



Programação Orientada a Objeto

Jackson Machado

Github

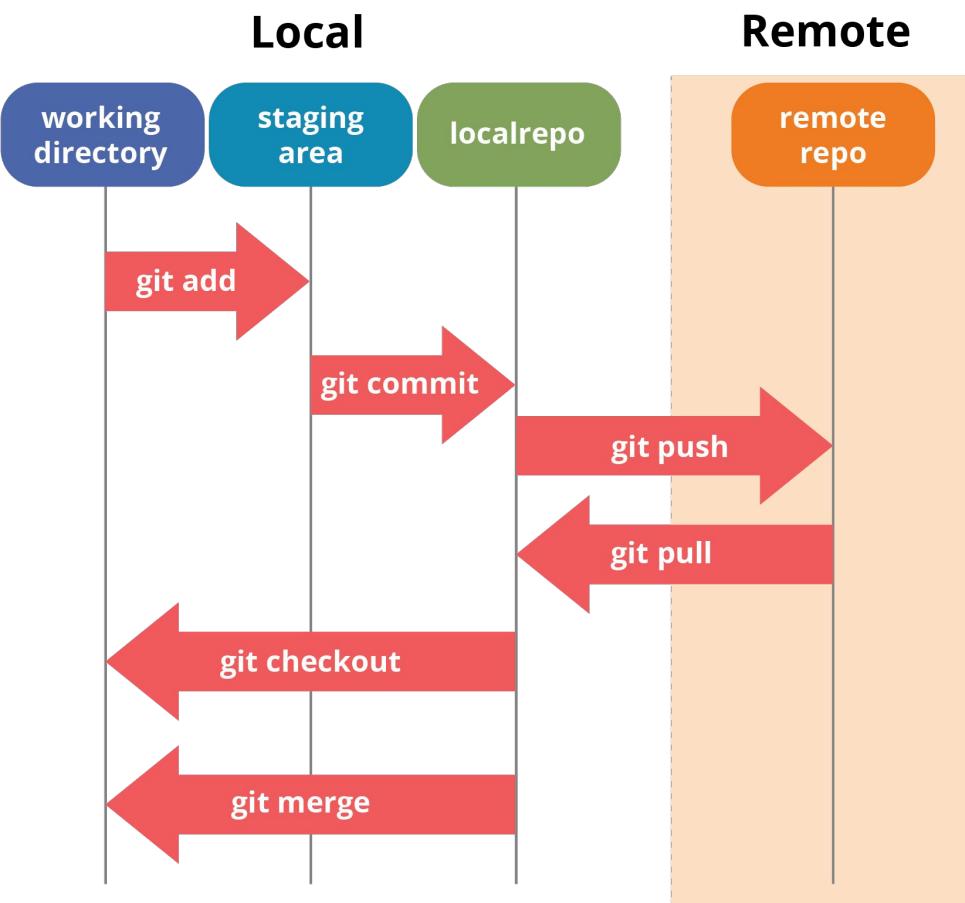
Conceito

- Plataforma de hospedagem de código-fonte
- Utilização da metodologia Git para versionamento
- Criação de projetos privados e *Open Source* acessível de qualquer lugar



Git

Fluxo Básico



Git

Common Git Commands



- `$git config`
- `$git init`
- `$git clone <path>`
- `$git add <file_name>`
- `$git commit`
- `$git status`
- `$git remote`
- `$git checkout <branch_name>`
- `$git branch`
- `$git push`
- `$git pull`
- `$git merge <branch_name>`
- `$git diff`
- `$git reset`
- `$git revert`
- `$git tag`
- `$git log`

Git

Fluxo Básico

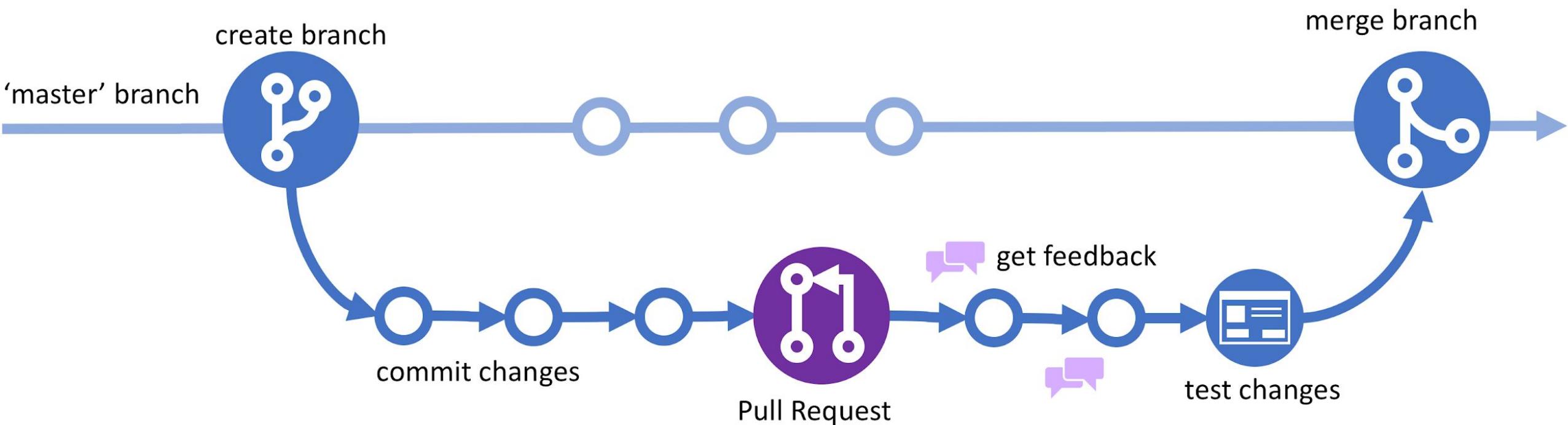
In case of fire



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

Github

GitHub Flow



Copyright © 2018 Build Azure LLC

<http://buildazure.com>

Iniciando no Git

Criando um repositório

A screenshot of the GitHub web interface. At the top left is the GitHub logo and a search bar with placeholder text "Search or jump to...". Below the search bar is a dark header with the word "Repositories" and a green "New" button featuring a document icon and the word "New". A light gray sidebar contains a search bar with placeholder text "Find a repository..." and a list of repository names. A callout bubble with a blue border and white background points from the "New" button towards the text "Cria um novo repositório REMOTO do Github".

Cria um novo repositório REMOTO do Github

Repositories

New

Find a repository...

- jacksjm/java-senac-2019.0520.001
- jacksjm/c-matrix-openmp
- jacksjm/c-matrix-pthread
- jacksjm/c-matrix
- jacksjm/c-vector
- jacksjm/rsa-python
- haskellcamargo/thetesto

Show more

Iniciando no Git

Criando um repositório

Caso não tenha ainda criado um repositório, aparecerá no local dos repositórios um texto em azul, para criar um novo repositório

Iniciando no Git

Criando um repositório

The screenshot shows the GitHub 'Create a new repository' interface. The 'Repository name' field is highlighted with a callout box containing the instruction: 'Digite um nome para o repositório (use sua imaginação :)... mas que descreve algo do projeto)'. Below it, the 'Description (optional)' field has a callout box: 'Descreve o intuito do seu projeto e o que estará contido nele'. Another callout points to the 'Initialize this repository with a README' checkbox: 'Marque esta opção para criar o README em Markdown padrão'. At the bottom, there's a 'Create repository' button with a callout: 'Crie seu projeto e seja feliz :)'. Other visible fields include 'Owner' (jacksjm), 'Repository name' (special-system), 'Description' (Descrição do Repositório), 'Visibility' (Public or Private), and 'Language' (Add .gitignore: Java). A note at the bottom says 'Skip this step if you're importing an existing repository.'

Digite um nome para o repositório (use sua imaginação :)... mas que descreve algo do projeto)

O Github sugere um nome se preferir...

