

**27^a
29**
**NOV
2019**



**X_CONGRESSO_
INTEGRADO_DA_
TECNOLOGIA_DA_
_INFORMAÇÃO.**

TECNOLOGIAS EMERGENTES: NOVOS DESAFIOS NA
EDUCAÇÃO E NA FORMAÇÃO DO PROFISSIONAL DE TI


**INSTITUTO
FEDERAL**
Fluminense
Campus
Campos Centro

Git e GitHub para iniciantes

Isadora Vasconcellos

Sumário

- Controle de Versão
- Git
- GitHub
- Exercícios práticos
- Referências



Controle de versão

➤ O que é o Git?

- É um *software* livre;
- Criado por Linus Torvalds (criador do Linux);
- É um sistema para controle de versões de arquivos;

Um sistema de **controle de versão**, no desenvolvimento de software, tem a finalidade de **gerenciar** diferentes versões do **código-fonte**.

Controle de versão

➤ Por que usar uma ferramenta de controle de versão?

✓ Permite ao programador...

- armazenar diversas cópias (versões) do código;
- restaurar versões anteriores;
- comparar mudanças ao longo do tempo;
- identificar modificações (quem, quando e o que);
- sincronizar o código entre diversos computadores;
- trabalhar de forma colaborativa com outros programadores (editando e criando novos arquivos simultaneamente).

Controle de versão

➤ Ferramentas de controle de versão:

- **Git**

<https://git-scm.com>

- **Subversion (SVN)**

<https://subversion.apache.org>

- **Mercurial**

<https://www.mercurial-scm.org>

Git



➤ **Download:**

- **Windows:** git-for-windows.github.io
- **MAC:** git-scm.com/download/mac
- **Linux:** book.git-scm.com/download/linux
 - `apt-get install git`

➤ **Manual em português:**

- <https://git-scm.com/book/pt-br/v1/Primeiros-passos-Sobre-Controle-de-Vers%C3%A3o>

➤ Configurações:

- O Git possui uma ferramenta chamada **“git config”** que permite definir variáveis que controlam aspectos de como o Git opera;
- Após a instalação, estas variáveis devem ser configuradas (usuário e e-mail), pois todos os “commits” no Git utilizam estas informações.

➤ Configurações:

- Para configurar o Git execute:

git config --global user.name meu nome

git config --global user.email meunome@email.com

Caso o parâmetro “**--global**” não seja informado, a configuração terá que ser repetida em cada repositório utilizado.

- Para verificar suas configurações execute:

git config --list

➤ Criando um novo repositório:

- Crie uma pasta:
mkdir minicurso
- Acesse a pasta pelo terminal:
cd minicurso
- Digite o comando:
git init

Pronto! Você já tem um novo repositório Git!

➤ Verificando o status do repositório:

- Para verificar o status de cada arquivo alterado/criado execute:

git status

- Adicione um novo arquivo ao repositório:

touch <nome_do_arquivo>

Ex.: touch personagem.txt

➤ Adicionando arquivo(s) no controle de versão:

- Adicione o arquivo executando o comando:

git add <nome_do_arquivo>

Ex.: git add personagem.txt

Caso queira adicionar todos os arquivos criados/modificados execute:

git add .

➤ Confirmando as alterações:

- Para confirmar as alterações, após adicionar o arquivo é necessário fazer um commit executando:

git commit -m “comentario_sobre_alterações”

Ex.: git commit -m “adicionado o arquivo personagem.txt”

Caso seja digitado apenas “git commit” será aberto o editor de texto padrão do sistema operacional no terminal para digitar o comentário.

➤ Visualizando logs:

- Para visualizar o log de alterações execute:

git log

- Para visualizar as modificações entre os logs dos commits execute:

git log -p

➤ Desfazendo alterações:

- Para desfazer alterações quando o arquivo já tiver sido adicionado (git add) execute:

git reset HEAD <nome_arquivo>

- Para desfazer alterações quando o arquivo já tiver sido commitado (git commit):

git reset --soft (id_do_commit)

- Volta para o estado de adicionado

git reset --mixed (id_do_commit)

- Volta para o estado de modificado

git reset --hard (id_do_commit)

- Remove as alterações do commit. Evitar!

➤ Ignorando arquivos:

- Caso não queira “versionar” algum arquivo ou pasta do projeto (temporários, logs, arquivo de senha, etc.) é possível criar um arquivo com uma lista de padrões que não serão monitorados pelo Git. Para isso execute na raiz pasta do projeto:

touch .gitignore

Dentro deste arquivo deve ser informado o nome ou padrão dos **arquivos/pastas** que devem ser **ignorados** pelo Git.

➤ Ignorando arquivos:

- Exemplos:

/teste1.txt

Ignora o arquivo “teste1.txt” na raiz.

