

FIAP GRADUAÇÃO

DISCIPLINA: COMPLIANCE & QUALITY ASSURANCE

AULA:

4 – EXEMPLIFICAÇÃO DO IMPACTO DE FERRAMENTAS NA QUALIDADE - GIT

PROFESSOR:

RENATO JARDIM PARDOCCI

PROFRENATO.PARDOCCI@FIAP.COM.BR

AGENDA DA AULA

- ✓ Gestão de conteúdo e versões



VAMOS REFORÇAR OS CONCEITOS DA GOVERNANÇA COM UM JOGO!

Introdução à Governança

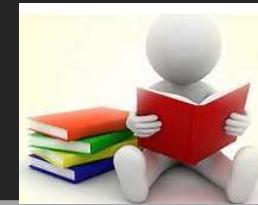
Kahoot!

by RenatoJP 13 seconds ago

Quiz 13 questions Private

GESTÃO DE CONTEÚDO E VERSIONAMENTO

ESTUDO DE CASO SIMULADO



Para atender a Governança de TI na empresa de Dilan e melhorar a condução dos projetos de software, você escolheu criar um processo de controle de versões de documentos e fontes de projetos e selecionou a ferramenta (recurso) GIT para implementar essa mudança.

Lembrando o nosso problema... a GD (Gerência de Desenvolvimento) da empresa de Dilan está sofrendo bastante com o controle de fontes de aplicação e de documentação de projeto.

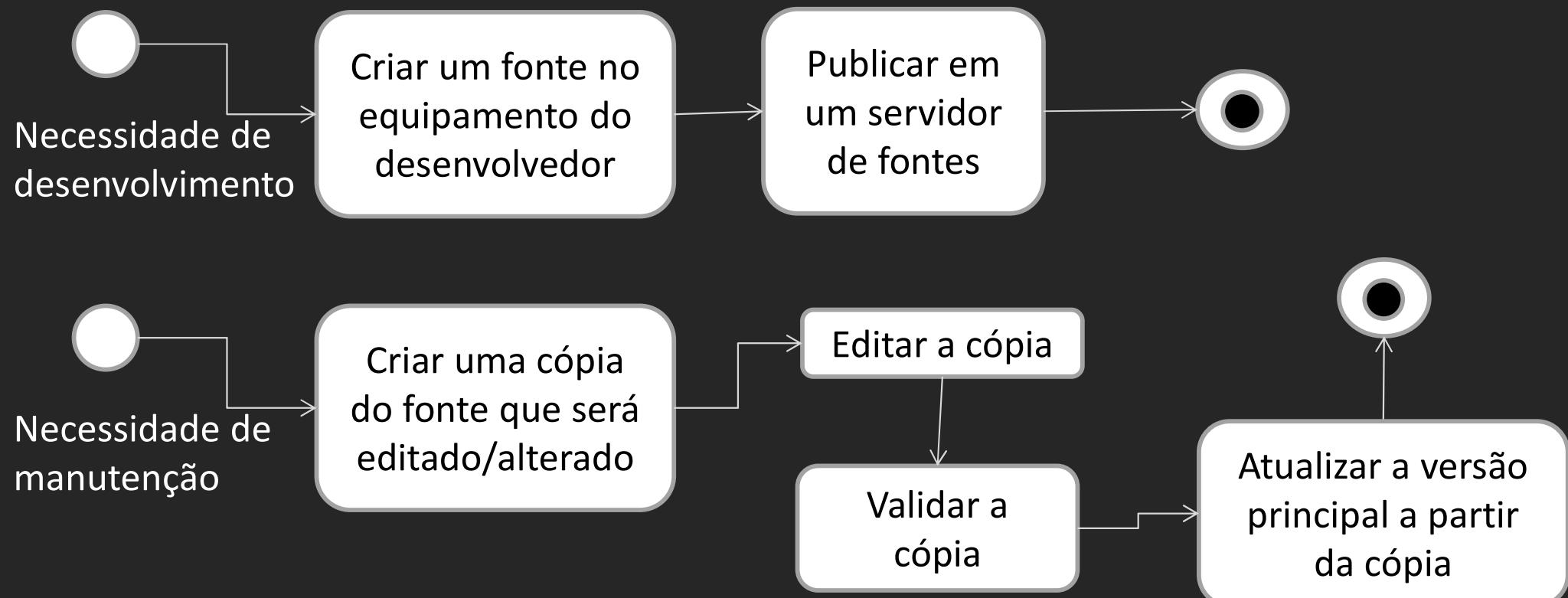
Cada grupo de projeto acaba definindo uma forma de trabalhar é não é incomum a sobreposição de fontes de forma errada, ou atualização de fonte em versão que não a última, gerando problemas em builds e releases.

Dilan gostaria que a sua empresa tivesse uma solução única para controlar fonte de programas e documentos, que permitisse a colaboração de desenvolvedores sobre um mesmo material, sem perder controle sobre a última versão oficial.

Um processo foi desenhado por Consuelo e ela preparou um treinamento sobre GIT que você vai realizar agora!

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Vamos trabalhar com a seguinte proposta de processo de gestão de mudança no desenvolvimento de software:



ASSISTA OS VÍDEOS NO CANAL DO PROFESSOR



<https://youtu.be/t1E-cbB4gFY>

A screenshot of a Windows File Explorer window. The left sidebar shows 'Acesso rápido' with icons for Área de Trabalho, Google Drive, Downloads, Documentos, OneDrive, Imagens, Material do prof, and Media. Below this is 'Área de Trabalho' with icons for RunMIDIA, Este Computador, Área de Treball, Documentos, Downloads, Imagens, Músicas, and Arquivos. The main pane displays a folder structure under 'RunMIDIA > Google Drive'. It lists three items: 'Metodologia' (modified 02/02/2016 11:56, type: Pasta de arquivos), 'Docencia (I)' (modified 14/03/2016 21:06, type: Pasta de arquivos), and 'Recibos 2016' (modified 11/07/2016 16:41, type: Pasta de arquivos). To the right of the file list is a video player window showing a man with a headset, likely the professor, speaking. The video player has standard controls (play/pause, volume, etc.) at the bottom.

Conteúdo didático complementar - Controle de Versão de Documentos



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Em seguida, vamos vamos adotar uma solução na nuvem como ferramenta para administrar os fontes – o GIT e GIT HUB:



Existem outras soluções, hoje menos populares que o GIT mas que funcionam bem na gestão de fontes e versões...



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Através da ferramenta de controle de versões de códigos e/ou documentos de software, é possível evitar os problemas de se trabalhar com a fonte errada em um determinado ponto do projeto, facilitando a colaboração no projeto.



Check out para fazer a manutenção

Check in após a manutenção



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Trabalhando com GIT, os arquivos fontes serão organizados em:

Cópia Master

Contém os arquivos na versão estável, que podem ser usados por outros desenvolvedores na integração de componentes ou com outros sistemas, ou podem ser usados para gerar um pacote de versão final do produto.

Cópia Branch

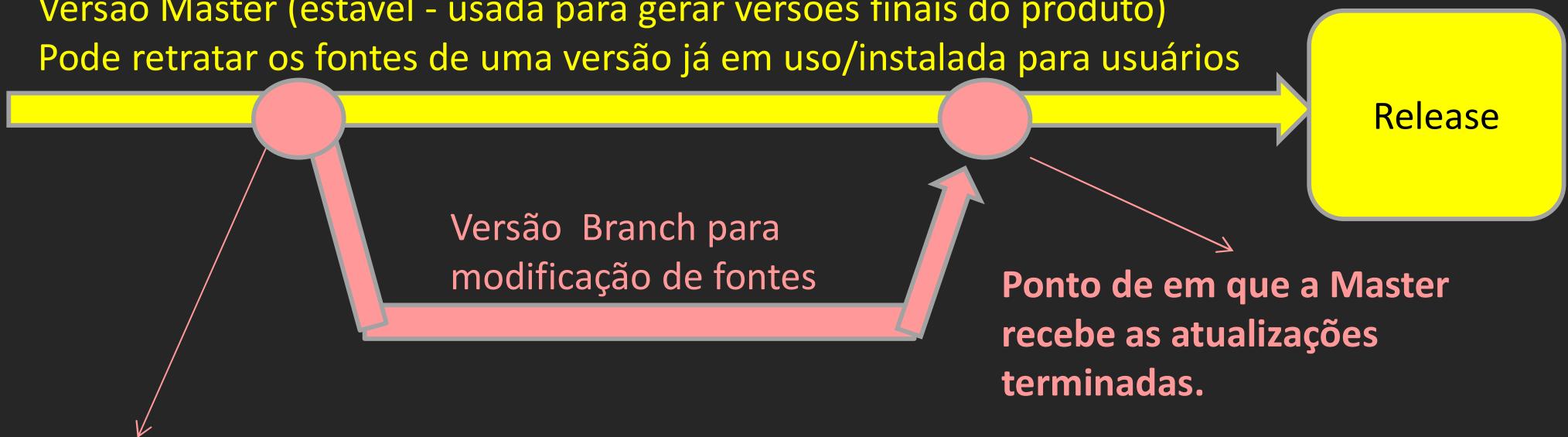
Contém os arquivos em uma versão de manutenção/atualização que não estão estáveis e não podem ser usados para gerar uma versão final do produto.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

O funcionamento da Branch.

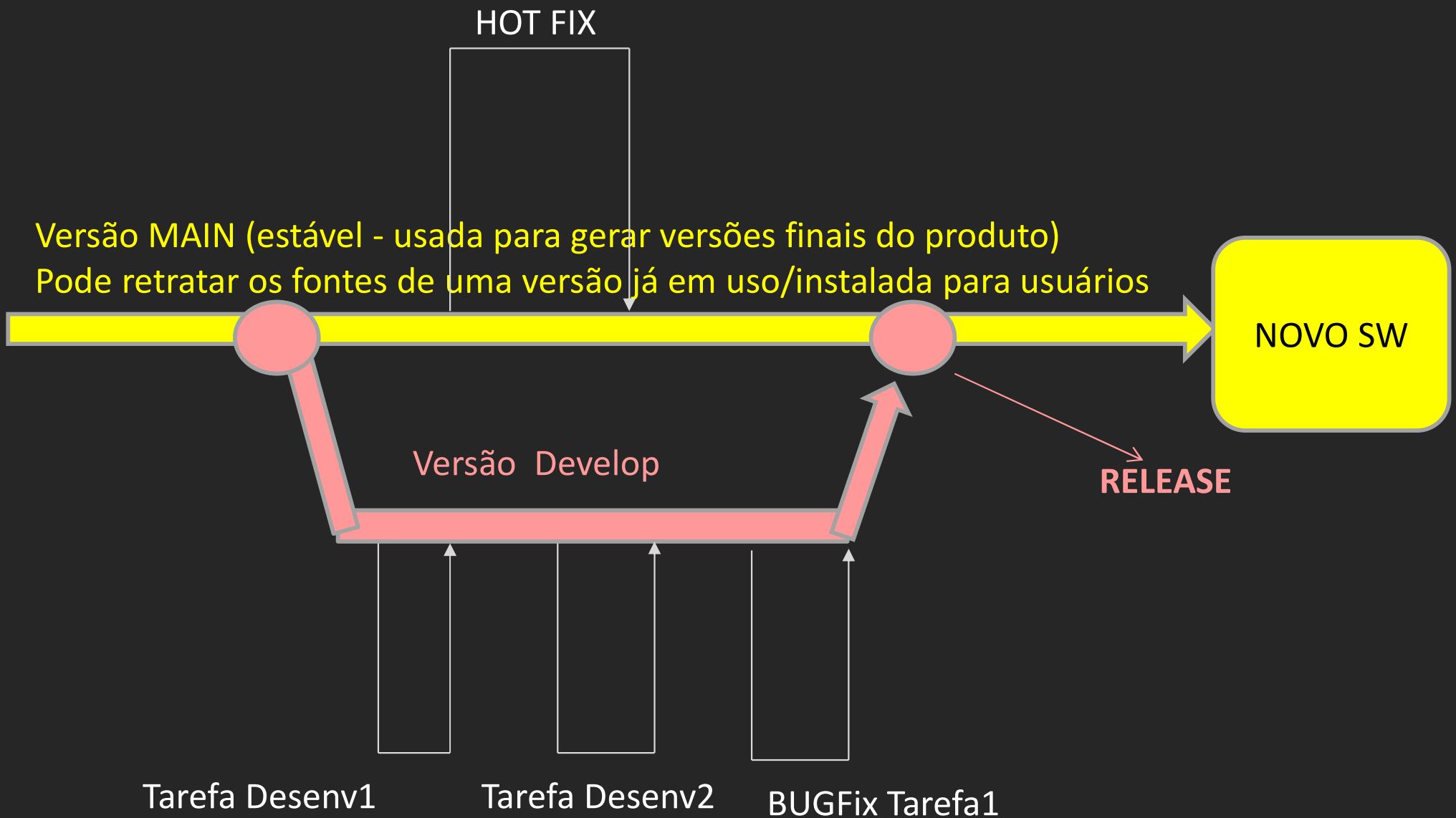
Versão Master (estável - usada para gerar versões finais do produto)

