

The slide features abstract green geometric shapes. On the left, a solid green triangle points downwards. On the right, a complex arrangement of overlapping translucent green triangles and polygons creates a dynamic, layered effect. The main title is centered in a large, bold, green sans-serif font.

Desenvolvimento Colaborativo de Software

2020/2021 - Apresentação

Instituto Politécnico do Cávado e do Ave

Pedro Emanuel Cardoso de Sousa
pesousa@ipca.pt

Git & GitHub

- ▶ Controlo de versões de ficheiros / programas
- ▶ Cenário:
 - ▶ 2 ou mais programadores estão a trabalhar no mesmo projeto;
 - ▶ A página da turma de DWM 2º Ano;
 - ▶ Com funcionalidades de consulta, mas também fórum de discussão sobre as unidades curriculares e outras atividades;
- ▶ Pergunta: como trabalhar no mesmo projeto, ao mesmo tempo e resolver os conflitos existentes?

Solução comum (pequenos projetos)

- ▶ Organizar uma estrutura de pastas;
- ▶ Dividir o projeto em pequenos módulos:
 - ▶ Funções core: 1 programador;
 - ▶ Módulo A: outro programador;
 - ▶ Módulo B: outro programador;
 - ▶ ...
- ▶ No final o responsável do projeto junta os códigos todos e resolve os problemas de integração.
- ▶ Este processo resulta sobretudo na 1ª versão da solução e em pequenas aplicações, com equipas pequenas e muito organizadas.

Crescimento da solução

- ▶ Problemas:
 - ▶ Agora é necessário adicionar mais funcionalidades;
 - ▶ Para o mesmo módulo é necessário ter 2 ou mais programadores;
 - ▶ É necessário fazer pequenos ajustes nas funções “core” e estas são feitas por programadores dos módulos;
 - ▶ É necessário ir buscar versões antigas das funções;
 - ▶ Perderam as cópias de segurança da aplicação;
 - ▶ ...
- ▶ Os sistemas de “versionamento” e trabalho colaborativo ajudam a resolver estes problemas.
- ▶ O **Git** e o **GitHub** são exemplo disso;

O que é o GIT

- ▶ É um sistema de controlo de versões;
- ▶ O Git permite manter histórico de todos os ficheiros, arquivos e projetos da nossa conta;
- ▶ Permite fazer “deploy” de várias versões do projeto, a partir da mesma conta;
- ▶ Limita as alterações entre utilizadores, ou seja, quando um utilizador altera a sua versão, não afeta a versão de outro;
- ▶ Permite que o projeto consiga voltar a uma versão estável.

O que é o GitHub

- ▶ É um serviço de alojamento de projetos em Git;
- ▶ Existem outros serviços como o GitLab, entre outros;
- ▶ Cria um ambiente de trabalho onde todos os envolvidos podem trabalhar;
- ▶ Implementa várias funcionalidades de trabalho colaborativo como equipa, projeto, documentação, etc.



Diferença entre Git e GitHub

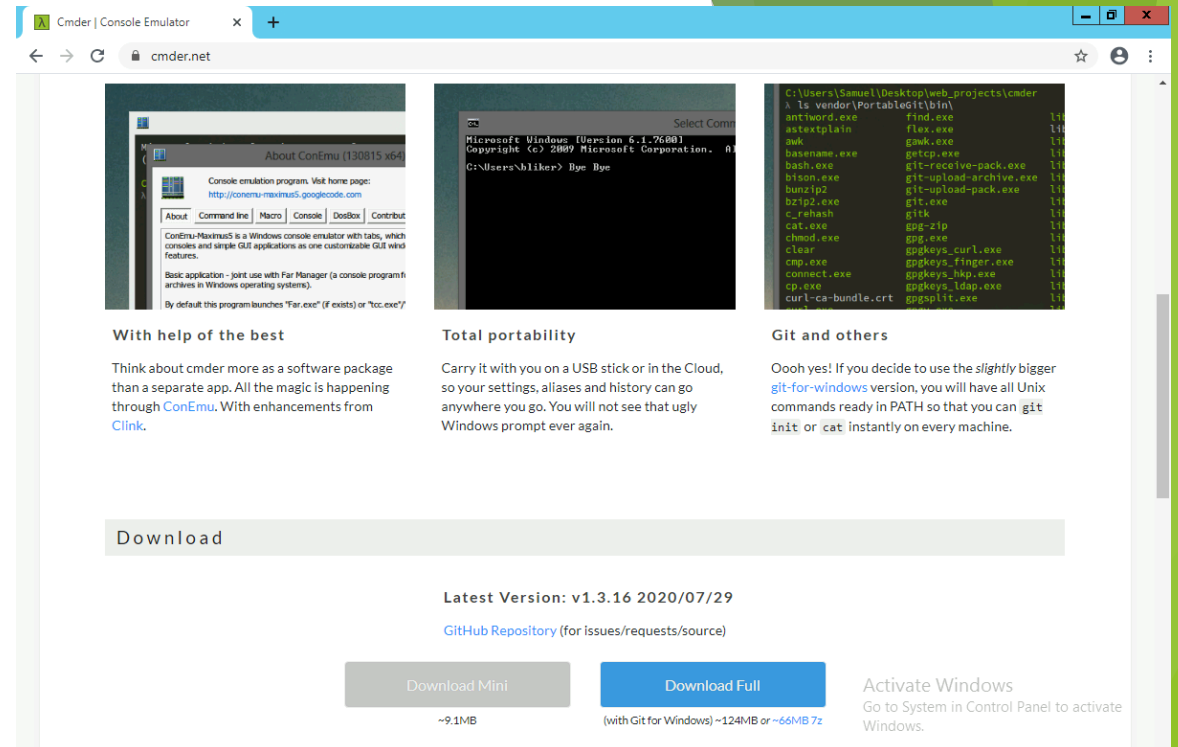
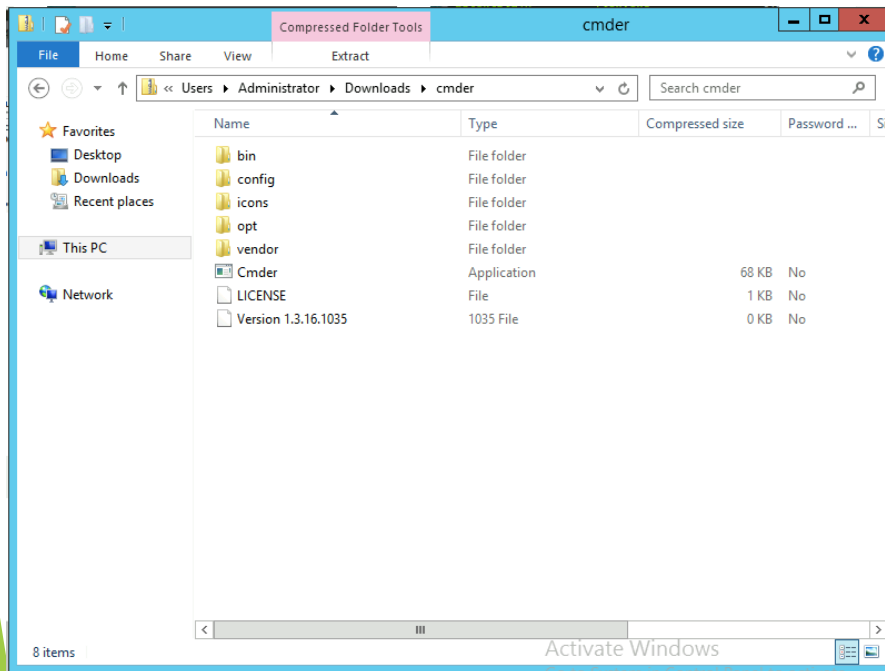