teste/

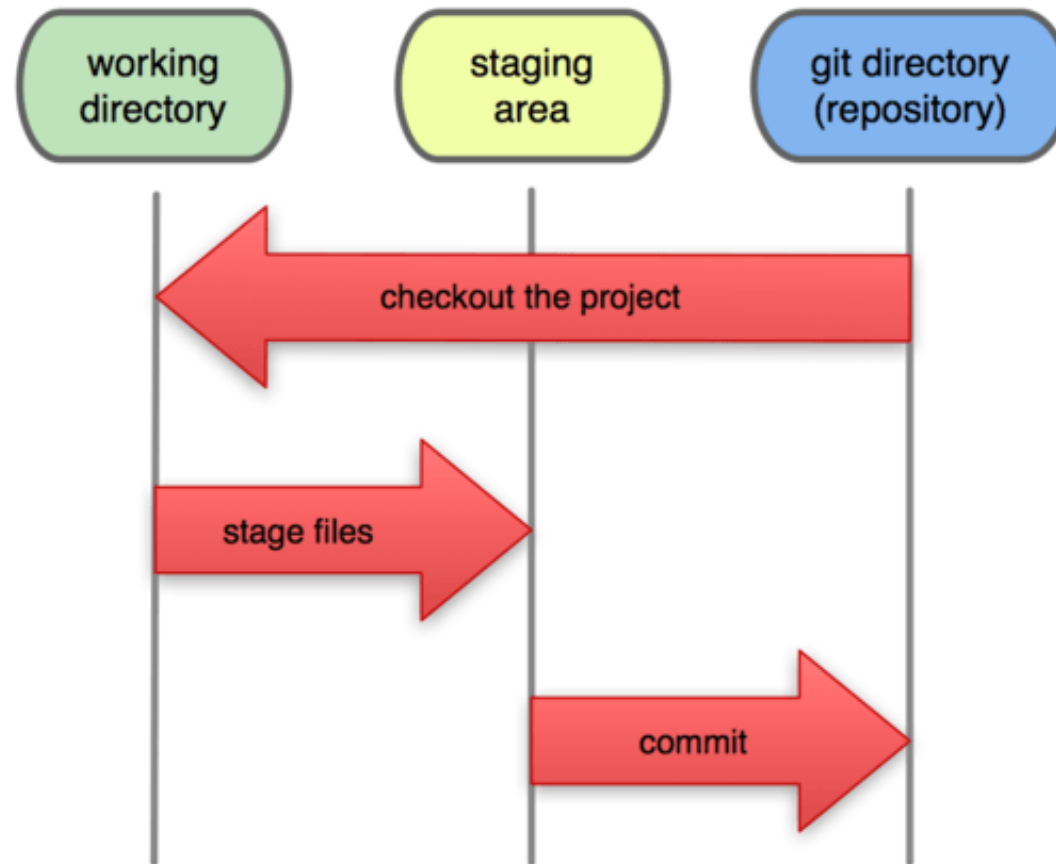
Ignora todos os arquivos do diretório “teste”.

teste/teste2.txt

Ignora o arquivo “teste2.txt” dentro do diretório “teste”.

➤ Funcionamento interno do Git:

Local Operations



➤ Locais do Git:

- ***Working directory* (diretório de trabalho):** são os arquivos do repositório existente na máquina local;
- ***Staging area* (área de preparação):** é um arquivo no diretório de trabalho que armazena informações sobre o que será enviado no próximo *commit* (*git add*);
- ***Git directory* (diretório do Git):** local onde o Git armazena os metadados e o banco de objetos do projeto (*git commit*).

➤ Estados dos arquivos do Git:

- **Modificado (*modified*):** quando um arquivo foi modificado, mas ainda não foi consolidado na base de dados local;
- **Preparado (*staged*):** quando o arquivo é marcado para fazer parte da consolidação (*git add*);
- **Consolidado (*committed*):** quando estão armazenados na base local de dados (*git commit*).

➤ Fases dos arquivos no Git:

1. Você **cria/modifica** os arquivos no diretório de trabalho;
2. Você seleciona os arquivos e os **adiciona** para sua **área de preparação** (*git add*);
3. Você faz um commit, levando os arquivos como eles estão da área de preparação e os **armazena** permanentemente no **diretório Git** (*git commit*).

➤ Revisão dos comandos:

- **git init:** inicializa um repositório git;
- **git status:** informa o status dos arquivos do repositório;
- **git add:** adiciona o arquivo criado/modificado na “área de preparo”;
- **git commit:** adiciona o arquivo preparado no “diretório git”;
- **git log:** exibe o log dos commits realizados;
- **git reset:** desfaz alterações nos arquivos “commitados”.

GitHub



➤ O que é o GitHub?

- É um serviço de **hospedagem** para projetos que utilizam o Git;
- É um **repositório *on-line*** de código fonte;
- Endereço: **github.com**

➤ Outros serviços de hospedagem de projetos Git:

- **Bitbucket**

<https://bitbucket.org>

- **GitLab**

<https://gitlab.com>

- **SourceForge**

<https://sourceforge.net>

➤ Cadastro no GitHub (Sign up):

- Informar:
 - Nome de usuário;
 - E-mail;
 - Senha;
- Selecionar um plano (*Free*):
- Informações pessoais sobre programação (Pular);
- Confirmar o cadastro por meio de e-mail recebido.

➤ Configuração de chave SSH:

- Necessária para não ter que informar a senha cadastrada sempre que o código for enviado/baixado no GitHub (*push/pull*);

- Para criar a chave SSH, execute:

```
cd ~
```

```
ssh-keygen -t rsa -C "comentário"
```

```
cat ~/.ssh/id_rsa.pub
```

➤ Configuração de chave SSH:

- No GitHub acessar:

Settings > SSH and GPG keys > New SSH key

- Preencha o nome da chave no campo “Title”;
- Copie a chave SSH do computador para o campo “Key”;
- Salva clicando em “Add SSH key”.

➤ Configuração de chave SSH:

The screenshot shows the GitHub web interface. At the top is a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. On the right of the navigation bar are notification and user icons. Below the navigation bar is a sidebar on the left with a list of settings: Personal settings, Profile, Account, Security, Emails, Notifications, Billing, SSH and GPG keys (highlighted with an orange bar), Blocked users, Repositories, Organizations, Saved replies, and Applications. The main content area is titled 'SSH keys / Add new'. It contains a 'Title' text input field and a 'Key' text area with a placeholder text: 'Begins with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521''. Below the key field is a green 'Add SSH key' button. On the right side, a user menu is open, showing 'Signed in as' followed by a 'Set status' button and a list of links: Your profile, Your repositories, Your projects, Your stars, Your gists, Feature preview, Help, Settings (highlighted in blue), and Sign out. At the bottom left, the browser address bar shows 'https://github.com/settings/profile'.

