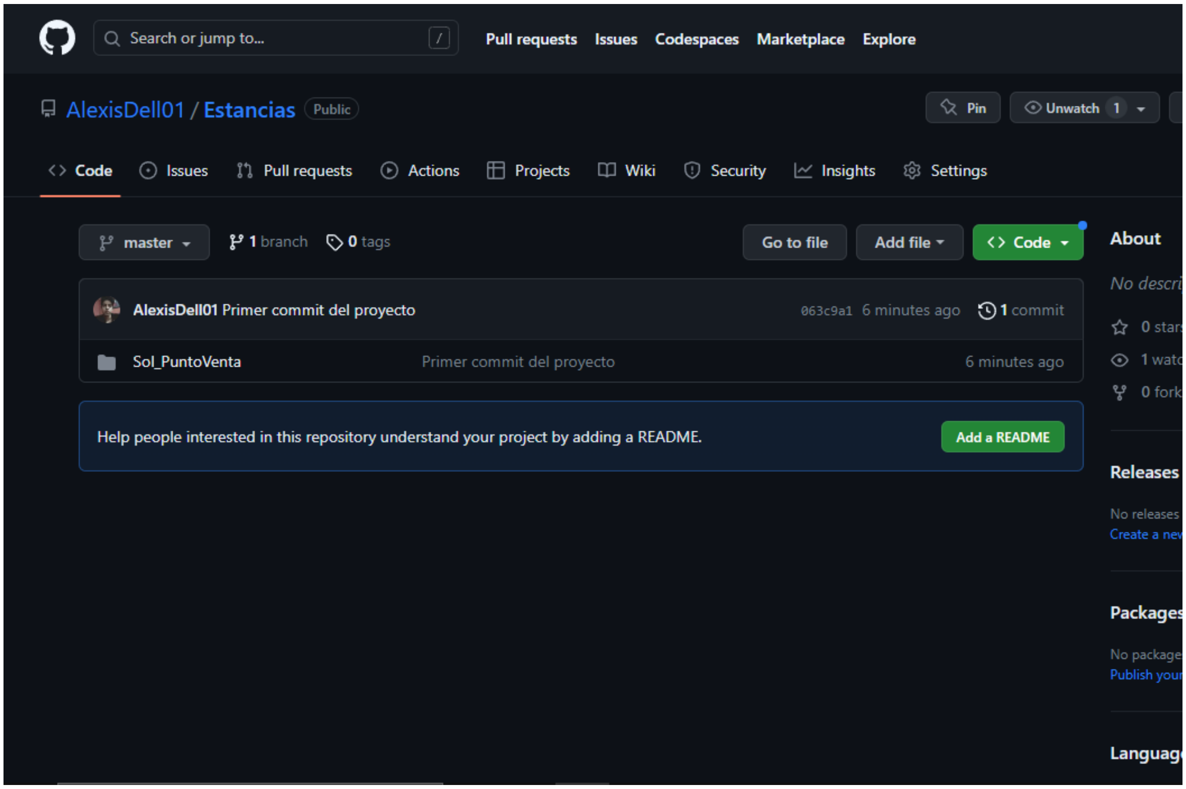
$git push origin master

This command sends the changes made in the local branch to the remote repository. It synchronizes the changes and updates the master branch in the remote repository with the latest commits from the local branch.



```
alexi@DESKTOP-EU28PFF MINGW64 ~/OneDrive/Escritorio/POS_Gacela_Express (master)
$ git push origin master
Enumerating objects: 385, done.
Counting objects: 100% (385/385), done.
Delta compression using up to 4 threads
Compressing objects: 100% (362/362), done.
Writing objects: 100% (385/385), 42.04 MiB | 594.00 KiB/s, done.
Total 385 (delta 175), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (175/175), done.
To https://github.com/AlexisDell01/Estancias.git
 * [new branch]      master -> master

alexi@DESKTOP-EU28PFF MINGW64 ~/OneDrive/Escritorio/POS_Gacela_Express (master)
$
```

Final Result:



AlexisDell01 / Estancias Public

1 branch 0 tags

AlexisDell01 Primer commit del proyecto 063c9a1 6 minutes ago 1 commit

Sol_PuntoVenta Primer commit del proyecto 6 minutes ago

Help people interested in this repository understand your project by adding a README.

