

Semana 1 - Aula 1

Git

FUTURE4

Big Picture e Sumário

Por que aprender Git?

Situação: escrevendo um trabalho

trabalho_da_escola.docx

trabalho_versao_final.docx

trabalho_final_final.docx

trabalho_final_AGORA_VAI.docx

trabalho_final_AGORA_VAI2.docx

Agora imagina um projeto de código...

- Alterações diárias
- Um errinho pode fazer tudo parar de funcionar
- Mais de uma pessoa alterando

Por isso foi criado o Git

- Criado por Linus Torvalds durante o desenvolvimento do sistema operacional Linux
- Sistema de controle de versões gratuito
- Armazena histórico de versões do projeto
- Desvantagem: não é muito eficiente para arquivos muito grandes

Mas e o Github?

- Plataforma para disponibilizar seu projeto
- Acesso de múltiplos computadores
- Projetos colaborativos
- "Rede social de código"

Pausa :)

Como funciona o Git

Fluxo usual



Analogia do Avião

- Avião parado na pista: diretório local
- Pessoas embarcando: seus arquivos
- Avião a caminho: Staging Area
- Avião chegou no destino: repositório remoto

O que seria o avião?

O avião é uma versão do seu projeto ou um
commit

Vamos entender por quê

Comandos do Git

Os comandos do Git serão feitos no
terminal

Windows - GitBash

Linux / MacOS - Bash

Comandos Locais

git **init**

- Configura todas as ferramentas que o Git precisa para localizar alterações feitas naquele diretório
 - Só é necessário uma vez por projeto
 - Só é necessário se você começar um projeto pelo seu diretório local
- O que não vamos fazer aqui!

git status

- Mostra as alterações feitas
 - arquivos criados
 - arquivos modificados
 - arquivos removidos

git *add nomeDoArquivo*

- Envia os arquivos criados/modificados/removidos do seu diretório de trabalho para a Staging Area
- Ainda está local!

git `commit -m "alguma mensagem"`

- Demarca uma versão do seu projeto com os arquivos que você moveu para a Staging Area
- Importante:
 - Não esquecer o `-m`
Caso contrário, o Vim vai ser aberto
Para sair: esc esc `:q!` enter
 - Não esquecer as aspas
 - Escreva uma mensagem que `faça sentido`

git log

- mostra uma lista dos commits antigos

O que é uma **branch**?

Um universo paralelo
do mesmo projeto

git branch

- Mostra uma lista de todas as branches do projeto local
- Destaca a branch em que você está
- A branch padrão do git é a *master*

git **branch** *nomeDaBranch*

- Cria uma nova branch com o nome que você escolher

git checkout *nomeDaBranch*

- Acessa uma branch já criada

git checkout -b *nomeDaBranch*

- Cria uma nova branch e a acessa

Comandos Remotos

git clone *localizacaoRemota*



Fala para o Git
onde
encontrar o
repositório
remoto

Cria uma pasta dentro da pasta local atual, com uma cópia do repositório remoto

git pull

- Atualiza **a branch atual do seu diretório local** com as atualizações feitas no repositório remoto

git push origin nomeDaBranch

- Envia suas alterações feitas na sua branch do diretório local (seu commit) para o repositório remoto

Github

Criando um novo repositório no Github

1) No canto superior esquerdo, clique em New

Repositories

New

Find a repository...

2) Dê um nome para o seu repositório, altere o Owner, selecione o repositório privado e clique em Criar

Create a new repository

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

Owner

future4code

Repository name *

Great repository names are short and memorable. Need inspiration? How about [automatic-waddle](#)?

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.

