

Semana 1 - Aula 1

Git

Labenu_



Sumário

Labenu_



O que vamos ver hoje?

- Vamos aprender como utilizar o **git** para facilitar o gerenciamento dos nossos códigos
- Enteremos a diferença entre git e **github**
- Tudo isso através de **poemas**



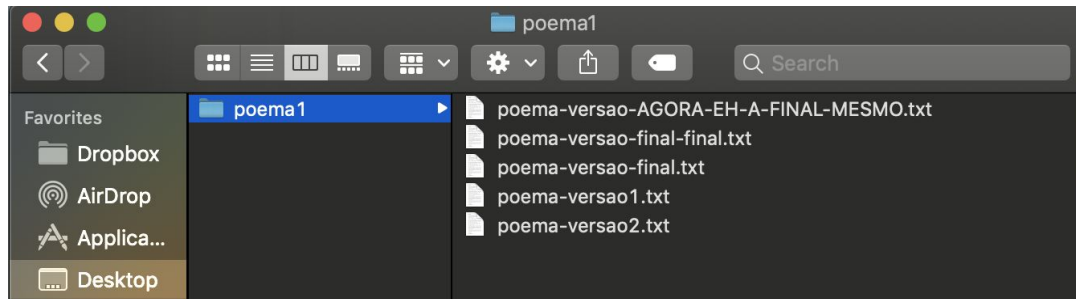
Motivação

Labenu_



Trabalhos e mais trabalhos

- Todos nós já tivemos que fazer diversos trabalhos de escola
- Antes do surgimento de plataformas Cloud (como Google Drive), tínhamos o costume de fazer assim



Trabalhos e mais trabalhos

- Para que guardamos versões dos nossos trabalhos?
 - Não perder **ideias antigas**
 - Poder **voltar atrás** em alguma decisão
 - Acompanhar **a evolução** do que estamos fazendo
- O **git** é uma ferramenta que permite fazermos o **gerenciamento de versão** de nossos projetos (de programação ou não)



Trabalhos e mais trabalhos

- Como vocês faziam projetos em grupo **antes das ferramentas Cloud**?
 - Mandando os arquivos separados em um e-mail
 - Tomando cuidado para não escrever onde seu colega está escrevendo
 - E, depois de tudo isso, alguém sozinho pegava o trabalho e formatava do jeito que tinha que ser



Trabalhos e mais trabalhos

- O **git** também facilita o trabalho **colaborativo**
- É muito fácil manter o **track** (rastreamento) de arquivos que são alterados por duas pessoas ao mesmo tempo



Um pouco de história

Labenu_



Um pouco de história

- Este problema de versionamento é algo que já preocupava a comunidade científica (em especial os devs) há bastante tempo
- Um dos primeiros sistemas a surgir foi o **bitkeeper**



Um pouco de história

- Em 1991, **Linus Torvalds** começou a elaborar o sistema operacional **Linux**
- A princípio, ele só **queria testar** seus conhecimentos de programação e criar o seu **próprio sistema operacional** que só funcionaria no seu próprio computador
- Queria fazer algo **simples**



Um pouco de história

Estou fazendo um sistema operacional (livre - apenas como um hobby, não será algo grande e profissional como o GNU) [...]

Eu gostaria de opiniões sobre coisas que as pessoas gostam/não gostam [...]

Mensagem enviada por Linus para divulgar seu projeto

La
Be
Nu_

Um pouco de história

- Com o tempo, o projeto foi crescendo e se tornando cada vez mais importante
- Por ser um projeto **open-source**, qualquer pessoa poderia **contribuir**
 - escrevendo código
 - sugerindo funcionalidades e melhorias



Um pouco de história

- O **bitkeeper** começou a **não** ser mais o **suficiente**
 - Era bastante lento
 - Virou pago
- Com isso, Linus e sua equipe decidiram criar o próprio **version control software** (VCS - software de controle de versão), o **git**



Git vs GitHub

Labenu_





- O **git** é a **ferramenta** que facilita o gerenciamento e colaboratividade dos projetos
- O **Github** é um **serviço cloud** que permite armazenar os projetos





- O projeto que está na nossa máquina chamados de **repositório** (ou repo) **do git local**
- O projeto que está no github, chamados de **repositório do git remoto**



Pausa para relaxar 🧘

- **PONTOS IMPORTANTES:**

- O **git** surgiu como uma **ferramenta** que propõe facilitar o **versionamento** e a **colaboração** em qualquer tipo de projeto
- **Github** é a **plataforma** que guarda os repositórios na **nuvem**



Comandos I

Começando o repositório

Labenu_



Começando o repositório

- Vamos começar **criando um repositório** no nosso **GitHub**

Overview

Repositories 38

Packages 0

Stars 2

Followers 3

Following 1

Gists 34

Find a repository...

Type: All ▼

Language: All ▼

 New



Começando o repositório

- Vamos começar **criando um repositório** no nosso **GitHub**

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository](#).

