

PyQt5 Workshop

University of Porto, SPIE Student Chapter, 2021

Nuno Azevedo Silva

Structure of the Workshop

13h30 – 14h00 (Optional)

Pre-requisites + Getting started with Git

14h00 – 15h30

PyQt5: Introduction to the basic components of an app

16h00 – 18h30

Deploying a real-world example

Pre-Requisites

Python, Anaconda and Git

Pre-requisites 1

Install Anaconda

<https://anaconda.org/anaconda>



Why Anaconda?

- + Allows to use multiple environments
- + Use the same environment across multiple IDE's (Integrated Development environment, e.g. Spyder, JupyterLab, PyCharm, etc)

Pre-requisites 1

Launch anaconda prompt

To open Anaconda Prompt:

1. Windows: Click Start, search, or select Anaconda Prompt from the menu.
2. macOS: Cmd+Space to open Spotlight Search and type “Navigator” to open the program.
3. Linux–CentOS: Open Applications - System Tools - terminal.

Pre-requisites 1

Create a new environment

```
>conda create --name new_env python=3.6
```

Change to (Activate) the new environment

```
(base) C:\Users\nunoa>conda activate new_env  
(new_env) C:\Users\nunoa>
```

Pre-requisites 1

Install necessary libraries

Packages necessary (in principle *numpy* and *matplotlib* are already installed):

PyQt5	conda install -c anaconda pyqt
Spyder	conda install spyder
Jupyter	conda install jupyter

Jupyter may need an additional package
conda install pywin32

Pre-requisites 1

Launch jupyter notebook

jupyter-notebook

Pre-requisites 2

Download GIT

<https://git-scm.com/>



Why GIT?

- Version Control, i.e. keep track of changes

- Remote – you can have your code (or any data you want) stored in a repository online

- Share with multiple users/contributors

Pre-requisites 2

Create an account on a remote host provider (we will use Github)

E.g.

GitLab (INESC TEC, but does not allow public repositories) https://gitlab.inescotec.pt/users/sign_in

Github, Bitbucket, etc

Pre-requisites 2

Setting up a ssh key to connect with an online server

Open cmd

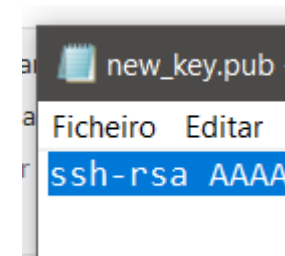
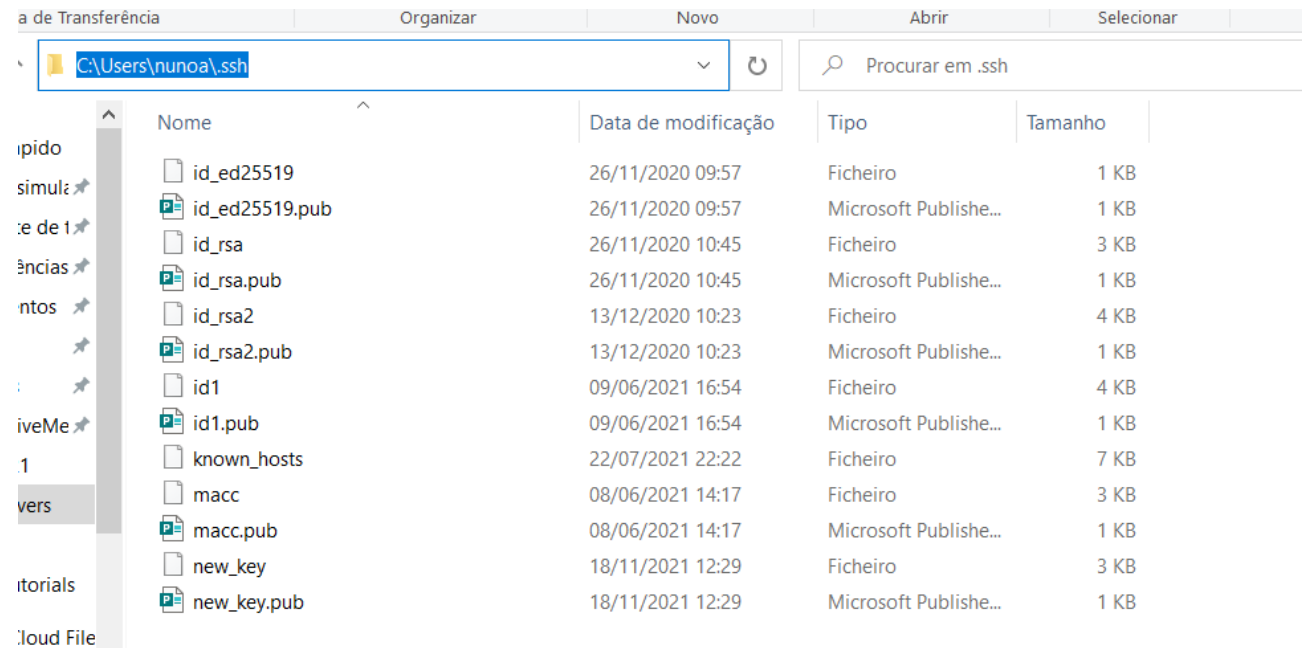
Type ssh-keygen