Personal settings

- Profile
- Account
- Security
- Emails
- Notifications
- Billing
- SSH and GPG keys
- Blocked users
- Repositories
- Organizations
- Saved replies
- Applications

SSH keys / Add new

Title

Key

Begins with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521'

Add SSH key

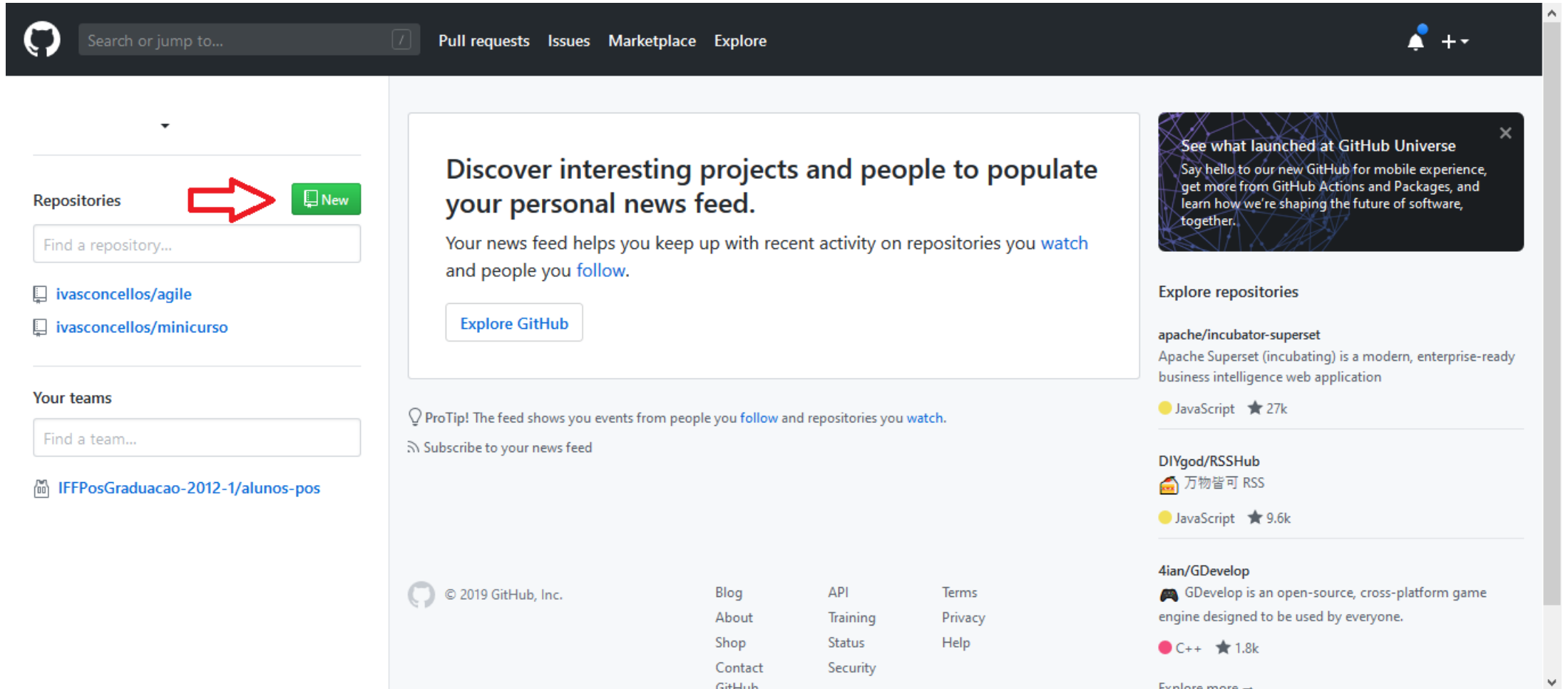
Signed in as

Set status

- Your profile
- Your repositories
- Your projects
- Your stars
- Your gists
- Feature preview
- Help
- Settings
- Sign out



https://github.com/settings/profile

➤ Criando um novo repositório no GitHub (New):





The screenshot shows the GitHub homepage. At the top is a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. On the left sidebar, under the 'Repositories' section, there is a search bar and a list of repositories. A red arrow points to a green 'New' button next to the search bar. Below the 'Repositories' section is the 'Your teams' section. The main content area features a large card titled 'Discover interesting projects and people to populate your personal news feed.' with a subtext about the news feed and an 'Explore GitHub' button. To the right of this card is a 'ProTip!' and a 'Subscribe to your news feed' link. On the far right, there is a 'See what launched at GitHub Universe' notification and a section titled 'Explore repositories' listing several repositories like 'apache/incubator-superset', 'DIYgod/RSSHub', and '4ian/GDevelop'.

Search or jump to... Pull requests Issues Marketplace Explore

Repositories  


Find a repository...

 [ivasconcellos/agile](#)

 [ivasconcellos/minicurso](#)

Your teams


Find a team...


 [IFFPosGraduacao-2012-1/alunos-pos](#)


Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you [watch](#) and people you [follow](#).

[Explore GitHub](#)

 ProTip! The feed shows you events from people you [follow](#) and repositories you [watch](#).

 [Subscribe to your news feed](#)

 © 2019 GitHub, Inc.

[Blog](#) [API](#) [Terms](#)

[About](#) [Training](#) [Privacy](#)

[Shop](#) [Status](#) [Help](#)

[Contact GitHub](#) [Security](#)



See what launched at GitHub Universe

Say hello to our new GitHub for mobile experience, get more from GitHub Actions and Packages, and learn how we're shaping the future of software, together.


Explore repositories



[apache/incubator-superset](#)

Apache Superset (incubating) is a modern, enterprise-ready business intelligence web application


 JavaScript  27k



[DIYgod/RSSHub](#)

 万物皆可 RSS

 JavaScript  9.6k

[4ian/GDevelop](#)

 GDevelop is an open-source, cross-platform game engine designed to be used by everyone.