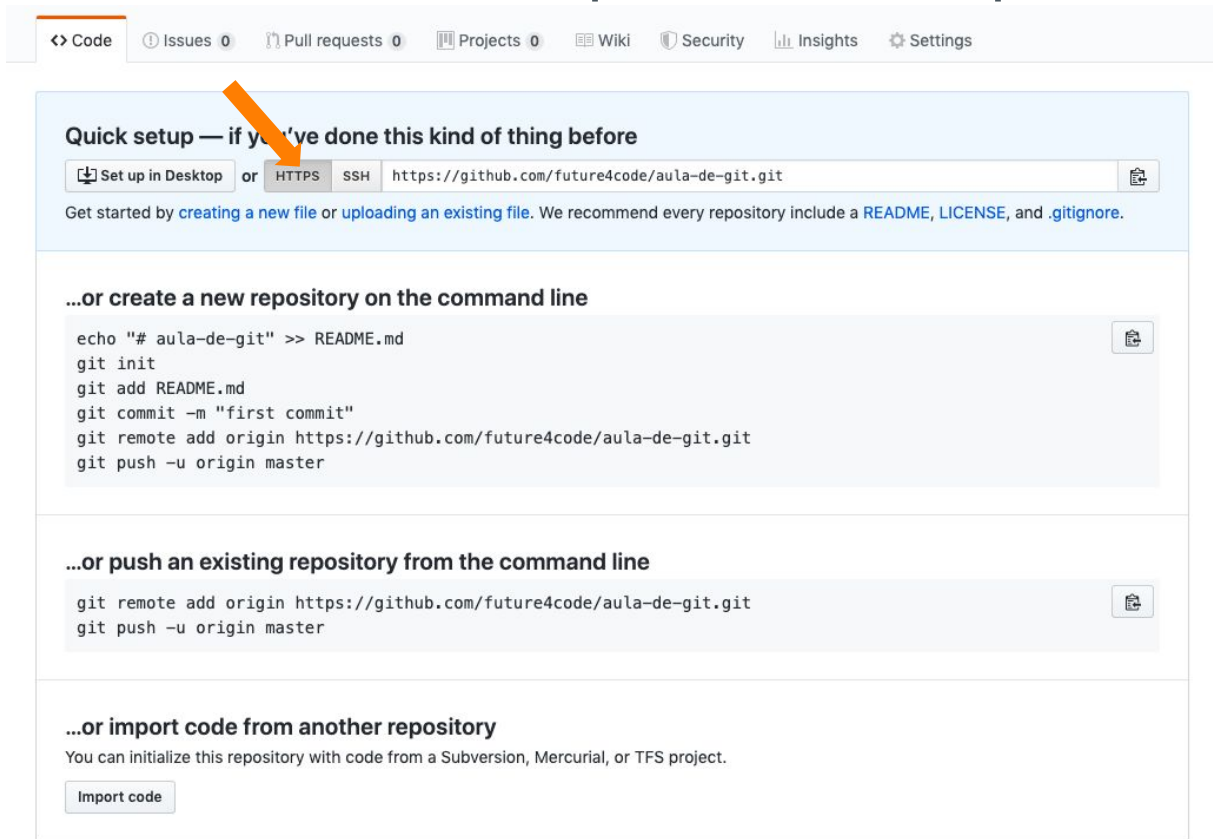
Add .gitignore: None

Add a license: None



Create repository

Clonando o repositório que você criou



<> Code ⓘ Issues 0 🔄 Pull requests 0 📁 Projects 0 📖 Wiki 🛡 Security 📊 Insights ⚙ Settings

Quick setup — if you've done this kind of thing before

☒ Set up in Desktop or **HTTPS** SSH

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# aula-de-git" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/future4code/aula-de-git.git
git push -u origin master
```

...or push an existing repository from the command line

```
git remote add origin https://github.com/future4code/aula-de-git.git
git push -u origin master
```

...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

Selecione a opção
HTTPS e copie o link

Clonando o repositório que você criou

No seu terminal, digite **git clone** e cole o link que você copiou

```
paulaarantes$ git clone https://github.com/future4code/aula-de-git.git
```

Assim que der enter, ele pedirá seu usuário e, em seguida, sua senha.

A senha não aparece escrita!