Enter path to save ssh key, enter on passphrase

```
C:\Users\nunoo>ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\nunoo/.ssh/id_rsa): C:\Users\nunoo/.ssh/new_key
```

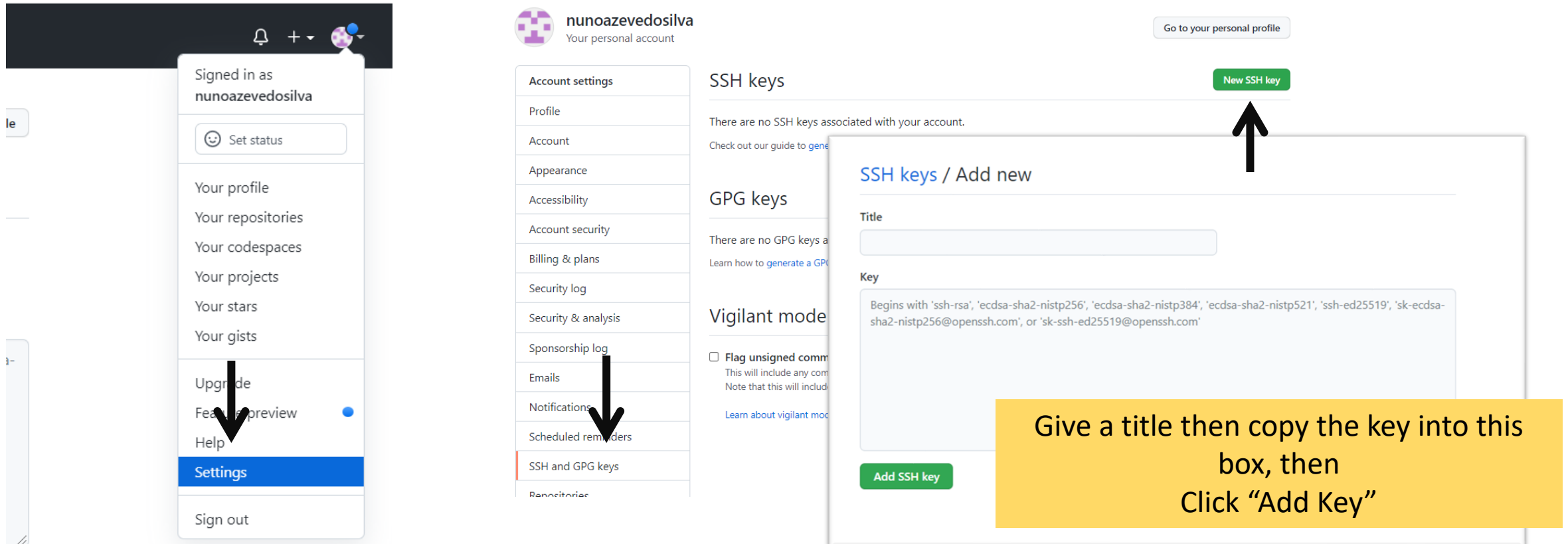
Pre-requisites 2

Find your key in path (in this example, new_key.pub), open in notepad and copy its contents