 C++  1.8k

[Explore more](#)

➤ Criando um novo repositório no GitHub:

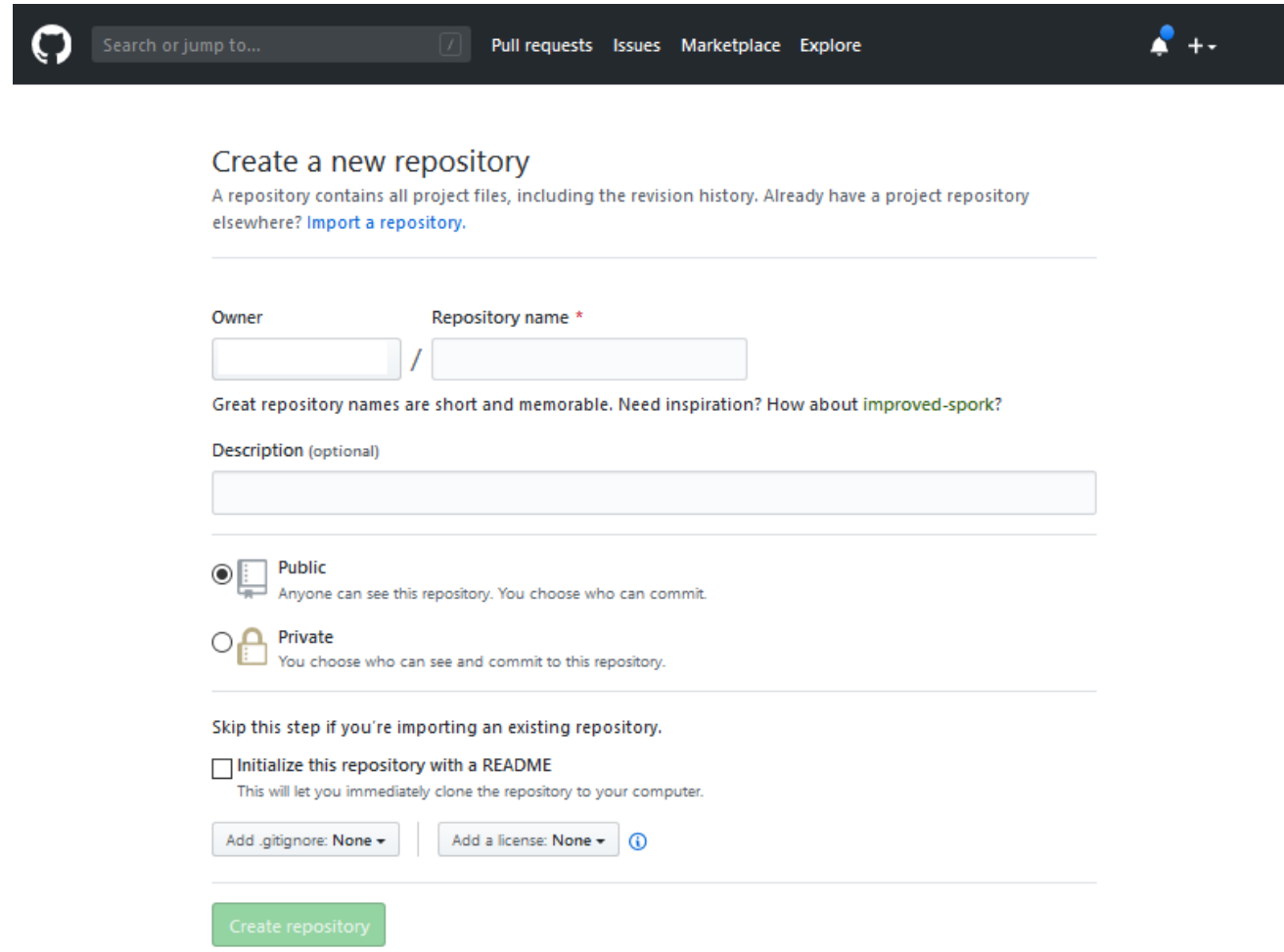
- Preencher campos obrigatórios:

- **Owner** (dono do repositório);
- **Repository name** (nome do repositório);
- **Public/Private** (público: visível a todos/privado: visível somente a membros do projeto);

- Campos opcionais:

- **Description** (descrição);
- **Initialize this repository with a README** (cria o arquivo Readme do repositório);
- **Add .gitignore** (selecionar linguagem de programação do projeto);
- **Add licence** (selecionar a licença de uso do projeto).

➤ Criando um novo repositório no GitHub:



The screenshot shows the GitHub interface for creating a new repository. At the top is a dark navigation bar with the GitHub logo, a search bar, and links for Pull requests, Issues, Marketplace, and Explore. Below this is the 'Create a new repository' section. It includes a sub-header, a descriptive paragraph, and a link to 'Import a repository'. The main form has two input fields: 'Owner' and 'Repository name *'. Below these is a tip about repository names. There is a 'Description (optional)' text area. The 'Visibility' section has two radio buttons: 'Public' (selected) and 'Private'. Below this is a note about skipping the step if importing an existing repository. There are checkboxes for 'Initialize this repository with a README' and 'Add a license'. At the bottom is a green 'Create repository' button.

Create a new repository

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

Owner / Repository name *

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

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

➤ Sincronizando o repositório local com o remoto:

- Copie a URL (SSH ou HTTPS) do repositório do GitHub;

The screenshot shows the GitHub interface for a repository named 'minicurso'. The repository has 10 commits, 1 branch, 0 packages, 0 releases, and 1 contributor. The 'Clone or download' button is highlighted with a red arrow. A modal window is open, showing the 'Clone with SSH' option. The modal contains a message: 'You don't have any public SSH keys in your GitHub account. You can add a new public key, or try cloning this repository via HTTPS.' Below this, there is a text input field with the URL 'git@github.com:...' and a button to 'Open in Desktop'. The modal also has a 'Download ZIP' button.

File	Commit Message
Questionario.txt	Create Questionario.txt
README.md	Initial commit
dupla1.txt	Rename dupla1 to dupla1.txt
dupla2.txt	Create dupla2.txt
dupla3.txt	Create dupla3.txt
dupla4.txt	Create dupla4.txt
dupla5.txt	Create dupla5.txt
dupla6.txt	Create dupla6.txt

➤ Sincronizando o repositório local com o remoto:

- Na pasta do repositório local (criado na máquina) execute:
git remote add origin <url_copiada>
- Envie o código do repositório local para o remoto executando:
git push -u origin master

➤ Sincronizando o repositório local com o remoto:

- Para visualizar a URL atual do repositório execute:

git remote -v

- Para alterar a URL do repositório execute:

git remote set-url origin <nova_url>

➤ Enviando as alterações para o repositório remoto:

- Para enviar as alterações, já “commitadas” para o repositório remoto execute:

git push origin master

➤ Obtendo um repositório remoto:

- Na página do repositório haverá a URL (caminho) do mesmo (SSH e HTTPS). Copie um dos endereços;

Clone or download → Clone with SSH/Clone with HTTPS

- No terminal, no local em que desejar colocar a cópia do projeto a ser copiado, execute:

git clone <endereço_copiado>

➤ Baixando atualizações do repositório remoto:

- Para obter as atualizações existentes no projeto do repositório (código atualizado) remoto execute:

git pull

➤ Resumo dos comandos:

- **git clone:** copia para a máquina local um projeto do GitHub;
- **git pull:** faz o download do código atualizado para a máquina local;
- **git push:** envia o código atualizado para o repositório no GitHub.