Problema dos projetos

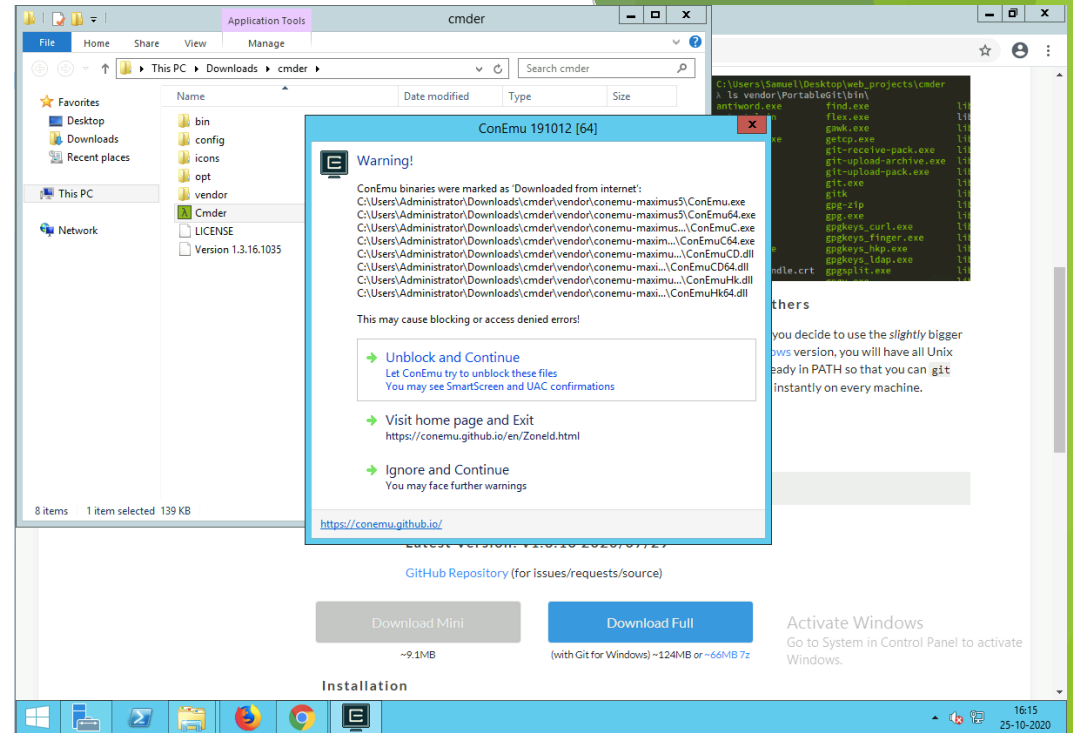
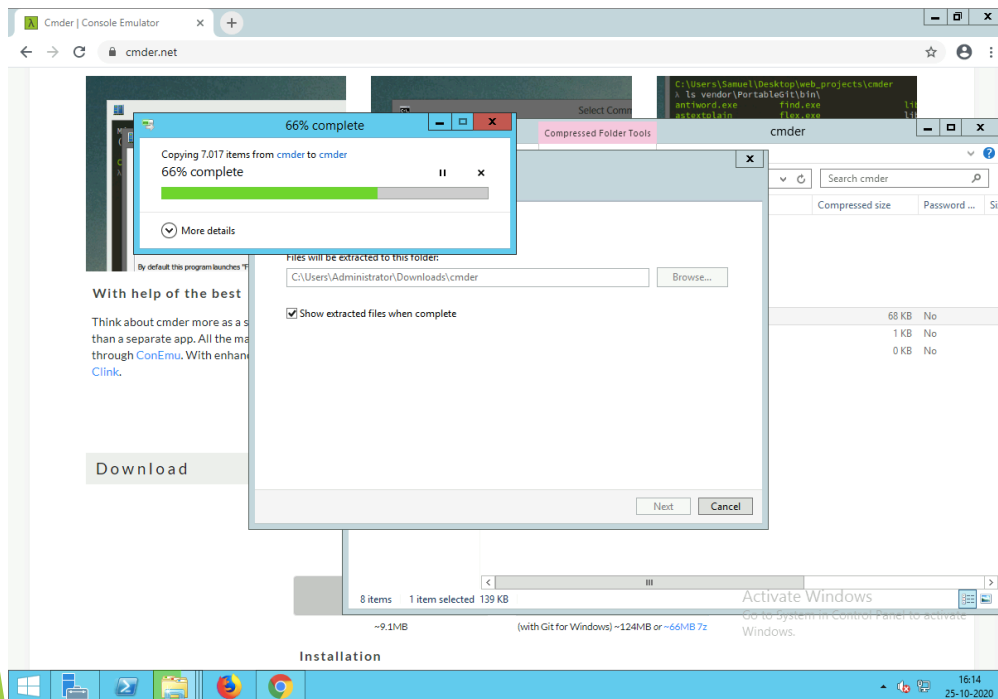


Trabalhar com o Git

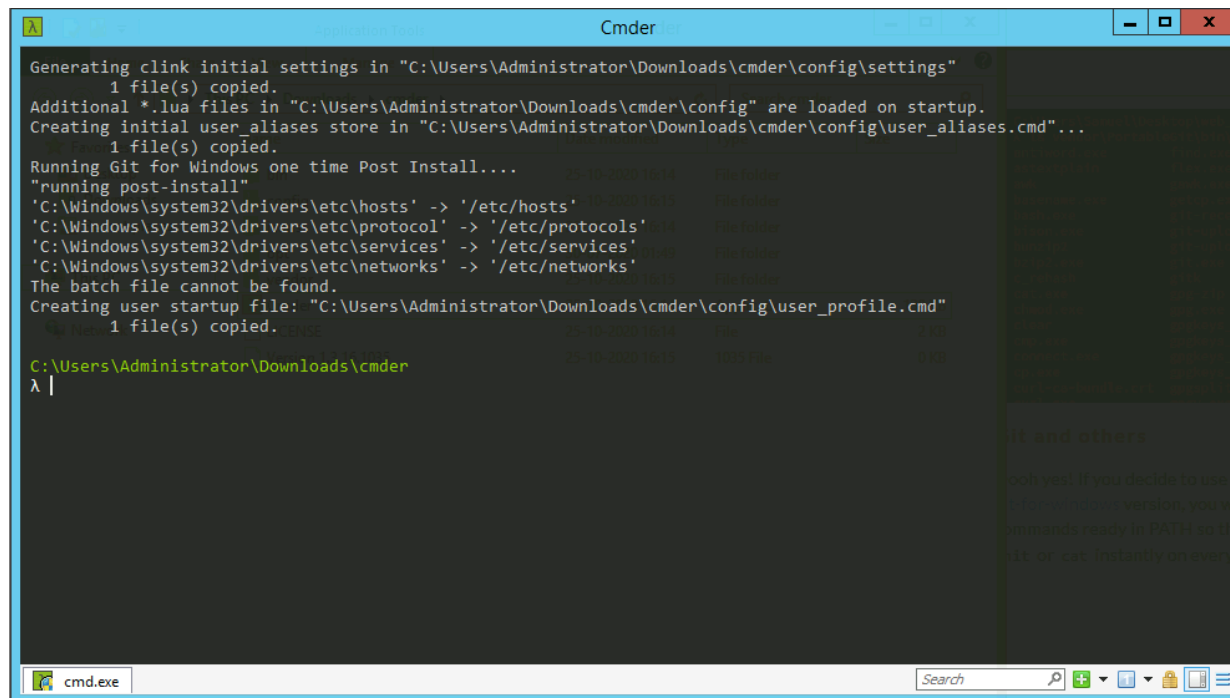
- ▶ Vamos usar o cmdr para emular o Git
- ▶ Extrair os dados



Git - Extrair a pasta



Git - linha de comandos

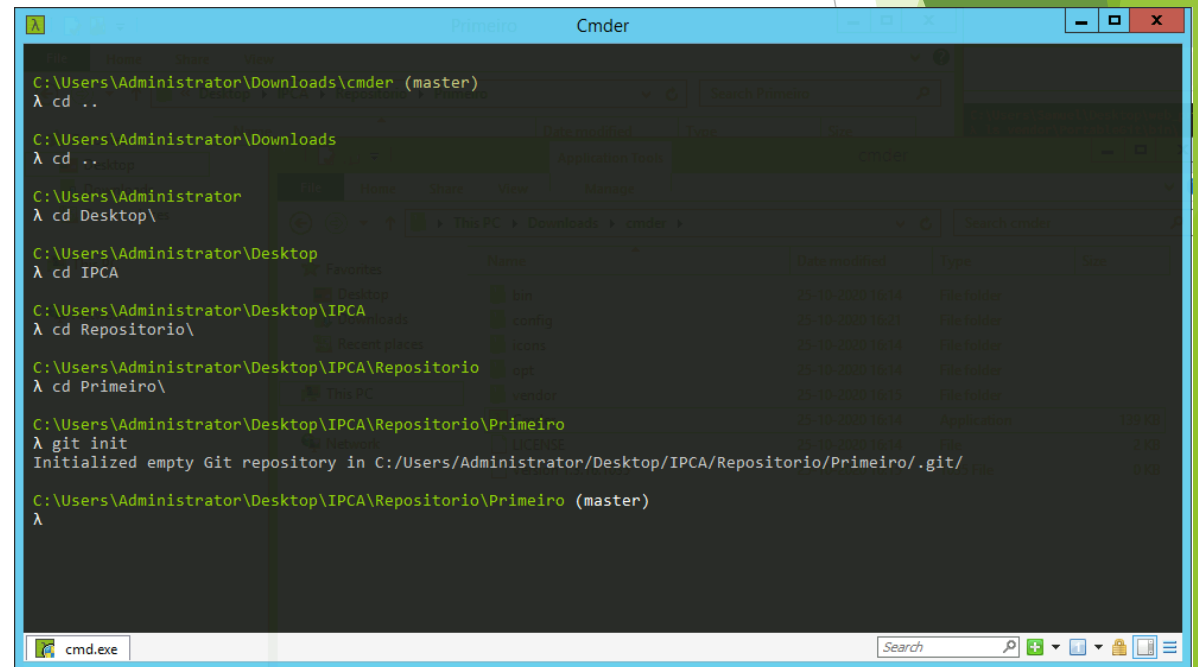


```
Generating clink initial settings in "C:\Users\Administrator\Downloads\cmder\config\settings"
1 file(s) copied.
Additional *.lua files in "C:\Users\Administrator\Downloads\cmder\config" are loaded on startup.
Creating initial user_aliases store in "C:\Users\Administrator\Downloads\cmder\config\user_aliases.cmd"...
1 file(s) copied.
Running Git for Windows one time Post Install....
"running post-install"
'C:\Windows\system32\drivers\etc\hosts' -> '/etc/hosts'
'C:\Windows\system32\drivers\etc\protocol' -> '/etc/protocols'
'C:\Windows\system32\drivers\etc\services' -> '/etc/services'
'C:\Windows\system32\drivers\etc\networks' -> '/etc/networks'
The batch file cannot be found.
Creating user startup file: "C:\Users\Administrator\Downloads\cmder\config\user_profile.cmd"
1 file(s) copied.

C:\Users\Administrator\Downloads\cmder>
λ |
```