Pre-requisites 2

Add key to user settings on the online provider page



The image shows a sequence of steps to add an SSH key to a GitHub account. On the left, a user profile dropdown menu is open, showing options like 'Your profile', 'Your repositories', and 'Settings'. A black arrow points to the 'Settings' option. In the center, the 'Settings' page is displayed, with a sidebar menu containing 'Account settings', 'Profile', 'Account', 'Appearance', 'Accessibility', 'Account security', 'Billing & plans', 'Security log', 'Security & analysis', 'Sponsorship log', 'Emails', 'Notifications', 'Scheduled reminders', 'SSH and GPG keys', and 'Repositories'. A black arrow points to the 'SSH and GPG keys' option. On the right, the 'SSH keys' page is shown, indicating that there are no SSH keys associated with the account. A green button labeled 'New SSH key' is visible. A black arrow points to this button. Below the 'SSH keys' page, a modal window titled 'SSH keys / Add new' is open, showing a form with a 'Title' field and a 'Key' field. A yellow callout box with black text says: 'Give a title then copy the key into this box, then Click "Add Key"'. The 'Add SSH key' button is at the bottom of the modal.

Signed in as **nunoazevedosilva**

Set status

Your profile

Your repositories

Your codespaces

Your projects

Your stars

Your gists

Upgrade

Feature preview

Help

Settings

Sign out

nunoazevedosilva
Your personal account

Go to your personal profile

Account settings

Profile

Account

Appearance

Accessibility

Account security

Billing & plans

Security log

Security & analysis

Sponsorship log

Emails

Notifications

Scheduled reminders

SSH and GPG keys

Repositories

SSH keys

There are no SSH keys associated with your account.

Check out our guide to [generate a GPG key](#)

GPG keys

There are no GPG keys associated with your account.

Learn how to [generate a GPG key](#)

Vigilant mode

☐ **Flag unsigned commits**

This will include any commits that are not signed by you. Note that this will include commits from forks.

[Learn about vigilant mode](#)

New SSH key

SSH keys / Add new

Title

Key

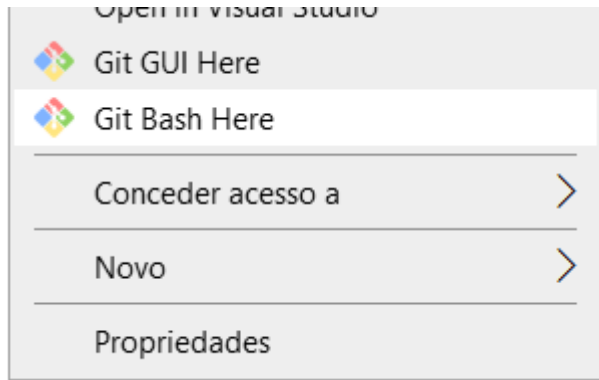
Begins with 'ssh-rsa', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'

Add SSH key

Give a title then copy the key into this box, then Click "Add Key"

Pre-requisites 2

In the folder you want to clone the repository, right-click and git Bash here, then start ssh agent and add your created key



```
nunoa@DESKTOP-GUBMLR6 MINGW64 ~/Pyqt5 Workshop  
$ exec ssh-agent bash
```

```
nunoa@DESKTOP-GUBMLR6 MINGW64 ~/Pyqt5 Workshop  
$ eval 'ssh-agent -s'
```

```
nunoa@DESKTOP-GUBMLR6 MINGW64 ~/Pyqt5 Workshop  
$ ssh-add C://Users/nunoa/.ssh/new_key  
Identity added: C://Users/nunoa/.ssh/new_key (nunoa@DESKTOP-GUBMLR6)
```

Put your key path here (note the // after C:)

Pre-requisites 2

Create a personal access token (for Github only)

Applications

Developer settings

Moderation settings

Blocked users

GitHub Apps

OAuth Apps

Personal access tokens

Personal access tokens

Tokens you have generated

Generate new token

New personal access token

Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Note

acc

What's this token for?

Expiration *

No expiration The token will never expire!