Descreve o intuito do seu projeto e o que estará contido nele

Marque esta opção para criar o README em Markdown padrão

Pode ser criado como Público ou Privado (recomendo FORTEMENTE a criação como Público)

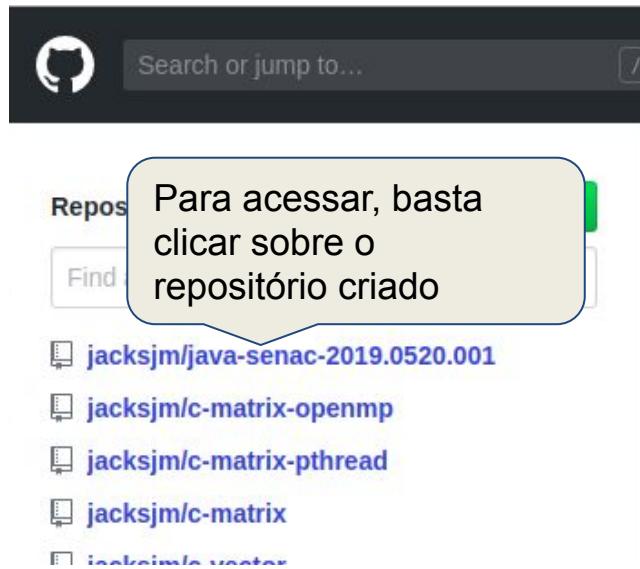
Crie seu projeto e seja feliz :)

Seleciona a linguagem que está trabalhando, para criação do .gitignore

Iniciando no Git

Clonando um repositório

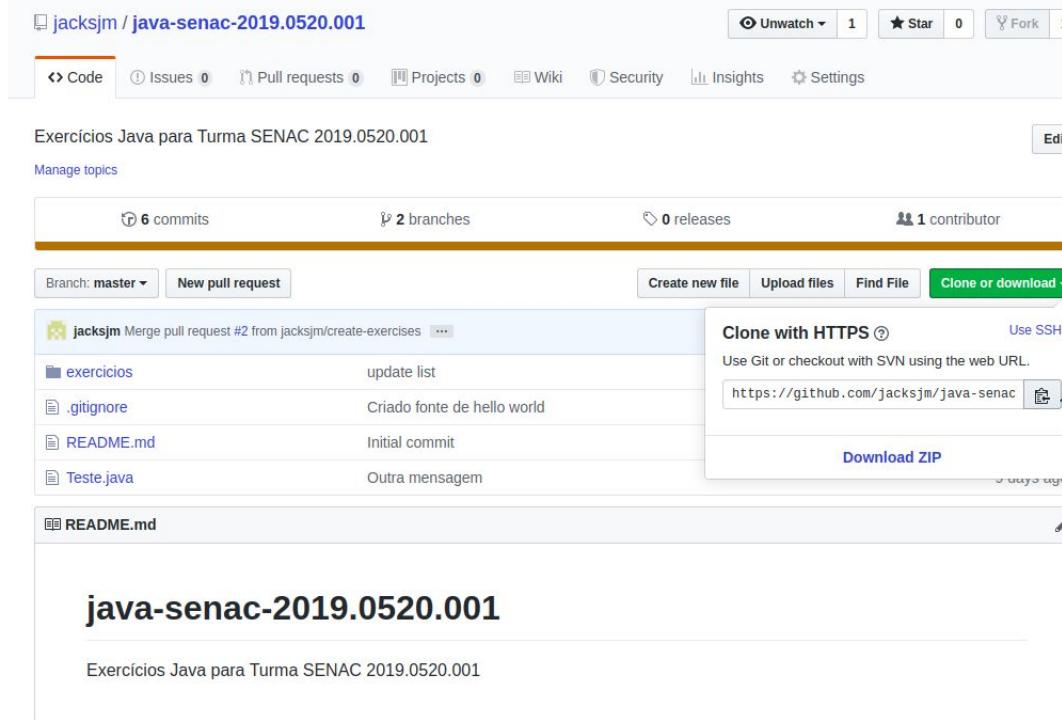
Para utilizar o repositório, é necessário cloná-lo para o computador local (workspace), para isso acesse o repositório.



Iniciando no Git

Clonando um repositório

Após acessar o repositório, selecione a opção **Clone or download** e copie o link apresentado.



Iniciando no Git

Clonando um repositório

Após copiar o link, acesse o Terminal (PowerShell, CMD, Terminal, o que achar melhor), posicione na pasta que desejar e execute o comando *git clone URL*, onde a URL é o endereço copiado.

```
jackson@jackson-ubuntu:~/Documentos/Java/Exemplo$ git clone https://github.com/jacksjm/java-senac-2019.0520.001.git
Cloning into 'java-senac-2019.0520.001'...
remote: Enumerating objects: 64, done.
remote: Counting objects: 100% (64/64), done.
remote: Compressing objects: 100% (51/51), done.
remote: Total 64 (delta 10), reused 58 (delta 9), pack-reused 0
Unpacking objects: 100% (64/64), done.
```

Iniciando no Git

Clonando um repositório

Com isso, o repositório remoto agora existirá fisicamente no computador, para acessá-lo (com o VsCode), pode executar a sequência de comandos:

*cd NOMEDOPROJETO
code .*

```
jackson@jackson-ubuntu:~/Documentos/Java/Exemplo$ cd java-senac-2019.0520.001/  
jackson@jackson-ubuntu:~/Documentos/Java/Exemplo/java-senac-2019.0520.001$ code .  
jackson@jackson-ubuntu:~/Documentos/Java/Exemplo/java-senac-2019.0520.001$ █
```

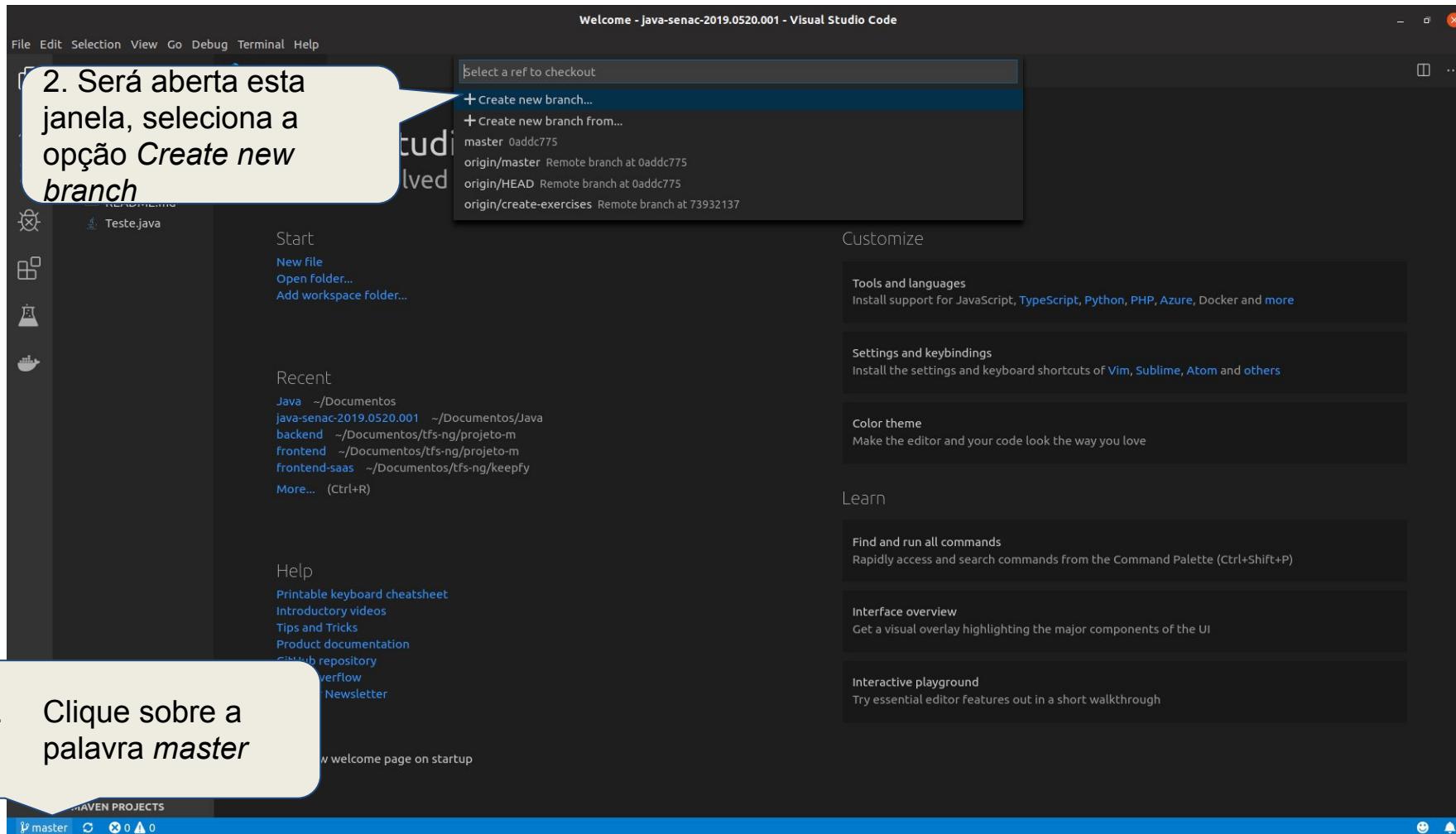
Iniciando no Git

Criando uma nova Branch

A partir disso, no VsCode é possível iniciar o desenvolvimento do projeto. Para isso, vamos criar primeiramente uma nova Branch