Git - Criar o primeiro repositório

- ▶ Criar uma pasta (Repositorio -> Primeiro)
- ▶ Navegar via consola do Git até essa pasta
- ▶ Executar o comando: **git init**
- ▶ Foi criado o primeiro repositório local



```
C:\Users\Administrator\Downloads\cmdr (master)
λ cd ..

C:\Users\Administrator\Downloads
λ cd ..

C:\Users\Administrator
λ cd Desktop\

C:\Users\Administrator\Desktop
λ cd IPCA

C:\Users\Administrator\Desktop\IPCA
λ cd Repositorio\

C:\Users\Administrator\Desktop\IPCA\Repositorio
λ cd Primeiro\

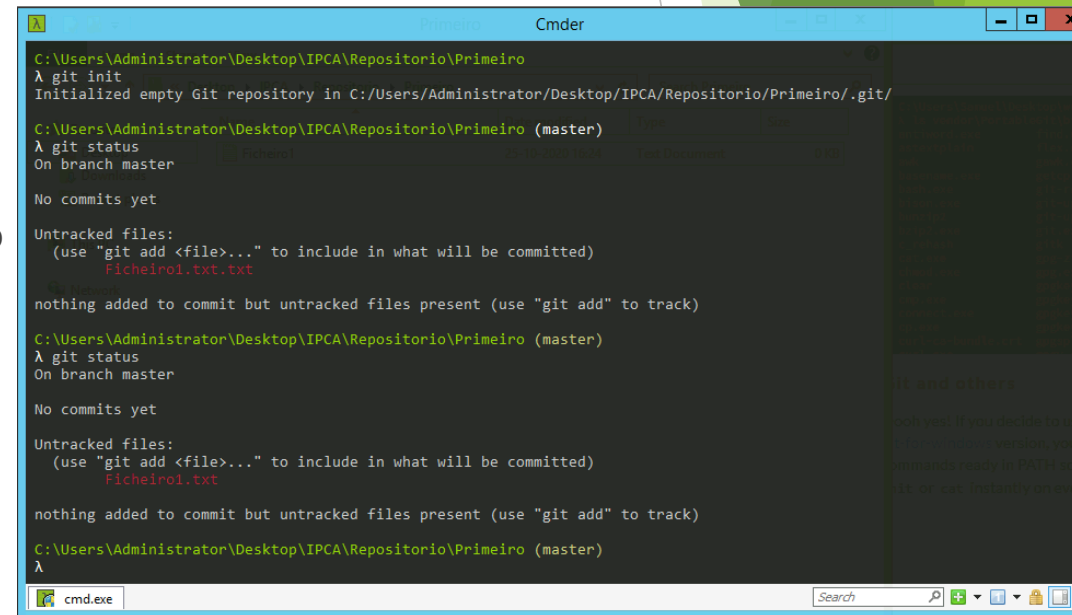
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro
λ git init
Initialized empty Git repository in C:/Users/Administrator/Desktop/IPCA/Repositorio/Primeiro/.git/

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ
```

The screenshot shows a Windows Command Prompt window titled 'cmdr' with a dark background. The command history shows the user navigating from the Downloads folder to the Desktop, then to a folder named 'IPCA', and finally to a subfolder 'Repositorio' and then 'Primeiro'. The final command executed is 'git init', which successfully initializes an empty Git repository in the current directory, as indicated by the output message: 'Initialized empty Git repository in C:/Users/Administrator/Desktop/IPCA/Repositorio/Primeiro/.git/'. The window title bar shows 'cmd.exe' and a search bar.

Git - Criar o primeiro repositório

- ▶ Criar um ficheiro na pasta com o nome Olamundo.txt
- ▶ Executar o comando **git status**
- ▶ Vermelho:
 - ▶ Ficheiros que nunca foram gravados no Git
 - ▶ Ficheiros que foram alterados mas ainda não estão gravados
- ▶ Verde:
 - ▶ Os ficheiros que estão sincronizados com o Git



```
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro
λ git init
Initialized empty Git repository in C:/Users/Administrator/Desktop/IPCA/Repositorio/Primeiro/.git/

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Ficheiro1.txt.txt

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

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git status
On branch master

No commits yet

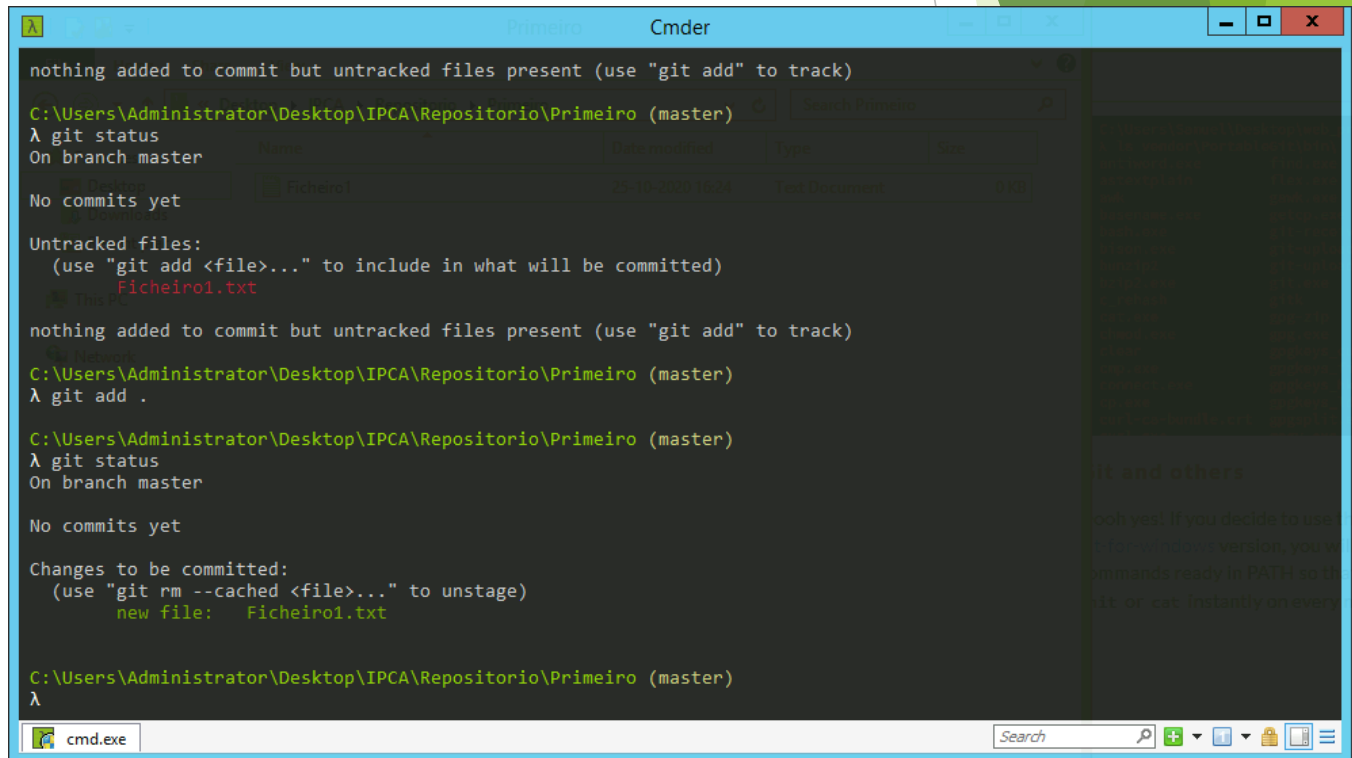
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Ficheiro1.txt

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

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ
```

Git - Criar o primeiro repositório

- ▶ Gravar o ficheiro Olamundo.txt
- ▶ Comando: `git add Olamundo.txt`
- ▶ Gravar todos os documentos
- ▶ Comando: `git add .`



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

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git status
On branch master
No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    Ficheiro1.txt

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

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git add .

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git status
On branch master
No commits yet

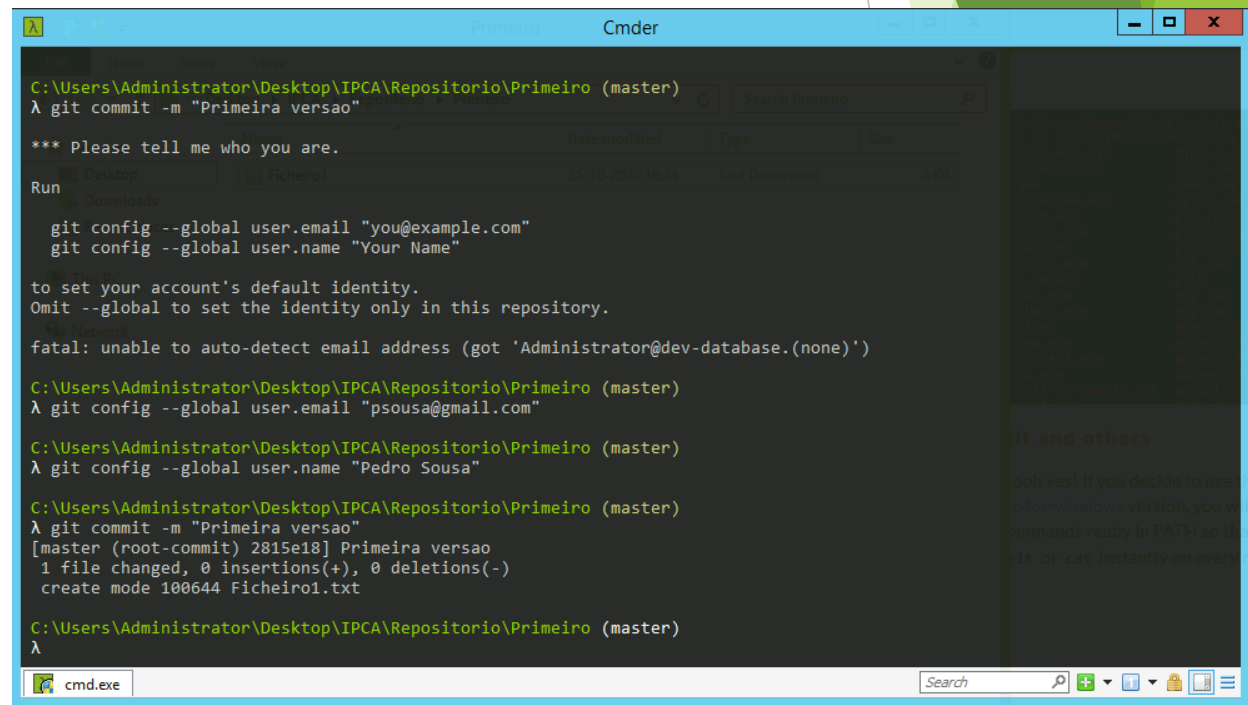
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Ficheiro1.txt