GitHub strongly recommends that you set an expiration date for your token to help keep your information secure. [Learn more](#)

Select scopes

Scopes define the access for personal tokens. [Read more about OAuth scopes.](#)

<input checked="" type="checkbox"/> repo	Full control of private repositories
<input checked="" type="checkbox"/> repostatus	Access commit status
<input checked="" type="checkbox"/> repo_deployment	Access deployment status
<input checked="" type="checkbox"/> public_repo	Access public repositories
<input checked="" type="checkbox"/> repoinvite	Access repository invitations
<input checked="" type="checkbox"/> security_events	Read and write security events
<input checked="" type="checkbox"/> workflow	Update GitHub Action workflows
<input checked="" type="checkbox"/> write:packages	Upload packages to GitHub Package Registry
<input checked="" type="checkbox"/> read:packages	Download packages from GitHub Package Registry

Select all

Pre-requisites 2

Clone the repository

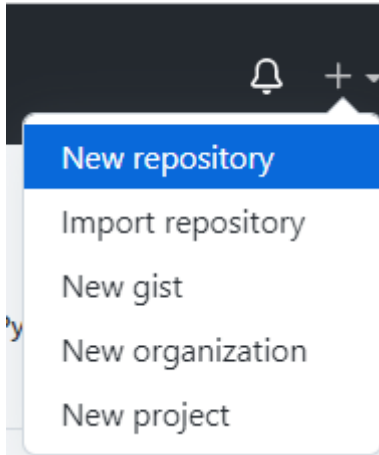
```
nunoo@DESKTOP-GUBMLR6 MINGW64 ~/Pyqt5 Workshop  
$ git clone git@github.com:nunoazevedosilva/GUI_WorkshopChapter2021.git
```

git clone git@github.com:nunoazevedosilva/GUI_WorkshopChapter2021.git

Additional Info - GIT



1. Start a new repository online





Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner * nunoazevedosilva / Repository name * my_new_repo ✓

Great repository names are short and unique. my_new_repo is available. Avoid duplication? How about [scaling-waddle?](#)

Description (optional)

- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☒ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

