



GIT & GITHUB

Versionamento de código

Por: Juliana Neres – 4ºADS

“

É um sistema criado para **gerenciar/controlar** as versões criadas de arquivos. A partir destas versões criadas é possível recuperar versões anteriores.

”

PROBLEMAS

Dificuldade de trabalhar em
equipe/ desenvolvimento paralelo

&

Falta de registros históricos sobre o
desenvolvimento (documentação)

PROBLEMAS

git

sistema de controle de versão,
usado para o **versionamento local**

!=

github

Site que utiliza do controle de versão
git para **manter repositórios remotos**

```
$ git config --list
```

Lista as configurações que já estão realizadas

```
$ git config --global user.name
```

Configura um nome de usuário globalmente

```
$ git config --global user.email
```

Configura um e-mail usuário globalmente

```
$ git init
```

Comando responsável por iniciar um repositório

ESTADOS DE UM REPOSITÓRIO

Modificado, preparado e consolidado

SESSÕES DE UM REPOSITÓRIO

Diretório de trabalho, área de preparação e diretório git

ARQUIVO README.md

USO BÁSICO

```
$ git add
```

Comando responsável por adicionar os arquivos à “área de espera”

```
$ git status
```

Comando responsável por verificar o estado do repositório

```
$ git commit
```

Comando responsável por “empacotar” os arquivos e coloca-los, de fato, no repositório.

```
$ git reset HEAD
```

Comando responsável por desfazer alterações do repositório

PARÂMETROS

```
$ git reset --soft
```

Retorna os arquivos para a área de preparação

```
$ git reset --mixed
```

Desfaz o commit já feito e depois retorna os arquivos para a área de preparação

```
$ git reset --hard
```

Desfaz o commit e apaga as modificações já realizadas

\$ git log

Comando responsável por apresentar o histórico do repositório

\$ git branch

Comando responsável por criar um novo “ramo” do repositório

\$ git checkout

Comando responsável por selecionar um novo “ramo”
Responsável também por descarte de modificações

```
$ git merge
```

Comando responsável por juntar as modificações de duas branches

LIDANDO COM CONFLITOS DE MERGE

```
C:\Users\jneres\Documents\oficina-git (master -> origin)
λ git merge nova-feature
Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

USO BÁSICO

LIDANDO COM CONFLITOS DE MERGE

```
<> index.html x
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <meta http-equiv="X-UA-Compatible" content="ie=edge">
7      <title>Document</title>
8  </head>
9  <body>
    Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
10  <<<<<<< HEAD (Current Change)
11      <h2>Novo cabeçalho master</h2>
12  =====
13      <h1>Novo cabeçalho - branche nova feature</h1>
14  >>>>>>> nova-feature (Incoming Change)
15  </body>
16  </html>
```

USO BÁSICO

LIDANDO COM CONFLITOS DE MERGE

```
C:\Users\jneres\Documents\oficina-git (master -> origin)
```

```
λ git status
```

```
On branch master
```

```
You have unmerged paths.
```

```
    (fix conflicts and run "git commit")
```

```
    (use "git merge --abort" to abort the merge)
```

```
Unmerged paths:
```

```
    (use "git add <file>..." to mark resolution)
```

```
        both modified:   index.html
```

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

USO BÁSICO

GITHUB

Trabalhando com repositórios remotos

Arquivo .gitignore

<https://www.gitignore.io/>

`$ git clone`

Comando responsável por copiar um repositório remoto para a máquina local.

`$ git fork`

Comando responsável por copiar um repositório remoto para seu perfil no github.

GITHUB