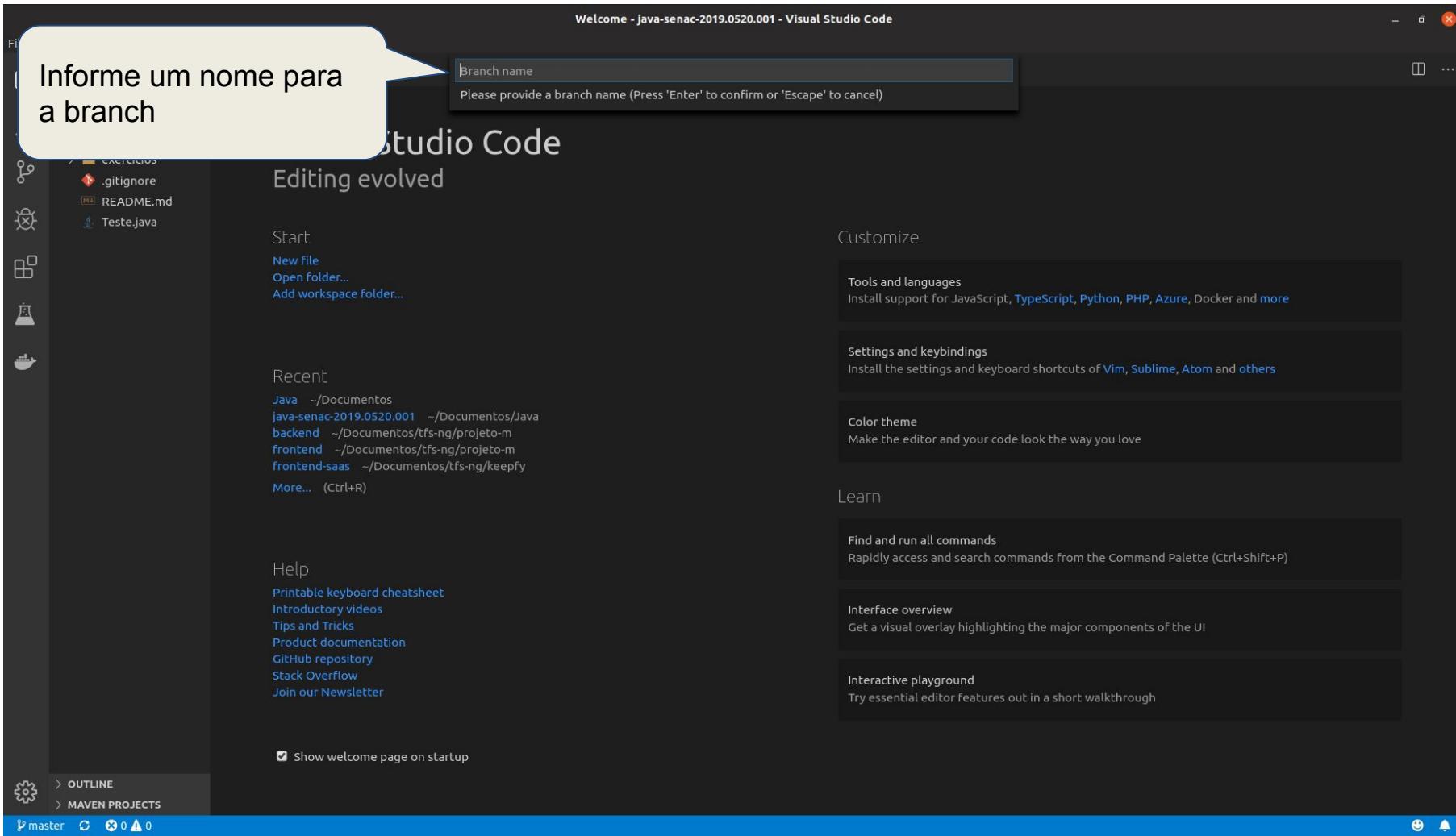
Iniciando no Git

Criando uma nova Branch



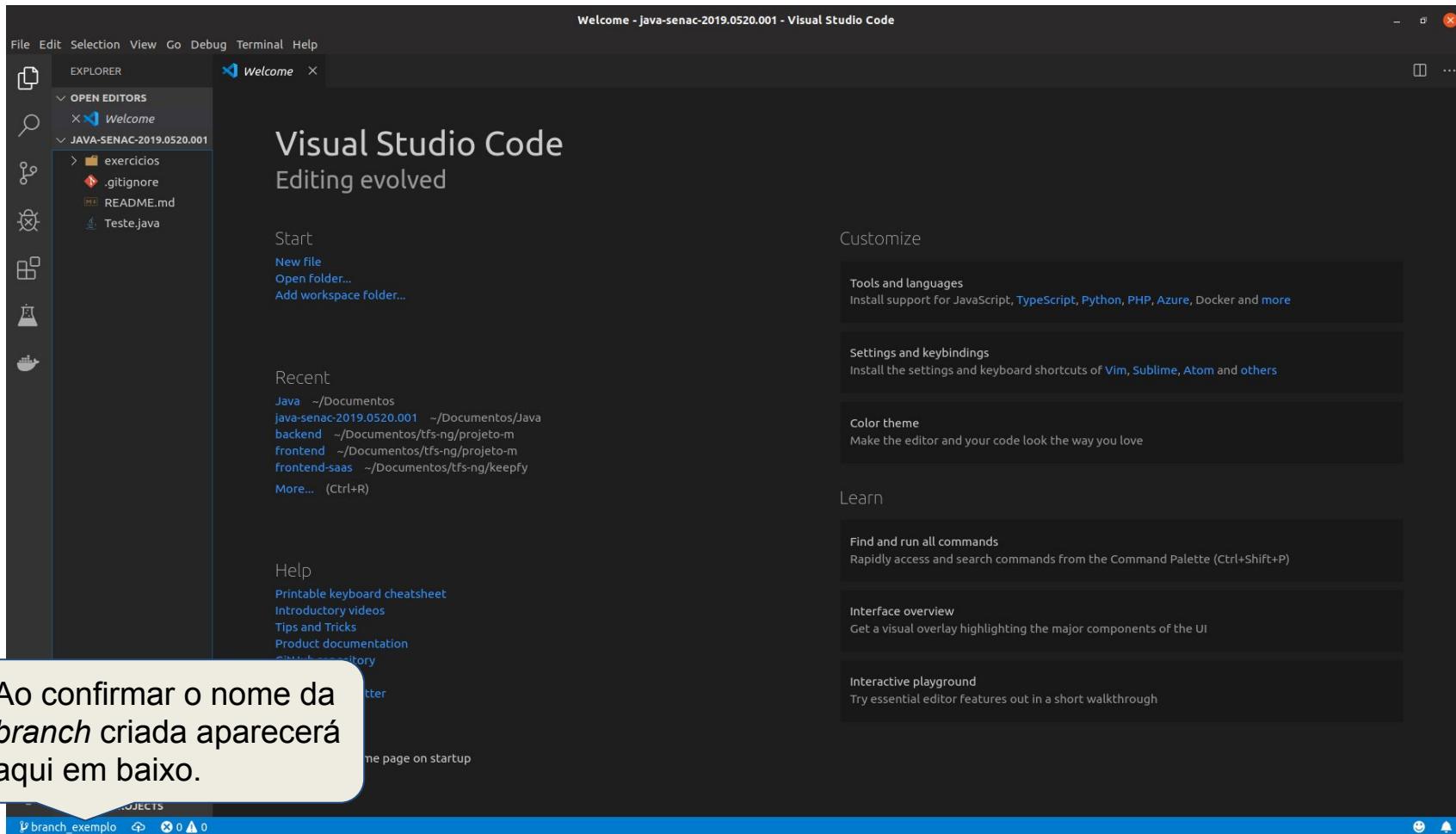
Iniciando no Git

Criando uma nova Branch



Iniciando no Git

Criando uma nova Branch



Iniciando no Git

Commitando alterações

Após a criação da branch, já é possível criar novos arquivos e realizar as alterações de fontes, sendo que as diferenças serão exibidas a partir da opção de Git do VsCode.