About

No descri

0 stars

1 watc

0 fork

Releases

No releases

Create a new

Packages

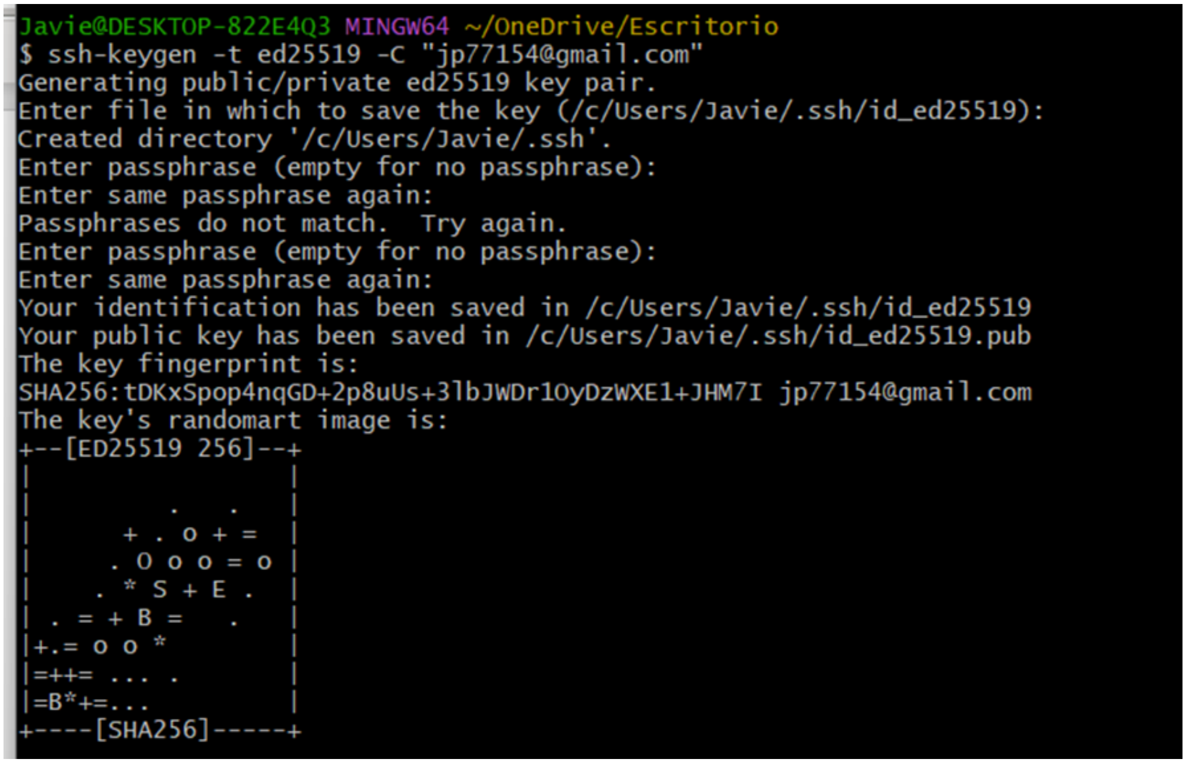
No package

Publish your

Languag

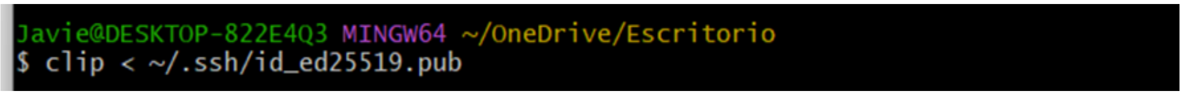
Create and collaborate within teams

The command ssh-keygen is used in Git to generate SSH keys. These keys are used to establish a secure connection between the local machine and a remote server, such as GitHub, allowing for secure authentication without the need to enter the password each time you interact with the server.



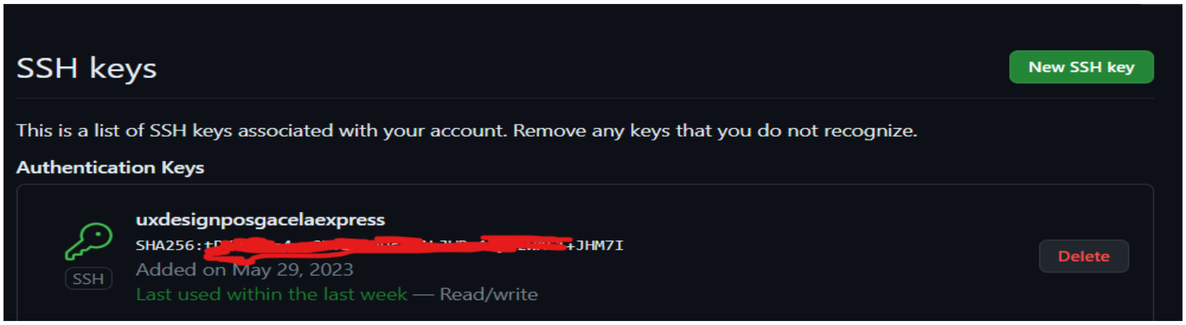
```
Javie@DESKTOP-822E4Q3 MINGW64 ~/OneDrive/Escritorio
$ ssh-keygen -t ed25519 -C "jp77154@gmail.com"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/c/Users/Javie/.ssh/id_ed25519):
Created directory '/c/Users/Javie/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match.  Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/Javie/.ssh/id_ed25519
Your public key has been saved in /c/Users/Javie/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:tDKxSpop4nqGD+2p8uUs+3lbJWDr1OyDzWXE1+JHM7I jp77154@gmail.com
The key's randomart image is:
+--[ED25519 256]--+
|                 |
|         .   .   |
|      + . o + =  |
|     . O o o = o |
|    . * S + E .  |
| . = + B =   .   |
|+.= o o *        |
|=++= ... .       |
|=B*+=...         |
+----[SHA256]-----+
```