Pode retratar os fontes de uma versão já em uso/instalada para usuários



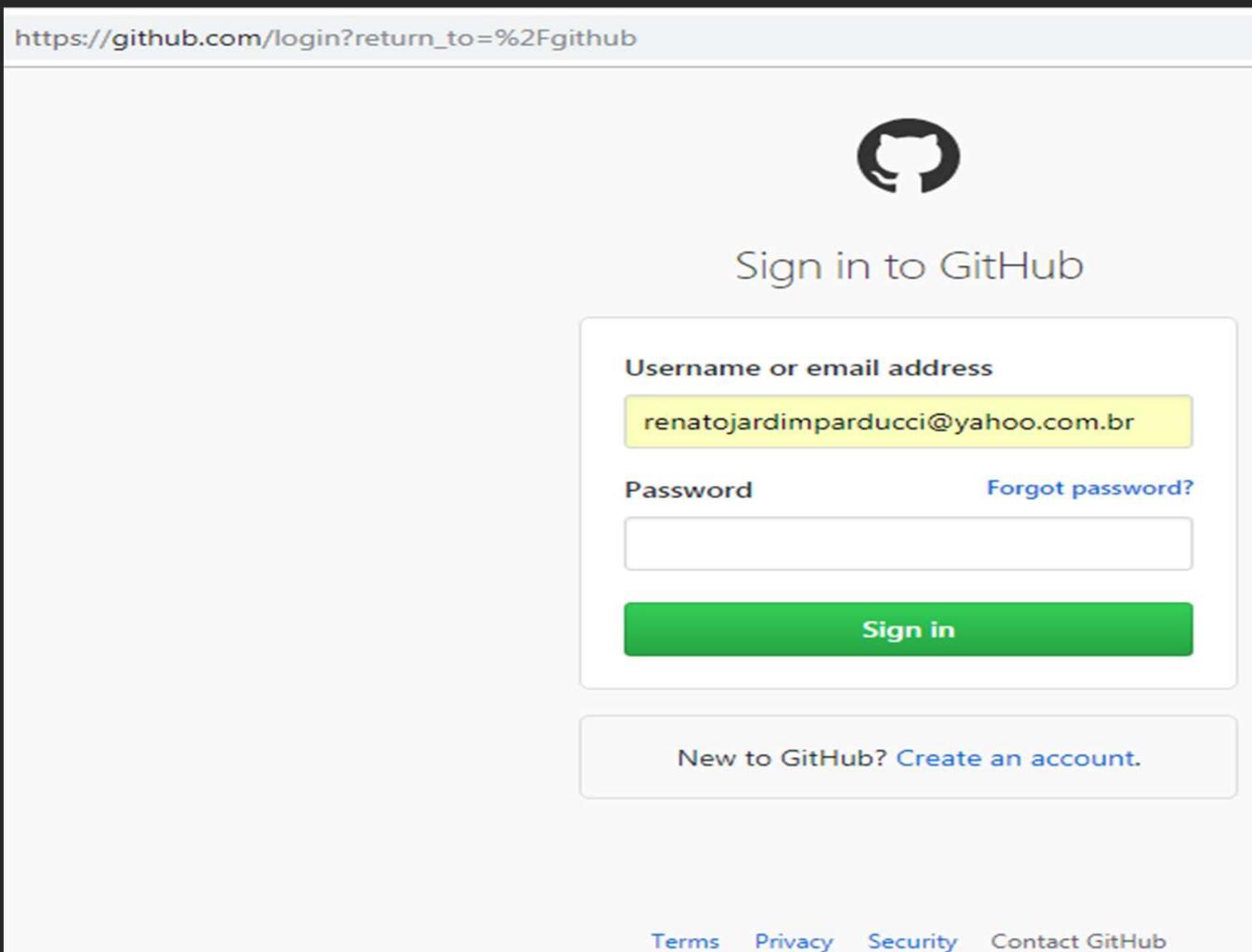
Ponto de necessidade de manutenção para:

- Corrigir o software/eliminar BUGs (manutenção CORRETIVA);
- Adaptar o software para novas regras de negócio (manutenção ADAPTATIVA);
- Prevenir contra possíveis problemas futuros (manutenção PREVENTIVA);
- Alcançar a perfeição na experiência do usuário (manutenção PREFECTIVA)



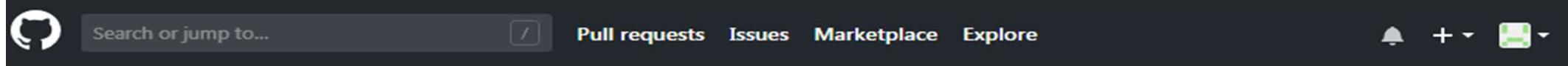
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Crie sua conta no GIT HUB.



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Faça o login para ver se está tudo Ok.



The GitHub homepage displays six pinned repositories:

- fetch**: A window.fetch JavaScript polyfill. (JavaScript, 20.8k stars, 1.9k forks)
- hub**: hub helps you win at git. (Go, 14.1k stars, 1.4k forks)
- training-kit**: Open source on demand courses and cheat sheets for Git and GitHub. (HTML, 1.9k stars, 1.9k forks)
- choosealicense.com**: A site to provide non-judgmental guidance on choosing a license for your open source project. (Ruby, 1.4k stars, 457 forks)
- scientist**: A Ruby library for carefully refactoring critical paths. (Ruby, 4.7k stars, 200 forks)
- gh-ost**: GitHub's Online Schema Migrations for MySQL. (Go, 5.9k stars, 453 forks)

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Logado no GITHUB, acesse a área de repositórios.

The screenshot shows a GitHub profile page for a user named RenatoJardimParducci. The top navigation bar includes links for Search or jump to..., Pull requests, Issues, Marketplace, and Explore. On the right, there's a user dropdown menu with options like Signed in as, Your profile, Your repositories (which is highlighted with a red box and a red arrow pointing to it), Your stars, Your gists, Help, Settings, and Sign out. The main content area displays the GitHub logo, the tagline "How people build software.", and the user's location as San Francisco, CA. It also shows contact information: https://github.com/about, support@github.com, and a Verified badge. Below this, there are three pinned repository cards: "fetch", "hub", and "training-kit".

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Seus repositórios de projetos de software aparecerão, caso você já tenha catalogado algum.

The screenshot shows a GitHub profile interface. At the top, there is a blue banner with the text "ProTip! Updating your profile with your name, location, and a profile picture helps other GitHub users get to know you." and a green "Edit profile" button. Below the banner, the profile navigation bar includes "Overview", "Repositories 12", "Stars 0", "Followers 0", and "Following 0". The "Repositories" tab is currently selected, indicated by an orange underline. Below the navigation bar is a search bar with the placeholder "Find a repository..." and filters for "Type: All" and "Language: All", along with a green "New" button. The main content area displays three repository cards: "1TDSJ" (updated on 26 Apr 2017), "RepExemplo" (updated on 17 Jan 2017), and "ExemploTDSS". Each card has a horizontal green progress bar underneath it.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Seus repositórios de projetos de software aparecerão, caso você já tenha catalogado algum.

The screenshot shows a GitHub profile interface. At the top, there is a blue banner with the text "ProTip! Updating your profile with your name, location, and a profile picture helps other GitHub users get to know you." and a green "Edit profile" button. Below the banner, the profile navigation bar includes "Overview", "Repositories 12", "Stars 0", "Followers 0", and "Following 0". The "Repositories" tab is currently selected, indicated by an orange underline. Below the navigation bar is a search bar with the placeholder "Find a repository..." and filters for "Type: All" and "Language: All", along with a green "New" button. The main content area displays three repository cards: "1TDSJ" (updated on 26 Apr 2017), "RepExemplo" (updated on 17 Jan 2017), and "ExemploTDSS". Each card has a horizontal green progress bar underneath it.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Crie um repositório.

Create a new repository

A repository contains all the files for your project, including the revision history.

Owner RenatoJardimParducci / Repository name TesteGITHUB ✓

Great repository names are short and memorable. Need inspiration? How about sturdy-spoon.

Description (optional)

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

 Private You choose who can see and commit to this repository.

Initialize this repository with a README This will let you immediately clone the repository to your computer. Skip this step if you're importing an existing repository.

Add .gitignore: None Add a license: None ⓘ

Create repository

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Entenda a tela do GitHub.

The screenshot shows a GitHub repository page for 'RenatoJardimParducci / TesteGITHUB'. The top navigation bar includes 'Unwatch' (1), 'Star' (0), and 'Fork' (0). Below the navigation are tabs for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Insights', and 'Settings'. A message 'No description, website, or topics provided.' is displayed, along with an 'Edit' button. A 'Manage topics' link is present. Key statistics are shown: 1 commit, 1 branch, 0 releases, and 1 contributor. A red box highlights the 'Branch: master ▾' dropdown (1) and the 'Upload files' button (3). Red arrows point from numbered callouts to these specific elements. The repository contains one file, 'README.md', which is also highlighted with a red box (2). The file details show it was created by 'RenatoJardimParducci' and is an 'Initial commit' made a minute ago. A red arrow points from callout 2 to the file name 'README.md' in the commit list. The repository title 'TesteGITHUB' is at the bottom left.

- 1 Seleciona a área/cópia de fontes para trabalho
- 2 Nomes dos arquivos que constam na área
- 3 Usado para carregar arquivos fonte no GIT

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Você pode criar pastas para separar tipos de arquivos

The screenshot shows a GitHub repository page for 'RenatoJardimParducci / TesteGITHUB'. The page includes navigation tabs for Code, Issues (0), Pull requests (0), Projects (0), Wiki, Insights, and Settings. A red callout box with the text 'Crie um nome para a pasta/nomde de um arquivo Readme que será criado' points to the 'Create new file' button in the main repository actions area. Below this, there are sections for commits, branches, releases, contributors, and file management options like New pull request, Upload files, Find file, and Clone or download.

RenatoJardimParducci / TesteGITHUB

No description, website, or topics provided.

Manage topics

1 commit | 1 branch | 0 releases | 1 contributor

Branch: master ▾ | New pull request | Create new file | Upload files | Find file | Clone or download ▾

RenatoJardimParducci Initial commit | Latest commit 7296d6c a minute ago

README.md | Initial commit | a minute ago

README.md | Edit

Crie um nome para a pasta/nomde de um arquivo Readme que será criado

TesteGITHUB

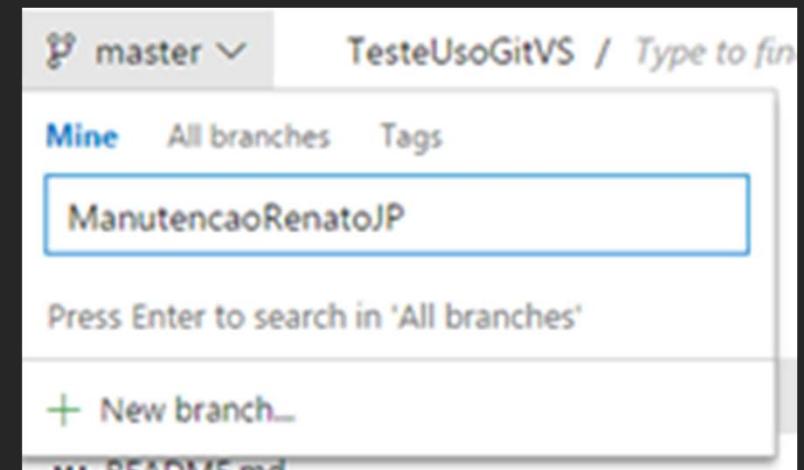
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Para incluir novos fontes ou alterar fontes publicados no GIT, crie uma Branch (cópia de desvio) da área Master.