Iniciando no Git

Commitando alterações



Iniciando no Git

Commitando alterações

Ao encerrar qualquer alteração é importante realizar o *commit* das informações. Há duas maneiras de se realizar o *commit*, confirmado todas as alterações ou confirmado apenas a alteração de um único (ou mais) fonte.

Iniciando no Git

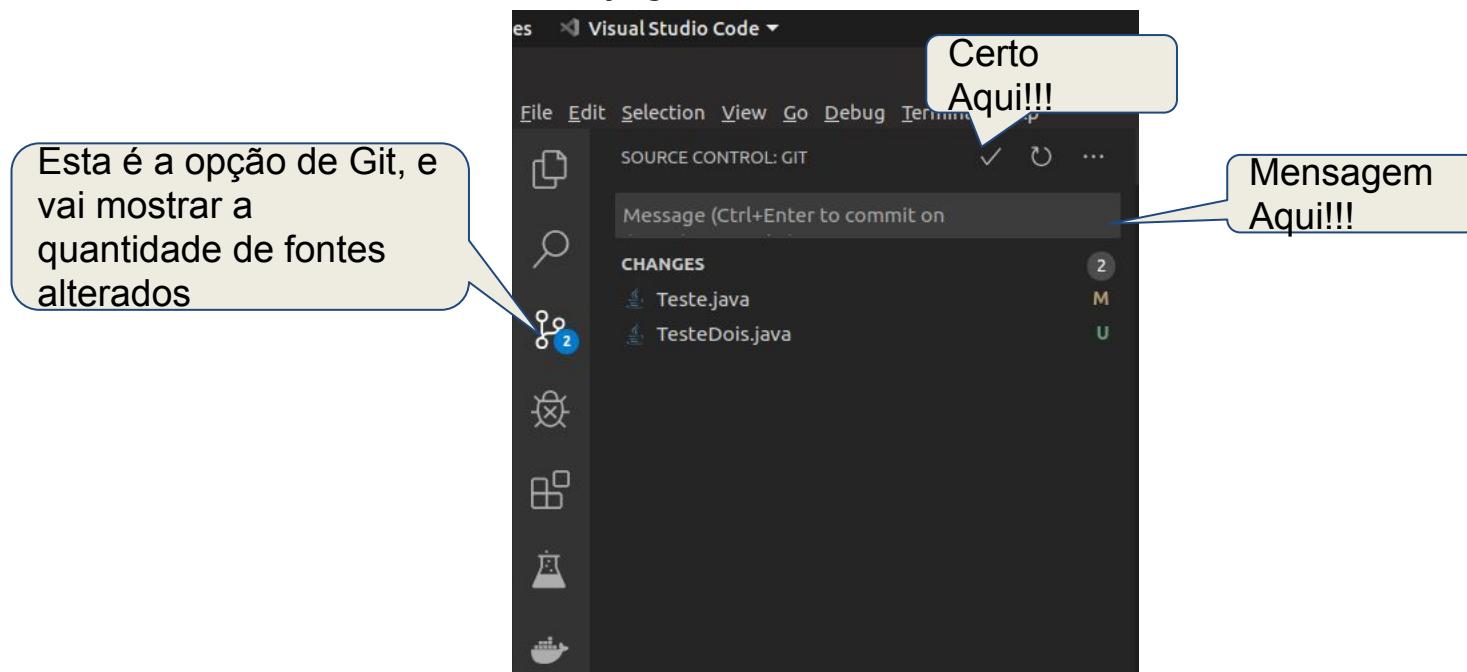
Commitando alterações

NUNCA REALIZE COMMIT NA MASTER, LEMBRE-SE
NUNCA REALIZE COMMIT NA MASTER

Iniciando no Git

Commitando alterações

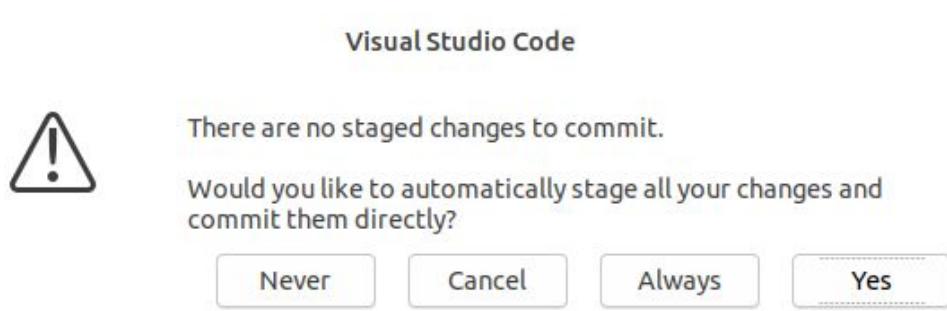
Para realizar o *commit* de todas as alterações, basta informar a mensagem (que deve ser o que foi feito) e selecionar o *certo* na opção de Git



Iniciando no Git

Commitando alterações

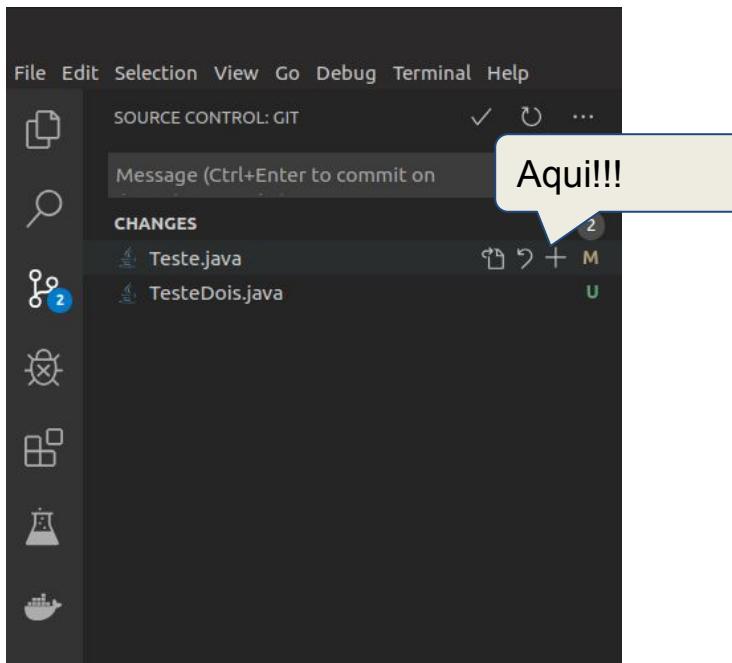
O VsCode irá confirmar se quer realmente confirmar todas as fontes, eu aconselho a selecionar sempre yes e não *allways*



Iniciando no Git

Commitando alterações

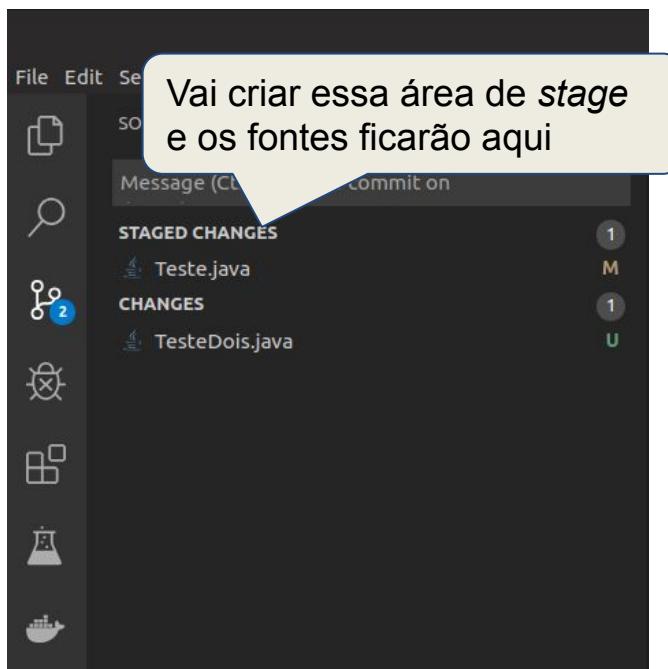
Caso queira subir apenas um (ou mais) fontes, mas não todos, é necessário colocar o fonte em *stage*. Para isso, clique no mais localizado no fonte