The command clip is not a Git-specific command but a Windows command-line command that allows you to copy the output of a command to the system clipboard. In the specific case you mentioned, "$ clip < ~/.ssh/id_ed25519.pub", this command is used to copy the content of the id_ed25519.pub file (which is usually the SSH public key) to the clipboard in a Windows operating system.



```
Javie@DESKTOP-822E4Q3 MINGW64 ~/OneDrive/Escritorio
$ clip < ~/.ssh/id_ed25519.pub
```

Once the ssh is copied, it will be placed on our GitHub



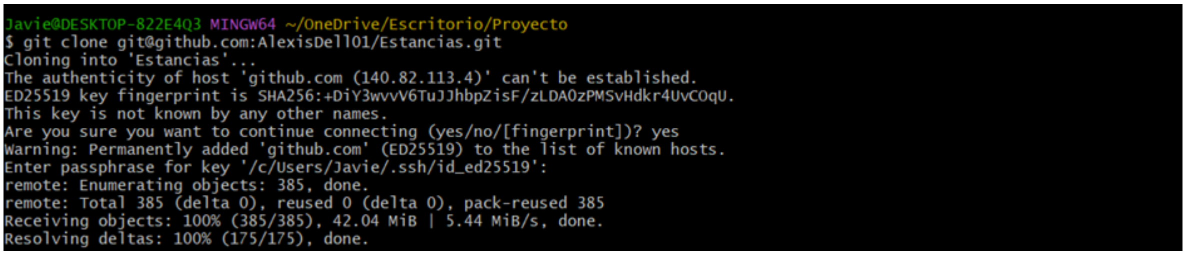
SSH keys

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

Authentication Keys

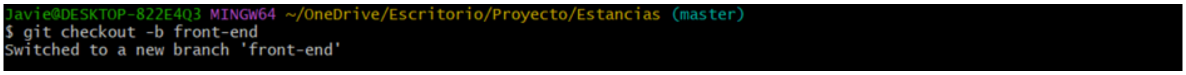
uxdesignposgacelaexpress
SHA256:[redacted]
Added on May 29, 2023
Last used within the last week — Read/write

When the git clone command is executed, Git creates a copy of the remote repository on the local machine, including branches, commits, files, and the entire history. It also establishes a connection between the local repository and the remote repository, enabling interaction with the remote repository using commands like git push and git pull.



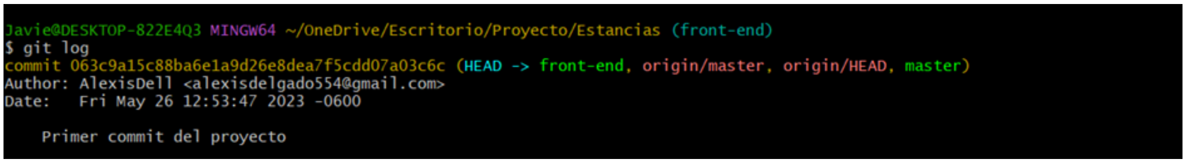
```
Javie@DESKTOP-822E4Q3 MINGW64 ~/OneDrive/Escritorio/Proyecto
$ git clone git@github.com:AlexisDell01/Estancias.git
Cloning into 'Estancias'...
The authenticity of host 'github.com (140.82.113.4)' can't be established.
ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'github.com' (ED25519) to the list of known hosts.
Enter passphrase for key '/c/Users/Javie/.ssh/id_ed25519':
remote: Enumerating objects: 385, done.
remote: Total 385 (delta 0), reused 0 (delta 0), pack-reused 385
Receiving objects: 100% (385/385), 42.04 MiB | 5.44 MiB/s, done.
Resolving deltas: 100% (175/175), done.
```

The Git command "$ git checkout -b <branch_name>" is used to create a new branch and switch to it in a single step. The command "$ git checkout -b front-end" is used to create a new branch named "front-end" and switch to it.



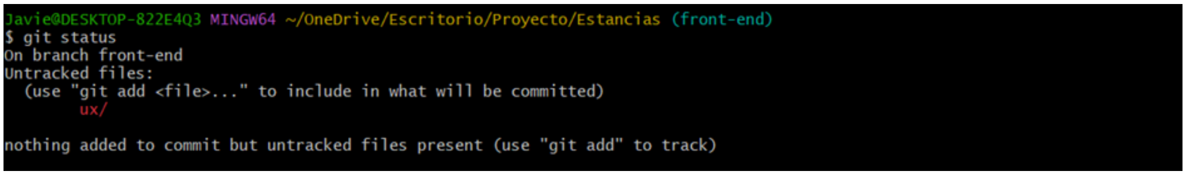
```
Javie@DESKTOP-822E4Q3 MINGW64 ~/OneDrive/Escritorio/Proyecto/Estancias (master)
$ git checkout -b front-end
Switched to a new branch 'front-end'
```

The git log command is used to view the commit history in a Git repository. It displays a detailed list of the commits made, including information such as the author, date and time, and the commit message.