Owner

 joaogolias ▾

Repository name *

Great repository names are short and memorable. Need inspiration? How about [scaling-octo-doodle](#)?

Description (optional)

☒  **Public**

Anyone can see this repository. You choose who can commit.

☐  **Private**

You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☐ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▾

Add a license: **None** ▾ ⓘ

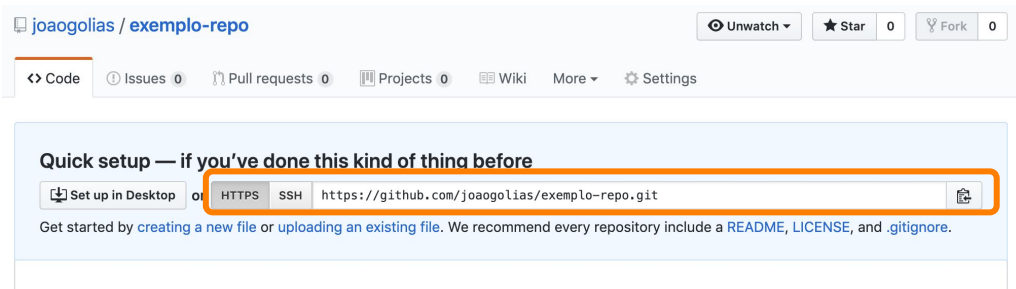
Create repository



Começando o repositório

- **git clone link-do-repo**

- É o comando que clona as informações do repositório remoto em uma pasta (repositório) na nossa máquina



Comandos II

Salvando localmente

Labenu_



Salvando localmente

- **git status**
 - Indica o status do repositório
 - Arquivos/pastas criados
 - Arquivos/pastas modificados
 - Arquivos/pastas removidos



Salvando localmente

- `git add nome-do-arquivo`



Salvando localmente

- **git add nome-do-arquivo**
 - Envia os arquivos modificados, removidos e criados para a Staging Area (que é local)
 - Podemos utilizar a opção --all para adicionar todos os arquivos
 - **Cuidado!** Tenha certeza que quer mesmo adicionar tudo



Salvando localmente

- **git commit -m "mensagem"**
 - Demarca uma versão do seu projeto com os arquivos que estiverem na Staging Area
 - A mensagem deve explicar as modificações, criações e deleções feitas



Salvando localmente

- **git commit -m "mensagem"**
 - Importante:
 - Não esquecer do -m
 - **Caso esqueça**, você vai entrar em uma parte do terminal, que, para sair, você deve digitar: **esc esc -q**
 - Não esquecer das aspas (")



Salvando localmente

- **git log**
 - Permite verificar o histórico de commits do projeto



Comandos III

Dividindo o trabalho

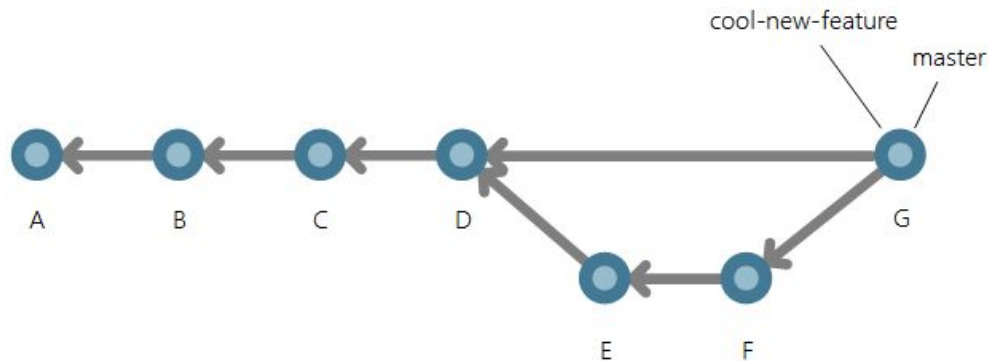
Labenu_



Dividindo o trabalho

- **git branch**

- Branch (ramo/galho) é uma ramificação do projeto principal



Dividindo o trabalho

- **git branch**

- Este comando em si mostra a lista de branches que estão no seu repositório local
- A branch padrão se chama master



Dividindo o trabalho

- **git branch nome-da-branch**
 - Permite criar uma nova branch, com o nome que você escolheu



Dividindo o trabalho

- **git checkout nome-da-branch**
 - Permite acessar uma branch que já foi criada (localmente ou no remoto)



Dividindo o trabalho

- **git checkout -b nome-da-branch**
 - É uma junção dos comandos anteriores
 - Ele cria uma nova branch e já acessa diretamente

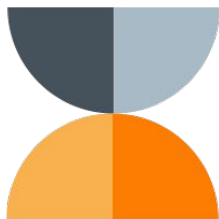


5 min

Pausa para relaxar 🧘

- **PONTOS IMPORTANTES:**

- git clone
- git status
- git add nome-do-arquivo
- git commit -m "mensagem"
- git log
- git branch
- git branch nome-da-branch
- git checkout
- git checkout -b nome-da-branch



