



Engenharia de Software

Introdução ao GIT e GitHub



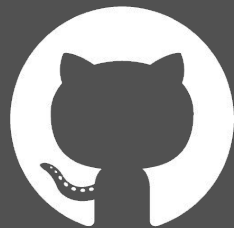
UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO

Monitoria 02/04/2019
Antônio Adelino
Armstrong Lohãns
Gabrielle Porto



git

≠



GitHub

GIT



O que é?

- Git é uma ferramenta de controle de versão.
- Facilita o desenvolvimento colaborativo de software.
- Permite que um mesmo arquivo seja editado ao mesmo tempo por pessoas diferentes, sem sobrescrevê-lo.
- Maior facilidade ao fazer deploy (disponibilizar um sistema para uso).

GIT



O que é?

- Registra alterações em arquivos ao longo do tempo.
- Permite voltar a uma versão antiga dos arquivos.
- Registra quem fez alterações nos arquivos.
- Esse é o papel mais básico de uma ferramenta de controle de versão: **gerenciar as alterações** feita em um arquivo e **guardar um histórico** delas, caso você queira ver como era antes, entender porque algo mudou ou até mesmo **voltar para uma versão anterior**.

GIT



O que é?

- Existem várias outras ferramentas de controle de versão além do Git: Subversion, TFS e Mercurial são algumas delas.
- O Git foi criado pelo Linus Torvalds em 2005 para desenvolvimento do Kernel do Linux e acabou se popularizando no mundo do desenvolvimento de software.
- O Git não depende do Github para ser utilizado. Na verdade, você pode usar ele até sem ter internet.

GitHub



O que é?

- Github é um site onde você pode subir seus projetos e compartilhar com o mundo. E você vai subir eles lá usando o Git.
- A grande maioria dos projetos open source estão hospedados no Github, inclusive o [código do Git](#) está lá no Github também.
- Você pode usar o Github como portfólio do seu código e outras pessoas podem ver seus projetos e baixá-los.

GitHub



O que é?

- Dá pra participar de projetos open source e seguir outros desenvolvedores conhecidos por lá pra ver o que eles estão fazendo.
- É tipo uma rede social para pessoas que programam :)
- Existem outros sites onde você pode subir seus projetos com Git, como o [Gitlab](#) e o [Bitbucket](#).



Conceitos GIT e GitHub

Conceitos GIT



- **Commit:** registra um conjunto de alterações feitas pelo desenvolvedor, cada commit possui uma hash única associada ao mesmo.
- **Branch:** é uma linha independente de desenvolvimento criada a partir do branch principal, o master.

Conceitos GIT



Estágios do GIT

- **Modified:** arquivos alterados, mas ainda não foram commitados;
- **Staged:** arquivos modificados que foram assinalados para o próximo commit que você fizer;
- **Committed:** dados armazenados corretamente no repositório local.

Conceitos GIT



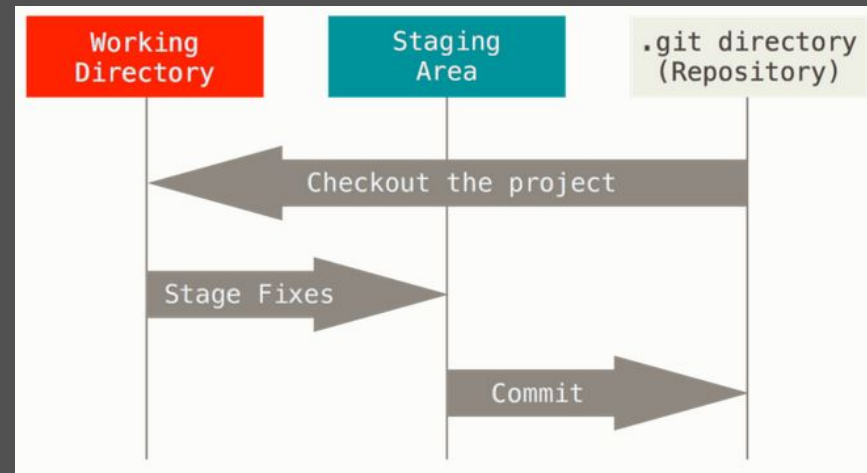
- **Conflito de merge:** duas pessoas alteraram a mesma linha do mesmo arquivo, qual alteração prevalece?
- Ocorre quando o repositório local está desatualizado;
- Necessita de intervenção manual do desenvolvedor.

```
<<<<<< HEAD: nomeArquivo
if (a > b) { ... }
=====
if (b > a) { ... }
>>>>>> hash do commit
```

Conceitos GIT



- Altera os arquivos no **Working Directory**;
- **Staging Area** é o arquivo que armazena informações sobre o que vai ser incluído no próximo commit;
- **.git directory** registra as alterações permanentemente após o commit.



Conceitos GitHub



- **Fork** – ao clicar em fork no repositório de alguém, ele pega o repositório dessa pessoa e copia para a sua conta. Lá, você pode editar esses arquivos e depois devolver a essa pessoa com as suas edições.
- **Pull request** – envia suas contribuições para o repositório principal, onde vai ficar pendente até a aceitação.



Trabalhando com o GIT e GitHub

Trabalhando com o GIT e GitHub



- Instalar o **git**:
 - Linux: `sudo apt get install git`
 - Windows: <https://git-scm.com/download/win>
- Criar conta no **github**: <https://github.com/join>
- Configurar o **git**:
 - `git config --global user.name "nomeDoUsuario"`
 - `git config --global user.email emaildousuario@gmail.com`

Trabalhando com o GIT e GitHub



- Realizar o fork de um repositório e cloná-lo, ou inicializar um novo repositório git;
- Criar um novo branch;
- Realizar alterações necessárias;
- Adicionar os arquivos alterados na staging area;
- Realizar commit;

Trabalhando com o GIT e GitHub



- Realizar push;
- Realizar pull request;
- Os commits agrupam arquivos associados;
- Um pull request pode possuir mais de um commit.



Comandos GIT

Comandos GIT



- **git clone url** – copia todos os arquivos de um repositório para a máquina local;
- **git init** – inicializa um repositório git;
- **git status** – apresenta os arquivos que foram modificados;
- **git add nomeArquivo** – adiciona o arquivo ao staging area;

Comandos GIT



- **git commit -m “mensagem do commit”** – faz o commit dos arquivos que foram colocados na staging area;
- **git log** – mostra o histórico de commits;
- **git reset HEAD nomeArquivo** – remove o arquivo da staging area;
- **git remote -v** – apresenta os “remotes” adicionados;
- **git remote add nomeRemote urlRemote** – adiciona um novo remote;

Comandos GIT



- **git fetch nomeRemote** – compara o seu remote (origin) com o remote passado;
- **git push nomeRemote nomeBranch** – envia as suas alterações para um remote;
- **git checkout -b nomeBranch** – cria um novo branch;
- **git checkout nomeBranch** – altera para um branch já criado;
- **git branch** – mostra todos os branches existentes;

Comandos GIT



- **git merge nomeBranch** – realiza o merge. Em alguns casos, também é possível realizar o merge no Github;
- **git pull nomeRemote nomeBranch** – atualiza o repositório local;

Links



- Noções básicas GIT:
<https://git-scm.com/book/pt-br/v1/Primeiros-passos-Noções-Básicas-de-Git>
- Configuração inicial GIT:
<https://git-scm.com/book/pt-br/v1/Primeiros-passos-Configuração-Inicial-do-Git>
- GIT e GitHub:
<https://tableless.com.br/tudo-que-voce-queria-saber-sobre-git-e-github-mas-tinha-vergonha-de-perguntar/>