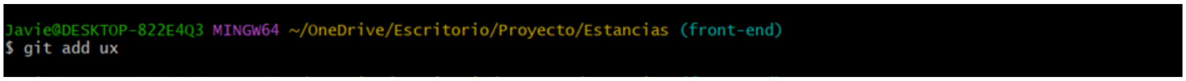
```
Javie@DESKTOP-822E4Q3 MINGW64 ~/OneDrive/Escritorio/Proyecto/Estancias (front-end)
$ git log
commit 063c9a15c88ba6e1a9d26e8dea7f5cdd07a03c6c (HEAD -> front-end, origin/master, origin/HEAD, master)
Author: AlexisDell <alexisdelgado554@gmail.com>
Date:   Fri May 26 12:53:47 2023 -0600

    Primer commit del proyecto
```

The git status command is used in Git to obtain information about the current state of the repository. It provides details on modified files, files added to the staging area, and the status of the current branch.



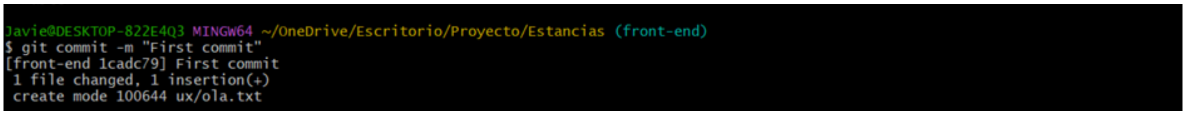
```
Javie@DESKTOP-822E4Q3 MINGW64 ~/OneDrive/Escritorio/Proyecto/Estancias (front-end)
$ git status
On branch front-end
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        ux/

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

After placing the command “git add ‘filename’” to prepare the document to be uploaded.



```
Javie@DESKTOP-822E4Q3 MINGW64 ~/OneDrive/Escritorio/Proyecto/Estancias (front-end)
$ git add ux
```

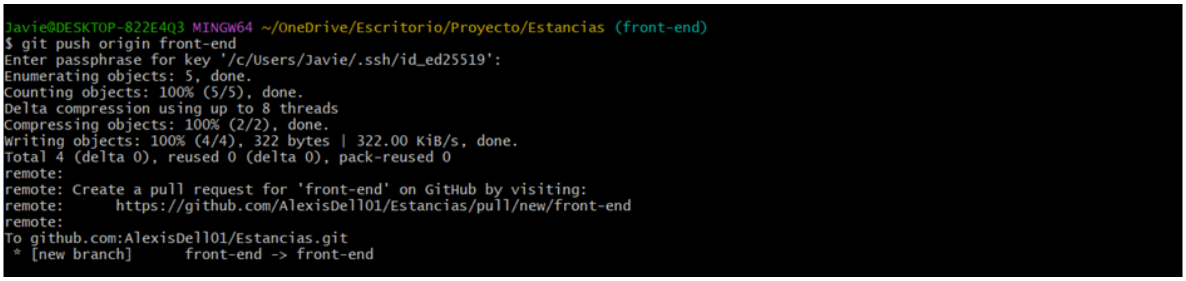
Next, we put the command "git commit -m 'commit name'" to grab the file we added earlier to be saved to the repository's history.



```
Javie@DESKTOP-822E4Q3 MINGW64 ~/OneDrive/Escritorio/Proyecto/Estancias (front-end)
$ git commit -m "First commit"
[front-end 1cadc79] First commit
 1 file changed, 1 insertion(+)
 create mode 100644 ux/ola.txt
```

"Git push origin 'branch name'"

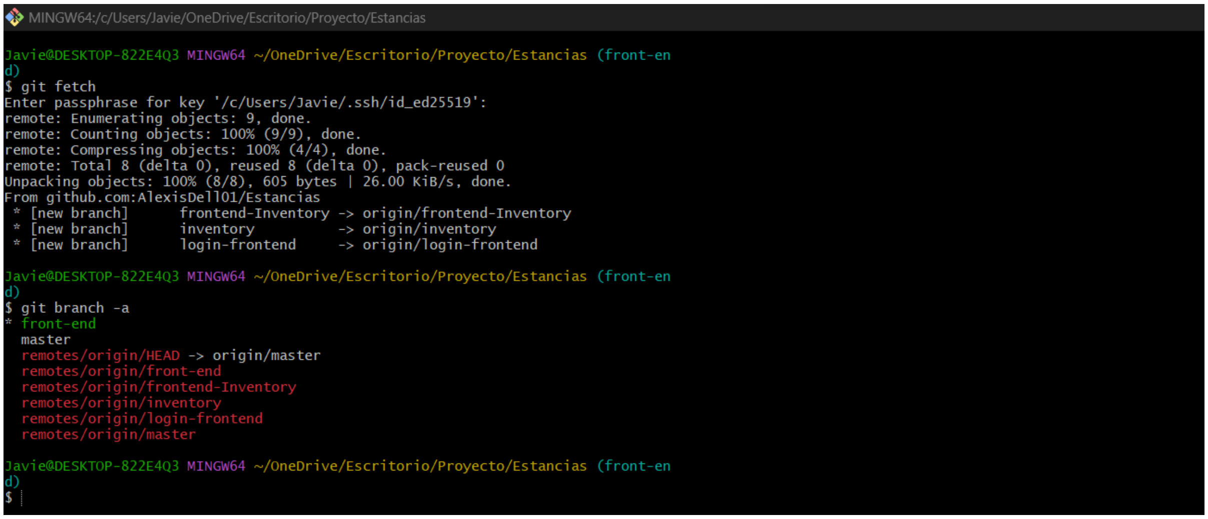
This command pushes the changes made in the previously created branch to the remote repository. Synchronizes the changes and updates the master branch in the remote repository with the latest commits from the created branch.