A Master deve conter apenas os códigos fonte estáveis, que podem ser usados por outros desenvolvedores.

A Branch é uma réplica da Master para que um ou mais programadores façam alterações nos programas e depois republiquem os arquivos atualizados na Master.

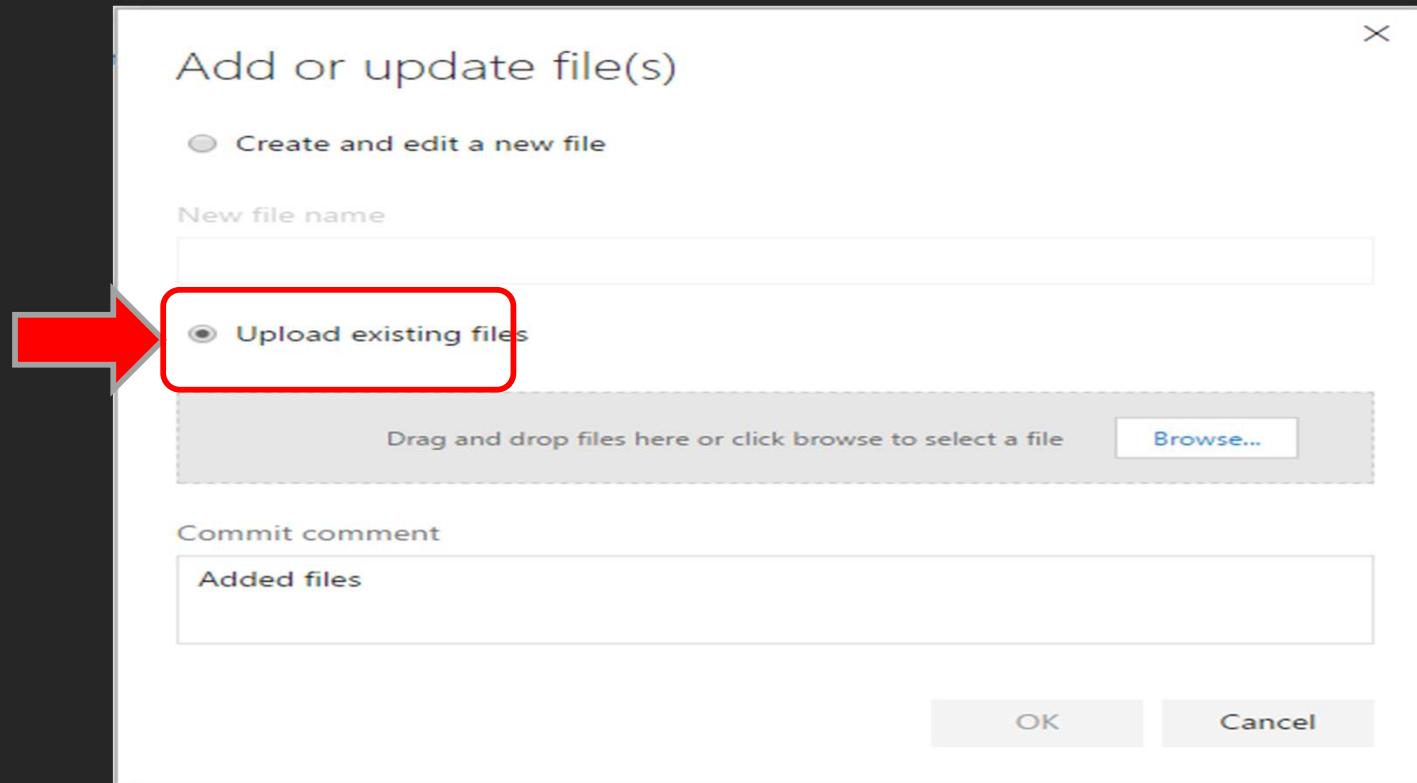
Crie uma Branch a partir da Master com o nome que desejar.



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Acesse o link de Upload para subir para o GitHub um arquivo do seu computador.

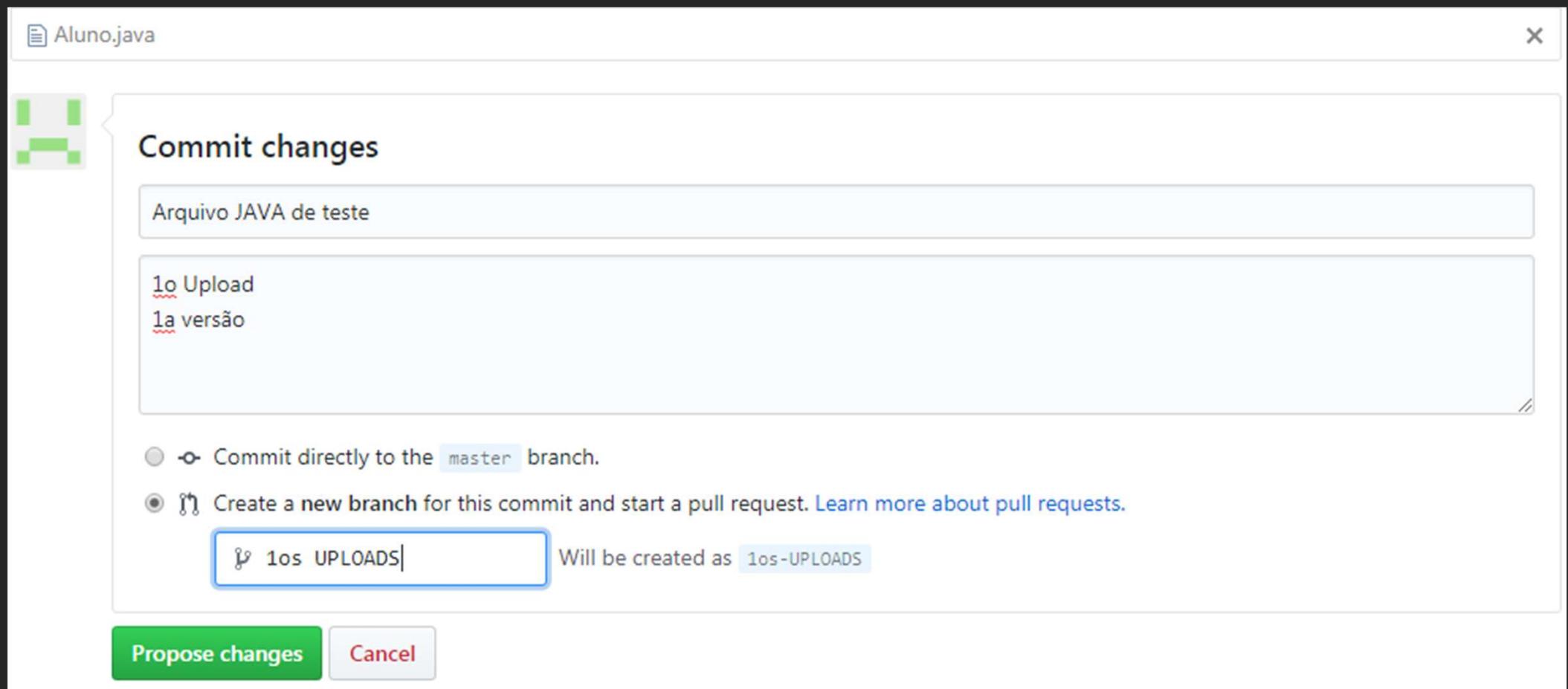
Suba um arquivo .JAVA ou .SQL para experimentar!



Como alternativa, você pode abrir a pasta com o seu arquivo no Windows Explorer e arrastá-lo para a página do GITHUB.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Cada inclusão ou alteração de arquivo pode ser comentada ao ser salva, facilitando a interpretação das versões.!



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Quando algo é modificado na área de trabalho/cópia, fica habilitada a possibilidade de criar uma Pull request para atualizar a Master a partir da Branch.!

The screenshot shows a GitHub repository interface. At the top, there's a header with the text "Your recently pushed branches:" followed by a yellow box containing "1 los-UPLOADS (1 minute ago)". To the right of this box is a green button with a white icon and the text "Compare & pull request". A large red arrow points from the text above down to this button. Below the yellow box, there are buttons for "Branch: master ▾", "New pull request", "Create new file", "Upload files", "Find file", and "Clone or download ▾". At the bottom, there's a commit history section showing "RenatoJardimParducci Initial commit" and "Latest commit 7296d6c 34 minutes ago".

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Depois que estiver certo que a mudança está completa e correta, publique a modificação na cópia Master, tornando-a disponível para todos os desenvolvedores usarem!

The screenshot shows a GitHub interface for a repository named 'TesteUsoGitVS'. The 'Pull Requests' tab is selected. A pull request titled 'Updated CalcHoras.js' is displayed. The 'Description' section contains a bullet point: '- Updated CalcHoras.js'. Below the description, it says 'Markdown supported. Use # to mention a work item or @ to mention a person.' and lists another bullet point: '• Updated CalcHoras.js'. The 'Reviewers' section shows '[ProjetoExemplo-2TBA-2016]\ProjetoExemplo-2TBA-2016 Team' and a search bar. The 'Work Items' section has a search bar with placeholder text 'Search work items by ID or title'. At the bottom left, a large red arrow points to a blue button labeled 'New pull request'.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Experimente criar uma nova Branch para fazer alterações.!

Faça as modificações e execute o Commit.

Crie uma Pull request e execute a atualização da Master (Merge).

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

A Master terá o histórico de todas as modificações feitas.

The screenshot shows a GitHub commit history for a file named 'Aluno.java' in a repository named 'TesteGITHUB'. The commit was made by 'RenatoJardimParducci' at 9 minutes ago with the message 'Update Aluno.java'. The commit hash is 'e92ad94'. A red arrow points from the text above to the 'History' button in the bottom right corner of the commit card. The code listing shows a single class definition:

```
1  public class Aluno1 extends Pessoa {  
2  
3      private Matricula matricula;  
4  
5      public void SolicitarMatricula() {  
6  
7          }  
8  
9      }
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

A Master terá o histórico de todas as modificações feitas.

The screenshot shows a GitHub repository page for 'RenatoJardimParducci / TesteGITHUB'. The top navigation bar includes 'Unwatch' (1), 'Star' (0), 'Fork' (0), and tabs for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Insights', and 'Settings'. A dropdown menu indicates the current branch is 'master'. The main content area displays a list of commits made on November 30, 2018:

- Merge pull request #1 from RenatoJardimParducci/1os-UPLOADS ...
RenatoJardimParducci committed 10 minutes ago
Verified | ba7d3c0 | ↗
- Update Aluno.java
RenatoJardimParducci committed 14 minutes ago
Verified | e92ad94 | ↗
- Arquivo JAVA de teste ...
RenatoJardimParducci committed 23 minutes ago
Verified | 55ecc9c | ↗
- Initial commit
RenatoJardimParducci committed an hour ago
Verified | 7296d6c | ↗

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Caso você precise recuperar uma versão anterior, basta selecioná-la.

The screenshot shows a GitHub repository page for 'RenatoJardimParducci / TesteGITHUB'. The 'Code' tab is selected. A tooltip 'Browse the repository at this point in the history' is displayed over the commit list. The commits are:

- Merge pull request #1 from RenatoJardimParducci/1os-UPLOADS ...
RenatoJardimParducci committed 9 minutes ago
- Update Aluno.java
RenatoJardimParducci committed 12 minutes ago
- Arquivo JAVA de teste ...
RenatoJardimParducci committed 21 minutes ago
- Initial commit
RenatoJardimParducci committed 44 minutes ago

At the bottom of the commit list, there is a link 'Browse the repository at this point in the history'.

Below the browser window, the taskbar shows several open applications: SourceTreeSetup-....exe, Git-2.19.2-32-bit (2).exe, and Git-2.19.2-32-bit.exe. The system tray shows the date and time as 11:43 30/11/2018.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Quando selecionada uma versão do fonte, o GIT mostra o que foi alterado para você ter certeza de qual versão está vendo.

The screenshot shows a GitHub commit history for the file `Aluno.java`. The commit was made by RenatoJardimParducci 12 minutes ago and is verified. It has one parent commit `55ecc9c` and a commit hash `e92ad942ca6b8d544ad4282d95dfcde8d1121567`. The commit message is not visible. The UI indicates "Showing 1 changed file with 1 addition and 1 deletion." The diff view shows the following changes:

```
2 2 Aluno.java
...
... ... @@ -1,4 +1,4 @@
 1 - public class Aluno extends Pessoa {
 1 + public class Aluno1 extends Pessoa {
 2 2
 3 3     private Matricula matricula;
 4 4
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Se precisar deletar uma Branch que ficou pra trás (não deletada no momento do MERGE), acesse na página principal do projeto, o número de branches e na janela que aparecer, clique na lata de lixo para eliminar a branch correspondente.