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ
```

The screenshot shows a Windows Command Prompt window titled "Primeiro Cmder". The window displays the output of several Git commands. It starts with a message indicating that there are untracked files. The user runs `git status`, which shows the current branch as master and lists `Ficheiro1.txt` as an untracked file. The user then runs `git add .` to stage the file. Finally, the user runs `git status` again, which shows that the file is now staged for commit under the heading "Changes to be committed".

Git - Criar o primeiro repositório

- ▶ Realizar o primeiro commit de dados
- ▶ Comando: `git commit -m "Primeira versão"`
- ▶ Criar a versão do nosso projeto
- ▶ Só são enviados os ficheiros que:
 - ▶ Estão na lista
 - ▶ Usando o comando `git add`



```
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git commit -m "Primeira versao"

*** Please tell me who you are.

Run: git config --global user.email "you@example.com"
      git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'Administrator@dev-database.(none)')

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git config --global user.email "psousa@gmail.com"

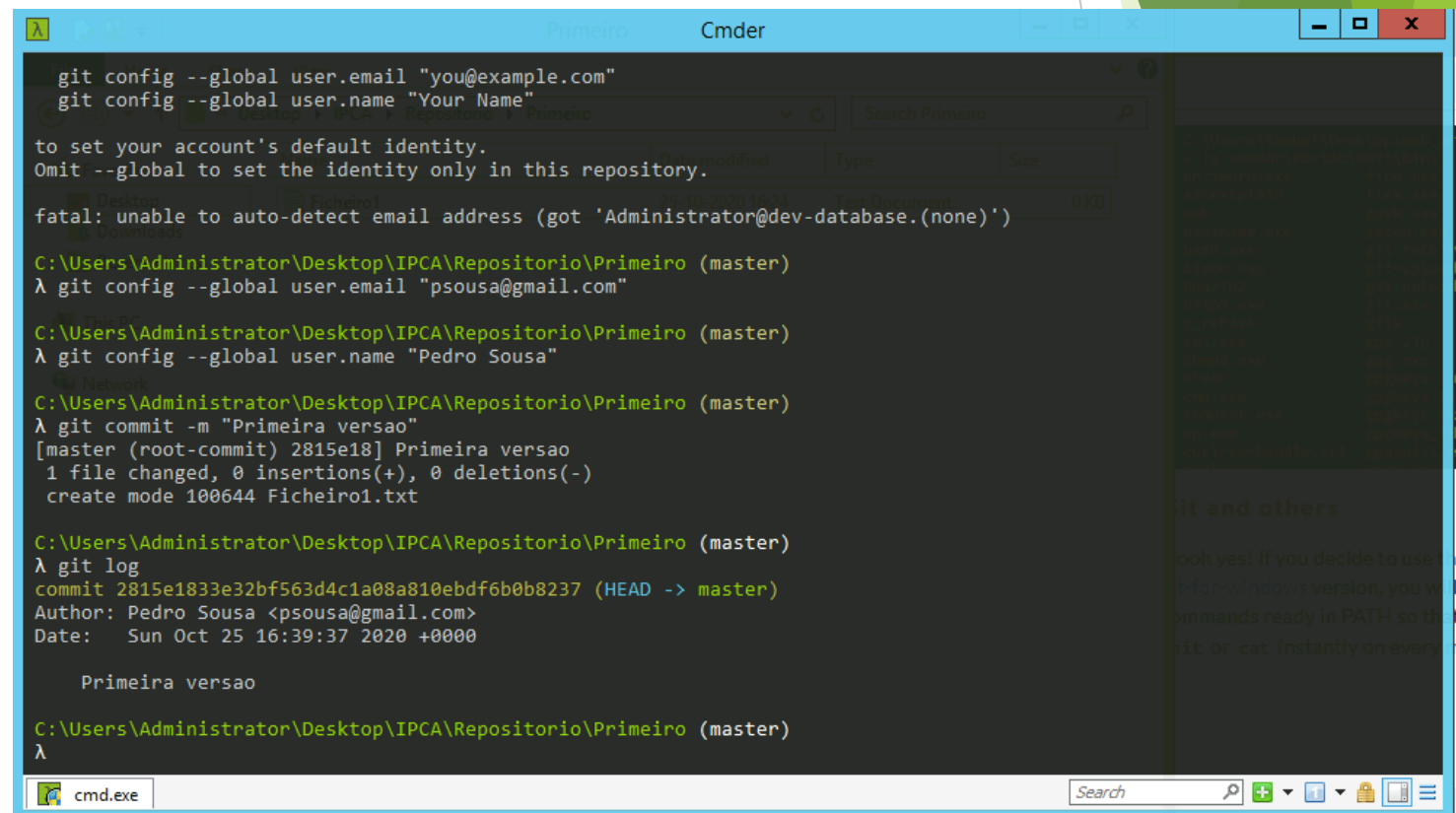
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git config --global user.name "Pedro Sousa"

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git commit -m "Primeira versao"
[master (root-commit) 2815e18] Primeira versao
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Ficheiro1.txt

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ
```

Git - Criar o primeiro repositório

- ▶ Ver os “commits” feitos
- ▶ Comando: `git log`



```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got 'Administrator@dev-database.(none)')

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git config --global user.email "psousa@gmail.com"

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git config --global user.name "Pedro Sousa"

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git commit -m "Primeira versao"
[master (root-commit) 2815e18] Primeira versao
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Ficheiro1.txt

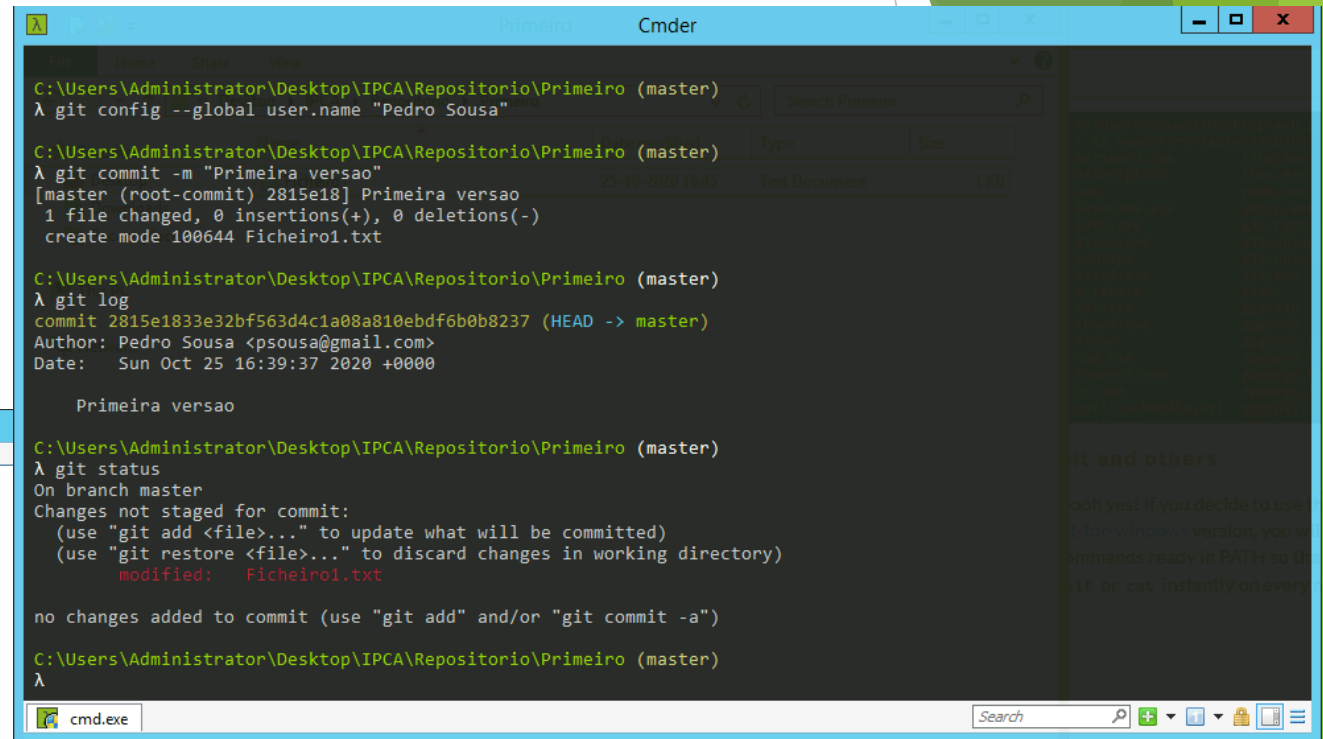
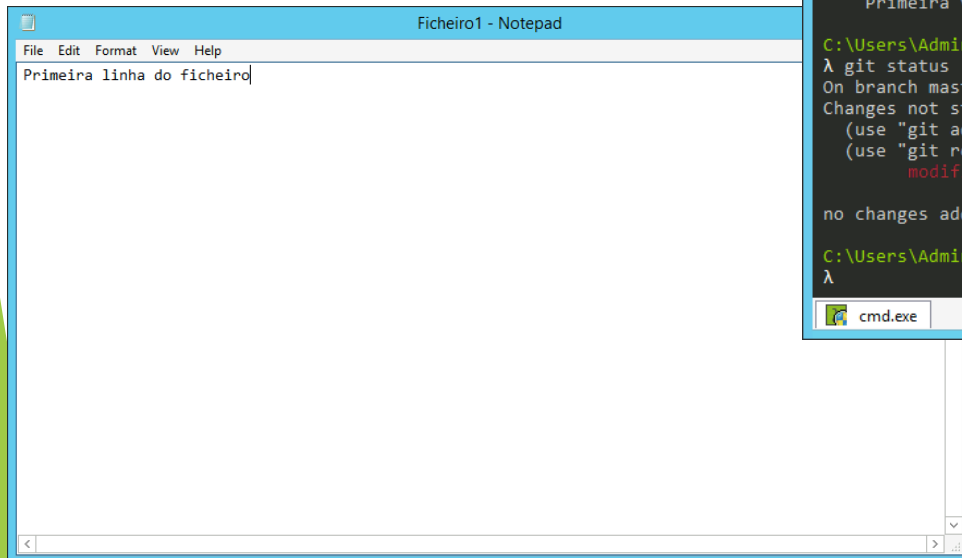
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git log
commit 2815e1833e32bf563d4c1a08a810ebdf6b0b8237 (HEAD -> master)
Author: Pedro Sousa <psousa@gmail.com>
Date:   Sun Oct 25 16:39:37 2020 +0000

