

Téc em Desenvolvimento  
de Sistemas Bilíngue

# Git e Github

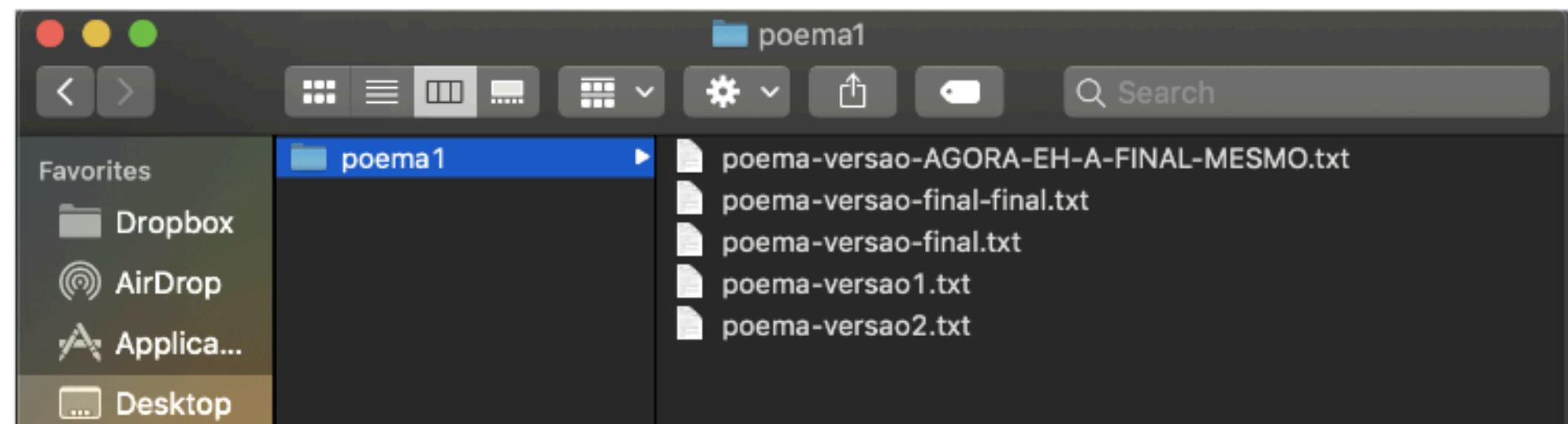
# Git e Github

- Gerenciamento de código com Git
- Diferença entre Git e Github
- Comandos do Git

# Git e Github

## Trabalhos e mais trabalhos

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



# Git e Github

- Como fazíamos projetos em grupo?
  - Mandando os arquivos **separados** em um e-mail
  - E depois, alguém **sozinho** pegava o trabalho e formatava do jeito que tinha que ser

# Git e Github

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

# Git e Github

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

Téc em Desenvolvimento  
de Sistemas Bilíngue

Um pouco de história

# Git e Github

- Este problema de versionamento é algo que já preocupava a comunidade científica (em especial, as pessoas desenvolvedoras) há bastante tempo
- Um dos primeiros sistemas a surgir foi o **bitkeeper**
- Mas o bitkeeper não era bom, e isso ficou evidente quando um dev começou a fazer um dos projetos open-source mais famosos da história...

# Git e Github

- 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**
- Segundo Linus, seria "algo simples"

## Git e Github

- O resultado é que o Linux se tornou o SO mais usado por pessoas desenvolvedoras no mundo
- 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 ou sugerindo funcionalidades e melhorias

# Git e Github

- O **bitkeeper** começou a **não** ser mais o **suficiente**:
  - Ele era bastante lento e passou a ser pago
- Com isso, Linus e sua equipe decidiram criar o próprio **version control software** (VCS - software de controle de versão)
- Surge daí o "**git**", o software de controle de versão que ninguém do mundo da tecnologia imagina viver sem...

Téc em Desenvolvimento  
de Sistemas Bilíngue

# Git VS Github

# Git e Github

## Git vs. Github

- O **git** é a **ferramenta** que gerencia as versões e colaborações em projetos
- O **Github** é um **serviço cloud** que permite armazenar os projetos
  - Existem outros, como Bitbucket e Gitlab. Todos usam a mesma ferramenta, o **git**.

# Git e Github

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

Téc em Desenvolvimento  
de Sistemas Bilíngue

# Começando o repositório

# Git e Github

## Começando o repositório 🎁

- Vamos começar **criando um repositório** no Github

The image shows two side-by-side screenshots. On the left is a screenshot of a GitHub user profile ('pedro-severo+') showing a list of repositories and recent activity. A red arrow points from the 'New' button in the repositories list to a tooltip on the right that says 'You've been invited to join an organization'. On the right is a screenshot of the 'Create a new repository' form. It includes fields for 'Owner' (set to 'joaogolias'), 'Repository name' (empty), 'Description (optional)' (empty), and a choice between 'Public' (selected) and 'Private'. At the bottom are buttons for 'Add .gitignore: None' and 'Add a license: None'.