The screenshot shows a GitHub repository page for 'RenatoJardimParducci / TesteGITHUB'. The top navigation bar includes 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', and 'Wiki'. Below the navigation, it says 'No description, website, or topics provided.' and has a 'Manage topics' link. On the left, there's a sidebar with '8 commits' and a 'New pull request' button. The main content area shows a summary: '2 branches' (with a red box and arrow pointing to the trash icon next to one of the branch names). Below this, under 'Default branch', is 'master' (Updated a minute ago by RenatoJardimParducci) with a 'Default' button and a 'Change default branch' link. Under 'Your branches', there's a single entry: 'SUPERBRANCH-01' (Updated 2 minutes ago by RenatoJardimParducci) with a status of '#3 Merged'. Under 'Active branches', there's another entry: 'SUPERBRANCH-01' (Updated 2 minutes ago by RenatoJardimParducci) with a status of '#3 Merged'. At the bottom right, there are 'Unwatch', 'Star 0', 'Fork 0', and a 'Search branches...' input field.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Você acabou de estudar um processo com atividades, definição de responsabilidades e ferramentas para gerenciar fontes de programas de aplicação em suas versões.

Seu 1º passo para garantir Qualidade em projetos de software e a Governança, através da garantia da continuidade dos negócios.

ASSISTA OS VÍDEOS NO CANAL DO PROFESSOR



<https://youtu.be/MYhIM0bk9aQ>

A screenshot of a video player displaying a GitHub tutorial. The video frame shows a man with a shaved head, wearing a dark t-shirt, looking directly at the camera. The video player interface includes a progress bar at 0:30 / 13:49, volume control, and other video controls. The background of the video frame shows the GitHub homepage with the text "Learn Git and GitHub without any code!" and "Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request." Below the video frame, there are two call-to-action buttons: "Read the guide" and "Start a project". A tooltip on the right side of the video frame says: "Visualize your project's community. A new graph is available in the Graphs tab to visualize your repository's data. You can now explore how repositories that contain Ruby gems relate to other repositories on GitHub." At the bottom of the video frame, there are links to "Your repositories" and various sharing icons. The overall theme of the slide is educational, specifically related to GitHub and its features.



Conteúdo didático complementar - GitHub

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Você pode ainda, baixar o aplicativo GIT para Windows ou outra plataforma em <https://git-scm.com/download/win>, para fazer comandos da sua máquina.

Downloading Git

The screenshot shows the 'Your download is starting...' screen of the Git for Windows download page. A large red arrow points to the '32-bit Git for Windows Setup' link, which is highlighted with a red border. Below it, a smaller red arrow points to the 'Download a GUI' section, also highlighted with a red border. The page includes links for 'Git for Windows Setup', '32-bit Git for Windows Setup', and '64-bit Git for Windows Setup'. To the right, there's a 'Now What?' section with three options: 'Read the Book', 'Download a GUI', and 'Get Involved'.

Your download is starting...

You are downloading the latest (2.19.2) 32-bit version of **Git for Windows**. This is the most recent [maintained build](#). It was released **9 days ago**, on 2018-11-21.

If your download hasn't started, [click here to download manually](#).

Other Git for Windows downloads

[Git for Windows Setup](#)

32-bit Git for Windows Setup

[64-bit Git for Windows Setup](#).

Now What?

Now that you have downloaded Git, it's time to start using it...

Read the Book

Dive into the Pro Git book and learn at your own pace.

Download a GUI

Several free and commercial GUI tools are available for the Windows platform.

Get Involved

A knowledgeable Git community is available to answer your questions.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Pode também selecionar um gerenciador de interface gráfica (Client – GIT) como o SourceTree para não ter que executar comandos GIT via linha de comando.

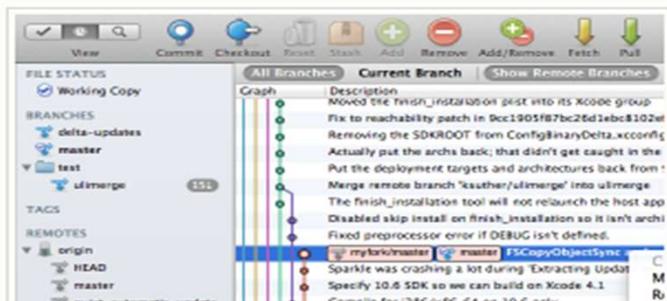
GUI Clients

Git comes with built-in GUI tools for committing ([git-gui](#)) and browsing ([gitk](#)), but there are several third-party tools for users looking for platform-specific experience.

If you want to add another GUI tool to this list, just follow the [instructions](#).

[All](#)[Windows](#)[Mac](#)[Linux](#)[Android](#)[iOS](#)

22 Windows GUIs are shown below ↓

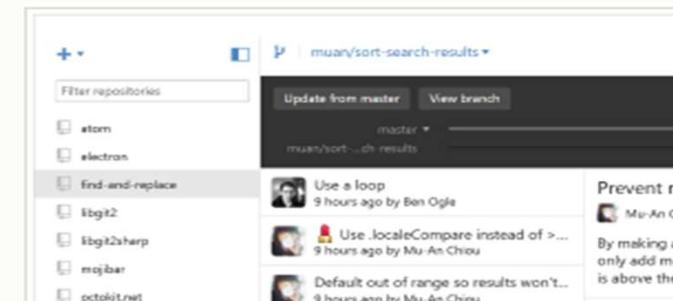


SourceTree

Platforms: Mac, Windows

Price: Free

License: Proprietary



GitHub Desktop

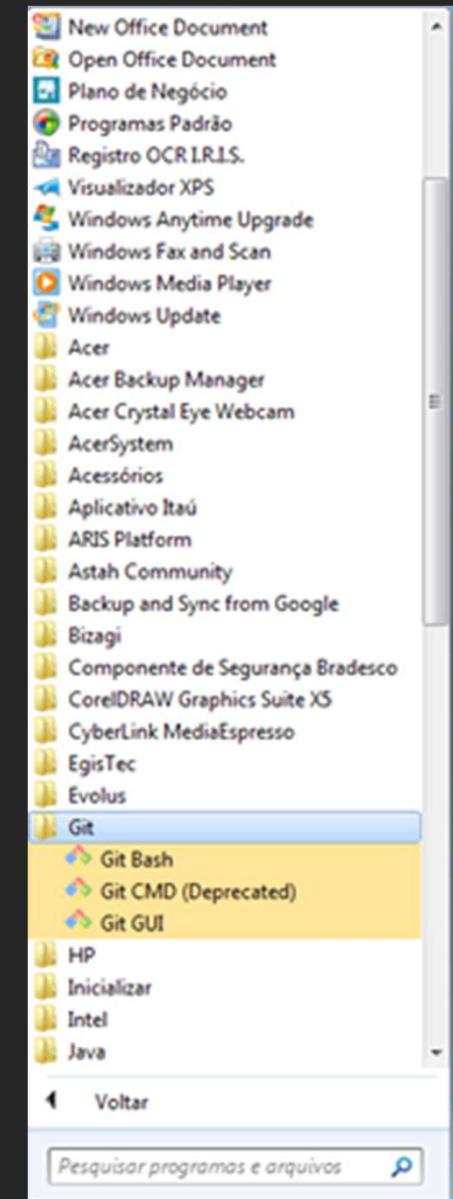
Platforms: Mac, Windows

Price: Free

License: MIT

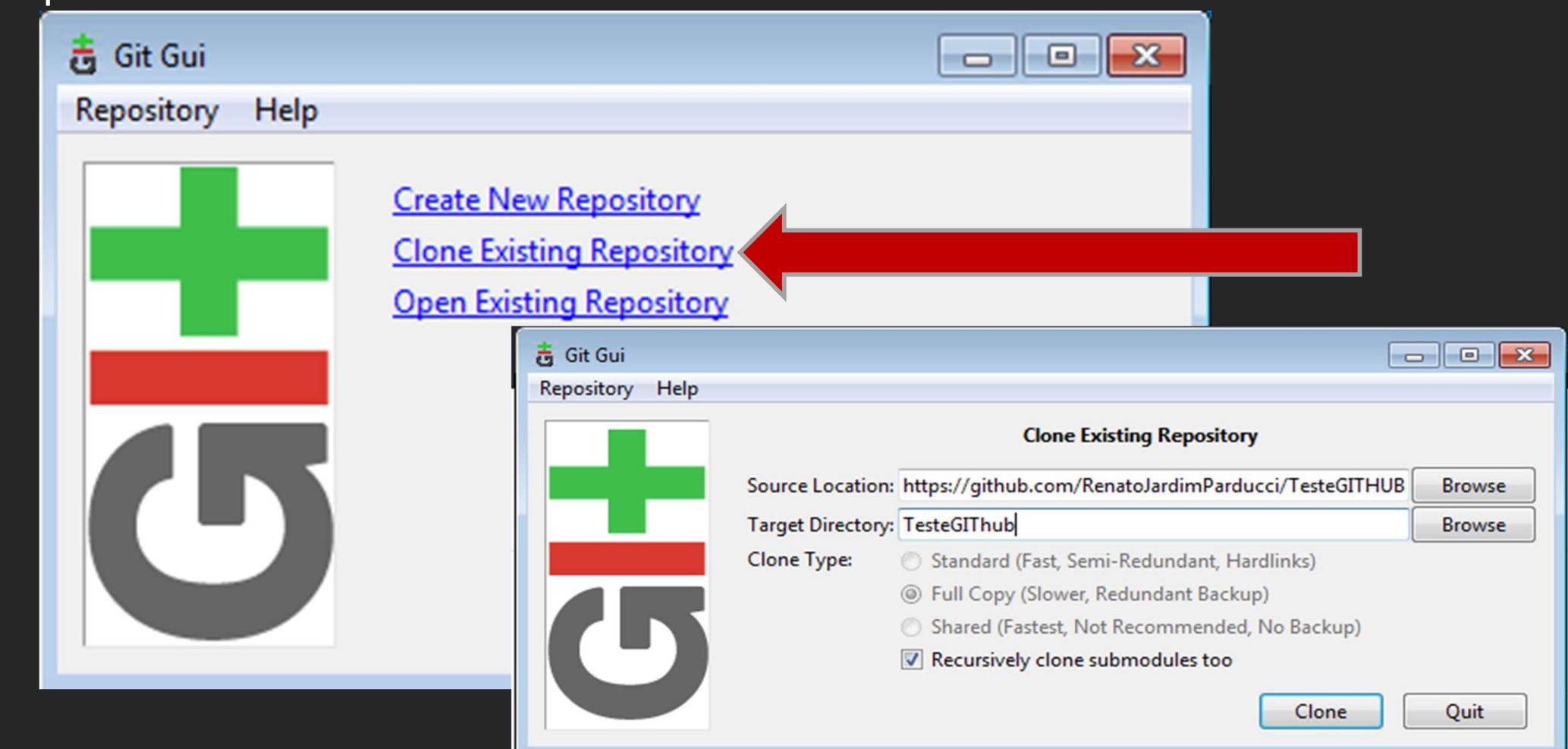
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Esses aplicativos adicionais permitem que você publique e puxe arquivos para manutenção usando somente o seu computador pessoal, sem necessidade de usar o navegador, como nós fizemos no nosso exemplo.