    Primeira versao

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ
```


Git - Criar o primeiro repositório

- ▶ Fazer alterações ao ficheiro
- ▶ Verificar o estado
- ▶ Comando: git status



```
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git config --global user.name "Pedro Sousa"

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git commit -m "Primeira versao"
[master (root-commit) 2815e18] Primeira versao
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Ficheiro1.txt

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git log
commit 2815e1833e32bf563d4c1a08a810ebdf6b0b8237 (HEAD -> master)
Author: Pedro Sousa <psousa@gmail.com>
Date:   Sun Oct 25 16:39:37 2020 +0000

    Primeira versao

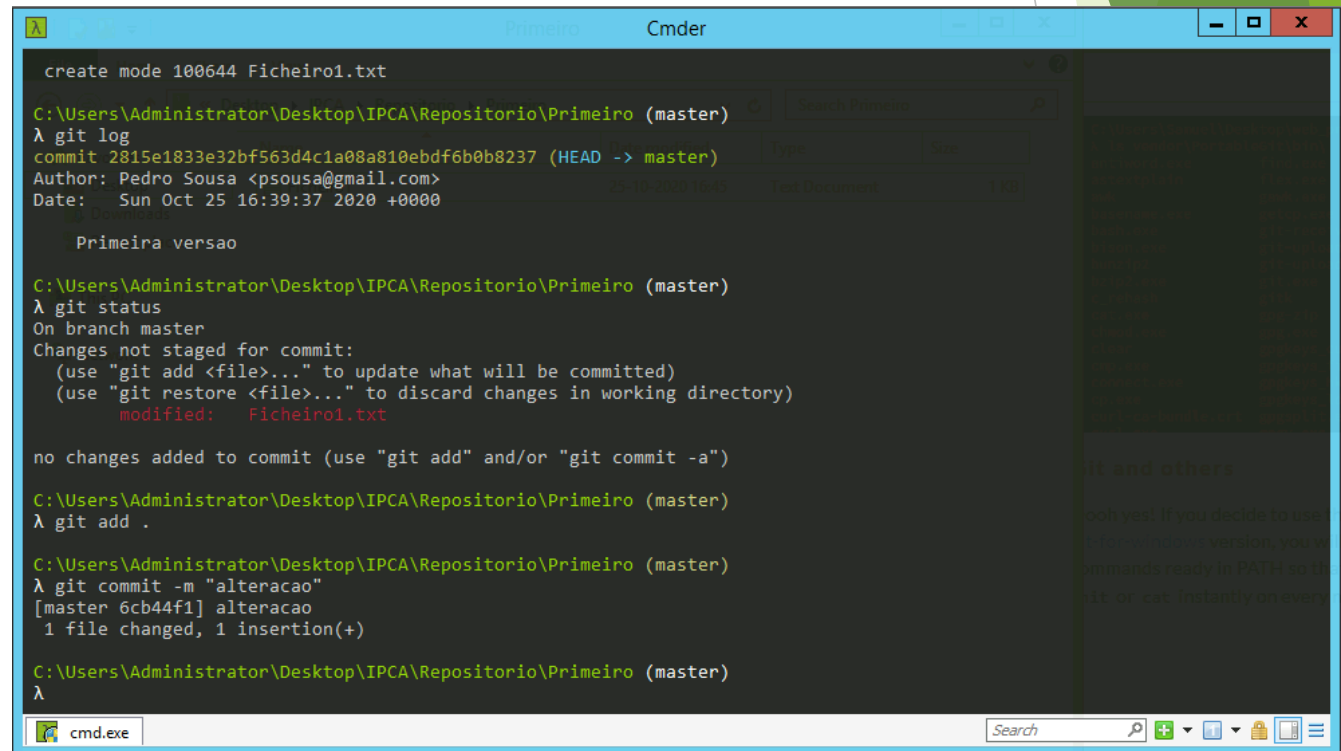
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Ficheiro1.txt

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ
```

Git - Criar o primeiro repositório

- Enviar as alterações efetuadas



```
create mode 100644 Ficheiro1.txt

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git log
commit 2815e1833e32bf563d4c1a08a810ebdf6b0b8237 (HEAD -> master)
Author: Pedro Sousa <psousa@gmail.com>
Date: Sun Oct 25 16:39:37 2020 +0000

    Primeira versao

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git status
On branch master
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Ficheiro1.txt

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git add .

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git commit -m "alteracao"
[master 6cb44f1] alteracao
1 file changed, 1 insertion(+)

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ
```

Git - Criar o primeiro repositório

- ▶ Executar nova alteração ao ficheiro
- ▶ Salvar e sincronizar o repositório
- ▶ Depois correr o comando: **gitk**
- ▶ Verificar o resultado
- ▶ Depois correr o comando: **git log -p filename**
- ▶ Verificar o resultado

Git - Criar o primeiro repositório

- ▶ Como recuperar uma versão de determinado ficheiro:
- ▶ Comandos:
 - ▶ `git log -p Ficheiro1.txt` (localizar o hash correto)
 - ▶ `git reset "hash" Ficheiro1.txt`
 - ▶ `git status` (ver o estado do repositório)
 - ▶ `git restore Ficheiro1.txt`
 - ▶ `git add .`
 - ▶ `git commit -m "comentário"`

Exercício prático

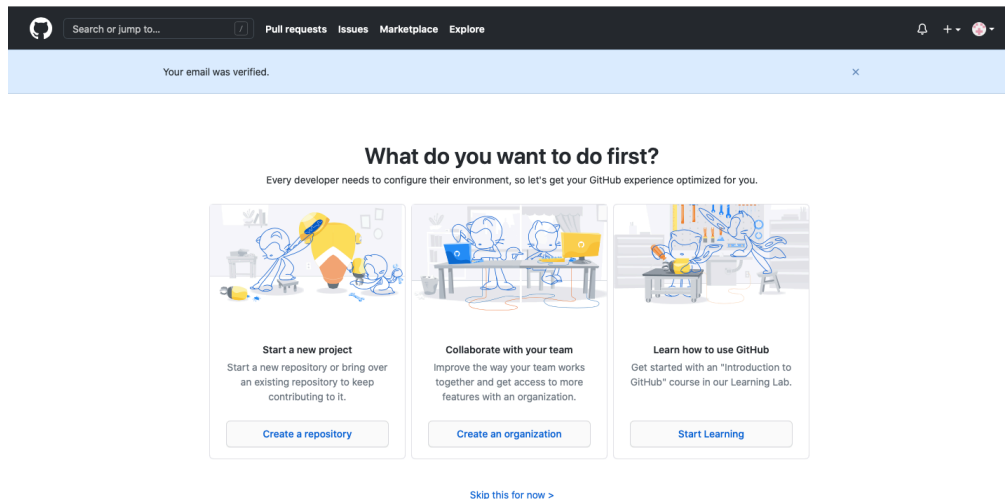
- ▶ Criar uma pasta com o nome “Segundo”
- ▶ Criar o ficheiro index.html
 - ▶ `<h1>Hello World</h1>`
 - ▶ `<p>O meu primeiro projeto com GIT</p>`
- ▶ Fazer o commit do projeto com a mensagem: segundo projeto
- ▶ Alterar o conteúdo para:
 - ▶ `<p>O meu segundo projeto com GIT</p>`
- ▶ Fazer um commit com a mensagem “corrigir o erro no ficheiro”
- ▶ Adicionar um novo ficheiro à pasta “Readme.txt” com alguma informação do projeto
- ▶ Fazer um commit com a mensagem “adicionar o ficheiro Readme.txt”

Exercício prático

- ▶ Alterar o conteúdo para:
 - ▶ `<p>O meu terceiro projeto com GIT</p>`
- ▶ Fazer um commit com a mensagem “correcao de novo erro”
- ▶ Agora vamos recuperar a versão 2 do ficheiro

Trabalhar com o GitHub

- ▶ Criar uma conta no GitHub (pode ser como estudante)
- ▶ Procurar a opção novo repositório
- ▶ Criar o novo repositório



Create a new repository

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

Owner ^{*} Repository name ^{*}

pedrocardososousa / primeiroGitHub ✓

Great repository names are short and memorable. Need inspiration? How about **fantastic-robot**?

Description (optional)

Primeiro Repositório de GitHub

☒ **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.](#)

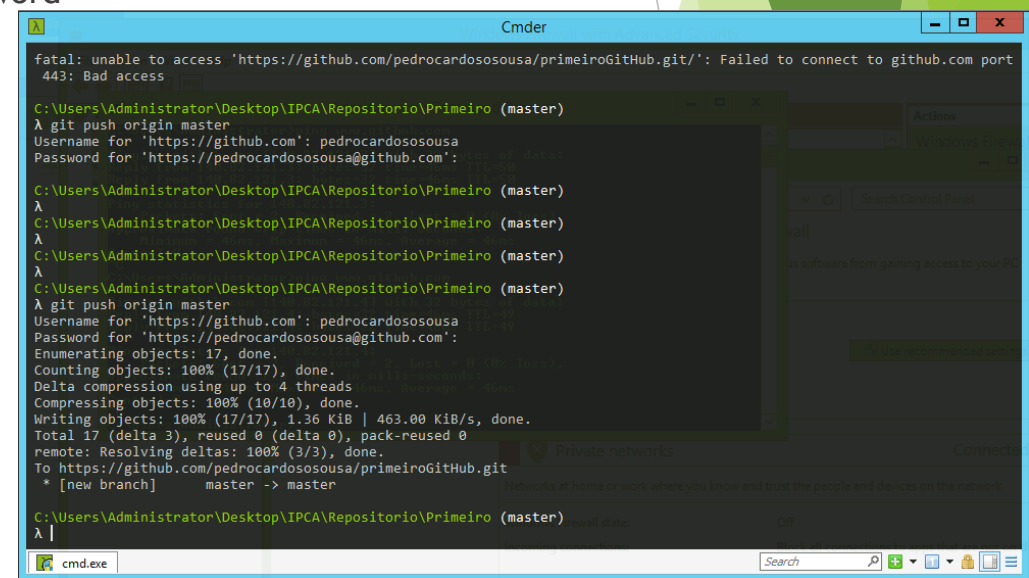
☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

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

Create repository

Trabalhar com o GitHub

- ▶ Guardar as credenciais do GitHub
 - ▶ Comando: `git config --global credential.helper store`
- ▶ Alterar as credenciais
 - ▶ Comando: `git config --global credential.username "username"`
 - ▶ Comando: `git config --global credential.password "password"`
- ▶ Alterar o repositório
 - ▶ Comando: `git remote add origin "url"`
- ▶ Ligar o repositório local ao repositório remoto
 - ▶ Comando: `git remote add origin "url_do_respositorio"`
 - ▶ Ou: `git remote set-url origin "url_do_respositorio"`
- ▶ Enviar os ficheiros para o GitHub
 - ▶ Comando: `git push origin master`