➤ **Branches:**

- São “ramos” utilizados para desenvolver **funcionalidades isoladas** umas das outras;
- O *branch* “**master**” é o *branch* “padrão” ao criar um repositório;
- Use outros *branches* para desenvolver funcionalidades e mescle-os (*merge*) ao *branch master* após a conclusão.

➤ Criando um *branch* local:

- Para criar um *branch* execute:

git checkout -b <nome_do_branch>

➤ Cria e troca automaticamente para o novo *branch*.

git branch <nome_do_branch>

➤ Apenas cria o *branch*, continuando no *branch* atual.

Os arquivos constantes no *novo branch* criado serão *iguais* a do *branch de origem* (que estava em uso no momento da criação).

➤ Verificando o *branch* atual e os existentes:

- Para verificar os *branches* existentes e o que está sendo usado no momento execute:

git branch

O *branch* com o **asterisco** e **destacado com cor diferente** (geralmente verde) é o que está sendo atualmente utilizado.

➤ Trocando de *branch*:

- Para trocar de um *branch* para outro ou para o *master* execute:

git checkout <nome_do_branch_ou_master>

➤ Enviando um *branch* local para um repositório remoto:

- Para enviar um *branch* local para um repositório remoto execute:

git push origin <nome_do_branch>

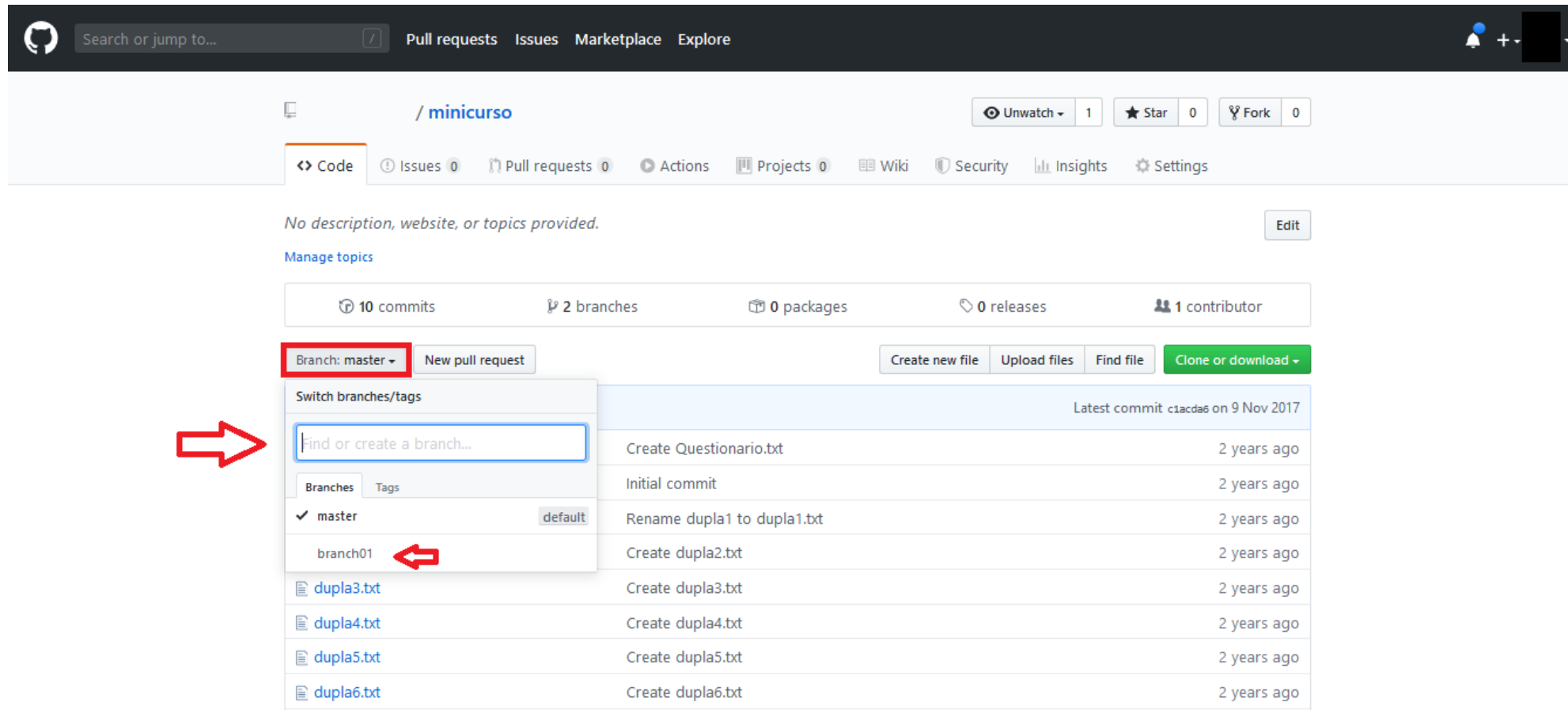
➤ Copiando um *branch* do GitHub para a máquina local:

- Para “baixar” um *branch* remoto para o repositório local execute:

git checkout -b <nome_branch> origin/<nome_branch_remoto>

➤ Criando um *branch* no GitHub:

- Digite o nome do novo *branch* e clique em “Enter”.



