escola britânica de artes criativas & tecnologia

Desenvolvedor Full Stack Python

Git e GitHub

escola britânica de artes criativa & tecnologia

AGENDA

- Instalando Git
- Ferramentas para controle de versão
- Gerenciando mudanças com git branches e tags
- Git na prática

Instalando Git

Windows https://gitforwindows.org/

Linux (Normalmente já vem instalado) ou sudo dnf install git-all

Mac (Normalmente já vem instalado) ou https://git-scm.com/download/mac

Quais controles de versões temos no mercado?

Controle de versão com suporte a Git com hospedagem de código:

- Github
- Gitlab
- Bitbucket

Criando uma conta no Github

Acesse: https://github.com/join?source=login

Github SSH

- Usada para aumentar a segurança de quem está usando um determinado repositório.
- Utilizado por diversas empresas.

Gerando uma chave SSH

1- Abra seu terminal

2 - Digite com ssh-keygen -t ed25519 -C “your_email@example.com” e substua com seu email

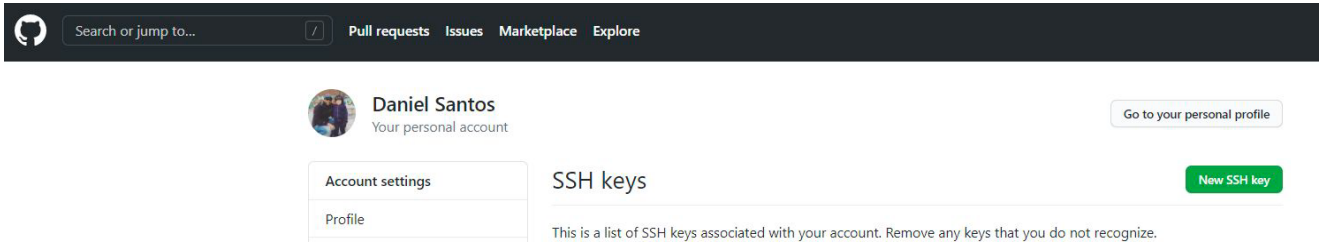
3 - De um enter caso queira manter o nome padrão do arquivo com a chave SSH

4 - Caso não queira dar um outro nome para o passphrase aperte Enter

Adicionando o SSH-key no Github

1 - Abra o terminal e digite **eval "$(ssh-agent -s)"**

2 - Adicione sua chave na sua conta do Github



Criando nosso primeiro repositório

Vamos criar nosso primeiro repositório.

Depois de criado vamos clonar nosso repositório usando o comando **clone:**

git clone git@github.com:<nome-do-user>/<nome-do-repo>.git

Fazendo nosso primeiro commit

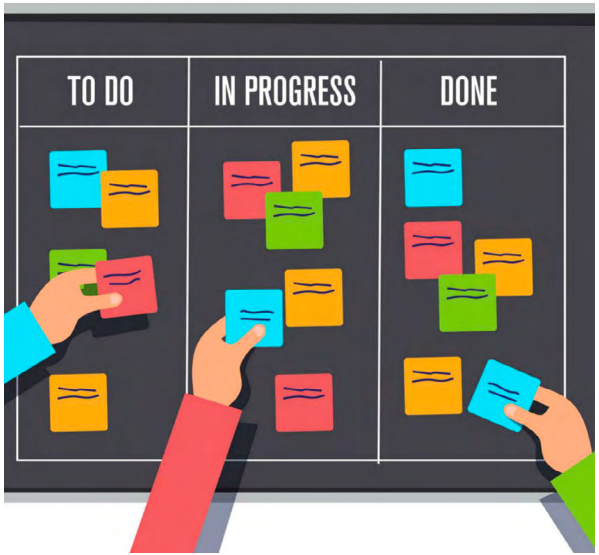
Vamos adicionar um arquivo de texto, commitar e enviar essa alteração para nosso repositório:

1. git add . (para adicionar todos os arquivos que foram alterados)
2. git commit -m “Nome da alteração”
3. git status (para visualizar o que será alterado)
4. git push origin HEAD