Iniciando no Git

Commitando alterações

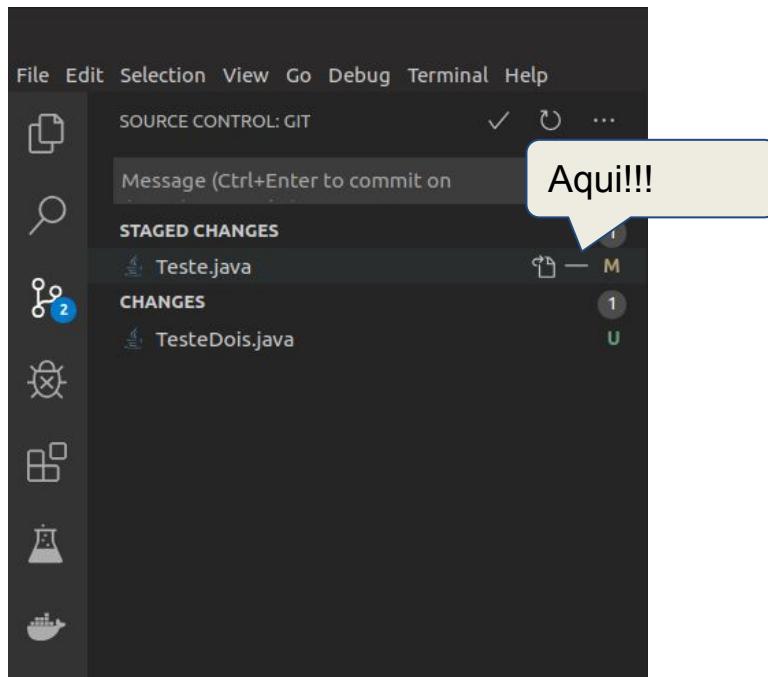
Após isso, o fonte adicionado ficará em *stage* e ao clicar no *certo*, será realizado o *commit* apenas desses fontes, e não será questionado nada.



Iniciando no Git

Commitando alterações

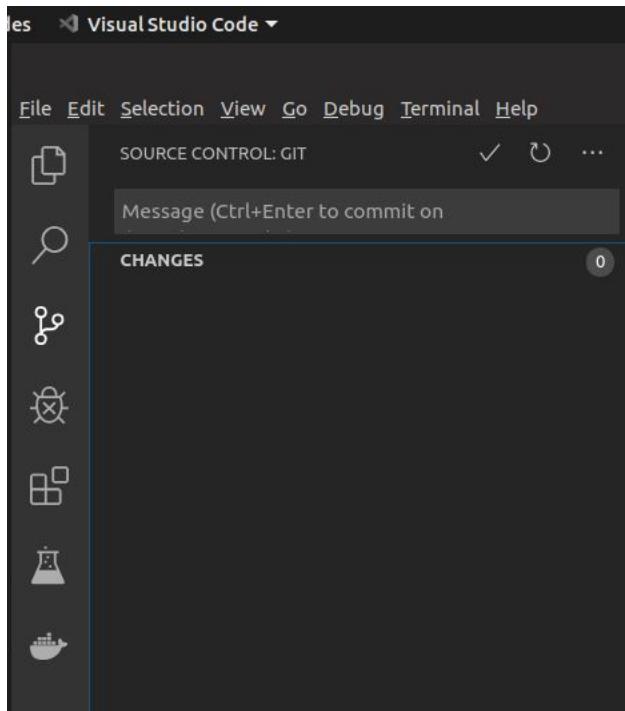
Caso tenha colocado o fonte errado em *stage*, basta removê-lo no botão de menos.



Iniciando no Git

Commitando alterações

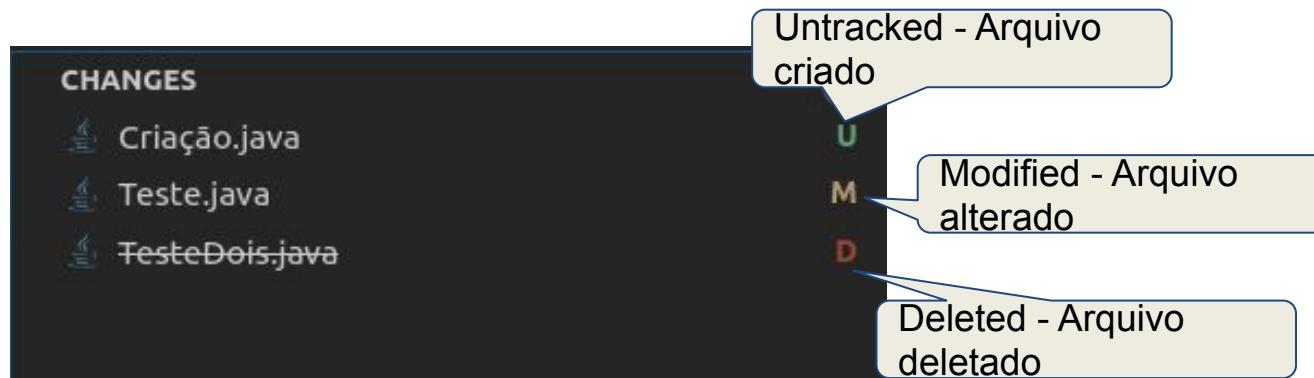
Após confirmar, não serão mais exibidas alterações na opção de Git e não serão mais apresentadas diferenças.



Iniciando no Git

Commitando alterações

Status

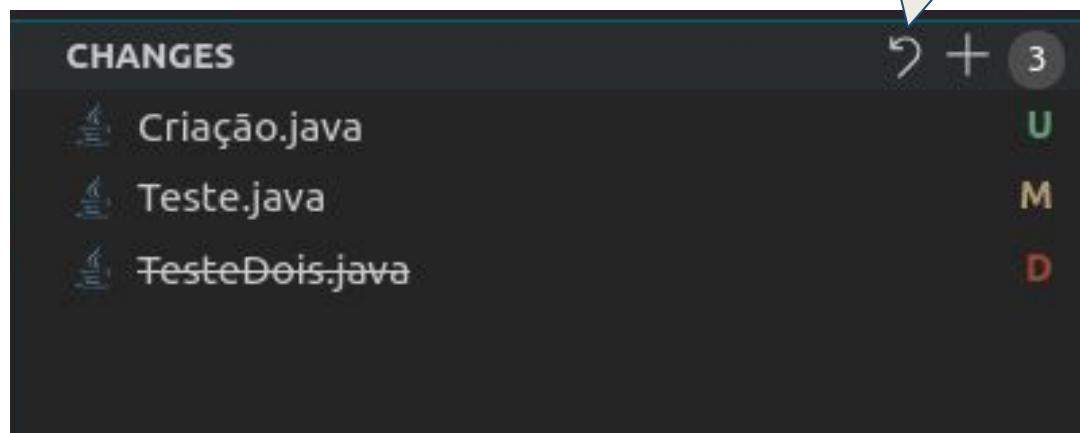


Iniciando no Git

Commitando alterações

Descartar alterações

Selecione esta opção para descartar todas as alterações, o selecione a mesma opção, mas sobre um fonte, para descartar as alterações deste fonte específico.