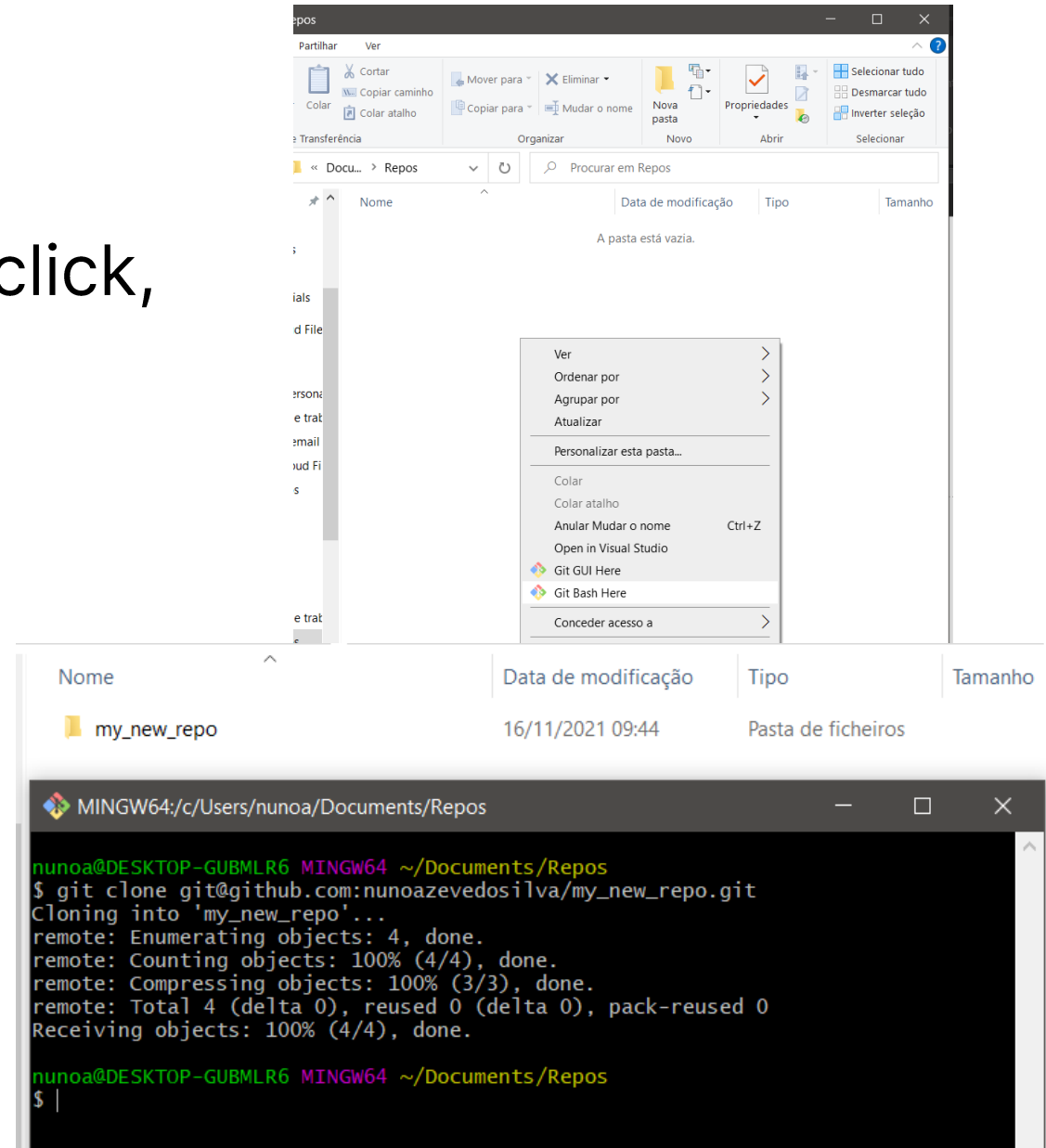
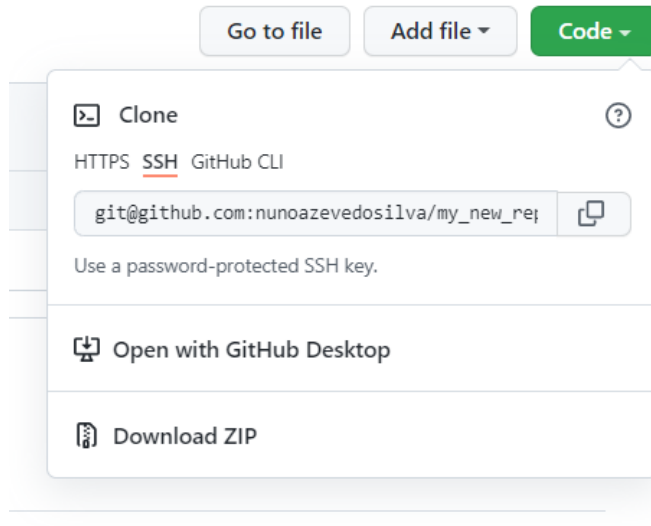
.gitignore template: Python

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set main as the default branch. Change the default name in your [settings](#).

Create repository

2. Clone it to your pc:
In the target directory, right click,
open Git Bash here
`git clone path`



3. Change to the repository directory

```
nunoo@DESKTOP-GUBMLR6 MINGW64 ~/Documents/Repos  
$ cd my_new_repo/
```

4. Create a new branch (optional)

git checkout -b *branch_name*

(without -b changes to existing branch)

```
nunoo@DESKTOP-GUBMLR6 MINGW64 ~/Documents/Repos/my_new_repo (main)  
$ git checkout -b new_branch  
Switched to a new branch 'new_branch'  
  
nunoo@DESKTOP-GUBMLR6 MINGW64 ~/Documents/Repos/my_new_repo (new_branch)  
$
```