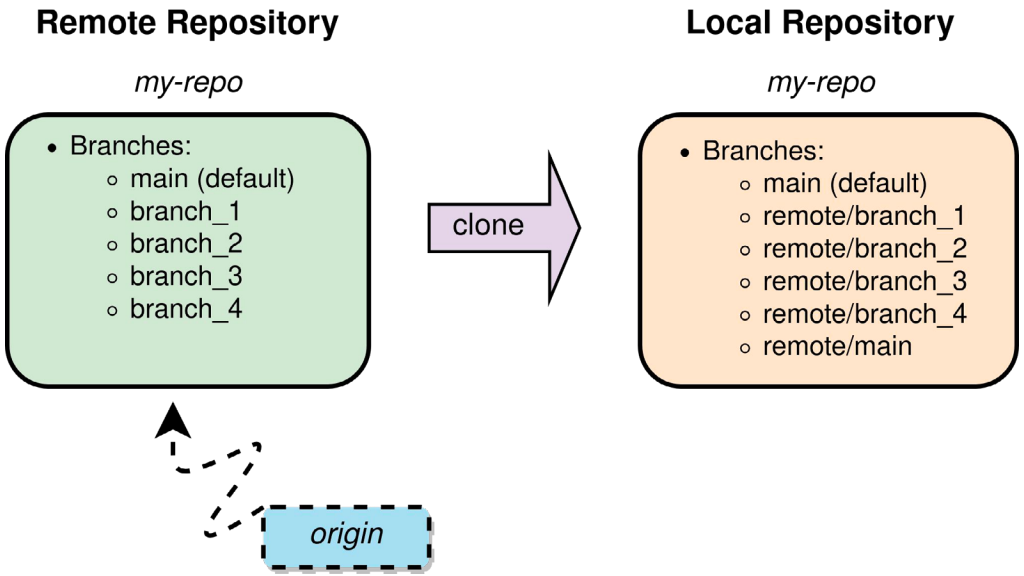
Gitflow

Sincronizar o trabalho entre equipes requer organização e controle de mudanças, veremos agora o que o **Git** oferece para suportar o trabalho em conjunto e em paralelo em um mesmo projeto.

Branches e Kanban



Local... Remote

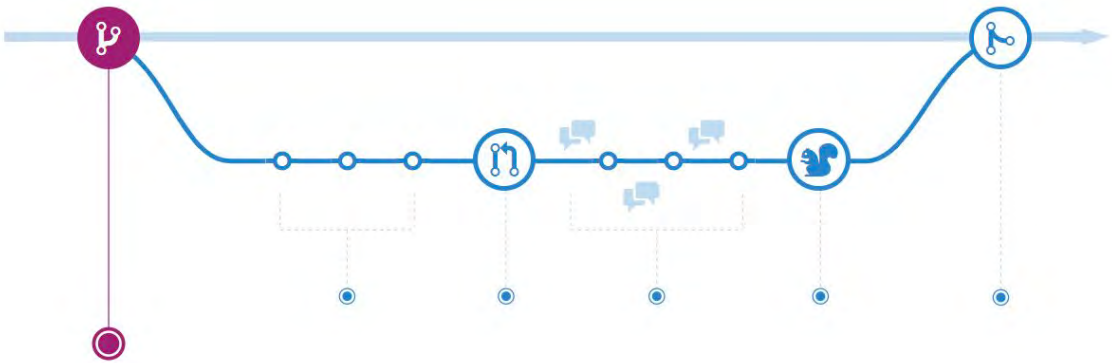


Criando uma branch

Trabalhando em times, normalmente trabalhamos em alguma mudança no sistema, seja a correção de um bug, uma nova feature de produto e normalmente quando fazemos isso, utilizamos **branches.**

Para criar uma branch use o comando:

git branch -b “nome-da-branch”

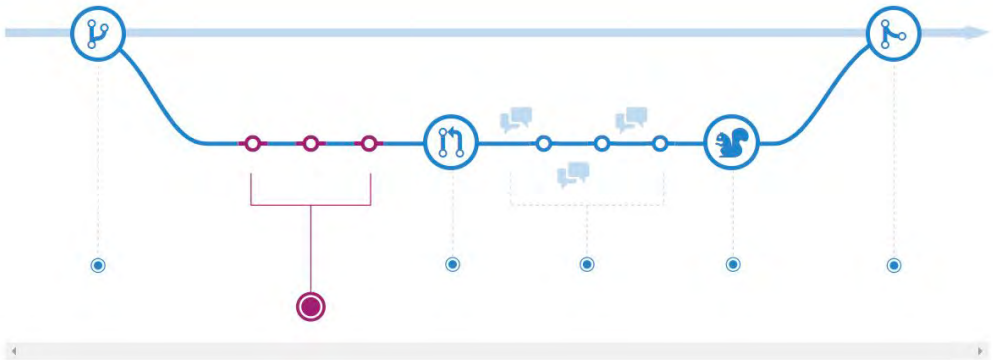


Adicionando commits

Depois que criamos nossa branch, chegou a hora de trabalharmos e adicionarmos arquivos que serão modificados.