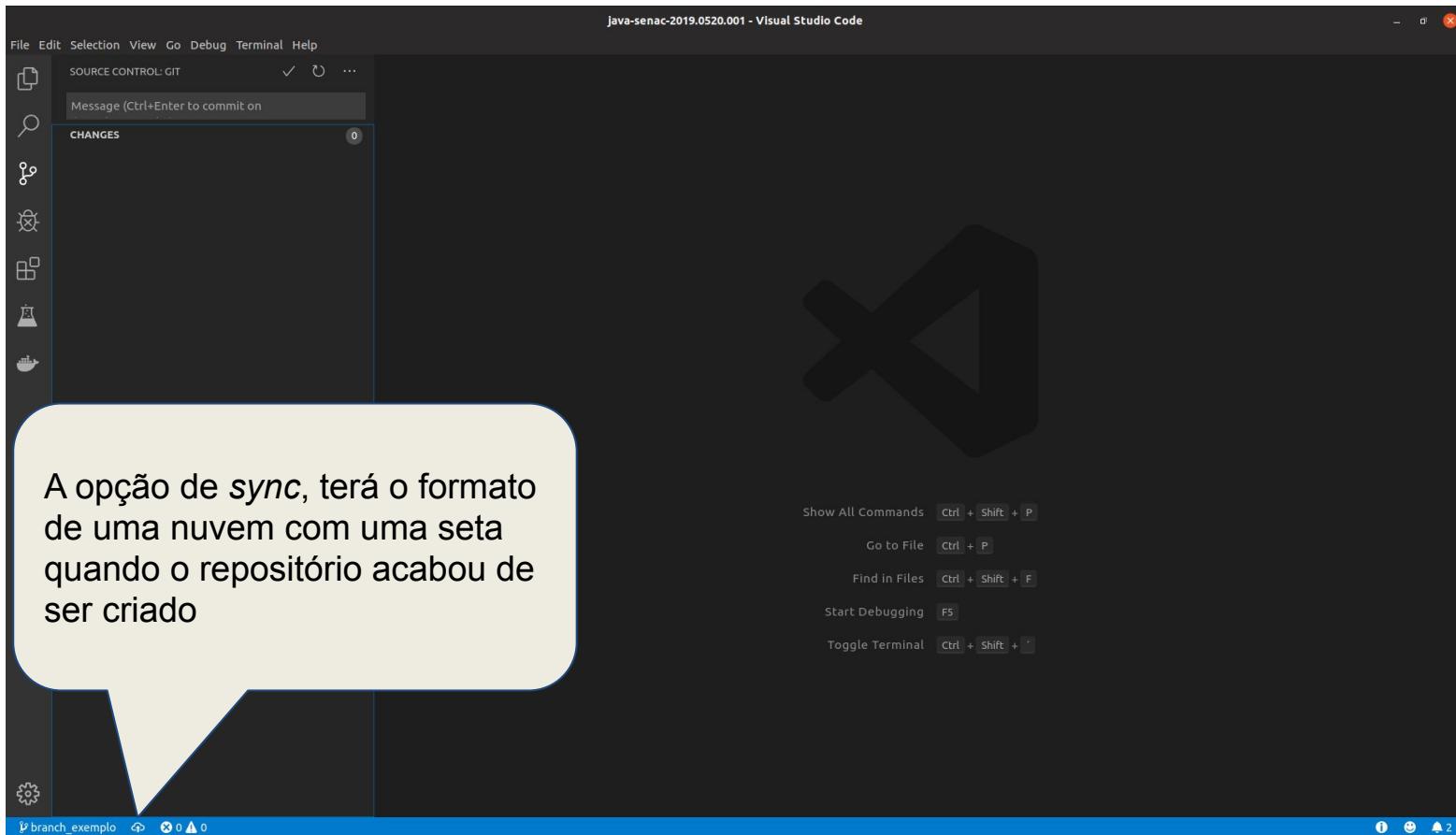
Iniciando no Git

Push para o repositório Remoto

Até este momento, todas as alterações estão no repositório local, ou seja, não estão sincronizadas com o repositório da nuvem (Github) para realizar a sincronização, é necessário selecionar a opção de sync. Para a sincronização, será solicitado usuário e senha do Github, na parte superior da tela.

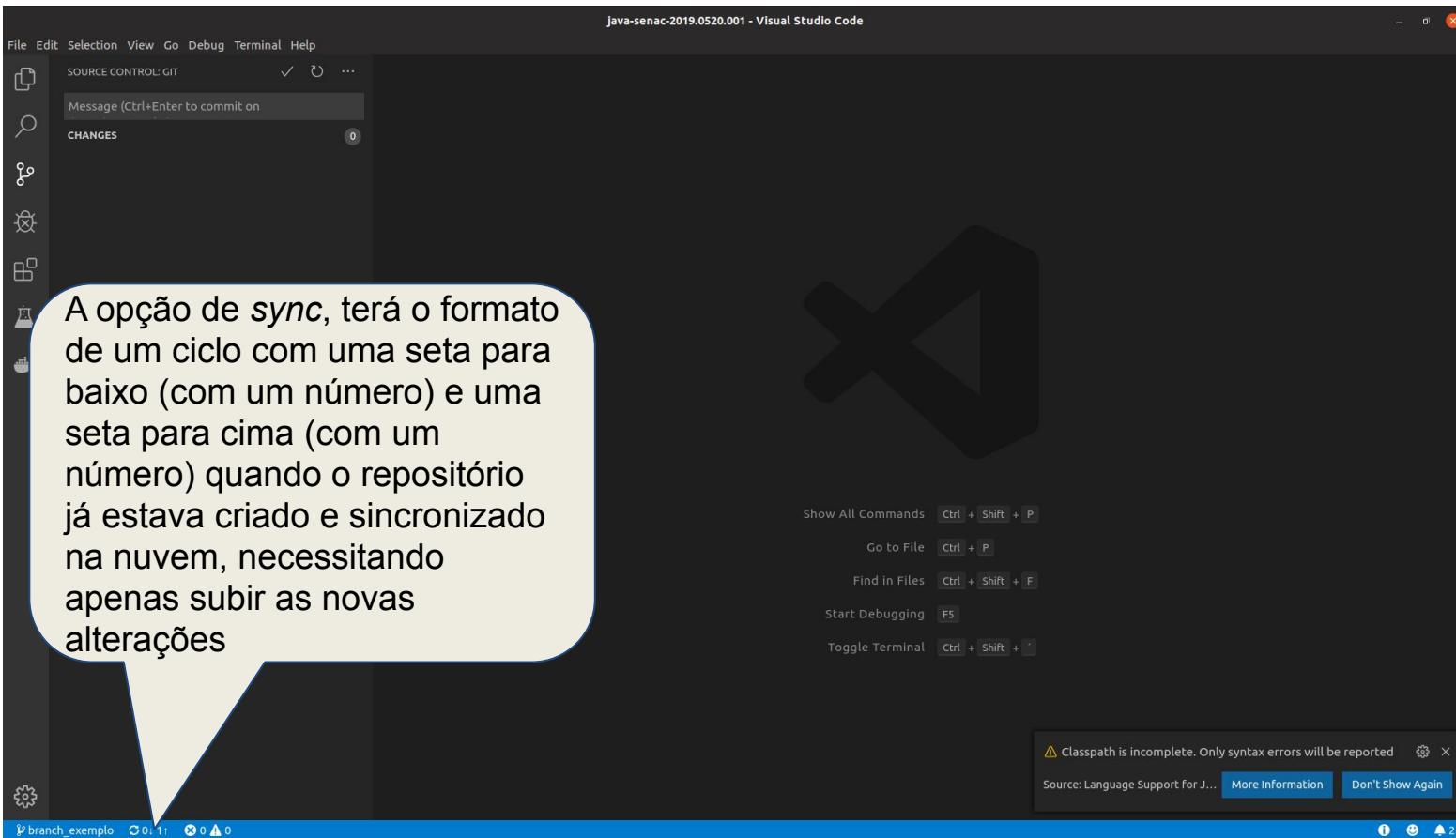
Iniciando no Git

Push para o repositório Remoto



Iniciando no Git

Push para o repositório Remoto



Iniciando no Git

Quando realizar o *Commit* e *Push*?

SEMPRE!

(Na verdade, sempre que julgar necessário, lembre-se
o *commit* é o *save* do jogo, quando quiser salvar só
realize o *commit* e o *push* é para jogar este *save* para
a nuvem)

Iniciando no Git

Liberando para a *master* - Criando Pull Request

Como não haverá *commit* na *master*, para liberarmos o fonte na versão oficial do projeto, utilizamos a ferramenta de *Pull Request*, que nos permite comparar o que foi feito com o *status* anterior do projeto.

Iniciando no Git

Liberando para a *master* - Criando Pull Request

Para isso, basta acessar as *branches* do projeto.

The screenshot shows a GitHub repository page for 'jacksjm / java-senac-2019.0520.001'. A callout box points to the '2 branches' link in the main statistics bar, with the text 'Acesse as branches aqui' (Access the branches here). The page displays a commit history with the following details:

File	Commit Message	Date
.gitignore	Criado fonte de hello world	9 days ago
README.md	Initial commit	9 days ago
Teste.java	Outra mensagem	9 days ago
jacksjm	Merge pull request #2 from jacksjm/create-exercises	2 days ago
exercicios	update list	2 days ago

Iniciando no Git

Liberando para a *master* - Criando Pull Request

Após, selecione a opção *new pull request*

The screenshot shows a GitHub repository page for 'jacksjm / java-senac-2019.0520.001'. The 'Code' tab is selected. At the top right, there are buttons for 'Unwatch' (1), 'Star' (0), and 'Fork' (1). Below the tabs, there are buttons for 'Overview' (selected), 'Yours', 'Active', 'Stale', and 'All branches'. A search bar for branches is also present. The 'Default branch' section shows the 'master' branch was updated 2 days ago by jacksjm. The 'Your branches' section lists 'branch_exemplo' (updated 16 minutes ago by Jackson Machado) and 'create-exercises' (updated 2 days ago by Jackson Machado). The 'Active branches' section also lists 'branch_exemplo' and 'create-exercises'. A callout bubble with the text 'Aqui!!!' points to the 'New pull request' button in the 'Your branches' section for the 'branch_exemplo' branch.