```
MacBook-Air-de-Paula:projetos paulaarantes$ git clone https://github.com/future4code/aula-de-git.git
Cloning into 'aula-de-git'...
Username for 'https://github.com': paula-aribeiroy
Password for 'https://paula-aribeiroy@github.com':
warning: You appear to have cloned an empty repository.
```

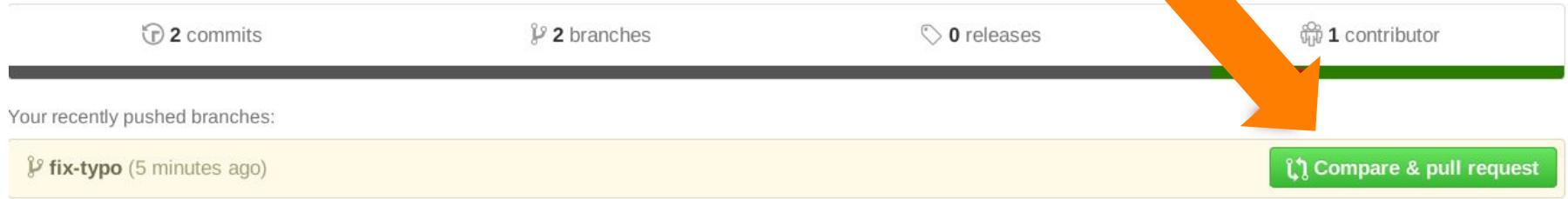
Esse warning é normal

Fork versus Clone

- O fork é a cópia de um repositório remoto para outro repositório remoto, pertencente ao seu usuário
- O clone é uma cópia de um repositório remoto no seu diretório local

Pull Request

- Usado para solicitar que as alterações que você fez em uma branch local entrem na branch master do seu repositório remoto



Pausa :)

Coding Together

Situação 1:

Você criando um novo projeto

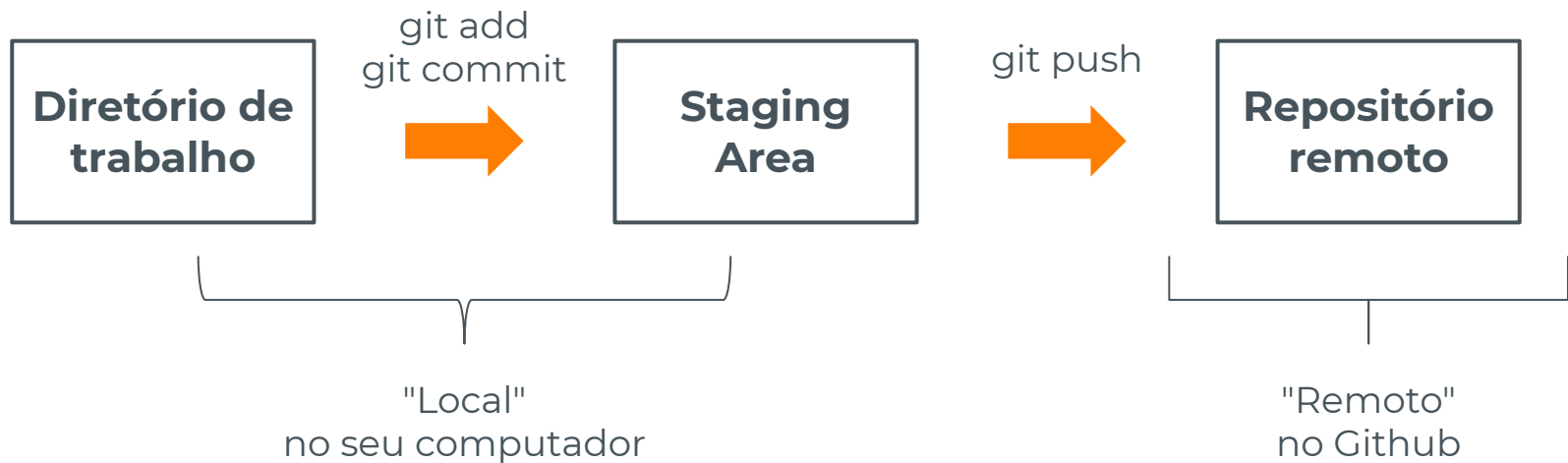
- Criar o repositório no Github
- Clonar o repositório para sua máquina (local)
- Criar arquivo
- Add, commit, push

Situação 2: Você fazendo alterações em um projeto já existente

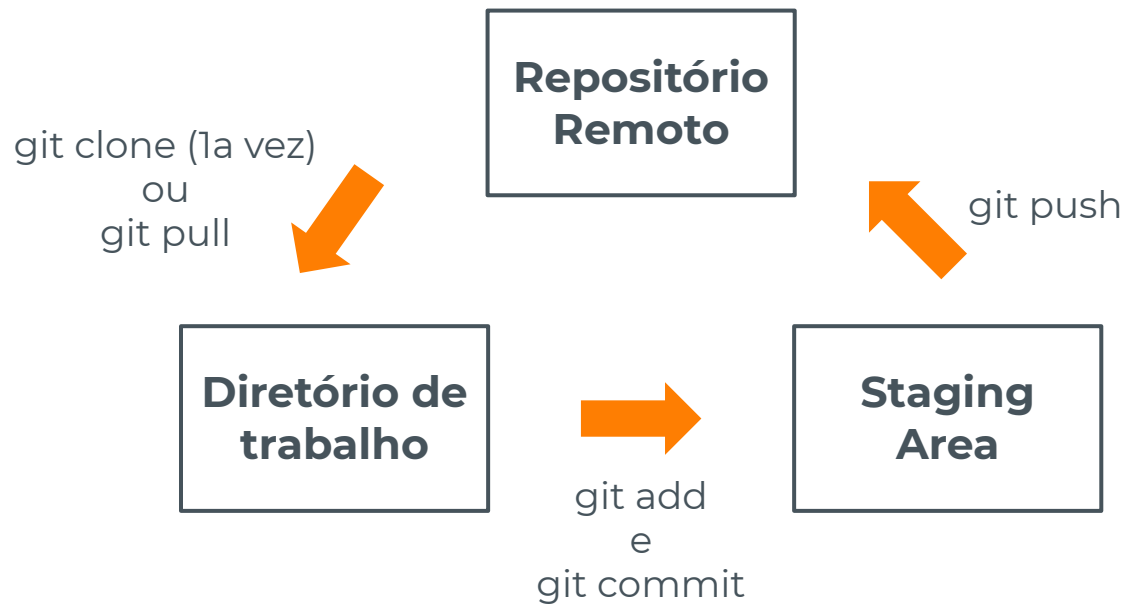
- Criar branch
- Alterar arquivo
- Add, commit, push origin branch
- Pull Request

Pausa :)

Review



Review



Não se preocupe em decorar todos os comandos!

Tabela de comandos básicos nas referências

In case of fire



 1. `git commit`

 2. `git push`

 3. `leave building`

Obrigada!