```
Javie@DESKTOP-822E4Q3 MINGW64 ~/OneDrive/Escritorio/Proyecto/Estancias (front-end)
$ git push origin front-end
Enter passphrase for key '/c/Users/Javie/.ssh/id_ed25519':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (4/4), 322 bytes | 322.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote: Create a pull request for 'front-end' on GitHub by visiting:
remote:      https://github.com/AlexisDell01/Estancias/pull/new/front-end
remote:
To github.com:AlexisDell01/Estancias.git
 * [new branch]      front-end -> front-end
```

Once this is done we can see that the branch to be worked on has been created in our repository, here are the branches created by each of the members:

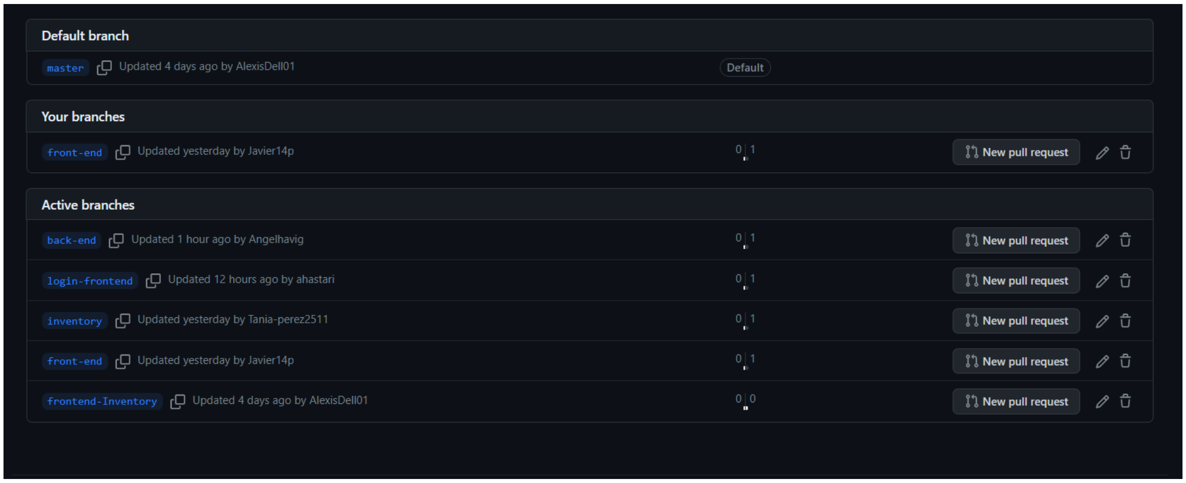
Using the command “git fetch” & “git branch -a”:



MINGW64:/c/Users/Javie/OneDrive/Escritorio/Proyecto/Estancias

```
Javie@DESKTOP-822E4Q3 MINGW64 ~/OneDrive/Escritorio/Proyecto/Estancias (front-en
d)
$ git fetch
Enter passphrase for key '/c/Users/Javie/.ssh/id_ed25519':
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 8 (delta 0), reused 8 (delta 0), pack-reused 0
Unpacking objects: 100% (8/8), 605 bytes | 26.00 KiB/s, done.
From github.com:AlexisDell01/Estancias
 * [new branch]      frontend-Inventory -> origin/frontend-Inventory
 * [new branch]      inventory          -> origin/inventory
 * [new branch]      login-frontend     -> origin/login-frontend

Javie@DESKTOP-822E4Q3 MINGW64 ~/OneDrive/Escritorio/Proyecto/Estancias (front-en
d)
$ git branch -a
* front-end
  master
  remotes/origin/HEAD -> origin/master
  remotes/origin/front-end
  remotes/origin/frontend-Inventory
  remotes/origin/inventory
  remotes/origin/login-frontend
  remotes/origin/master