➤ Renomeando um *branch*:

- Para renomear um *branch* local a partir dele execute:

git branch -m <novo_nome_branch>

- Para renomear um *branch* a partir de outro *branch* execute:

git branch -m <antigo_nome_branch> <novo_nome_branch>

➤ Removendo um *branch*:

- Para remover um *branch* local execute:

git branch -D <nome_do_branch_local>

Se o branch já tiver sido enviado para um **repositório remoto** continuará existindo neste, sendo removida apenas a cópia local.

- Para remover um *branch* remoto execute:

git push origin --delete <nome_do_branch_remoto>

➤ Verificando diferenças entre *branches*:

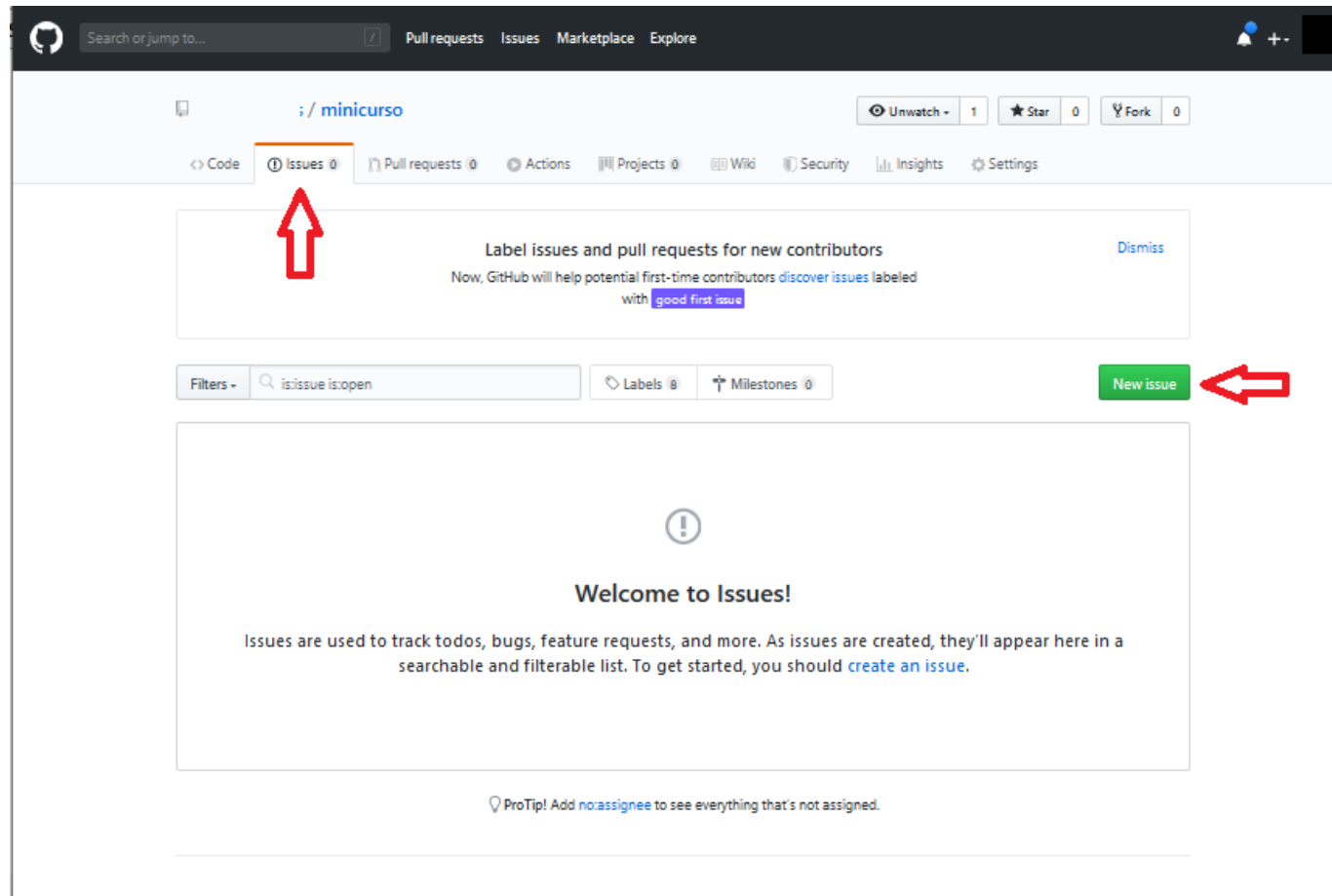
- É possível verificar as diferenças entre dois *branches* execute:

git diff <branch origem> <branch destino>

➤ Issues:

- São “tarefas” criadas para desenvolver uma funcionalidade ou resolver um problema;
- É uma boa prática criar para cada issue um *branch*.

➤ Criando uma nova Issue: Issues → New issue



➤ Criando uma nova Issue:

- Deve conter um **título** e, preferencialmente, uma **descrição** detalhada do que deve ser feito;
- Pode conter as pessoas envolvidas (*assignees*), etiquetas (*labels*), projetos relacionados (*projects*) e um marco (*milestone*).

➤ Criando uma nova Issue:

The screenshot shows the GitHub web interface for the repository `minicurso`. The 'Issues' tab is active, and a modal for creating a new issue is displayed. The modal includes a title input field, a rich text editor with 'Write' and 'Preview' tabs, and a 'Submit new issue' button. The right sidebar shows settings for Assignees, Labels, Projects, and Milestones.

Search or jump to... Pull requests Issues Marketplace Explore

/ minicurso Unwatch 1 Star 0 Fork 0

<> Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

+ Related Issues Beta Try it.

Write Preview AA B i “ <> @ Submit new issue

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Styling with Markdown is supported

Assignees No one—assign yourself

Labels None yet

Projects None yet

Milestone No milestone

➤ Criando uma nova Issue:

The screenshot shows the GitHub interface for a repository named 'minicurso'. The top navigation bar includes links for Pull requests, Issues, Marketplace, and Explore. The repository header shows 'minicurso' with 1 Unwatch, 0 Stars, and 0 Forks. The 'Issues' tab is selected, showing a list of issues. A new issue titled '+Cadastro de personagens #1' is being created. A red arrow points to the 'Open' button next to the issue title. The issue content area shows a text input field with the placeholder 'Nome do personagem.' and a rich text editor with a 'Write' tab. A red arrow points to the 'Close issue' button at the bottom of the issue editor. The right sidebar contains metadata for the issue, including Assignees, Labels, Projects, Milestone, and Notifications.

