

Minicurso de Git

Julio Pinto Coelho

April 17, 2019

1 Introdução

Trabalhando em uma organização, é imprescindível que todos envolvidos em um determinado projeto tenham acesso à toda informação necessária sobre o mesmo. Isso muitas vezes inclui códigos, arquivos, documentos. Para possibilitar essa colaboração, utilizamos alguma ferramenta de controle de versão. As mais comuns são mercurial, subversion e git. Nesse artigo, veremos alguns comandos de git.

2 Comandos

2.1 git-init

O primeiro comando para se inicializar um repositório do tipo git. A sintaxe é simples:

```
git init
```

2.2 git-remote

Após inicializar o repositório git, precisamos vinculá-lo a um servidor remoto. Caso contrário, todo trabalho será apenas local.

```
git remote add origin url-to-remote-server/path-to-git-repo
```

2.3 git-clone

Caso o repositório já exista em algum servidor remoto, podemos, ao invés de git-init, usar o git-clone, da seguinte forma:

```
git clone https://url-to-git-server/repo.git
```

E, com isso, teremos já o repositório pronto

2.4 git-status

Este comando serve para aferir se houve modificações, ou qual é o estado do repositório local em relação ao remoto.

```
git status
```

A saída desse comando é algo semelhante a:

```
On branch master
```

```
Your branch is up to date with 'origin/master'.
```

```
Changes not staged for commit:
```

```
(use "git add <file>..." to update what will be committed)
```

```
(use "git checkout -- <file>..." to discard changes in working directory)
```

```
modified:   player/__pycache__/__init__.cpython-36.pyc
```

```
modified:   player/__pycache__/player.cpython-36.pyc
```

```
modified:   scenario/__pycache__/floor.cpython-36.pyc
```

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

Podemos ver que houve modificações, e elas não foram atualizadas no repositório remoto.

2.5 git-add

Antes de poder enviar as modificações para o servidor remoto, precisamos atualizar o repositório local com as novas modificações. Para isso, selecionamos quais arquivos serão atualizados, usando o seguinte comando:

```
git add path/to/file.extension
```

```
git add .
```

O primeiro comando adiciona um arquivo específico a ser atualizado, enquanto que o segundo adiciona todos os arquivos modificados a serem atualizados.

2.6 git-commit

Após selecionar os arquivos a serem atualizados, precisamos confirmar a atualização. Ao fazermos isso, definimos uma mensagem ou breve título a essa atualização, de forma a facilitar as futuras buscas.

```
git commit -m "Mensagem de atualização do repositório"
```

Com isso, o repositório local está atualizado. Nada mais é necessário, caso esteja trabalhando apenas com repositório local. Caso contrário, os próximos comandos são bastante importantes.

2.7 git-push

Após a atualização do repositório local, deve-se atualizar o remoto, de forma a possibilitar que as outras pessoas trabalhando no mesmo projeto possam ter acesso às novas modificações.

```
git push
git push origin master
git push branch-local branch-remoto
```

Quando se usa esse comando sem argumentos, o programa procurará configurações que indiquem quais branches utilizar. Caso nada seja encontrado, será utilizada a segunda chamada.

2.8 git-pull

As modificações enviadas ao servidor remoto não são automaticamente atualizadas nos repositórios locais individuais. Para tal, é necessário utilizar esse comando.

```
git pull
```

Será exibida uma série de informações na tela, e, ao final da execução, o repositório local estará sincronizado com o remoto.

3 Clientes GUI

Os comandos listados acima, apesar de não serem complexos, requerem algum treinamento por parte do usuário. Ao invés disso, pode-se usar ferramentas gráficas para gerenciar os repositórios. Alguns dos clientes mais comuns são:

- GitKraken
Software gratuito para uso pessoal, possui versão paga. Pode ser usada com vários serviços de controle de versão (GitHub, GitLab, BitBucket...)
- GitHub Desktop
Gratuito, vincula diretamente à conta do GitHub

4 Conclusões

Agora que eu dei uma aula, e escrevi esse artigo da forma mais profissional possível, vocês não têm mais o direito de não atualizarem os repositórios no github do TauraBots. Larguem de ser preguiçosos, e mostrem o que estão fazendo!

Enquanto isso, eu não atualizo nada lá pq tô fazendo meu TCC

5 Agradecimentos

Valeu!

Thanks!

Gracias!

Dankon!

Danke Schön!