Javie@DESKTOP-822E4Q3 MINGW64 ~/OneDrive/Escritorio/Proyecto/Estancias (front-en
d)
$
```

View the branches in github:



Default branch

master	Updated 4 days ago by AlexisDell01	Default

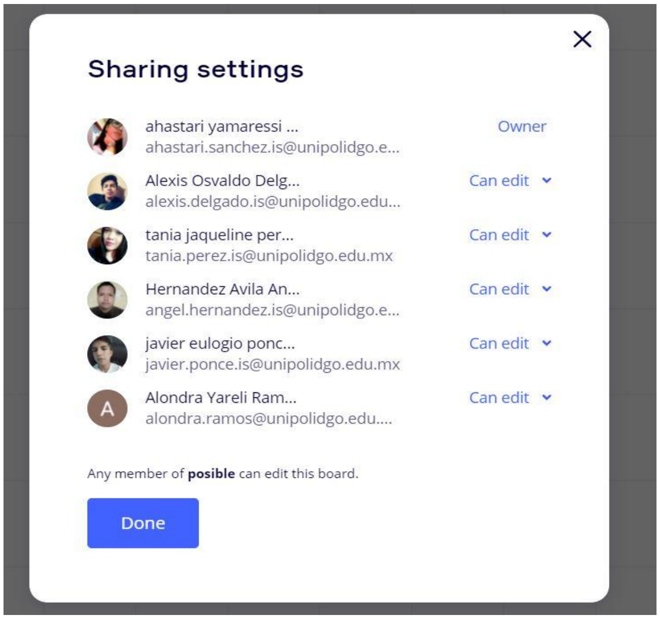
Your branches

front-end	Updated yesterday by Javier14p	0 | 1	New pull request

Active branches

back-end	Updated 1 hour ago by Angelhavig	0 | 1	New pull request
login-frontend	Updated 12 hours ago by ahastari	0 | 1	New pull request
inventory	Updated yesterday by Tania-perez2511	0 | 1	New pull request
front-end	Updated yesterday by Javier14p	0 | 1	New pull request
frontend-Inventory	Updated 4 days ago by AlexisDell01	0 | 0	New pull request

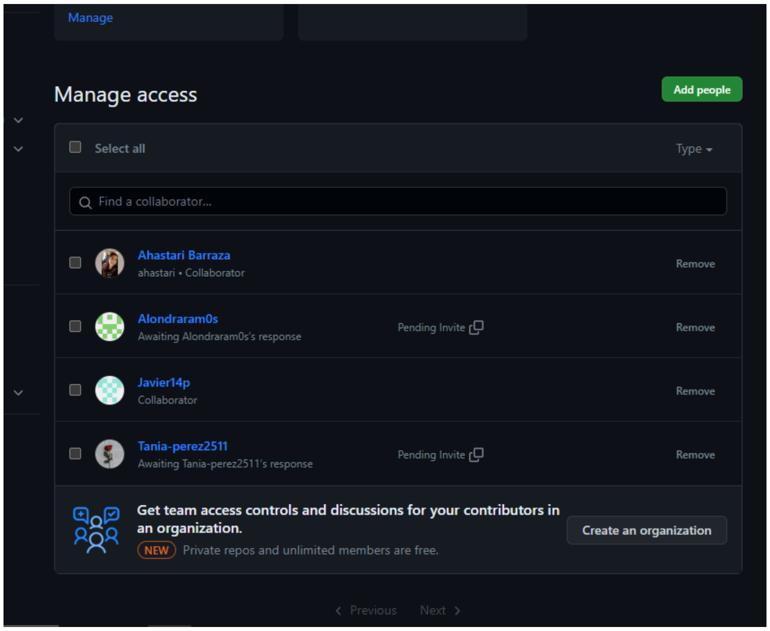
Invitation to the Miro tool:



Sharing settings

ahastari yamaressi ... ahastari.sanchez.is@unipolidgo.e...	Owner
Alexis Osvaldo Delg... alexis.delgado.is@unipolidgo.edu...	Can edit
tania jaqueline per... tania.perez.is@unipolidgo.edu.mx	Can edit
Hernandez Avila An... angel.hernandez.is@unipolidgo.e...	Can edit
javier eulogio ponc... javier.ponce.is@unipolidgo.edu.mx	Can edit