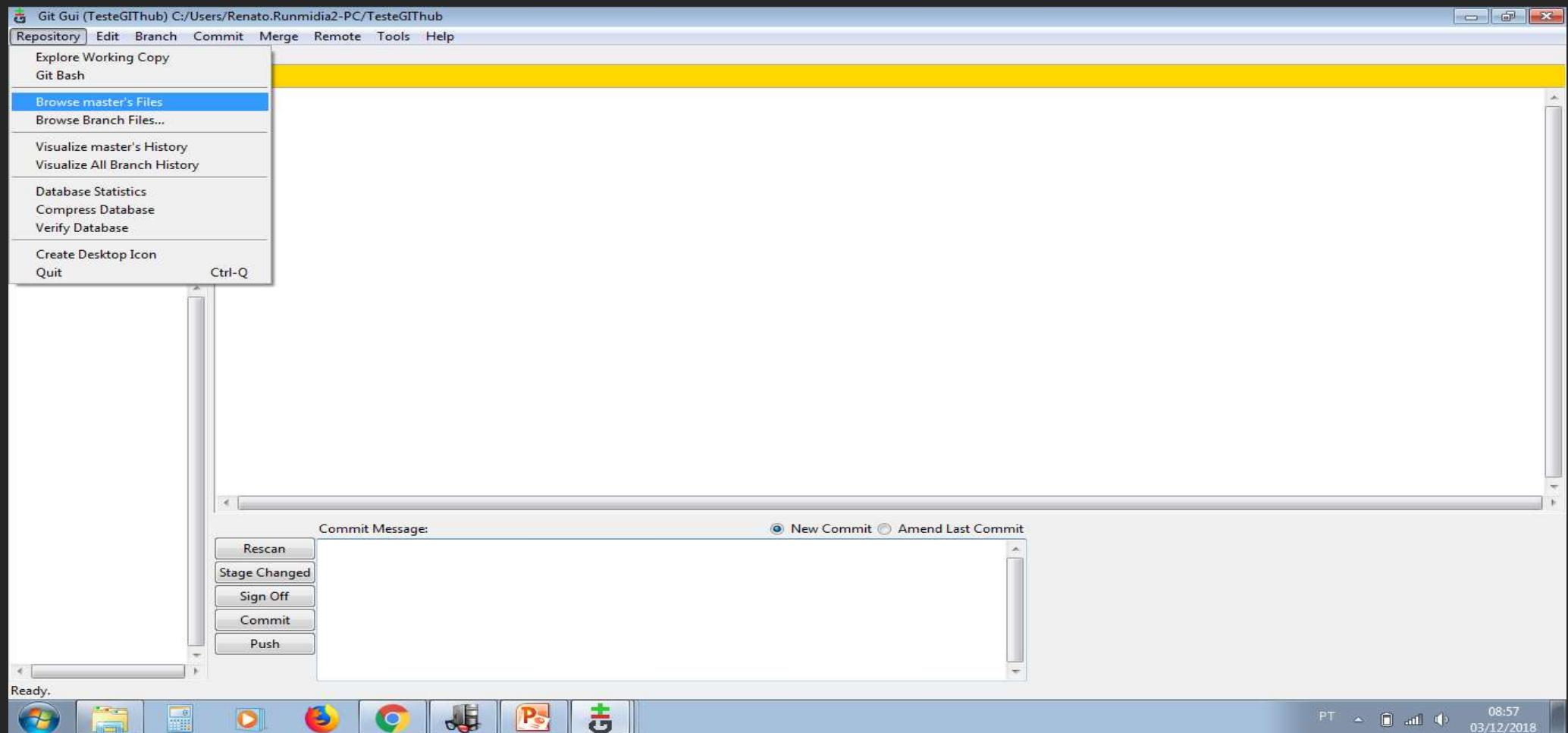
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Você pode sincronizar os repositórios em nuvem com o computador pessoal.



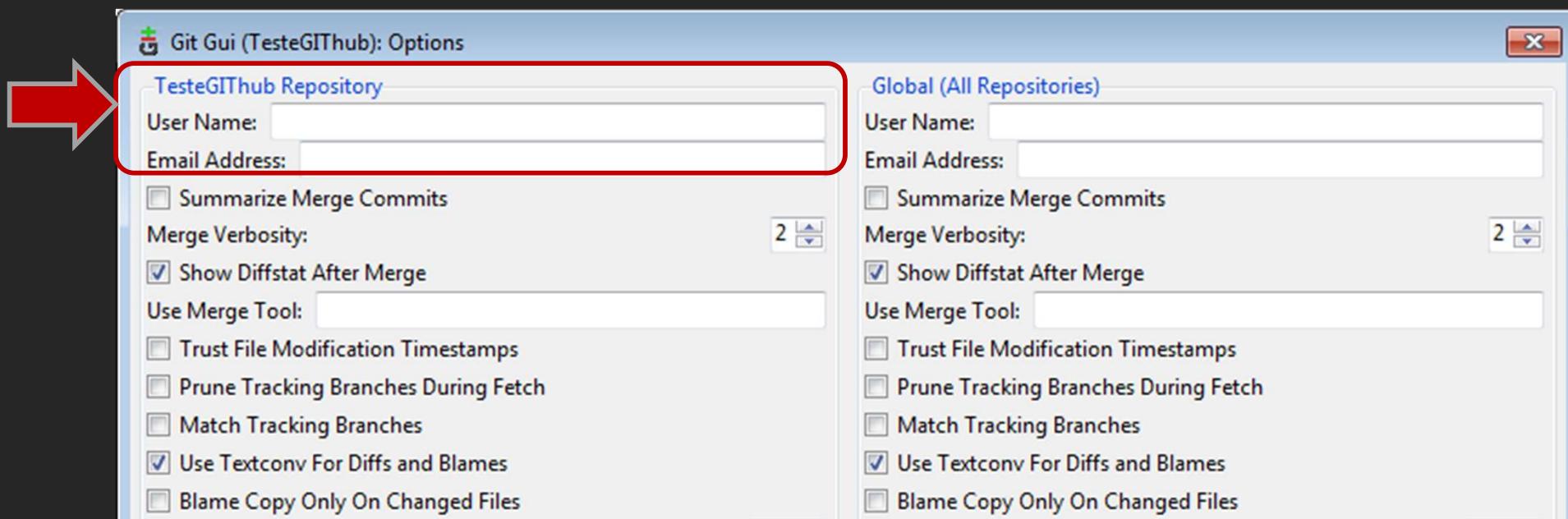
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Com o GitGUI, por exemplo, você pode acessar as áreas Master, Branches e pastas pelos menus o seu PC..



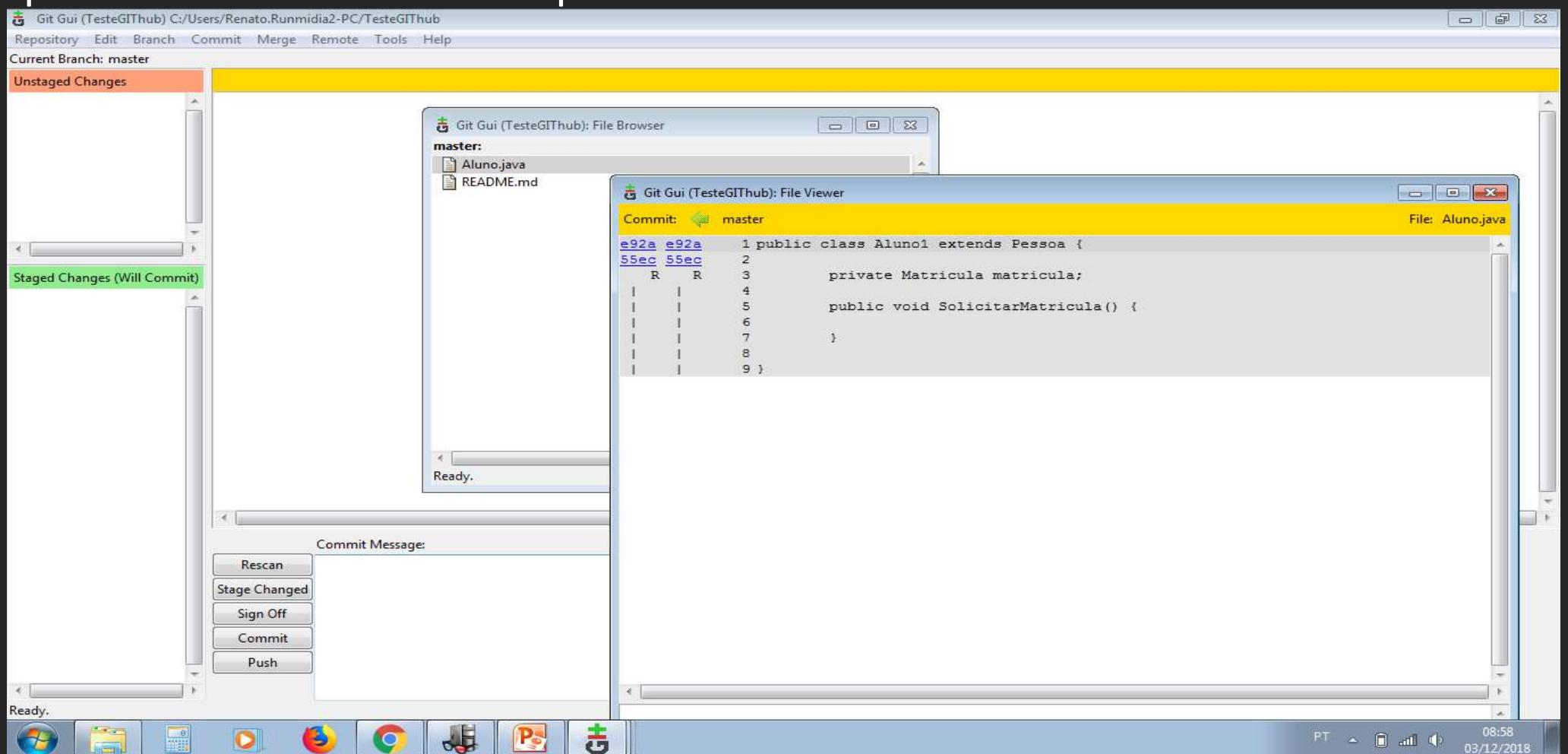
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Ele pode sincronizar com o GitHub.com se você configurar o Submenu Edit/Options com os dados do seu login na WEB.