Comandos IV

Salvando no remoto

Labenu_



Salvando no remoto

- **git push origin nome-da-branch**
 - Envia as suas alterações feitas para a branch no repositório remoto
 - Ele só envia as alterações que foram commitadas



Salvando no remoto

- **git pull origin nome-da-branch**
 - Atualiza a branch em questão no seu repositório local com as alterações commitadas na branch remota
 - Se você já estiver acessando a branch que deseja atualizar, o comando pode ser reduzido a git pull



PR

Labenu_





- **Pull Requests**

- Depois de fazer todas as alterações na sua branch, você deve querer que elas sejam mescladas com a branch principal (a master)
- A esta mesclagem, damos o nome de merge








- **Pull Requests**

- Para fazer um merge no GitHub, nós devemos criar um Pull Request (ou PR) antes


Compare changes

Compare changes across branches, commits, tags, and more below. If you need to, you can also [compare across forks](#).

 base: master ▾  compare: master ▾

 Create pull request

Choose different branches or forks above to discuss and review changes.







- **Pull Requests**

- Quando trabalhamos em equipe, os membros dela avaliam os nossos PRs
 - Pedindo correções no código
 - Sugerindo alterações
- Após o processo de **Code Review** (CR); e o seu código estiver **aprovado**, ele pode ser **mergeado** na master



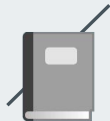
Resumo



- O git é uma ferramenta que ajuda muito o dia a dia de desenvolvedores, porque:
 - Permite gerenciar facilmente várias versões do código
 - Facilita o trabalho colaborativo em equipes
- GitHub é um sistema cloud que permite que guardemos os nossos repositórios remotos



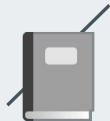
Resumo



- **Staging area:**



Resumo



- Comandos básicos
 - Começando o repositório
 - `git clone link-do-repo`
 - Salvando localmente
 - `git status`
 - `git add nome-do-arquivo`
 - `git commit -m "mensagem"`
 - `git log`



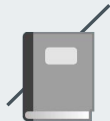
Resumo



- Sempre queremos que as alterações de uma branch nossa sejam mescladas com as informações que já estão na master (merge)
 - Para isso , devemos criar um PR
 - Solicitando aos nossos colegas de trabalho que avaliem o nosso código, dando sugestões de melhoria



Resumo



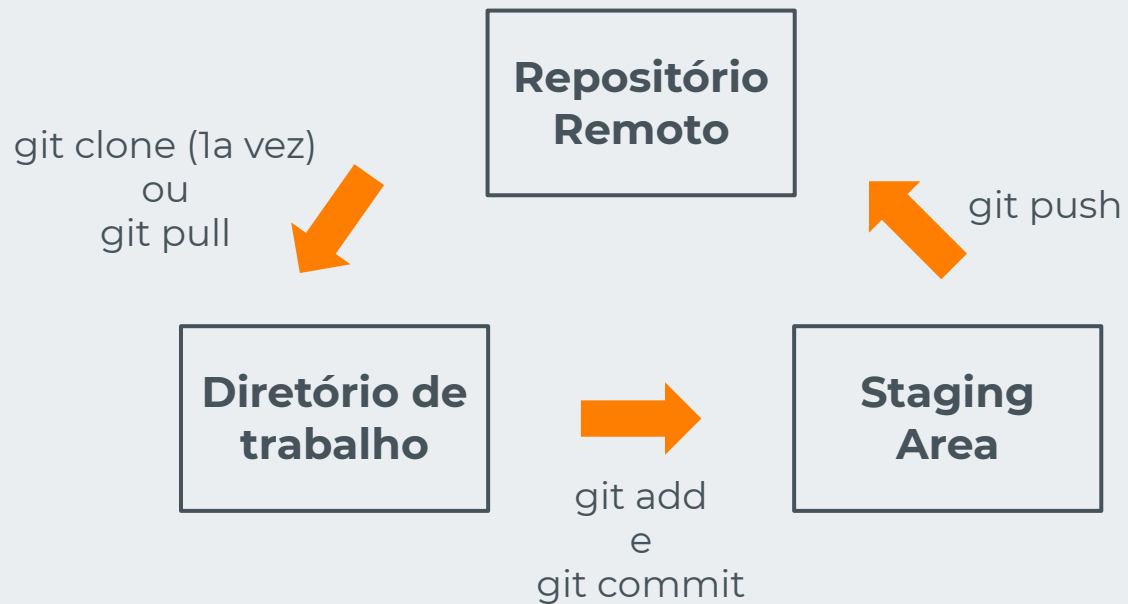
- Importante: comandos de git **não são** o mesmo que comandos do terminal!

Ex: git mkdir ❌

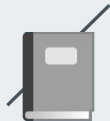
- Importante 2: **branch não é pasta!**



Resumo



Resumo



In case of fire



1. git commit
2. git push
3. leave building



Dúvidas?

Labenu_





Obrigado!