Alondra Yareli Ram... alondra.ramos@unipolidgo.edu....	Can edit

Any member of **posible** can edit this board.

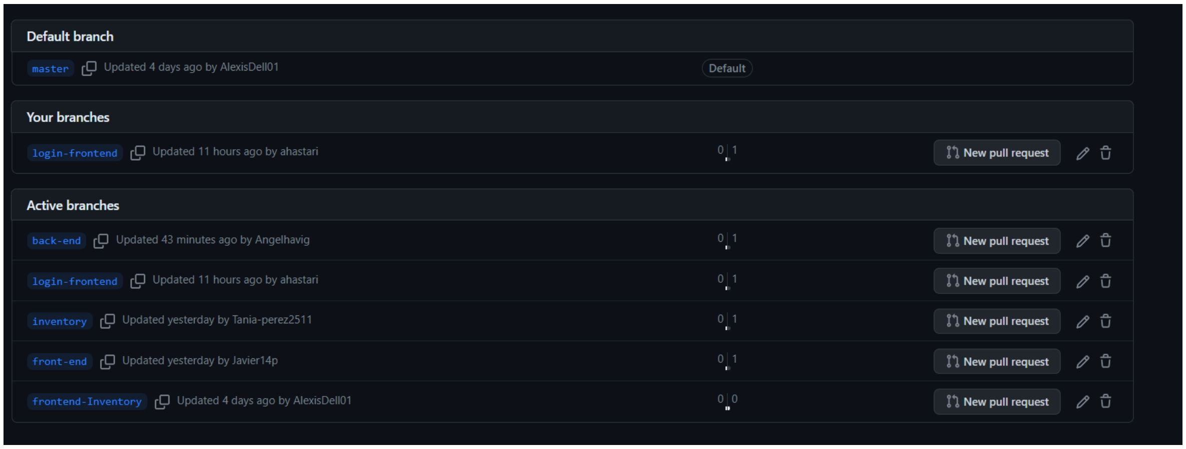


Manage access

Ahastari Barraza ahastari • Collaborator		Remove
Alondraram0s Awaiting Alondraram0s's response	Pending Invite	Remove
Javier14p Collaborator		Remove
Tania-perez2511 Awaiting Tania-perez2511's response	Pending Invite	Remove

Get team access controls and discussions for your contributors in an organization.

NEW Private repos and unlimited members are free.



Default branch

master Updated 4 days ago by AlexisDell01 Default

Your branches

login-frontend Updated 11 hours ago by ahastari 0 | 1 New pull request

Active branches

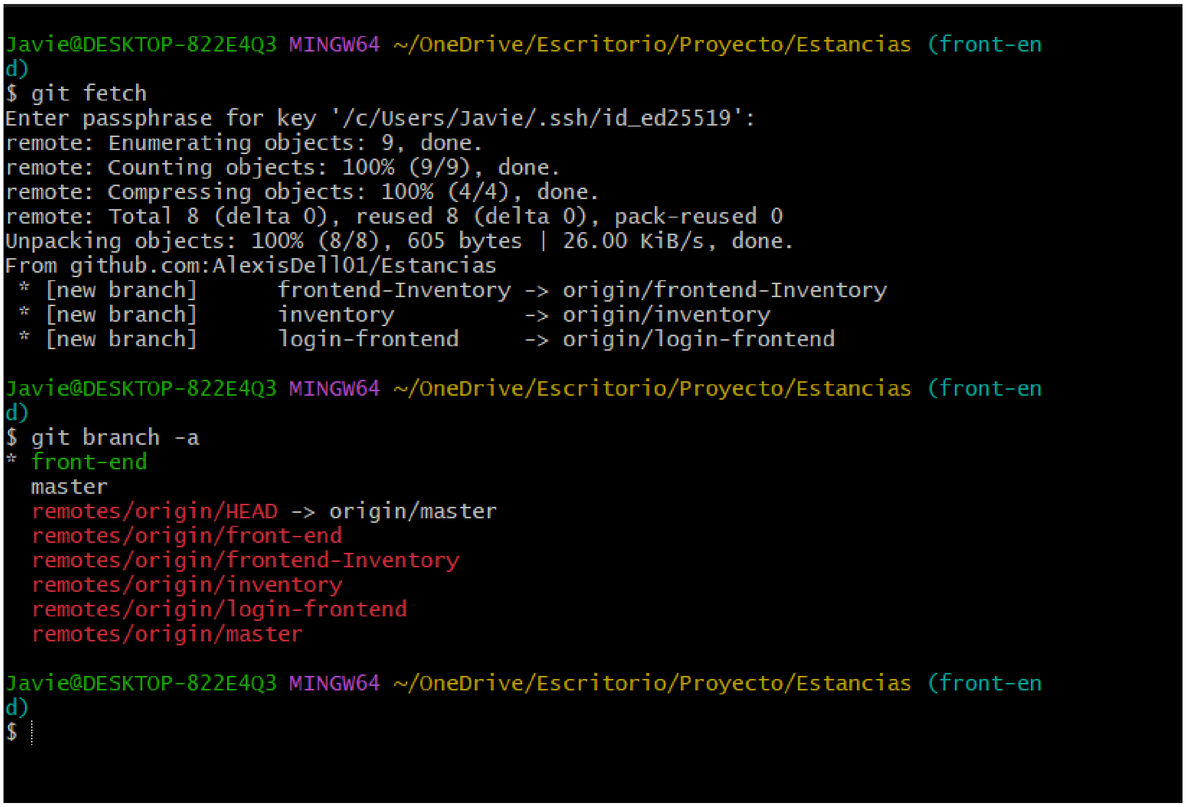
back-end Updated 43 minutes ago by Angelhavig 0 | 1 New pull request

login-frontend Updated 11 hours ago by ahastari 0 | 1 New pull request

inventory Updated yesterday by Tania-perez2511 0 | 1 New pull request

front-end Updated yesterday by Javier14p 0 | 1 New pull request

frontend-Inventory Updated 4 days ago by AlexisDell01 0 | 0 New pull request