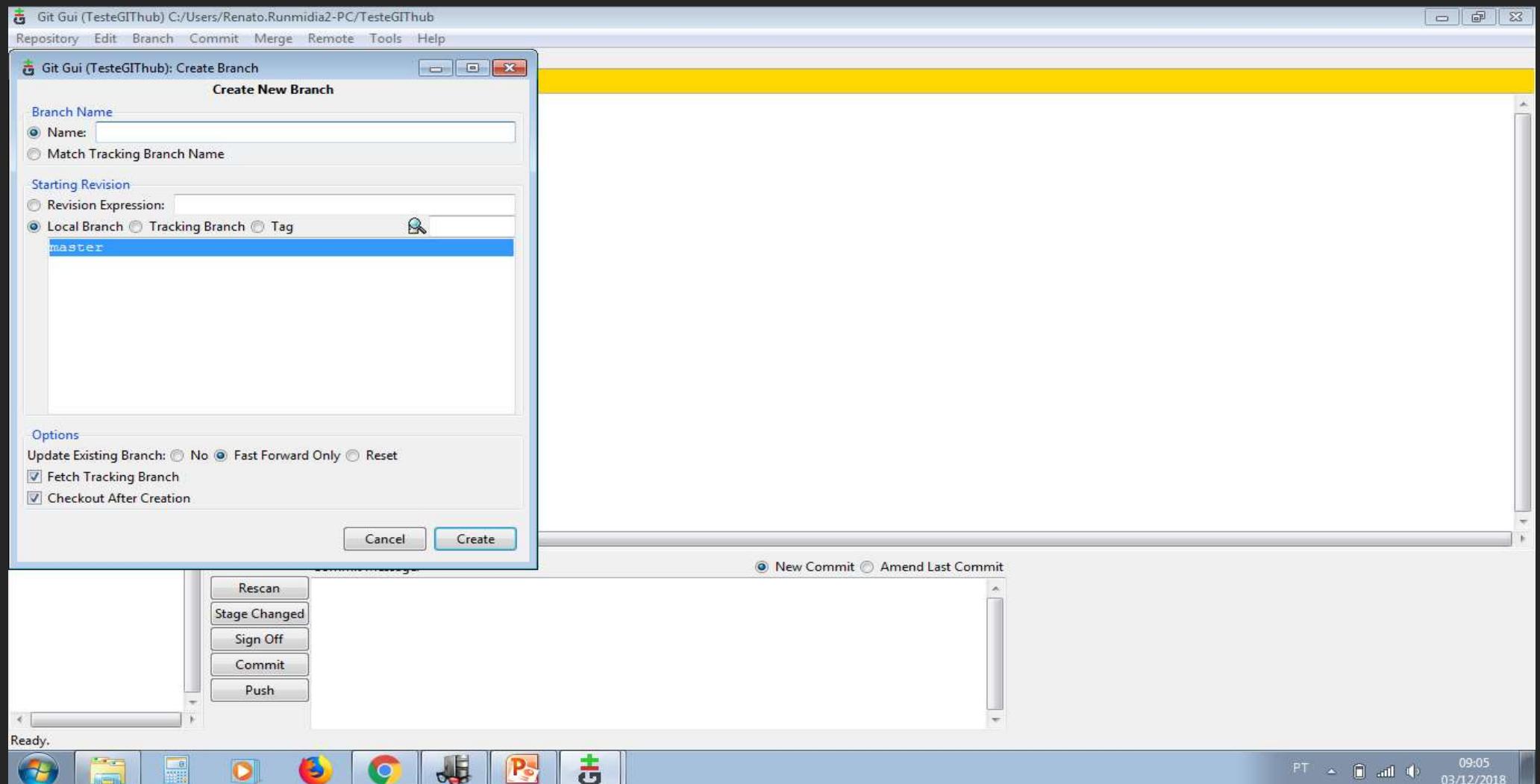
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Você pode sincronizar os repositórios em nuvem com o computador pessoal e visualizar seus repositórios.



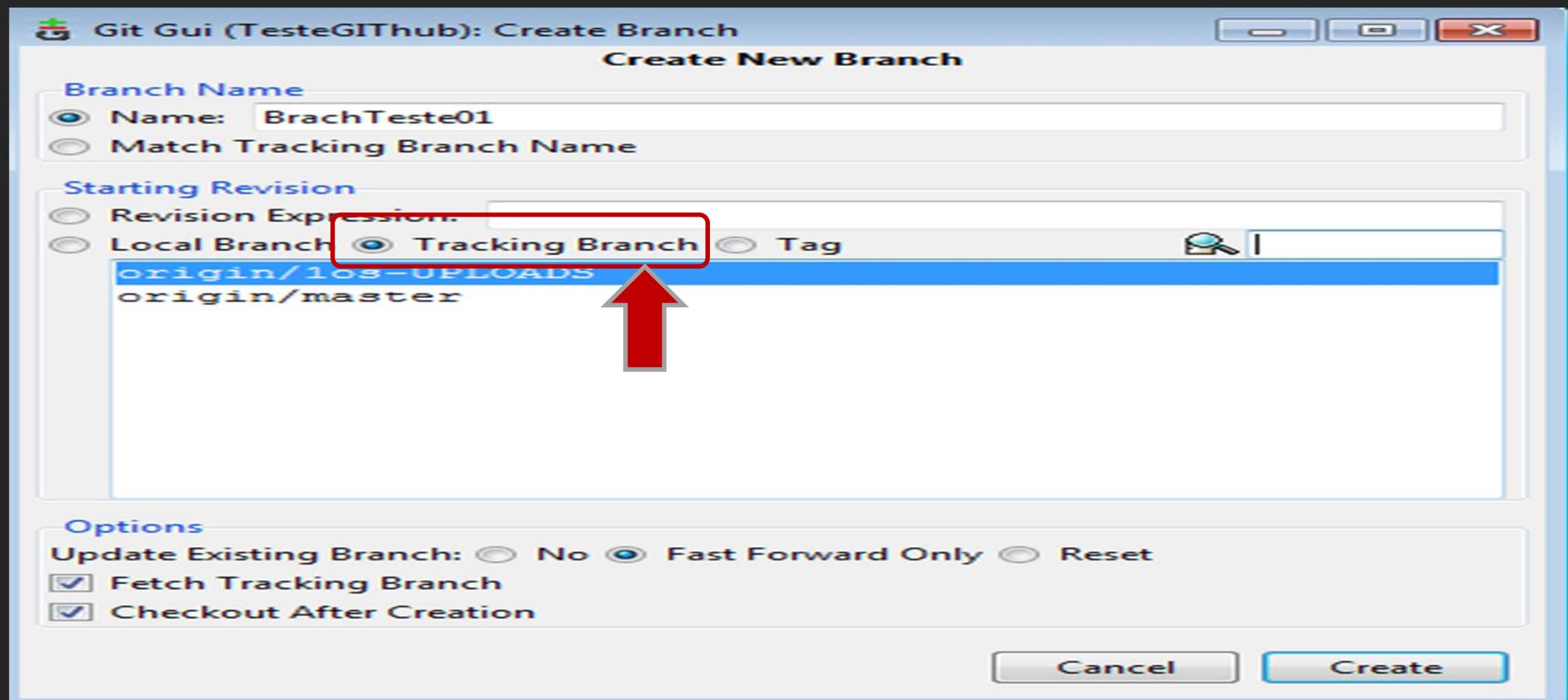
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Pode ainda criar branch...



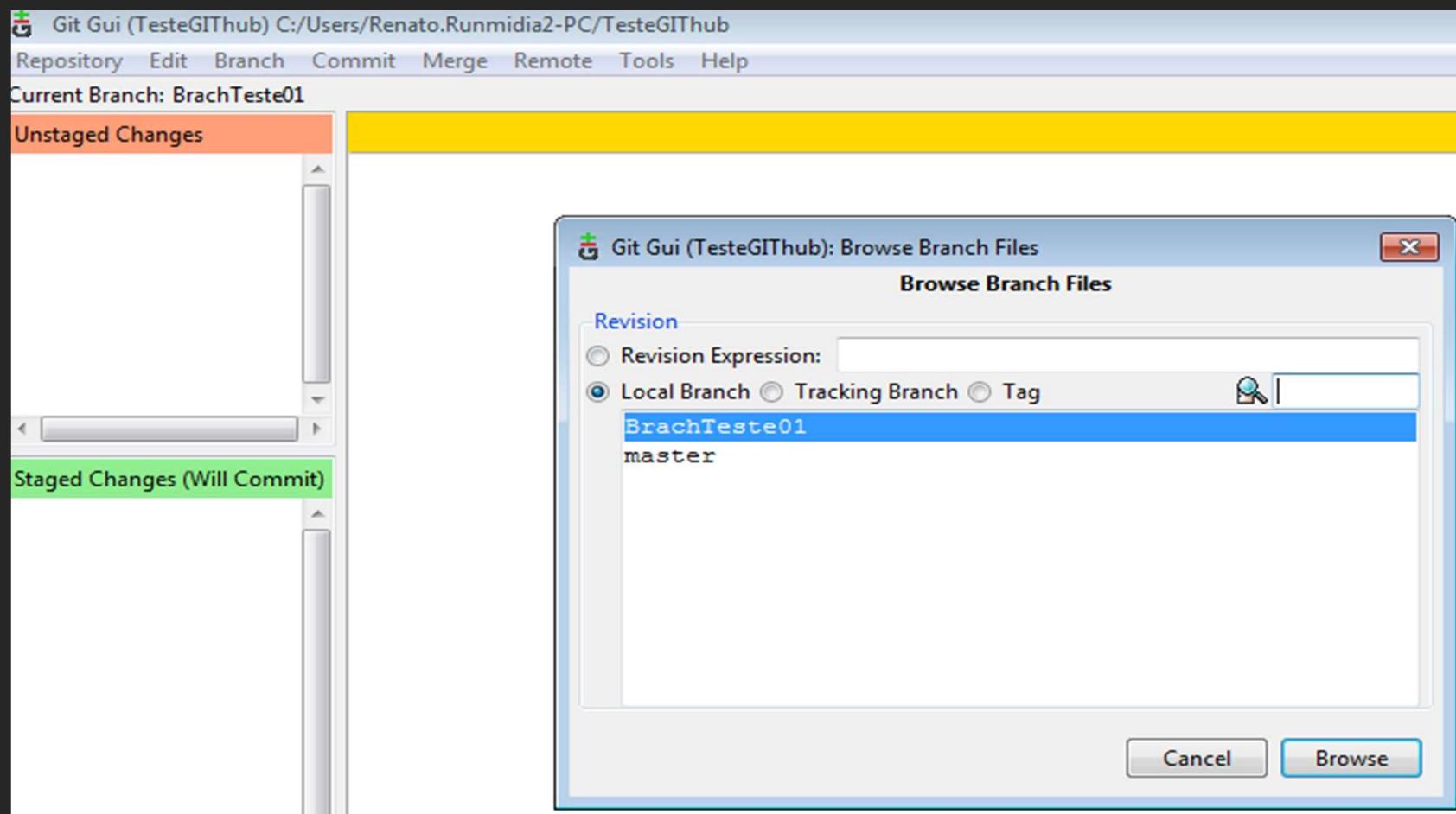
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

E essa Branch pode ser local (só no seu computador) ou rastreada (track), ligada com o repositório remoto na Internet (associada ao GitHub).