Search or jump to... Pull requests Issues Marketplace Explore

minicurso Unwatch 1 Star 0 Fork 0

Code Issues 1 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

+Cadastro de personagens #1 Edit New issue

Open opened this issue 1 minute ago · 0 comments

commented 1 minute ago · edited Owner + ...

Nome do personagem.

Write Preview AA B i “ <> @

Leave a comment

Attach files by dragging & dropping, selecting or pasting them.

Close issue Comment

Assignees No one—assign yourself

Labels None yet

Projects None yet

Milestone No milestone

Notifications Customize

Unsubscribe

➤ Merge:

- É a **junção** do código de dois **branches**;
- Pode ser feito entre **branches** criados pelo usuário ou entre um **branch** e o **master** (*branch* principal padrão dos projetos Git).

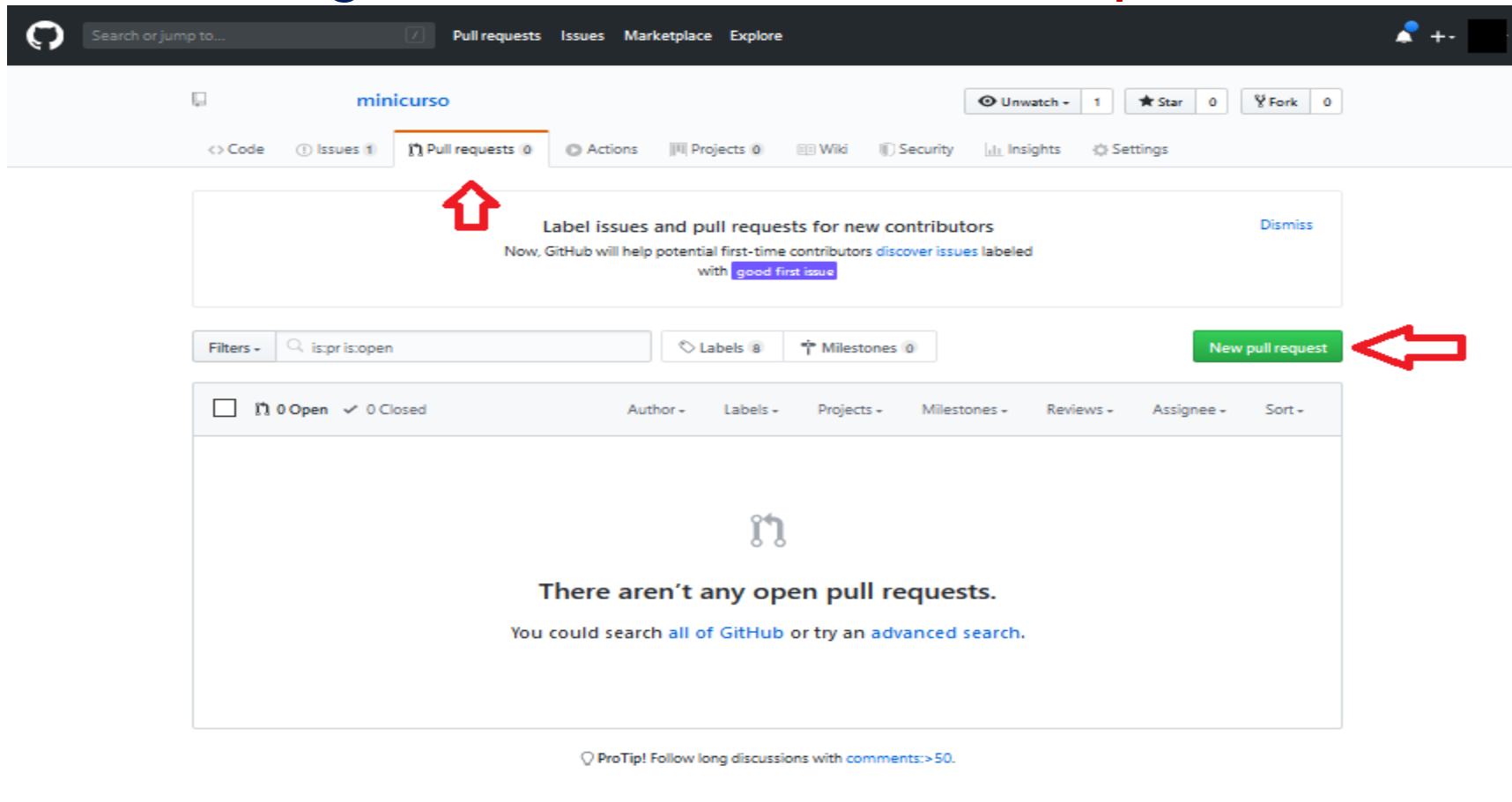
➤ Fazendo um *Merge* local:

- Para fazer um *merge* local execute o comando:
git merge <nome_outro_branch>

Fazer um Merge pode gerar **conflitos** que terão que ser **resolvidos manualmente**.

➤ Fazendo um *Merge* remoto:

- No GitHub o *merge* é chamado de “Pull Request”.



➤ Fazendo um Merge remoto:

Search or jump to... Pull requests Issues Marketplace Explore

ivasconcellos / minicurso Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).

base: master + compare: issue1 ✓ Able to merge. These branches can be automatically merged.

Create pull request Discuss and review the changes in this comparison with others.

1 commit 1 file changed 0 commit comments 1 contributor

Commits on Nov 26, 2019

Adicionado arquivo issue1.txt Verified 3e6fd11

Showing 1 changed file with 1 addition and 0 deletions. Unified Split

1 issue1.txt

```
... @@ -0,0 +1 @@
1 + Simulação de pull request
```


➤ Fazendo um Merge remoto:

Search or jump to... Pull requests Issues Marketplace Explore

/ minicurso Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: master compare: issue1 ✓ Able to merge. These branches can be automatically merged.