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.

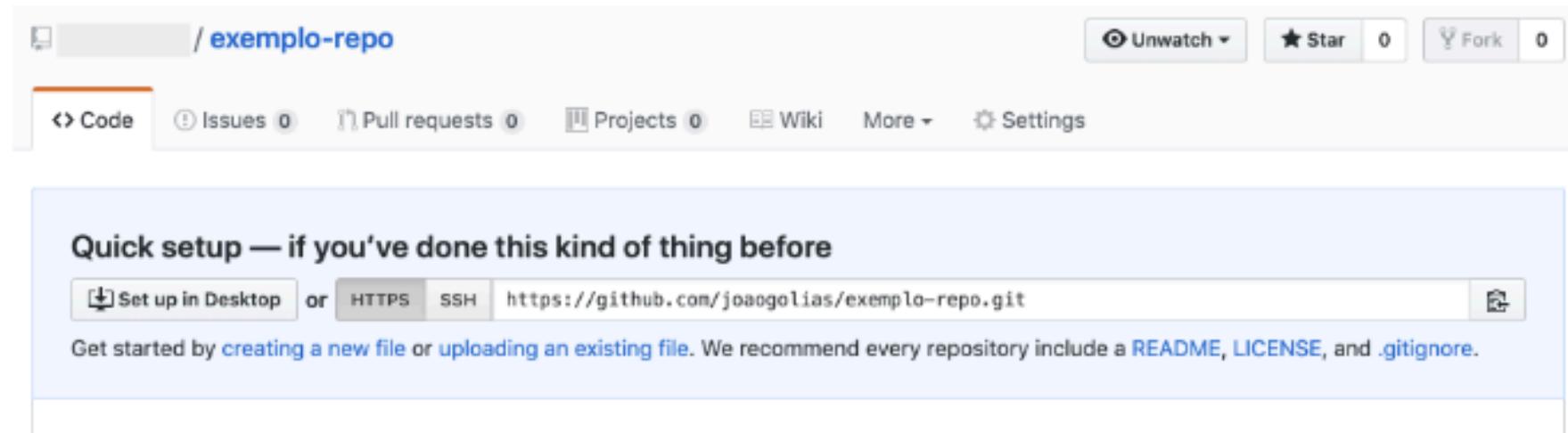
Initialize this repository with a README This will let you immediately clone the repository to your computer.

Add .gitignore: None Add a license: None

Create repository

# Git e Github

- **git clone link-do-repo**
  - É o comando que clona as informações do repositório remoto em uma pasta (repositório) na nossa máquina



# Git e Github

## Fixação

- 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**

Téc em Desenvolvimento  
de Sistemas Bilíngue

Salvando localmente

## Git e Github

### Salvando Localmente

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

# Git e Github

- **git add nome-do-arquivo**
  - Envia os arquivos modificados, removidos e criados para a Staging Area (que é local)
  - Também podemos utilizar a opção **git add --all** para adicionar todos os arquivos do repositório;
  - Ou a opção **git add .** para adicionar todos os arquivos da pasta onde você se encontra;

# Git e Github

- **git add .**



# Git e Github

- **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

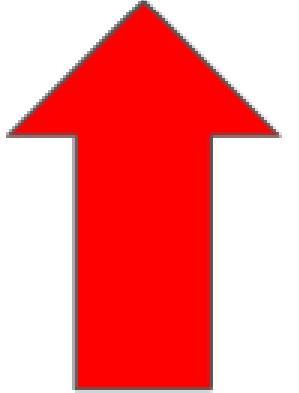
# Git e Github

- 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 ("")

# Git e Github

- REPETINDO PQ É MTO IMPORTANTE:

- Não esquecer do -m



# Git e Github

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

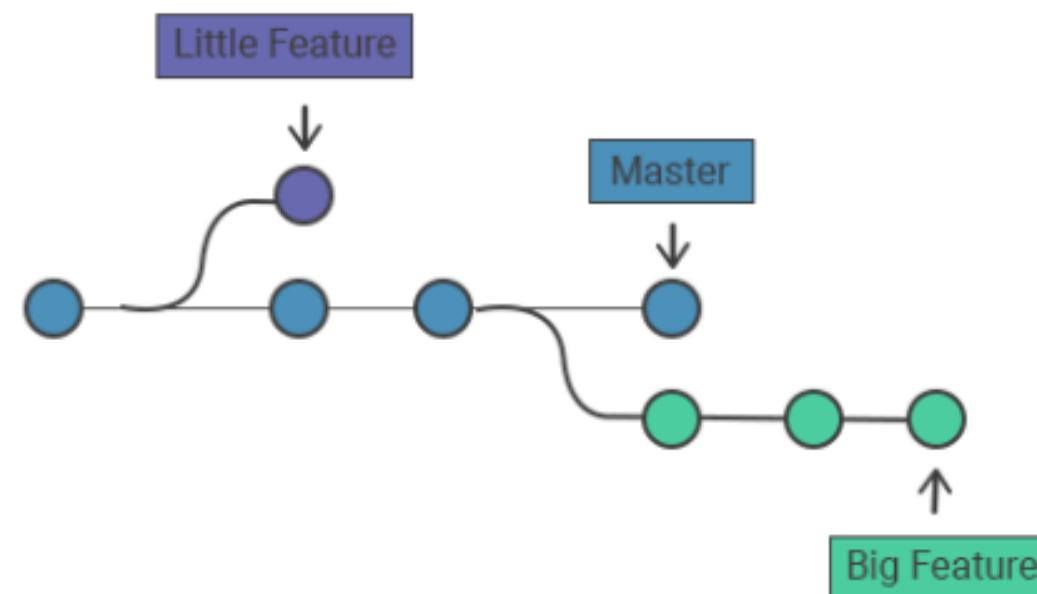
Téc em Desenvolvimento  
de Sistemas Bilíngue