```
fatal: unable to access 'https://github.com/pedrocardososousa/primeiroGitHub.git/': Failed to connect to github.com port 443: Bad access

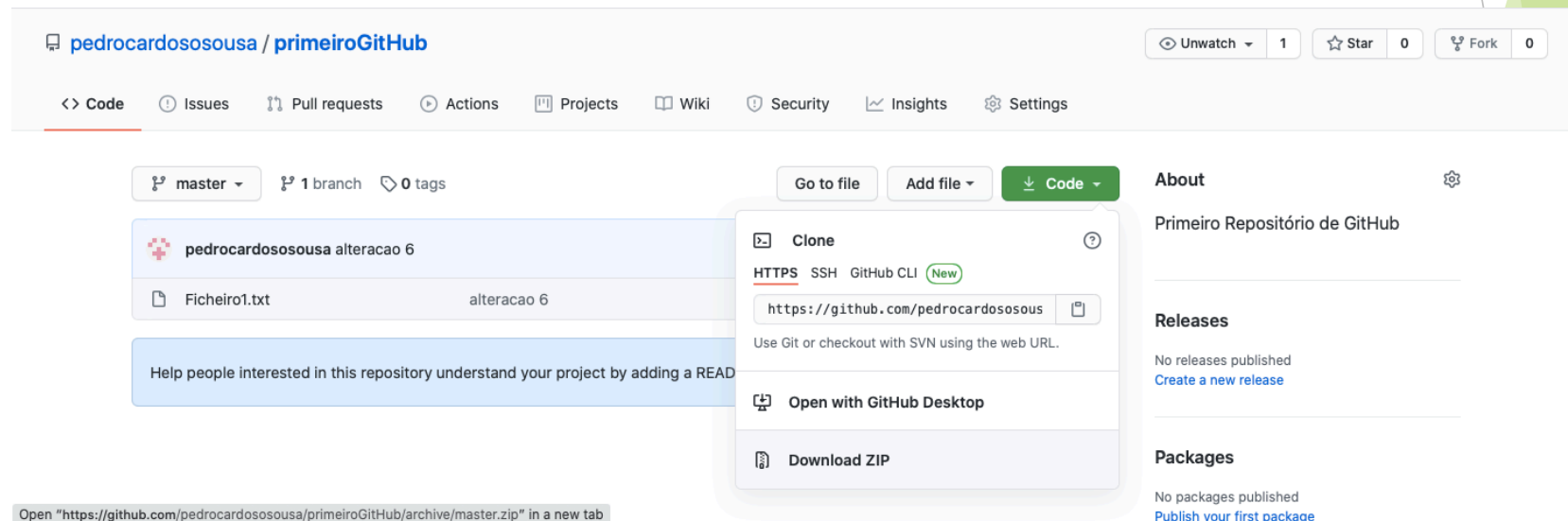
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git push origin master
Username for 'https://github.com': pedrocardososousa
Password for 'https://pedrocardososousa@github.com':

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git push origin master
Username for 'https://github.com': pedrocardososousa
Password for 'https://pedrocardososousa@github.com':
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 4 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (17/17), 1.36 KiB | 463.00 KiB/s, done.
Total 17 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/pedrocardososousa/primeiroGitHub.git
 * [new branch]      master -> master

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ
```


Trabalhar com o GitHub

- ▶ Fazer download de um projeto no interface web é simples, seguindo a imagem.
- ▶ Na linha de comando podes usar:
 - ▶ Comando: `git clone "url"`



Exercício prático

- ▶ Criar um novo repositório no GitHub para o exercício 1
- ▶ Ligar o repositório local ao repositório remoto (GitHub)
- ▶ Alterar o ficheiro “Readme.txt”: adicionar a informação do URL do GitHub
- ▶ Fazer novo commit

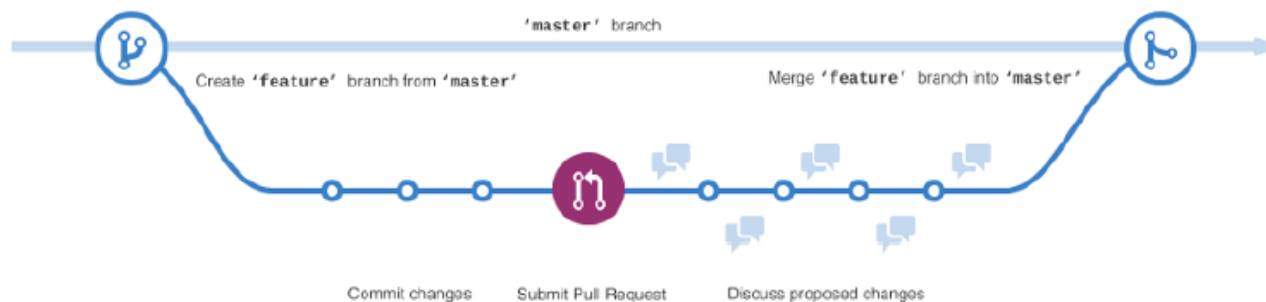
Trabalhar com “branches”

- ▶ Esta é uma funcionalidade interessante dos sistemas Git, ou seja, Ramos ou “Branches”.
- ▶ A melhor forma de definir uma “branch” é: um ambiente diferente para o mesmo projeto
- ▶ Por padrão o Git define o “Branch” com o nome “master”.
- ▶ Este é o “branch” principal e neste devemos ter sempre a última versão estável do projeto



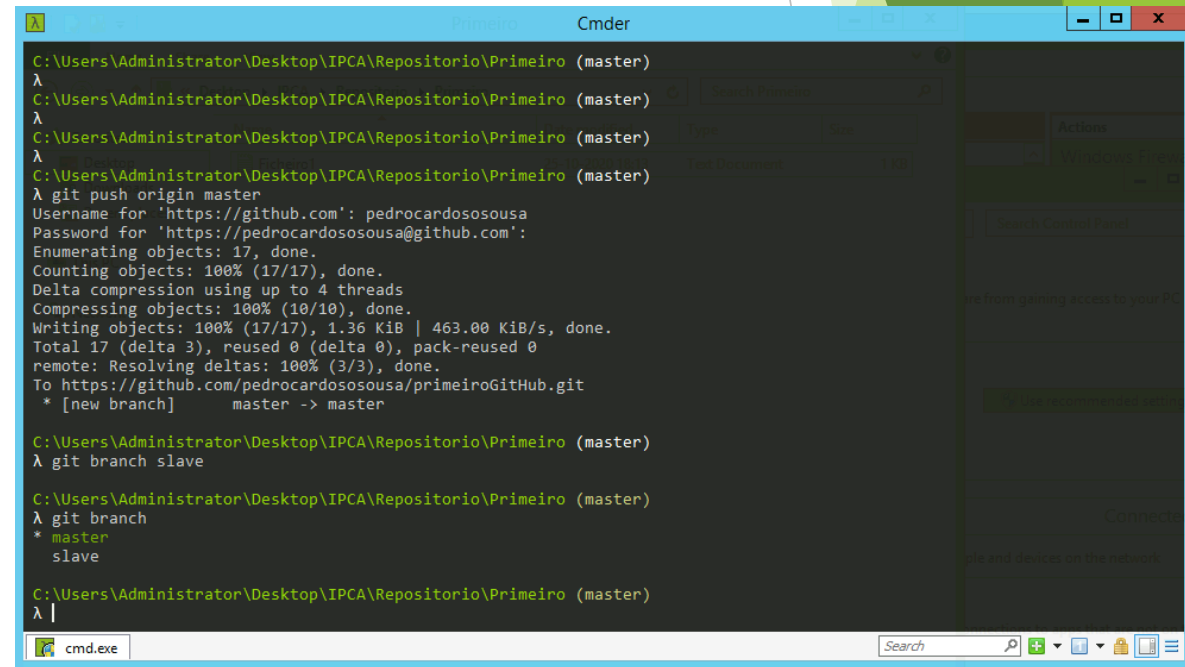
Trabalhar com “branches”

- ▶ As boas práticas dizem que sempre que pretendemos fazer alterações, devemos criar uma nova “branch”.
- ▶ Todas as alterações e “commit” devem ser feitas neste “branch”.
- ▶ No final realizar um pedido para incorporar estas alterações no “branch master”.
- ▶ Os processos de revisão de projeto são realizados neste momento de integração de versões na “branch master”.



Criar uma “branch”

- ▶ No Ficheiro1.txt vamos adicionar uma linha com a seguinte descrição: “trabalho com branch”
- ▶ Não vamos fazer commit, pois este seria na “branch master”
- ▶ Vamos criar um “branch” novo
 - ▶ Comando: git branch “nome_do_branch”