Iniciando no Git

Liberando para a *master* - Criando Pull Request

Caso acabou de sincronizar um *branch* irá apresentar uma mensagem na entrada do projeto, podendo selecionar a opção *Compare & pull request*.

The screenshot shows a GitHub repository page for 'jacksjm / java-senac-2019.0520.001'. The repository name is at the top left, followed by statistics: 1 unwatched, 0 stars, and 1 fork. Below the header are navigation links: Code, Issues (0), Pull requests (0), Projects (0), Wiki, Security, Insights, and Settings. The main content area displays the repository details: 'Exercícios Java para Turma SENAC 2019.0520.001', 'Edit', 'Manage topics', '6 commits', '2 branches', and '0 releases'. A callout bubble points to the 'Compare & pull request' button, which is highlighted with a green box. Below this, a section titled 'Your recently pushed branches:' shows 'branch_exemplo' (less than a minute ago). At the bottom, there's a commit list with the latest commit being a merge pull request from 'jacksjm/create-exercises' into 'master'. The commit message is 'Merge pull request #2 from jacksjm/create-exercises ...'. The commit list includes files like 'exercicios', '.gitignore', 'README.md', and 'Teste.java' with their respective commit messages and dates.

File	Commit Message	Date
exercicios	update list	2 days ago
.gitignore	Criado fonte de hello world	9 days ago
README.md	Initial commit	9 days ago
Teste.java	Outra mensagem	9 days ago

Iniciando no Git

Liberando para a *master* - Criando Pull Request

A screenshot of a GitHub pull request creation interface. The top bar shows the repository 'jacksjm / java-senac-2019.05'. Below it, there are tabs for 'Code', 'Issues 0', and 'Pull requests'. A button 'Open a pull request' is visible. The main area shows a comparison between 'base: master' and 'compare: branch_exemplo'. A green checkmark indicates 'Able to merge'. A large text input field contains the placeholder 'Commit de Teste'. To the right, there are sections for 'Assignees' (No one—assign yourself), 'Labels' (None yet), and 'Projects' (None). At the bottom, a green button says 'Create pull request'.

Automaticamente o PR será gerado da *branch* para a *master*

É possível também colocar um comentário a respeito da alteração também

Esta área demonstrará um resumo da alteração

É necessário informar uma descrição da alteração, se houver apenas um único *commit* a descrição deste será colocada automaticamente, se houverem mais, uma descrição genérica será adicionada

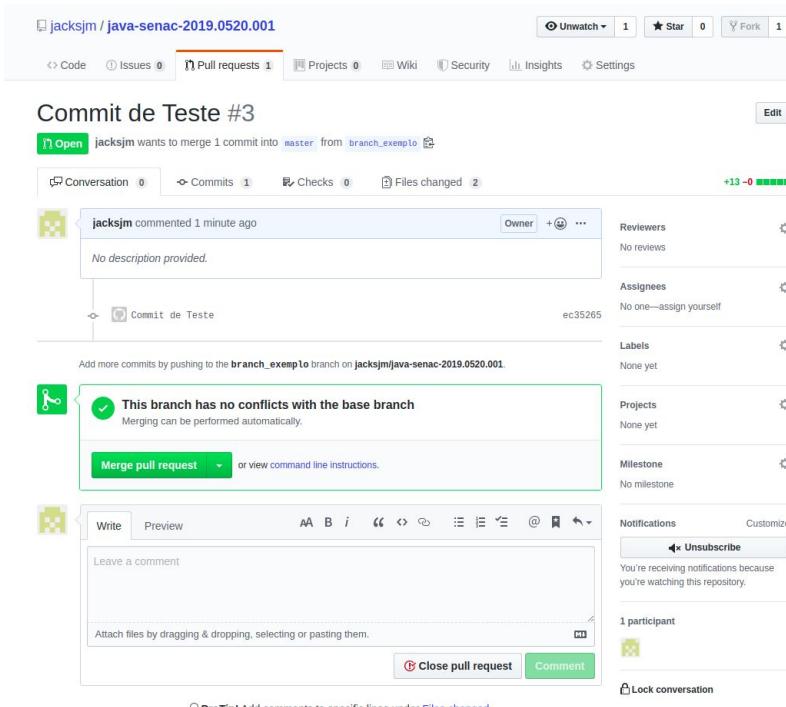
Para confirmar, basta clicar na opção *Create pull request*.

Esta área demonstrará a alteração realizada

Iniciando no Git

Liberando para a *master* - Criando Pull Request

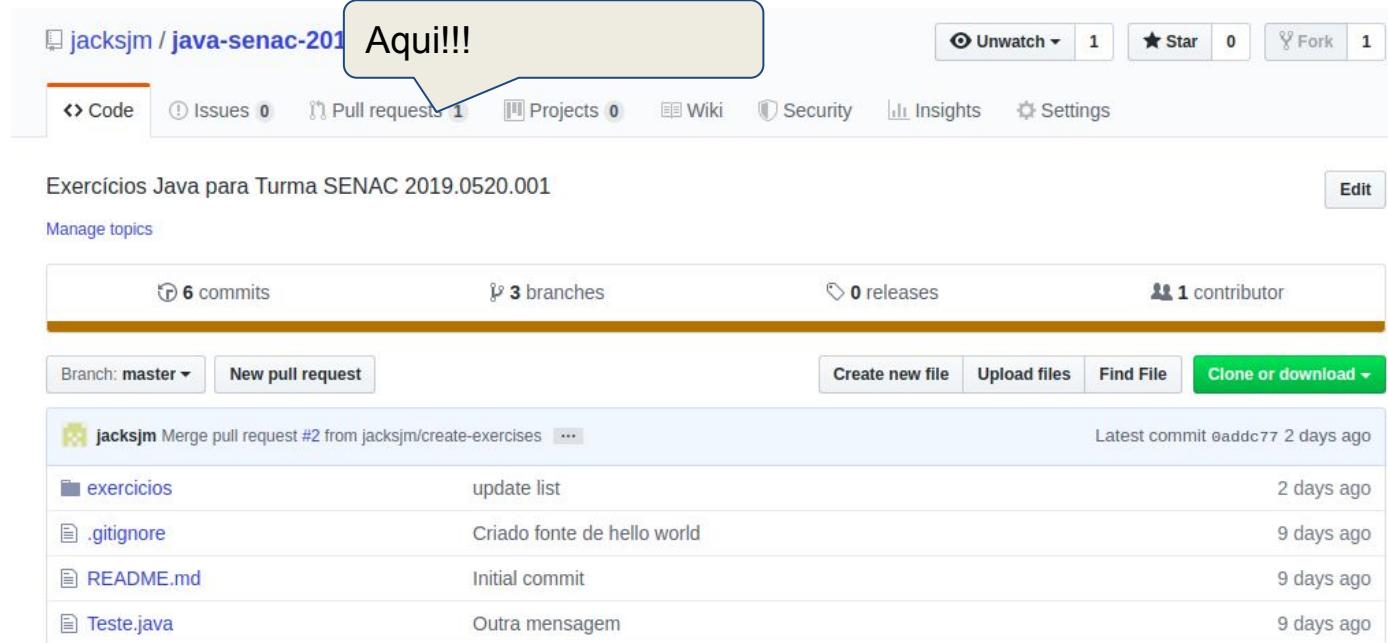
Após isso, será criado um *Pull Request* (PR) que estará disponível para revisão pela comunidade.



Iniciando no Git

Liberando para a *master* - Criando Pull Request

Para reacessar o PR, basta selecionar a opção *Pull request* no menu do projeto



A screenshot of a GitHub repository page for 'jacksjm / java-senac-201'. A callout bubble points to the 'Pull requests' tab in the navigation bar, which is highlighted in blue. The repository name 'jacksjm / java-senac-201' is at the top left. The navigation bar also includes 'Code', 'Issues 0', 'Projects 0', 'Wiki', 'Security', 'Insights', and 'Settings'. Below the navigation bar, the repository name 'Exercícios Java para Turma SENAC 2019.0520.001' is displayed, along with an 'Edit' button. A 'Manage topics' link is also present. A summary bar shows '6 commits', '3 branches', '0 releases', and '1 contributor'. The 'Branch: master' dropdown is set to 'master'. A green 'Clone or download' button is visible. The main content area lists recent commits:

Commit	Message	Date
jacksjm Merge pull request #2 from jacksjm/create-exercises	...	Latest commit eaddc77 2 days ago
exercicios	update list	2 days ago
.gitignore	Criado fonte de hello world	9 days ago
README.md	Initial commit	9 days ago
Teste.java	Outra mensagem	9 days ago

Iniciando no Git

Liberando para a *master* - Criando Pull Request

Será exibida uma lista dos PRs abertos

The screenshot shows a GitHub repository page for 'jacksjm / java-senac-2019.0520.001'. The 'Pull requests' tab is selected, showing 1 open pull request. A modal window titled 'Label issues and pull requests for new contributors' is displayed, explaining GitHub's help for first-time contributors. Below the modal, the pull request list includes a filter bar with 'is:pr is:open', a 'New pull request' button, and sorting options. One pull request is visible: '#3 Commit de Teste'.