```
$ git pull
```

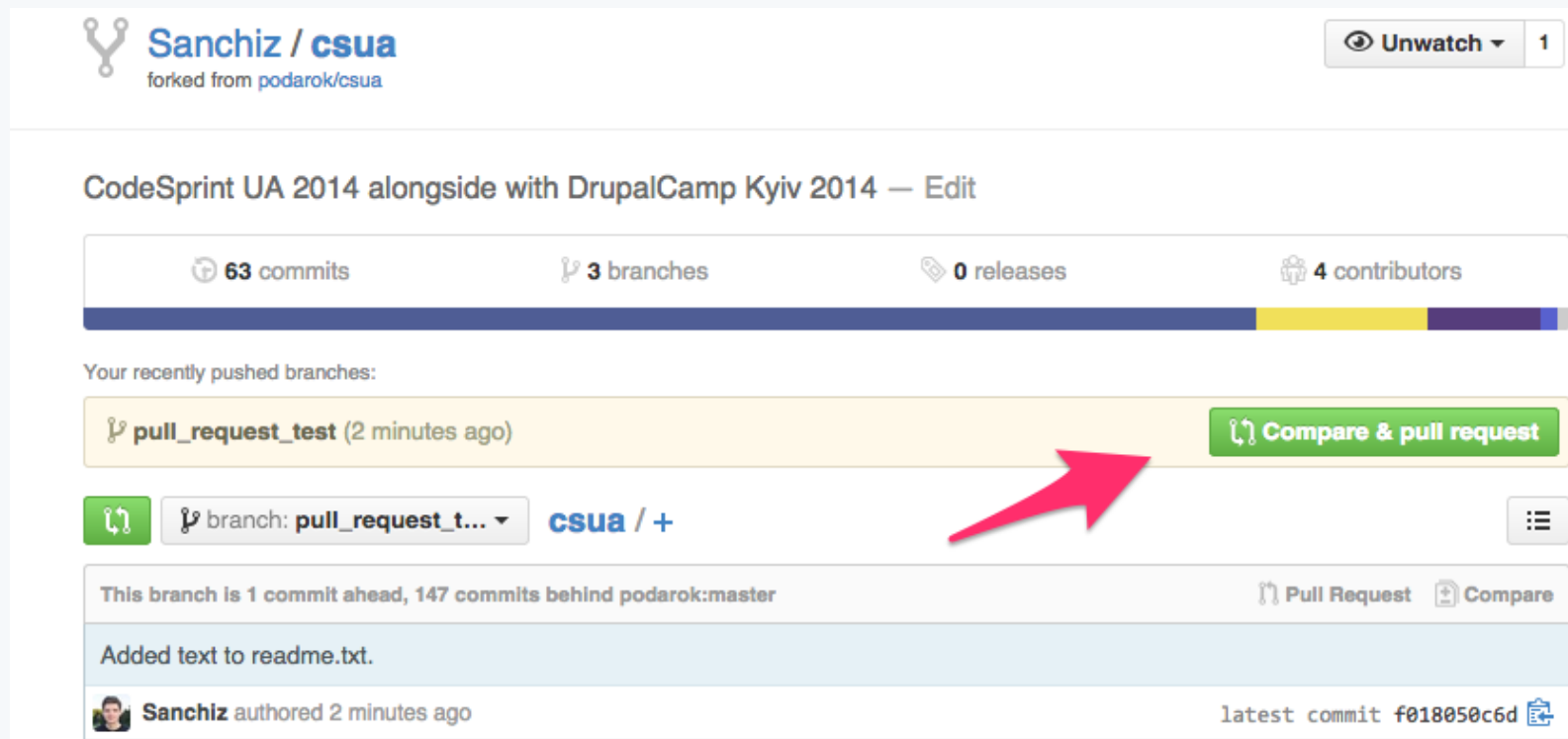
Comando responsável por baixar as atualizações de um repositório remoto para o repositório local.

```
$ git push
```

Comando responsável enviar as alterações locais para o repositório remoto.

Pull Request

É a solicitação para enviar modificações para repositórios de terceiros.



The screenshot shows a GitHub Pull Request page for the repository 'Sanchiz / csua', which is forked from 'podarok/csua'. The page title is 'CodeSprint UA 2014 alongside with DrupalCamp Kyiv 2014'. It displays repository statistics: 63 commits, 3 branches, 0 releases, and 4 contributors. A section titled 'Your recently pushed branches:' lists a branch named 'pull_request_test' pushed 2 minutes ago. A red arrow points to a green button labeled 'Compare & pull request' next to this branch. Below this, a dropdown menu shows the current branch as 'pull_request_t...' and the base branch as 'csua / +'. A status bar indicates 'This branch is 1 commit ahead, 147 commits behind podarok:master'. The commit message 'Added text to readme.txt.' is shown, along with the author 'Sanchiz' and the latest commit hash 'f018050c6d'.

REFERÊNCIAS

Gitbook:

<https://git-scm.com/book/pt-br/v1/Primeiros-passos>

GitHub Learning Lab:

<https://lab.github.com>

Curso de Git e Github para iniciante:

<https://www.udemy.com/git-e-github-para-iniciantes/>

Screencast – Git e github para iniciantes

<https://www.youtube.com/watch?v=UMhskLXJuq4>

PARA ESTUDOS



OBRIGADA