

Usando o Git e GitHub - Tutorial

O que é o Git?

O Git é um sistema de controle de versão (ou versionamento), VCS (do inglês version control system) ou ainda SCM (do inglês source code management) distribuído que está sendo cada vez mais usado, pois foca em velocidade e robustez. O desenvolvedor inicial do Git foi Linus Torvalds (criador do Linux).

Um sistema de controle de versão tem a finalidade de gerenciar as diferentes versões de um documento. É muito utilizado por empresas de desenvolvimento de software que possuem grandes equipes e precisam manter organizado, documentado e arquivado tudo o que foi e está sendo implementado no software.

Como exemplo de projetos que utilizam o Git temos: Arch Linux, Android, Debian, Digg, Eclipse, Fedora, GNOME, jQuery, openSUSE, Ruby on Rails, Samba, Yahoo UI Library, entre outros.

site oficial: <http://git-scm.com/>

O que é o Github?

O Github é um serviço de hospedagem distribuído desenvolvido em Ruby on Rails para projetos que utilizam o controle de versão Git. Assim, é utilizado como repositório online de códigos fonte para projetos de código aberto.

Podemos encontrar nele informações sobre todos os commits (atualizações) dos projetos que o utilizam, uma rede social que possibilita que outras pessoas acompanhem o desenvolvimento de seu projeto, um recurso para visualizar gráficos de quantas atualizações cada pessoa no projeto está realizando, entre outras novidades.

site oficial: <http://github.com>

1- Instalando Git

- Podemos fazer o download do Git pelo site: <http://git-scm.com/download>, atualmente funciona em Linux, Mac OSX e Windows.
- Para usuários GNU/Linux Debian o download do Git pode ser feito pelo Gerenciador de Pacotes com o comando:

```
$ sudo apt-get install git-core git-gui git-doc
```

2- Criar um usuário no Github

- Acessar <http://github.com/> -> Pricing and Signup -> Escolher um plano e “Create an account” (Possui planos free) -> Escrever seus dados e seguir os passos de costume.

3- Criando as chaves públicas

- Acessar <http://github.com/>
- Fazer login
- Account Settings
- SSH Public Keys
- Add another public key
- No lado direito tem um Help: “Need help with public keys?” - Siga os passos lá citados
- Use o comando a seguir no Terminal para gerar a chave
`$ cat ~/.ssh/id_rsa.pub`
- Cole essa chave no campo Key e escolha um Title de sua preferência (Ex: Minha chave)
- Clique em Add Key e a chave será criada

Pode-se testar se a chave foi corretamente criada com o comando:

```
$ ssh git@github.com
```

4- Configurando o Git

- Configure seu usuário Git com os comandos
`$ git config --global user.name "Your Name"`
`$ git config --global user.email youremail@gmail.com`

5- Criando seu projeto e seu repositório

- Vá em <http://www.github.com> em sua Dashboard e procure por “New Repository” e preencha os campos.
- Agora você deve criar o diretório para o repositório criado em seu computador. Use os comandos:

```
$ mkdir ~/YourProjectName  
$ cd ~/YourProjectName  
$ git init  
$ touch README  
$ git add README  
$ git commit -m 'first commit'  
$ git remote add origin git@github.com:username/Hello-World.git  
$ git push origin master
```

5- Clonando um projeto do Github (com direitos de R/W no repositório)

- Criar uma pasta no seu diretório do usuário (Ex: MeuRepositorio)
- Entre nesse diretório pelo Terminal
- Usar o git clone para clonar um projeto da web (Github) para seu pc, exemplo (*1): git clone git@github.com:taboca/TelaSocial.git

6- Utilizando um projeto do Github

De dentro do diretório criado, você pode usar os comandos:

- `$ git pull`
Atualiza os arquivos do seu pc conforme o que está no github (sempre lembre de usá-lo se imaginar que alguém possa ter feito algum commit)

Fazendo commits:

- Adicione o arquivo modificado no diretório criado
- `$ git status`
Mostra quais arquivos sofreram modificações (modificados ou inseridos ou deletados)
- `$ git add .`
Se novos arquivos foram adicionados esse comando irá adicioná-los ao projeto, do diretório atual pra baixo.
- `$ git commit -m "Comments ... ex: Changing the background color of file1..."`
Esse comando irá fazer o commit no repositório do Github, com o comentário que está entre aspas.
- `$ git push origin master`
Após executar esse comando deve aparecer na página do github, do projeto em que está trabalhando, suas modificações/commits.

Desfazendo commits:

Em alguns momentos talvez seja necessário desfazer algum commit, por exemplo, se você fez um commit de uma coisa errada, que não era pra ter feito.

- `$ git log`
Esse comando irá mostrar o histórico de commits
Nesse histórico copie o código (um gigante) referente ao commit que quer desfazer
- `$ git revert codigoDoCommit`
Assim que executar esse comando abrirá um editor padrão para confirmar que quer reverter o commit e qual mensagem deseja que apareça nessa reversão (normalmente é só fechar o editor sem fazer modificações, e não precisa salvar o documento que abriu no editor)
Com isso o commit será revertido, aparecerá em seu terminal uma mensagem parecida com a de quando um commit é feito
- `$ git push origin master`
Esse comando fará com que o revert apareça no histórico de commits no Github.

(*1) - Para que isso funcione o dono do projeto que desejamos clonar deve ter dado permissão para seu usuário no Github.

Se ele tiver dado permissão e mesmo assim estiver ocorrendo algum problema, tente usar isso (use case):

- `ssh-add _arquivo_da_key_mas_o_private_nao_o_public`

Referências:

Git community book: <http://book.git-scm.com/index.html>

Git Official website: <http://git-scm.com/>

Git na Wikipedia: <http://pt.wikipedia.org/wiki/Git>

Github na Wikipedia: <http://pt.wikipedia.org/wiki/GitHub>

Github website: <http://github.com>



Esta obra de [Rafael Sartori](#), foi licenciada com uma Licença [Creative Commons - Atribuição - Partilha nos Mesmos Termos 3.0 Não Adaptada](#).