# Dividindo o trabalho

# Git e Github

- **git branch**

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



# Git e Github

- **git branch**

- Este comando em si mostra a lista de branches que estão no seu repositório local
- A branch padrão se chama main\* e, a princípio, apenas ela vai existir no seu repositório

\* Anteriormente a branch padrão se chamava master, hoje em dia apenas repositórios antigos permanecem com esse nome.

# Git e Github

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

# Git e Github

- **git checkout nome-da-branch**
  - Permite acessar uma branch que já foi criada (localmente ou remota)

# Git e Github

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

# Git e Github

## Fixação

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

Téc em Desenvolvimento  
de Sistemas Bilíngue

Salvando remotamente

# Git e Github

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

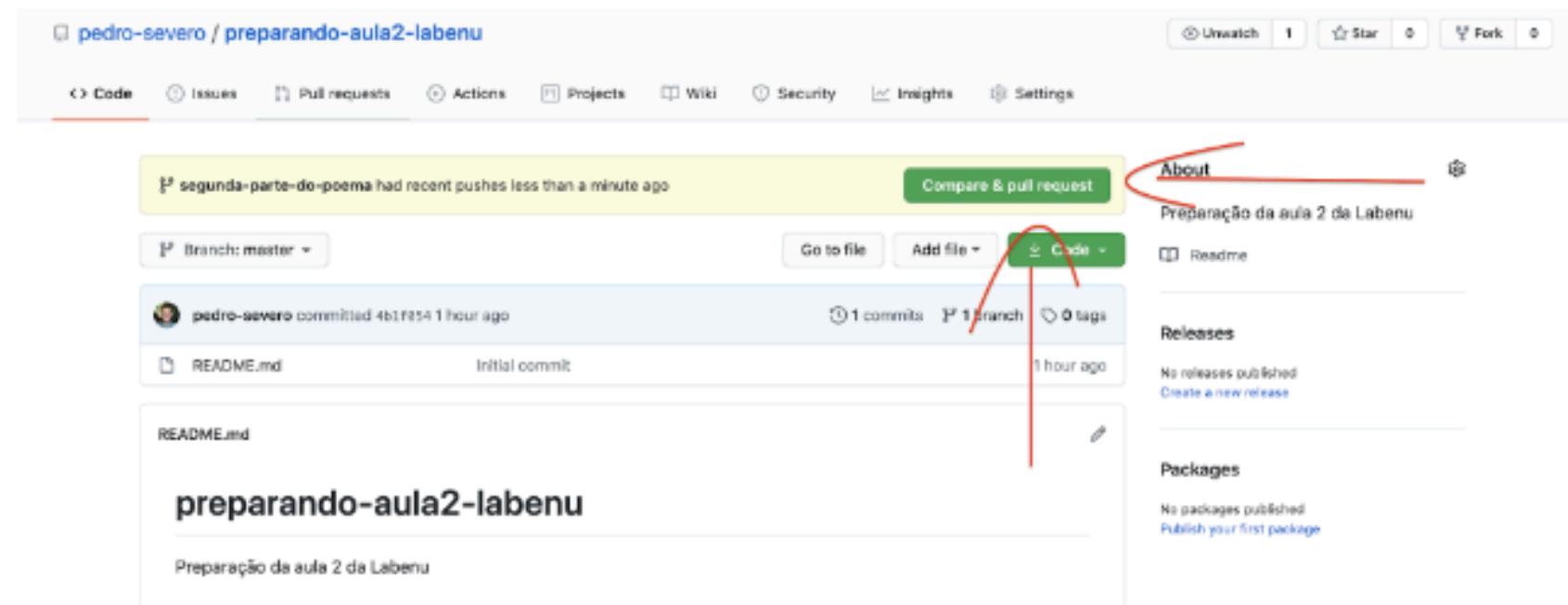
# Git e Github

## Pull Request (PR)

- 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**

# Git e Github

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



# Git e Github

- 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 main

Téc em Desenvolvimento  
de Sistemas Bilíngue

Atualizando localmente

# Git e Github

- **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