Adicionado arquivo issue1.txt

Write Preview AA B i “ <> ☰ ☷ ☹ @ 📌 ↶

Criando o pull request

Attach files by dragging & dropping, selecting or pasting them.

➡ Create pull request

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

➤ Fazendo um Merge remoto:

Search or jump to... Pull requests Issues Marketplace Explore

Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 1 Actions Projects 0 Wiki Security Insights Settings

Adicionado arquivo issue1.txt #1

Open ivasconcellos wants to merge 1 commit into master from issue1

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -0

ivasconcellos commented 2 minutes ago

Owner + Add + More

Criando o pull request

Adicionado arquivo issue1.txt Verified 3e6fd11

Add more commits by pushing to the issue1 branch on ivasconcellos/minicurso.

Continuous integration has not been set up
GitHub Actions and several other apps can be used to automatically catch bugs and enforce style.

✓ This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request - You can also open this in GitHub Desktop or view command line instructions.

Reviewers: No reviews

Assignees: No one—assign yourself

Labels: None yet

Projects: None yet

Milestone: No milestone

Notifications: Customize

Unsubscribe

➤ Fazendo um Merge remoto:

Search or jump to... Pull requests Issues Marketplace Explore

/ minicurso Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 1 Actions Projects 0 Wiki Security Insights Settings

Adicionado arquivo issue1.txt #1

Open wants to merge 1 commit into master from issue1

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -0

commented 5 minutes ago Owner + ...

Criando o pull request

Adicionado arquivo issue1.txt Verified 3e6fd11

Add more commits by pushing to the **issue1** branch on ivasconcellos/minicurso.

Merge pull request #1 from /issue1

Adicionado arquivo issue1.txt

Confirm merge Cancel

➤ Fazendo um Merge remoto:

The screenshot shows a GitHub pull request for the repository `ivasconcellos / minicurso`. The title of the pull request is "Adicionado arquivo issue1.txt #1". The status is "Merged", indicated by a green checkmark and the word "Merged" in a red box. The pull request was created by `ivasconcellos` and merged into the `master` branch from the `issue1` branch. The commit message is "Adicionado arquivo issue1.txt". The pull request is now closed, and a message at the bottom states: "Pull request successfully merged and closed. You're all set—the `issue1` branch can be safely deleted." The right sidebar shows the pull request details, including reviewers, assignees, labels, projects, and milestones.

Search or jump to... Pull requests Issues Marketplace Explore

ivasconcellos / minicurso Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 1 Actions Projects 0 Wiki Security Insights Settings

Adicionado arquivo issue1.txt #1

Merged ivasconcellos merged 1 commit into `master` from `issue1` now

Conversation 0 Commits 1 Checks 0 Files changed 1 +1 -0

ivasconcellos commented 8 minutes ago Owner + ...

Criando o pull request

Adicionado arquivo issue1.txt Verified 3e6fd11

merged commit 54c0ceb into `master` now Revert

Pull request successfully merged and closed Delete branch

You're all set—the `issue1` branch can be safely deleted.

Reviewers: No reviews

Assignees: No one—assign yourself

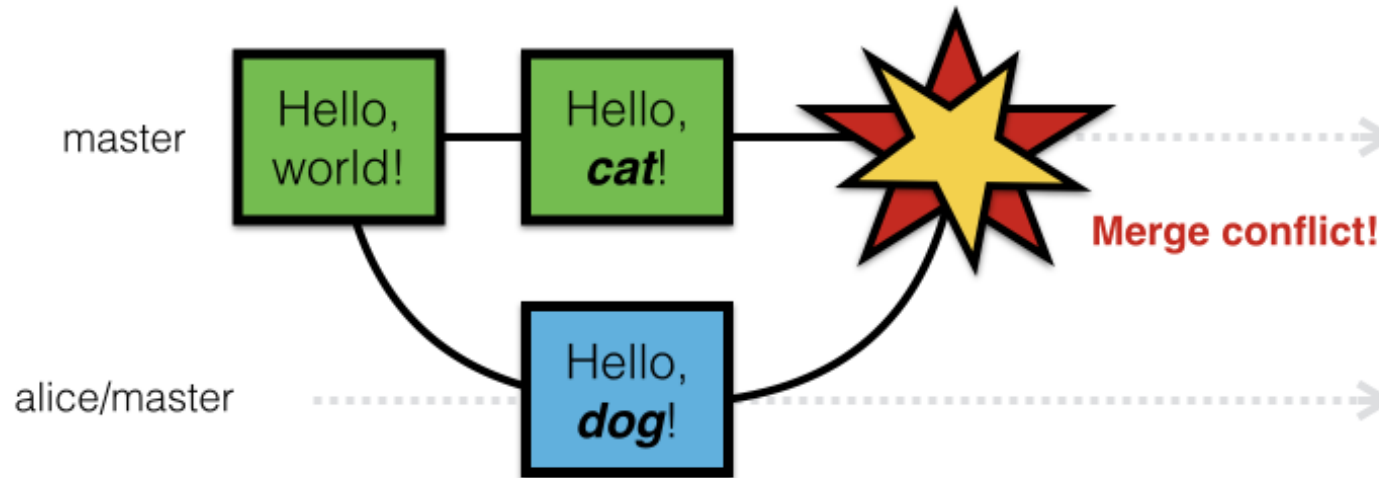
Labels: None yet

Projects: None yet

Milestone: None yet

➤ Conflitos:

- Ocorre quando **alterações concorrentes** são feitas nas mesmas linhas ou linhas próximas de um arquivo;



➤ Conflitos:

- Pode ocorrer durante a execução do comando *push* ou *merge* ou ao realizar um *pull request*;
- Precisa ser resolvido *manualmente*, pois o Git/GitHub não consegue identificar qual é o código correto;
- Após resolver o conflito deve ser feito um novo *commit* e *push*.

Dúvidas???

Questionário

<https://forms.gle/RSHzJYJGuXd1oh9J6>

Referências

- Primeiros passos - Noções Básicas de Git. Disponível em: git-scm.com/book/pt-br/v1/Primeiros-passos-No%C3%A7%C3%B5es-B%C3%A1sicas-de-Git. Acesso em: 15 nov. 2019.