```
Javie@DESKTOP-822E4Q3 MINGW64 ~/OneDrive/Escritorio/Proyecto/Estancias (front-en
d)
$ git fetch
Enter passphrase for key '/c/Users/Javie/.ssh/id_ed25519':
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 8 (delta 0), reused 8 (delta 0), pack-reused 0
Unpacking objects: 100% (8/8), 605 bytes | 26.00 KiB/s, done.
From github.com:AlexisDell01/Estancias
 * [new branch]      frontend-Inventory -> origin/frontend-Inventory
 * [new branch]      inventory          -> origin/inventory
 * [new branch]      login-frontend     -> origin/login-frontend

Javie@DESKTOP-822E4Q3 MINGW64 ~/OneDrive/Escritorio/Proyecto/Estancias (front-en
d)
$ git branch -a
* front-end
  master
  remotes/origin/HEAD -> origin/master
  remotes/origin/front-end
  remotes/origin/frontend-Inventory
  remotes/origin/inventory
  remotes/origin/login-frontend
  remotes/origin/master

Javie@DESKTOP-822E4Q3 MINGW64 ~/OneDrive/Escritorio/Proyecto/Estancias (front-en
d)
$
```

Conclusions

Sanchez Barraza Ahastari Yamaressi

In conclusion I think it is good to give maintenance to the product for better operation and performance to the user, plus it helps us to continue to improve our skills and gain more knowledge and experience.

The maintenance of a software project involves the detection and resolution of errors (bugs), the incorporation of new functionalities and improvements, as well as the update and adaptation to changes in the technological and business environment. In addition, documentation, testing and monitoring tasks are carried out to ensure that the software meets the users' requirements and expectations.

It is important to note that the maintenance of a software project involves not only technical aspects, but also the effective management of resources, version control, communication with users and the planning of maintenance activities. Good maintenance management can help maximize software life, minimize downtime and optimize associated costs.

Delgado Espinoza Alexis Osvaldo

In my opinion, using GitHub for software maintenance and a project management tool has many advantages. GitHub helps keep track of changes, collaborate with team members, and fix bugs efficiently. A project management tool enables task tracking, progress monitoring, and team coordination. To upload a project to GitHub, create a repository, connect it to your local repository, and regularly update it. To set up a management environment for the tool, define roles, break tasks into smaller ones, and configure the tool to reflect the project structure. By utilizing these tools effectively, you can improve productivity and ensure project success. These are two powerful tools in order to make better projects and have a good track record.

Pérez Calzada Tania Jaqueline.

In conclusion, from my point of view, I think that github is very useful for teamwork since it helps us a lot in a certain part to have order and version control, in this case it will be very useful since we are a team and we will be working on the same project where everyone can collaborate within their branch and review or approve the changes that are made over time.

For this project, each one of the members followed certain commands that we saw during class to be able to create each one their branch and thus be able to work without problems, each one with the branch they chose.

On the other hand, we will make use of the Miro platform which allows all team members to distribute their tasks and times to work together effectively, which helps the organization and that all team members participate equally within the project, but In the same way we can do some brainstorming in case it is necessary to define an idea and see what is the best option, in this case my team and I already have a workspace where we will be working.

Both tools are important within software maintenance because git is first used as a problem tracker, which provides us with a record of said problems and errors found in the software so that later we can see the real changes in the code and be able to know where they came from. errors could come from and apply their due correction. Regarding Miro, it is not especially focused on software maintenance but serves as a tool for online collaboration, documentation and team planning in a more efficient and collaborative way.

javier eulogio ponce barraza

In my personal opinion, it is important to bring some tool that helps us to be able to monitor and keep a better track of the activities that we could be doing as a team, in this case, our randomly selected tool was Miro, but I think it is a very good option. Despite the fact that some features of Miro are not free, working in the free plan gives you the necessary tools to be able to carry out a fairly good project management, since this tool can provide you with brainstorming layouts , drawings for explanations in emphasis, diagrams, concept maps, including other frameworks such as Kanban, is a fairly complete tool and in our case, it helps us a lot by providing very limited but sufficient tools to work as a team. As for Git & Github, it is important to know even a little about how these tools work, since despite the fact that we are not part of the coding, it is very likely that we have to work with system versions and therefore we need this, For me it's something quite new since my only interaction with git & github was a couple of times where I pushed a project to a repository, but I didn't do branches or anything like that, so these activities are quite new for me, but I hope to learn enough because as I mentioned before, it is important to know these tools and at least know how to use them in their basic form, since even though we plan not to use them, it may happen that we need them in some job, and with these activities we could already have a little experience so that it is not so complicated.

Hernandez Avila Angel Guadalupe

Version control is essential in software development as it enables working on the same project. Git is a version control tool and one of the most widely used when developing projects due to its ease of use and ability to handle projects. Version control allows developers to track changes made in the code and manage software versions. Additionally, version control allows developers to revert changes and examine the history of changes made.

For example, Git enables developers to work in a decentralized manner, meaning there doesn't need to be a centralized server for version control.

In conclusion, using a similar tool not only helps to avoid data loss issues but also allows for identifying when a problem arose and recovering a previous version. All systems and applications have different versions, yet they all should have undergone versioning for better control.

Bibliography

Miro: https://miro.com/

Mural: https://www.mural.co/

Microsoft Whiteboard:
https://www.microsoft.com/en-us/microsoft-365/microsoft-whiteboard/digital-whiteboard-app

Lucidchart: https://www.lucidchart.com/

"Miro Review" - Capterra: https://www.capterra.com/p/156256/Miro/

"Miro Pricing" - Miro: https://miro.com/pricing/

"Miro: Collaboration Made Easy, Fun, and Intuitive" - TrustRadius:
https://www.trustradius.com/products/miro/reviews