

Guião - Introdução ao Git

1. Configurar git no PC
 - a. <https://docs.github.com/en/get-started/quickstart/set-up-git#setting-up-git>
 - b. <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>
 - c. <https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account>
2. Criar repositório no Github
3. Dar clone com ssh - **git clone git@...**
4. Adicionar um ficheiro **touch README.md** "Hello World"
5. Criar um .gitignore e explicar a diferença entre **git add .** ou **git add ficheiro**. Porque é que os .gitignore são muito importantes para projetos

**/*logs	logs/debug.log logs/monday/foo.bar build/logs/debug.log
**/*logs/debug.log	logs/debug.log build/logs/debug.log <i>but not</i> logs/build/debug.log
*.log !important.log	debug.log <i>but not</i> logs/debug.log
debug.log	debug.log logs/debug.log
logs/	logs/debug.log logs/latest/foo.bar build/logs/foo.bar build/logs/latest/debug.log
logs/**/debug.log	logs/debug.log logs/monday/debug.log logs/monday/pm/debug.log
logs/debug.log	logs/debug.log <i>but not</i>

	debug.log build/logs/debug.log
--	-----------------------------------

6. Ver que com .gitignores diferentes aparece ou não o README.md no **git status**
7. **git add README.md, git commit -m ""**
8. **git push**

Agora vamos tentar simular uns conflitos

9. Alterar o README.md, **git add README.md, git commit -m ""** mas não git push
10. Fazer alterações também online
11. Tentar **git push** e ver o erro
12. Vamos fazer **git pull --no-rebase** para ele criar um merge e escolhermos o que queremos deixar ou não
13. **git add README.md, git commit -m ""** e **git push**

Agora vamos simular um conflito sem dar o pull

14. Alterar o README.md
15. Fazer alterações também online
16. Tentar dar **git pull** e ver o erro
17. Vamos fazer **git stash, git pull** e **git stash pop** e vai meio que acontecer um merge porque as alterações foram feitas em linhas diferentes
18. **git add README.md, git commit -m ""** e **git push**

Agora vamos fazer uma branch e dar merge da mesma

19. Criar uma branch
 - a. **git branch new-feature**
 - b. **git checkout new-feature**
 - c. Criar um ficheiro **touch feature**
 - d. Ir para a main e ver que as changes estão lá, isto é porque não definimos o upstream, para o github não existe ainda uma segunda branch, so localmente
 - e. Fazer **git add feature, git commit -m "add feature"** e **git push** e ver o erro que dá
 - f. Dar set da upstream
20. Ir à main ver não aparecas coisas da outra branch, vamos criar um ficheiro **touch file, git add file, git commit -m "added file", git push**

21. Fazer **git merge new-feature** e **git push**, não é preciso um commit
22. Vamos remover a branch que já não é necessária **git branch -d new-feature** e **git push origin -d new-feature**

Coisas adicionais do Github

23. Corrigir um issue
 - a. Criar issue no Github
 - b. Criar uma commit que contenha “close #number”
24. O que é um fork?
25. Criar um release final com o conteúdo da apresentação em .pdf