Labels issues and pull requests for new contributors

Now, GitHub will help potential first-time contributors discover issues labeled with [help wanted](#) or [good first issue](#)

Dismiss

Filters is:pr is:open Labels 9 Milestones 0 New pull request

Clique no PR para acessar

#3 Commit de Teste #3 opened 5 minutes ago by jacksjm

Iniciando no Git

Quando criar o *Pull Request*?

Todo *pull request* é realizado ao terminar um *checkpoint* do projeto, ou seja, ao encerrar uma funcionalidade ou um ponto funcional do sistema, é possível criar um *pull request* para liberar ao mercado.

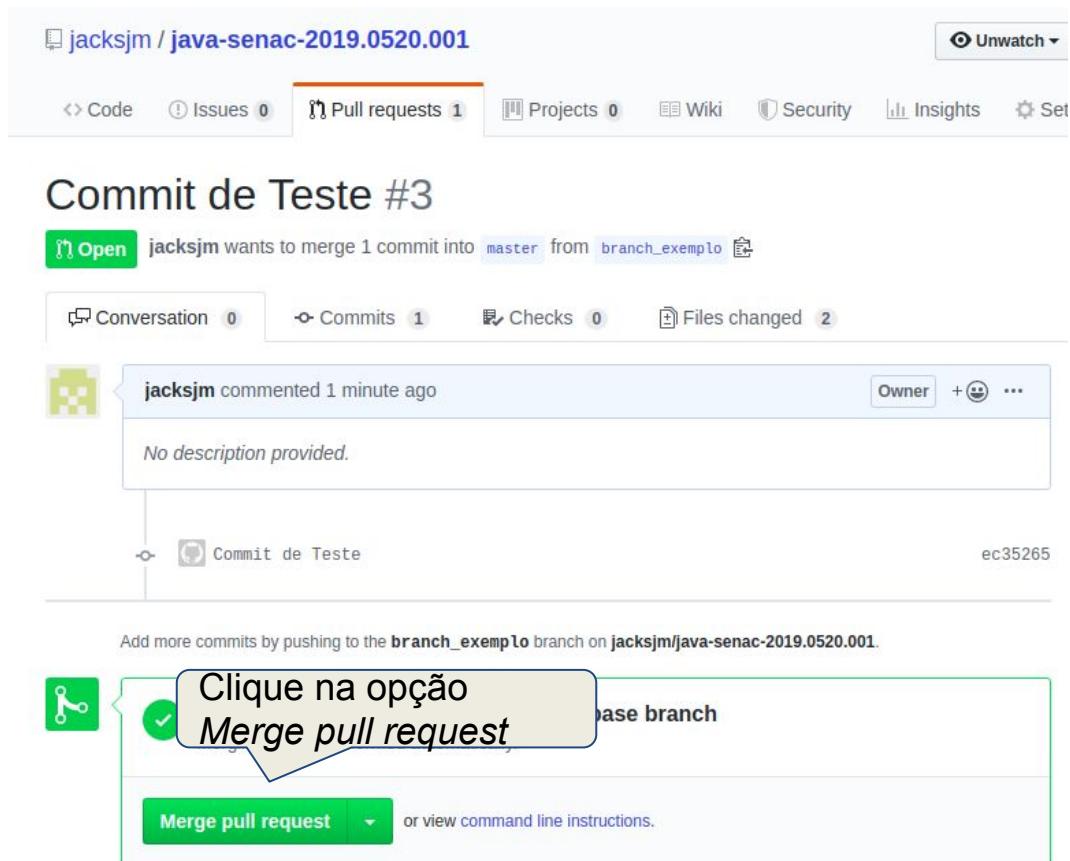
Iniciando no Git

Liberando para a *master* - *Merge* para a *master*

Ao se completarem as revisões, para que o código faça parte do produto oficial, é necessário realizar o *merge* do que foi criado com o que já estava na *master*.

Iniciando no Git

Liberando para a *master* - *Merge* para a *master*



Iniciando no Git

Liberando para a *master* - *Merge* para a *master*

A screenshot of a GitHub pull request merge dialog. The title bar shows the repository name "jacksjm / java-senac-2019.0520.001". The top navigation bar includes "Code", "Issues 0", "Pull requests 1" (which is highlighted), "Projects 0", "Wiki", "Security", "Insights", and "Settings". The main content area is titled "Commit de Teste #3" and shows a green "Open" button. Below it, a message states "jacksjm wants to merge 1 commit into master from branch_exemplo". The commit details show "Conversation 0", "Commits 1", "Checks 0", and "Files changed 2". The commit itself has a green icon and the text "jacksjm commented 8 minutes ago" followed by "No description provided.". To the right of the commit, there are sections for "Reviewers" (No reviews), "Assignees" (No one—assign yourself), "Labels" (None yet), "Projects" (None yet), "Milestone" (No milestone), and "Notifications" (Customize). A blue callout box points to the "No description provided." text with the instruction "Coloque uma descrição para o merge". Another blue callout box points to the "Confirm merge" button with the instruction "Coloque um comentário, se necessário". At the bottom left, there is a note "Add more commits by pushing to" followed by a "Merge pull request #3 from jacksjm/branch" button.

Iniciando no Git

Liberando para a *master* - *Merge* para a *master*

A partir deste momento, a *branch* deverá ser inativada, alterações.

É um bom momento para excluí-la e pode ser realizada pela operação *Delete branch*

The screenshot shows a GitHub repository page for 'jacksjm / java-senac-2019.0520.001'. The 'Pull requests' tab is selected, showing a single merged pull request. The commit message is 'Commit de Teste #3' and it includes the note 'No description provided.' A callout bubble points to the merge message with the text 'Aqui!!!'. Below the commit, a message states 'Pull request successfully merged and closed' and 'You're all set—the branch_exemplo branch can be safely deleted.' A 'Delete branch' button is visible.

Iniciando no Git

Master atual

Com isso, a *master* passa a estar com a versão mais atualizada dos fontes.

The screenshot shows a GitHub repository page for 'jacksjm / java-senac-2019.0520.001'. The repository name is at the top left, followed by statistics: 1 unwatched, 0 stars, and 1 fork. Below the header are navigation links: Code (highlighted), Issues (0), Pull requests (0), Projects (0), Wiki, Security, Insights, and Settings. The main content area displays the repository details: 'Exercícios Java para Turma SENAC 2019.0520.001' and 'Edit'. Below this are metrics: 8 commits, 2 branches, 0 releases, and 1 contributor. A 'Manage topics' link is also present. At the bottom, there are buttons for Branch: master (selected), New pull request, Create new file, Upload files, Find File, and Clone or download. The commit history table lists the following commits:

Author	Commit Message	Time Ago
jacksjm	Merge pull request #3 from jacksjm/branch_exemplo ...	Latest commit d56c91c 3 minutes ago
exercicios	update list	2 days ago
.gitignore	Criado fonte de hello world	9 days ago
	Initial commit	9 days ago
	Commit de Teste	37 minutes ago
	Commit de Teste	37 minutes ago
	Fonte criado agora	(This commit is highlighted with a blue callout box)
TesteDois.java		

Iniciando no Git

Dicas

- Crie o projeto em um local de fácil acesso;
- Abra o VsCode sempre no projeto criado;
- NUNCA DE *COMMIT NA MASTER*;
- Há várias outras funcionalidades no Github, os passos aqui contidos são os mais simples para criação de projeto, seguindo eles criteriosamente serão apresentadas as mesmas telas e processos

Iniciando no Git

Dicas

- Lembre-se, você estará trabalhando com dois repositórios, um local e um remoto, é necessário que haja sempre um *sync* entre os dois (*git pull* e *git push*)
- Inicie sempre o projeto com a linguagem (para criar o *.gitignore*) e com o README, pois desta forma, o repositório será criado já com a versão master
- É uma boa prática que o README contenha informações significativas sobre o projeto