```
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git push origin master
Username for 'https://github.com': pedrocardososousa
Password for 'https://pedrocardososousa@github.com':
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 4 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (17/17), 1.36 KiB | 463.00 KiB/s, done.
Total 17 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/pedrocardososousa/primeiroGitHub.git
 * [new branch]      master -> master

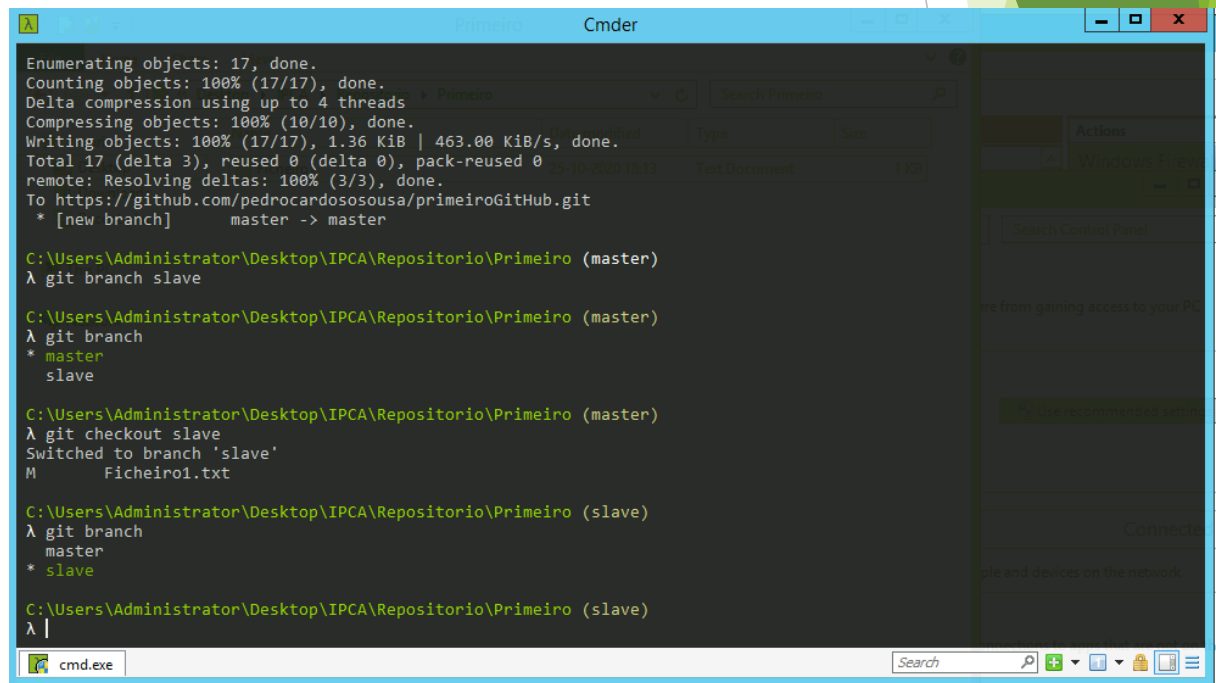
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git branch slave

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git branch
* master
  slave

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ |
```

Criar uma “branch”

- Mudar de “branch”:
 - Comando: `git checkout “nome_do_branch”`



```
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 4 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (17/17), 1.36 KiB | 463.00 KiB/s, done.
Total 17 (delta 3), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (3/3), done.
To https://github.com/pedrocardososousa/primeiroGitHub.git
 * [new branch]      master -> master

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git branch
λ git branch
 * master
  slave

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (master)
λ git checkout slave
Switched to branch 'slave'
M       Ficheiro1.txt

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (slave)
λ git branch
  master
 * slave

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (slave)
λ
```

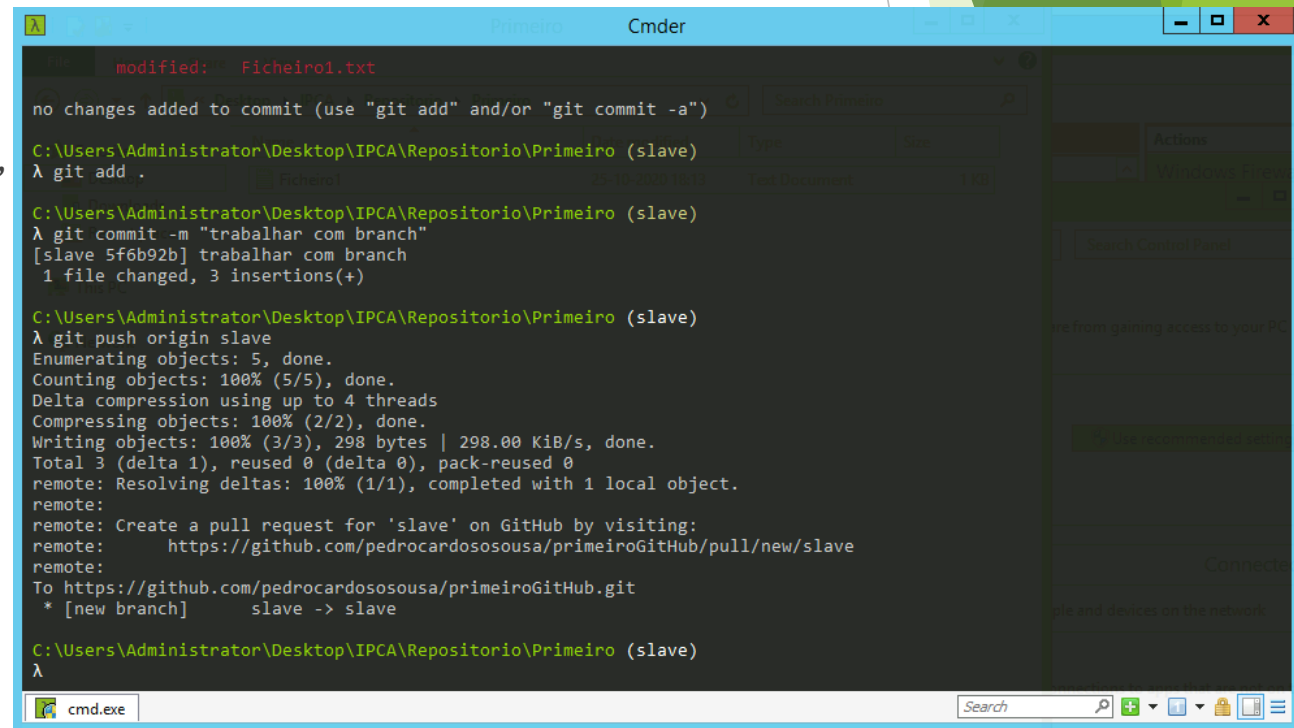
Criar uma “branch”

- ▶ Criar o “branch” e fazer “checkout”
 - ▶ Comando: `git checkout -b “nome_do_branch”`



Criar uma “branch”

- ▶ Fazer o “commit” do ficheiro
 - ▶ Comandos:
 - ▶ git status
 - ▶ git add .
 - ▶ git commit -m “trabalho com branch”
 - ▶ git push origin “nome_do_branch”



```
modified: Ficheiro1.txt

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (slave)
λ git add .