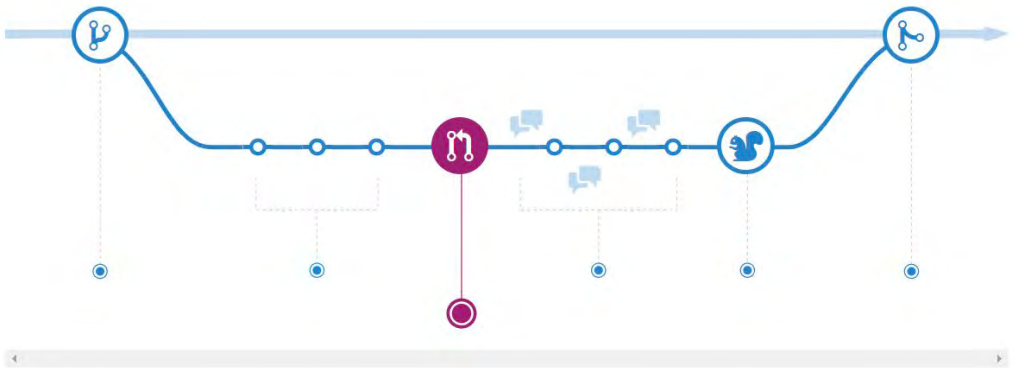
Para isso utilize o comando:

git add <nome-do-arquivo>



Abrindo um Pull Request

O último passo para nosso fluxo desenvolvimento é a abertura de um **pull request.** O pull request é onde iniciamos as discussões de sobre o objetivo da mudança e normalmente adicionamos membros do time para revisar o PR e uma vez aprovado seguimos para o merge e deploy da mudança feita.



Abrindo um Pull Request

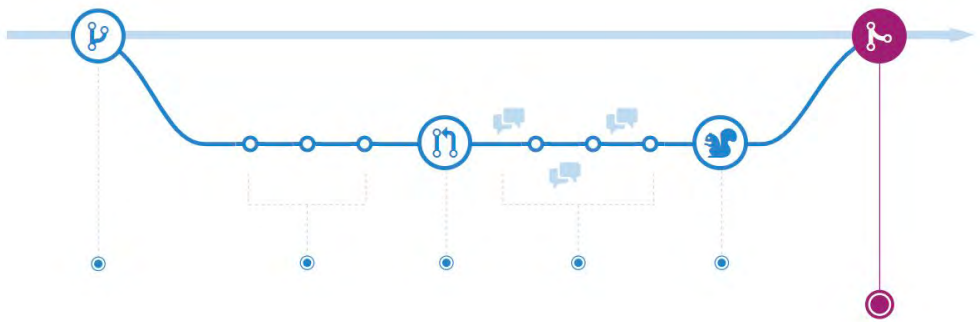
Para criar um PR, temos que seguir os passos a seguir

1. Enviamos nossas alterações locais para nossa branch remota com o comando: git push origin head

2. Dentro do git abrimos um PR e adicionamos as pessoas que vão revisar as alterações propostas.

Merge

Após a revisão e aprovação dos revisores do pull request, normalmente fazemos o **merge** da branch para a main ou master branch.



Fluxo de Desenvolvimento

Não existe regra ou receita para um modelo ideal de ciclo de desenvolvimento, **cada empresa/equipe trabalha em algum modelo específico.**

Tags

Tags normalmente são geradas durante um deploy e seu principal objetivo serve para **marcarmos** o que foi para produção e além disso tornar a mudança entre versões em produção de forma mais fácil.

Boas Práticas no Git

- Commits com verbos e com descrição objetiva:

  exemplo: git commit –m “add title e h1”

- Commits com poucos arquivos (se possível).
- Normalmente adicionamos o nome ou link da historia que estamos trabalhando.

Resumo

- Aprendemos o que é o Git
- Git branches
- Tags
- Boas práticas

Exercício

escola britânica de artes criativa & tecnologia

Nesse exercício vamos armazenar o projeto que trabalhamos no módulo de Python avançado no Github e para isso vamos criar um repositório, seguido de uma branch e por fim um pull request.

Você pode compartilhar o link do PR na plataforma da EBAC.