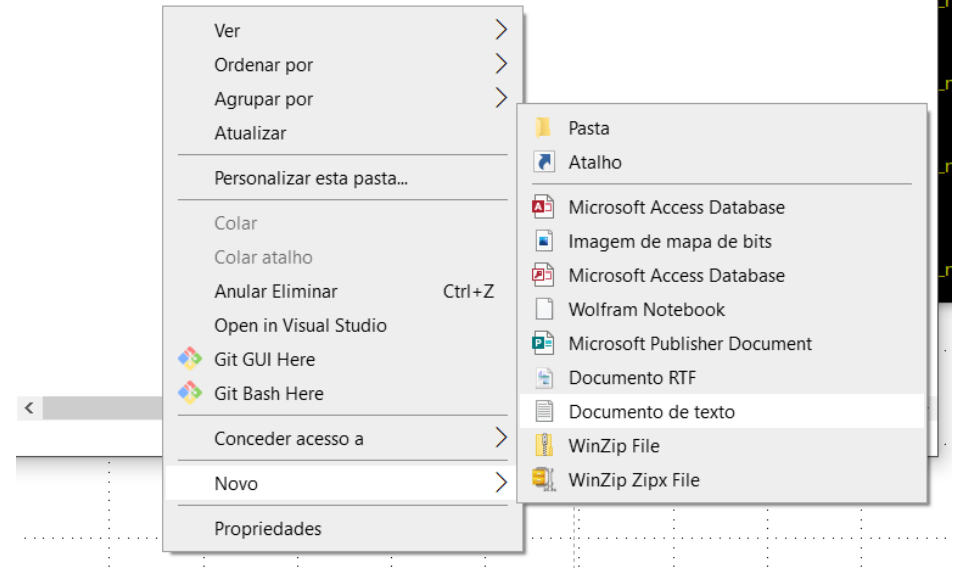
5. Add some files

6. Add changes to the staging area (prepare to commit)

`git add .`

```
nunoa@DESKTOP-GUBMLR6 MINGW64 ~/Documents/Repos/my_new_repo (main)
$ git add .
```

Nome	Data de modificação	Tipo	Tamanho
.git	16/11/2021 09:51	Pasta de ficheiros	
.gitignore	16/11/2021 09:44	Ficheiro GITIGNORE	2 KB
README.md	16/11/2021 09:44	Ficheiro MD	1 KB



7. Commit the changes

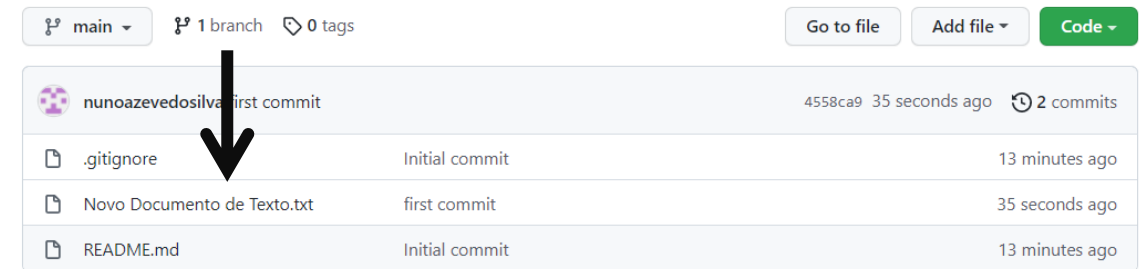
`git commit -m "message of this commit"`

```
nunoa@DESKTOP-GUBMLR6 MINGW64 ~/Documents/Repos/my_new_repo (main)
$ git commit -m "first commit"
[main 4558ca9] first commit
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Novo Documento de Texto.txt
```

8. Push it to the online repository

`git push origin branch_name`

```
nunoa@DESKTOP-GUBMLR6 MINGW64 ~/Documents/Repos/my_new_repo (main)
$ git push origin main
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 8 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 330 bytes | 66.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To github.com:nunoazevedosilva/my_new_repo.git
c05df7e..4558ca9  main -> main
```



The screenshot shows the GitHub interface for the repository 'nunoazevedosilva/my_new_repo'. At the top, there are buttons for 'Go to file', 'Add file', and 'Code'. Below these, the commit history is displayed as a table. The first commit, 'first commit' (4558ca9), is highlighted. A black arrow points from the 'main' branch dropdown to this commit. The commit list includes files like '.gitignore', 'Novo Documento de Texto.txt', and 'README.md'.

File	Commit Message	Commit Hash	Time Ago
.gitignore	Initial commit	4558ca9	13 minutes ago
Novo Documento de Texto.txt	first commit	4558ca9	35 seconds ago
README.md	Initial commit	4558ca9	13 minutes ago

Structure of the Workshop

13h30 – 14h00 (Optional)

Pre-requisites + Getting started with Git

14h00 – 15h30

PyQt5: Introduction to the basic components of an app

16h00 – 18h30

Deploying a real-world example