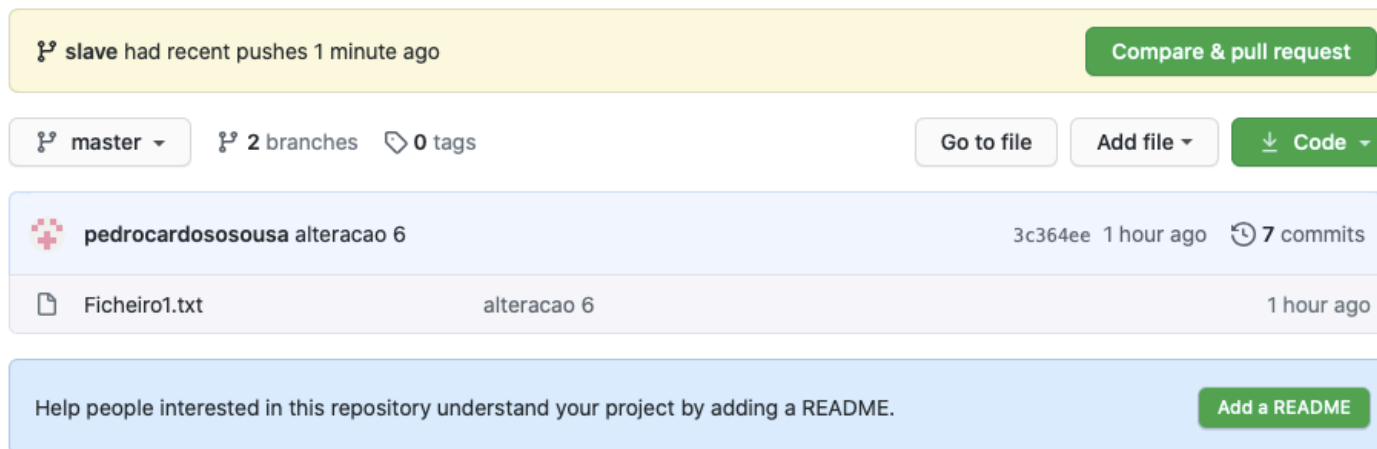
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (slave)
λ git commit -m "trabalhar com branch"
[slave 5f6b92b] trabalhar com branch
1 file changed, 3 insertions(+)

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (slave)
λ git push origin slave
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 298 bytes | 298.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
remote:
remote: Create a pull request for 'slave' on GitHub by visiting:
remote:   https://github.com/pedrocardososousa/primeiroGitHub/pull/new/slave
remote:
To https://github.com/pedrocardososousa/primeiroGitHub.git
 * [new branch]      slave -> slave

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (slave)
λ
```


Criar uma “branch”

- ▶ Neste momento temos 2 “branches” na nossa conta de GitHub
- ▶ Podemos fazer um pedido de “pull-request”

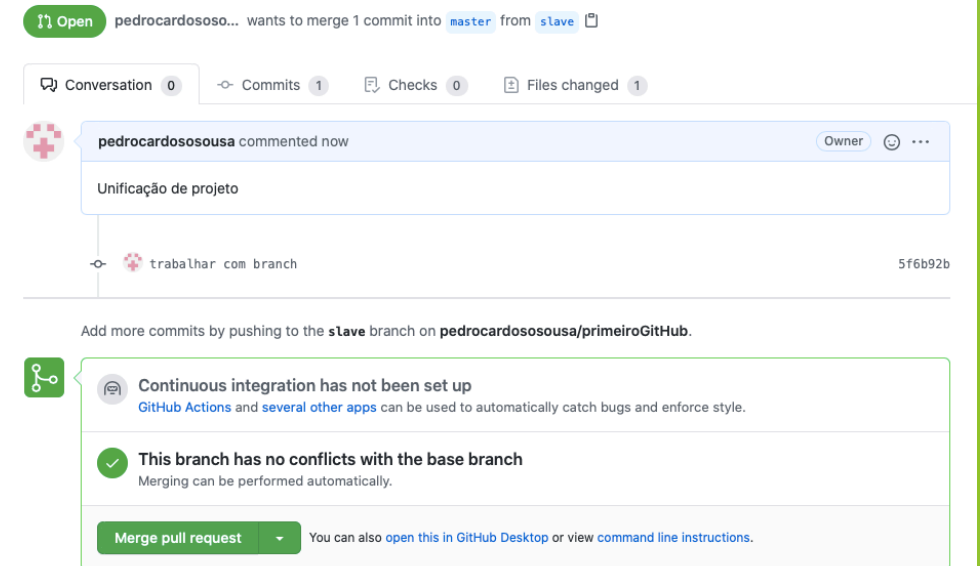


Pull-request é o processo de unificar o projeto entre os vários desenvolvimentos.

Criar uma “branch”

- ▶ O GitHub irá verificar as duas “branch” e verificar se existe algum conflito
- ▶ Caso não exista conflito será possível unificar o projeto
- ▶ Em desenvolvimento, inicia-se a fase de revisão de código

trabalhar com branch #1



The screenshot displays a GitHub pull request interface. At the top, a green button labeled 'Open' is followed by the text 'pedrocardososousa... wants to merge 1 commit into master from slave'. Below this, a navigation bar shows 'Conversation 0', 'Commits 1', 'Checks 0', and 'Files changed 1'. A comment from 'pedrocardososousa' is visible, stating 'Unificação de projeto'. The commit history shows a single commit titled 'trabalhar com branch' with the hash '5f6b92b'. A message below the commit history reads: 'Add more commits by pushing to the slave branch on pedrocardososousa/primeiroGitHub.' The bottom section contains two status messages: 'Continuous integration has not been set up' (with a link to GitHub Actions) and 'This branch has no conflicts with the base branch' (indicating automatic merging is possible). At the bottom, there is a 'Merge pull request' button and a link to GitHub Desktop instructions.

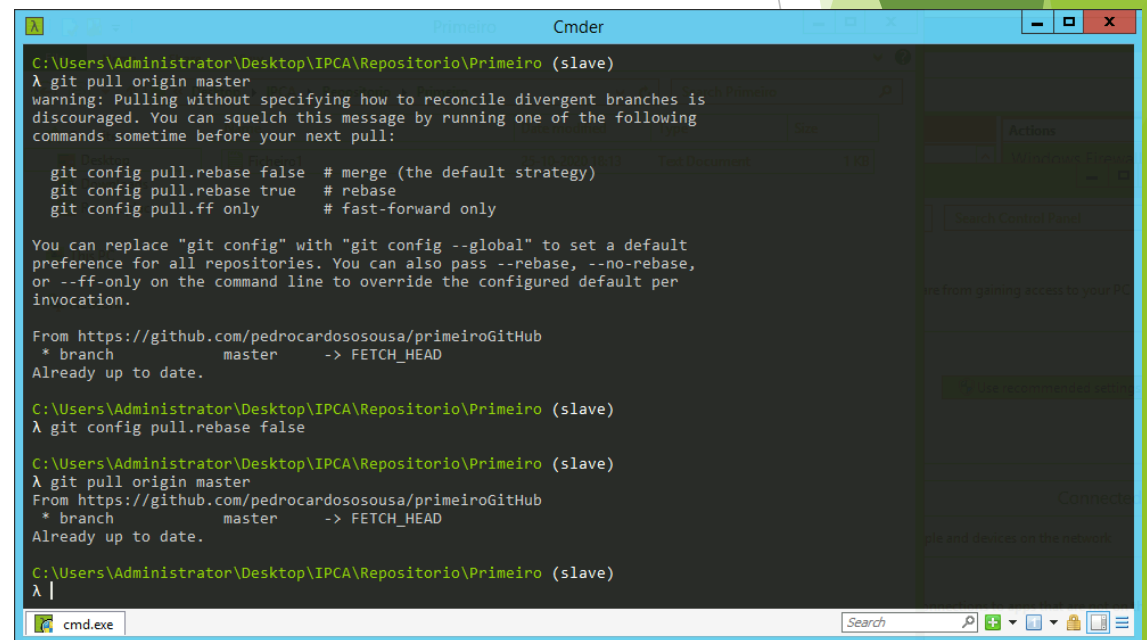
Criar uma “branch”

- Se a unificação for efetuada com sucesso é possível remover a “branch”

The screenshot displays a GitHub pull request interface. At the top, there are tabs for 'Conversation' (0), 'Commits' (1), 'Checks' (0), and 'Files changed' (1). Below the tabs, a comment from 'pedrocardososousa' is visible, stating 'Unificação de projeto'. The commit history shows a commit 'trabalhar com branch' by 'pedrocardososousa' with hash '5f6b92b'. Below this, a message indicates that the pull request was successfully merged into the 'master' branch, with a 'Revert' button. At the bottom, a purple box contains the message 'Pull request successfully merged and closed' and 'You're all set—the slave branch can be safely deleted.', along with a 'Delete branch' button.

Criar uma “branch”

- ▶ Sincronizar as alterações para o Git local
 - ▶ Comando: `git pull origin master`
- ▶ Recomenda-se ter todos os “branch” atualizados



```
C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (slave)
λ git pull origin master
warning: Pulling without specifying how to reconcile divergent branches is
discouraged. You can squelch this message by running one of the following
commands sometime before your next pull:

    git config pull.rebase false  # merge (the default strategy)
    git config pull.rebase true   # rebase
    git config pull.ff only       # fast-forward only

You can replace "git config" with "git config --global" to set a default
preference for all repositories. You can also pass --rebase, --no-rebase,
or --ff-only on the command line to override the configured default per
invocation.

From https://github.com/pedrocardososousa/primeiroGitHub
 * branch                master       -> FETCH_HEAD
Already up to date.

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (slave)
λ git config pull.rebase false

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (slave)
λ git pull origin master
From https://github.com/pedrocardososousa/primeiroGitHub
 * branch                master       -> FETCH_HEAD
Already up to date.

C:\Users\Administrator\Desktop\IPCA\Repositorio\Primeiro (slave)
λ
```

Criar uma “branch”

- ▶ Unificar “branches” diferentes
 - ▶ Comando: `git merge A B`



Exercício prático

- ▶ Criar um branch com o nome “page1” e adicionar um novo ficheiro page1.html
- ▶ Fazer commit deste novo ficheiro
- ▶ Alterar o ficheiro index.html e adicionar um link para o page1.html
- ▶ Fazer commit da alteração ao ficheiro index.html
- ▶ Criar um branch com o nome “page2” e adicionar um novo ficheiro page2.html
- ▶ Fazer commit deste novo ficheiro
- ▶ Fazer merge do page2 para o page1
- ▶ Abrir o PR e fazer o merge do branch page1 para o master
- ▶ Fazer pull do master

Outras informações

- ▶ Lista de comandos Git
 - ▶ Link: <https://gist.github.com/leocomelli/2545add34e4fec21ec16>
- ▶ Aprender comandos de Git
 - ▶ Link: <https://learngitbranching.js.org>

Git em Ambiente Gráfico

Programa	Windows	MacOSx	Linux
GitKraken	X	X	X
GitHub Desktop	X	X	
SourceTree	X	X	
Fork	X	X	