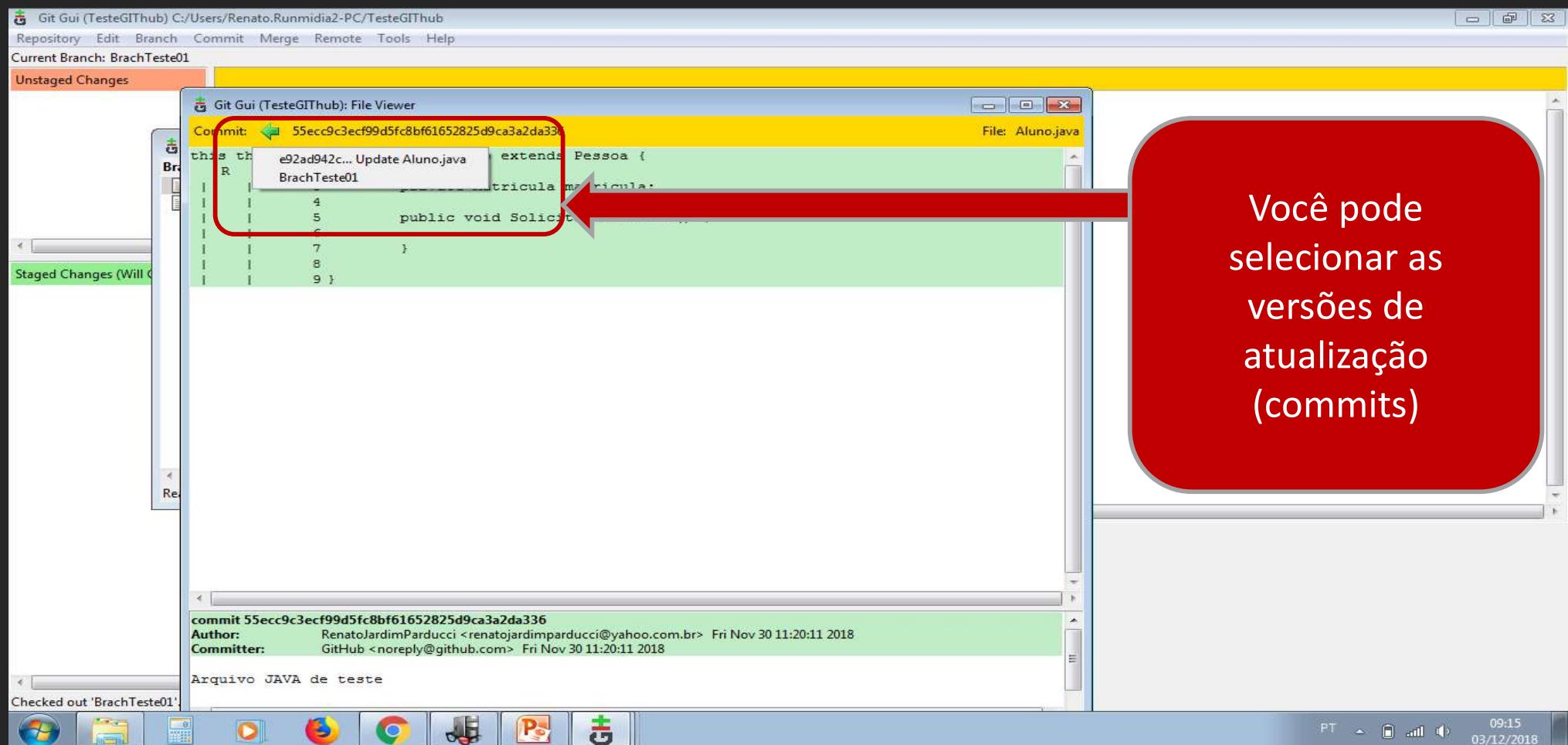
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Após criada a Branch, a navegação de acesso aos fontes é via menu da aplicação GUI.



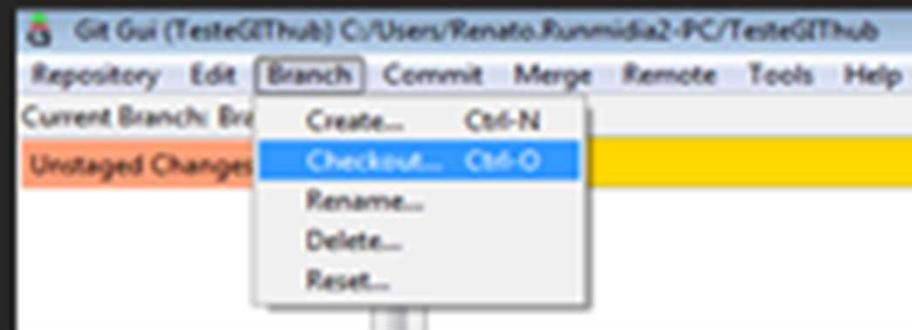
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Após criada a Branch, a navegação de acesso aos fontes é via menu da aplicação GUI.



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

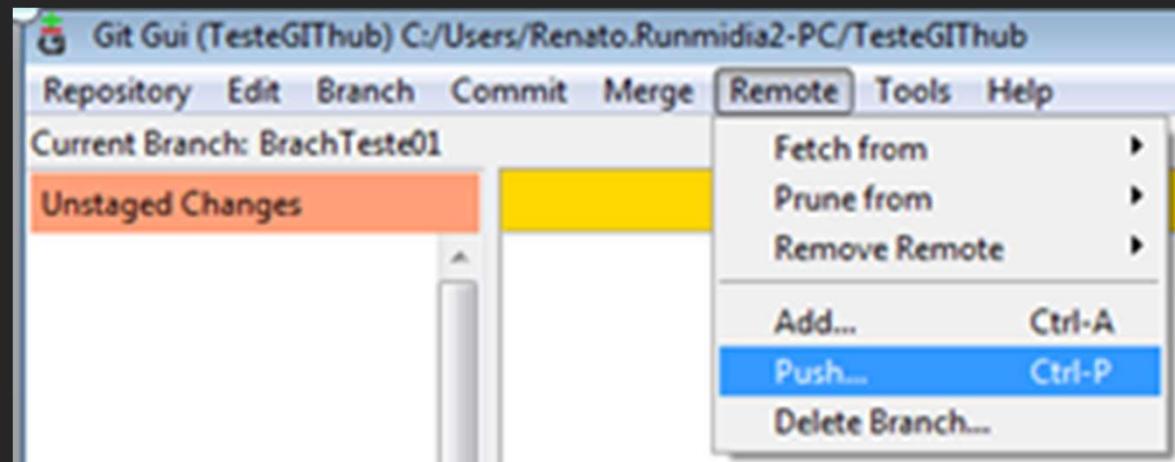
Para editar arquivos na Branch, faça um Checkout!



Use a opção REPOSITORY/EXPLORE WORKING COPY para acessar os documentos no repositório local e editá-los!

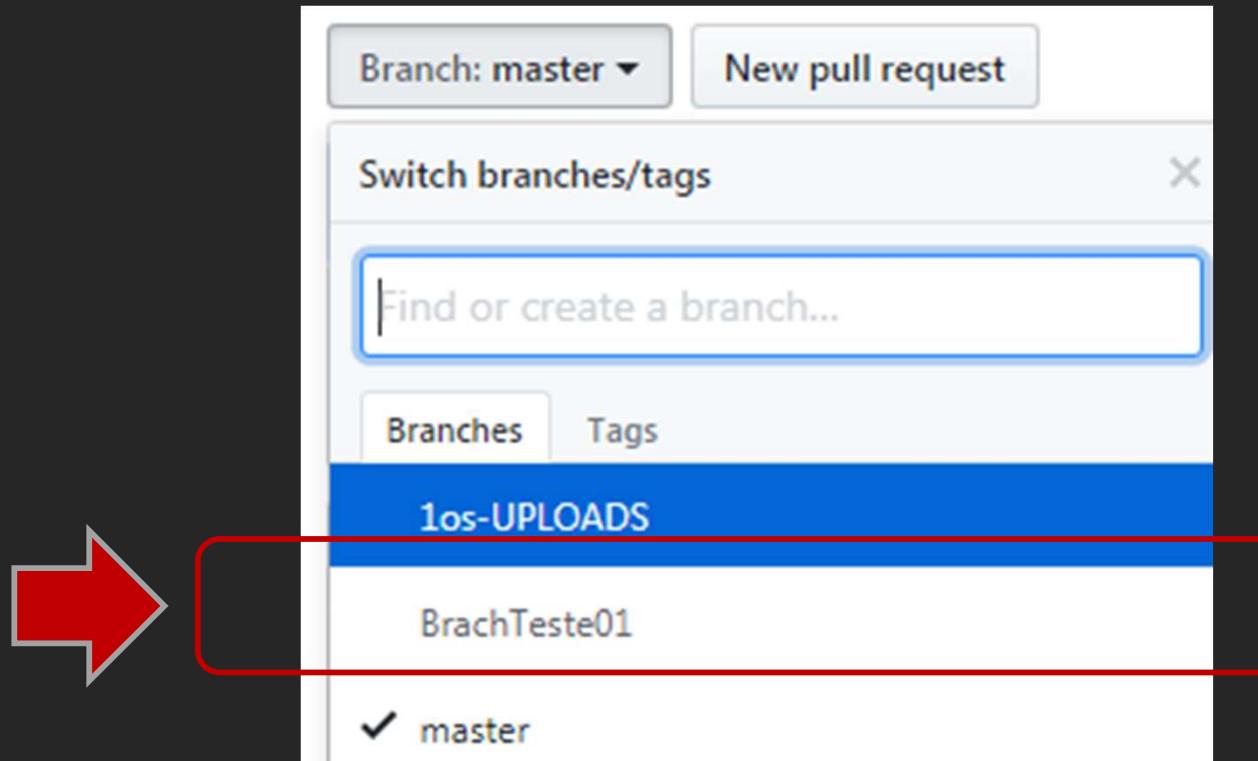
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Se você desejar publicar a Branch criada no seu PC no GitHub.com, faça um PUSH no submenu Remote (será exigido o Login do GitHub para essa operação)!



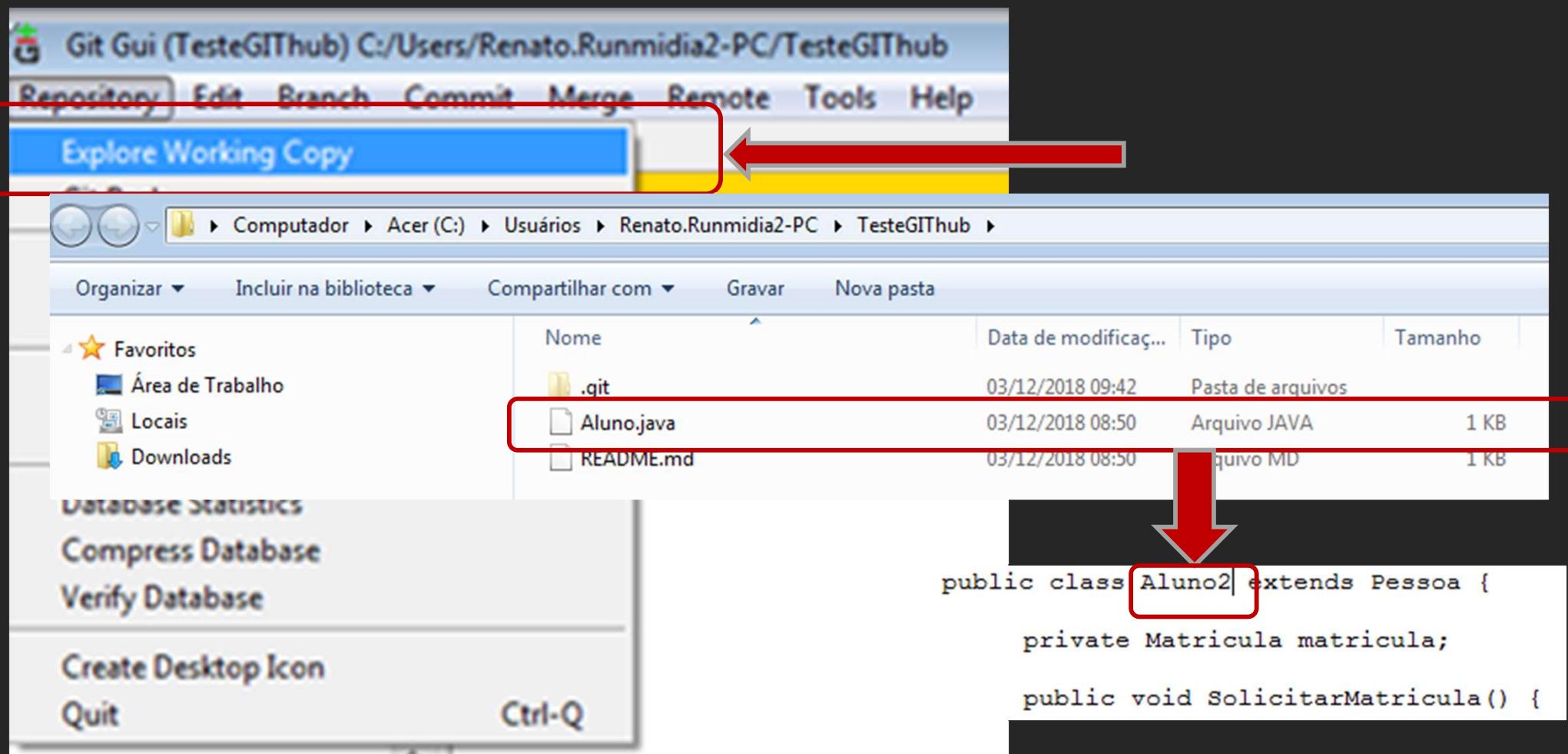
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Após o Push, você poderá visualizar e operar a Branch e seus arquivos na WEB:



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

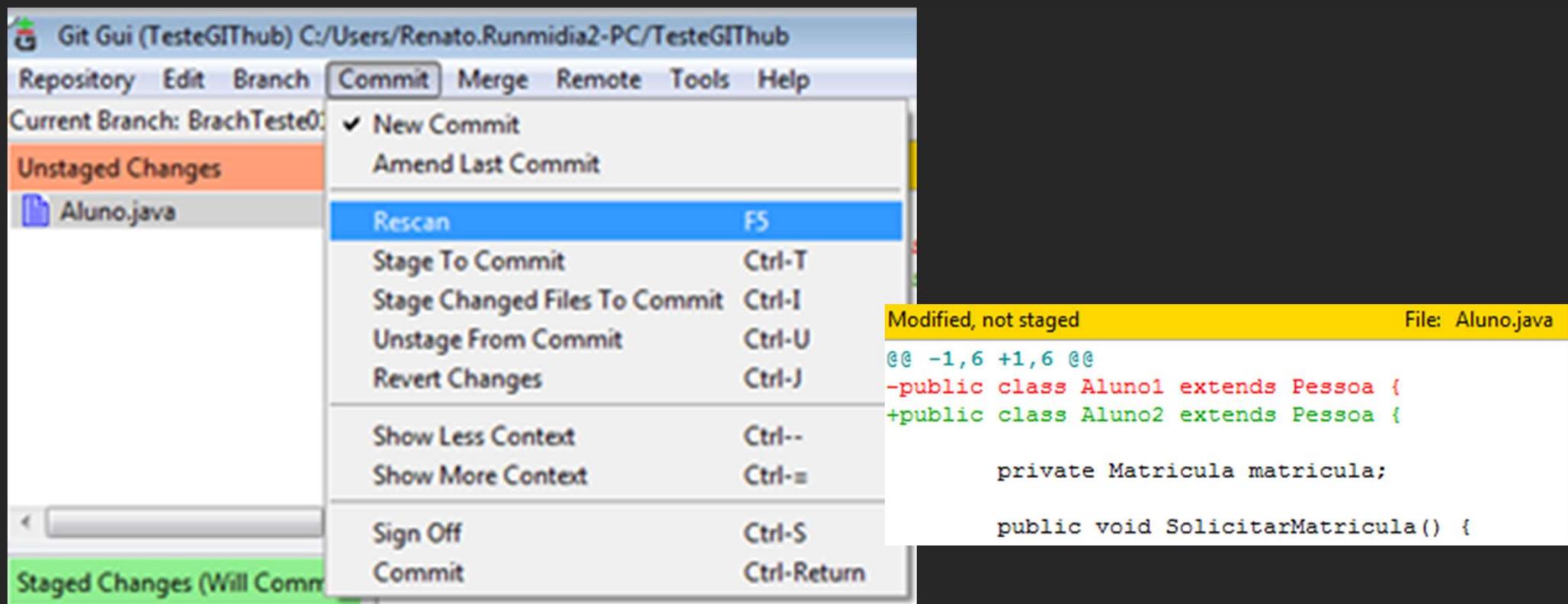
Você pode editar os arquivos no seu PC:



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

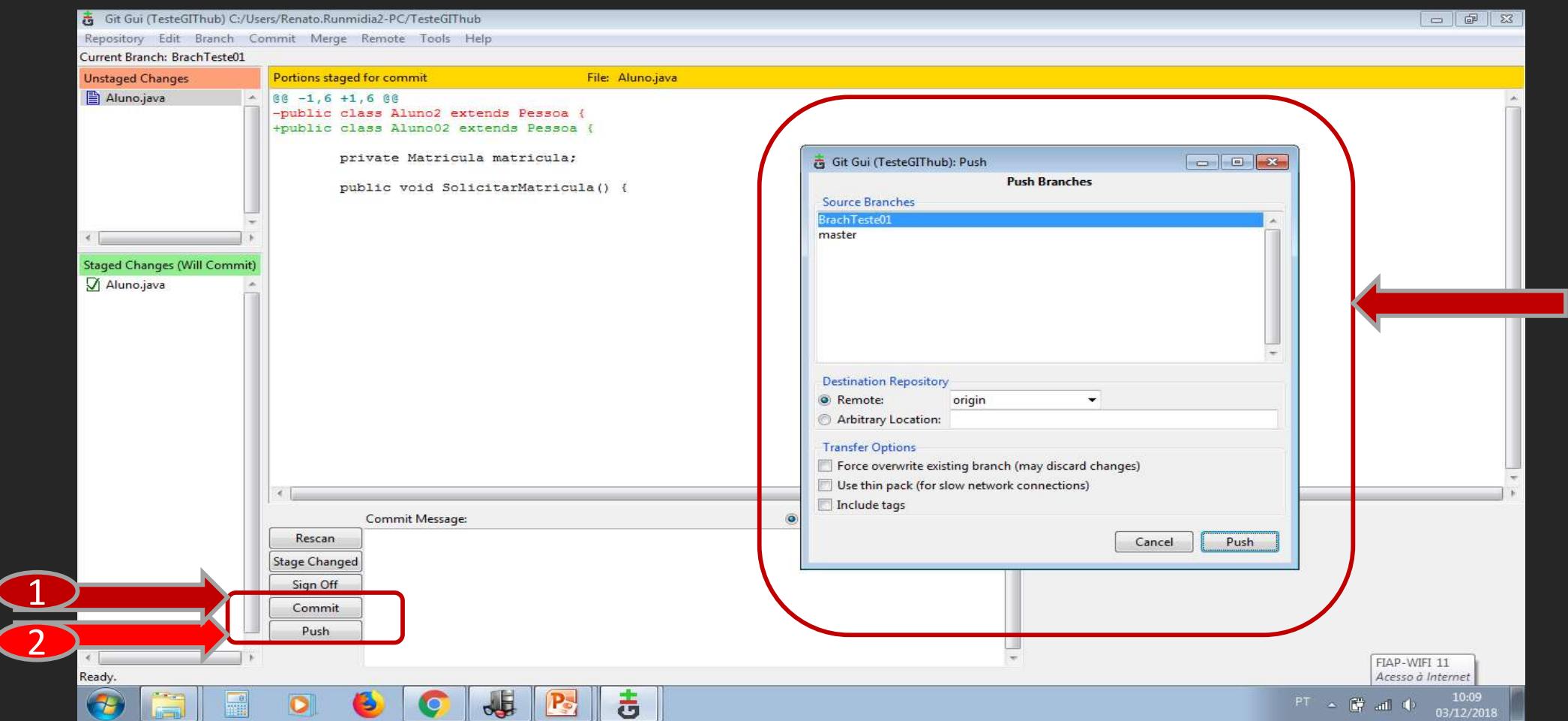
As edições são refletidas no GitGui e podem ser atualizadas por Push no GitHub!

Vá no submenu Commit/Rescan para descobrir as mudanças feitas, antes de fazer o Push!



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Executando o Push via Menu Commit: execute o Commit e depois o Push!!



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

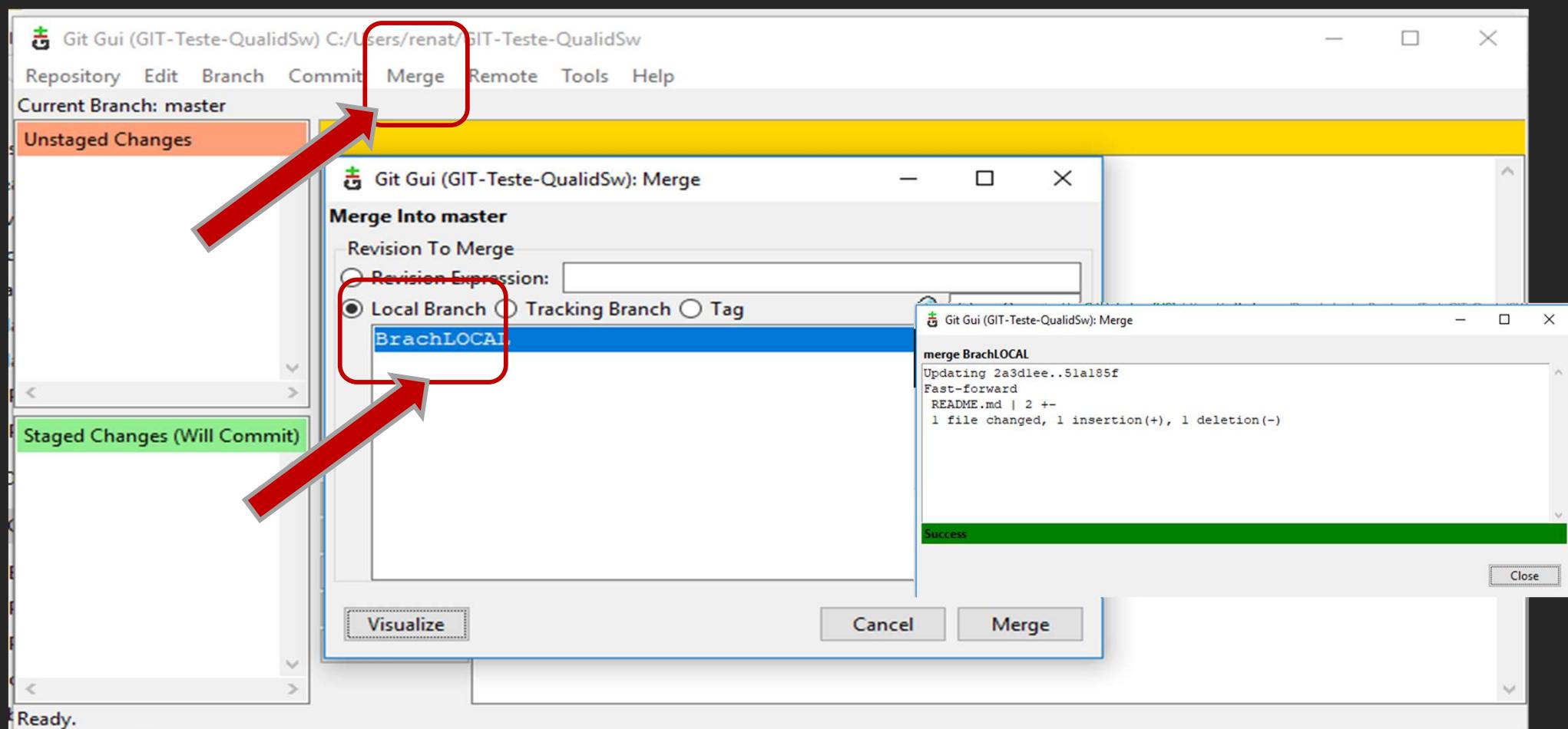
Com esses procedimentos, você consegue fazer Check out e Check in (Push) , mantendo sincronia entre o ambiente remoto dos fontes e o seu ambiente local de trabalho.

Após os Pushes, confira sempre se o GitHub.com foi atualizado corretamente.

The screenshot shows a GitHub repository page for 'RenatoJardimParducci / TesteGITHUB'. At the top, there's a search bar, navigation links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore', and a notification bell icon. Below the header, the repository name 'RenatoJardimParducci / TesteGITHUB' is displayed along with statistics: 'Unwatch' (1), 'Star' (0), and 'Fork' (0). A yellow banner at the top indicates a recent push to the 'BrachTeste01' branch 'less than a minute ago'. On the left, a sidebar lists 'Your recently pushed branches' with 'BrachTeste01' highlighted. The main content area shows the repository's structure with files like 'Aluno.java' and commit details such as 'Mudança via Cliente GUI' and 'Latest commit b700069 5 minutes ago'. Buttons for 'Create new file', 'Upload files', 'Find file', and 'Clone or download' are also visible.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

IMPORTANTE: o fato de você ter atualizado a Branch e Master no GITHUB não implica que a Master Local tenha sido atualizada.
Para atualizar a Master Local com as modificações, use o menu MERGE.



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

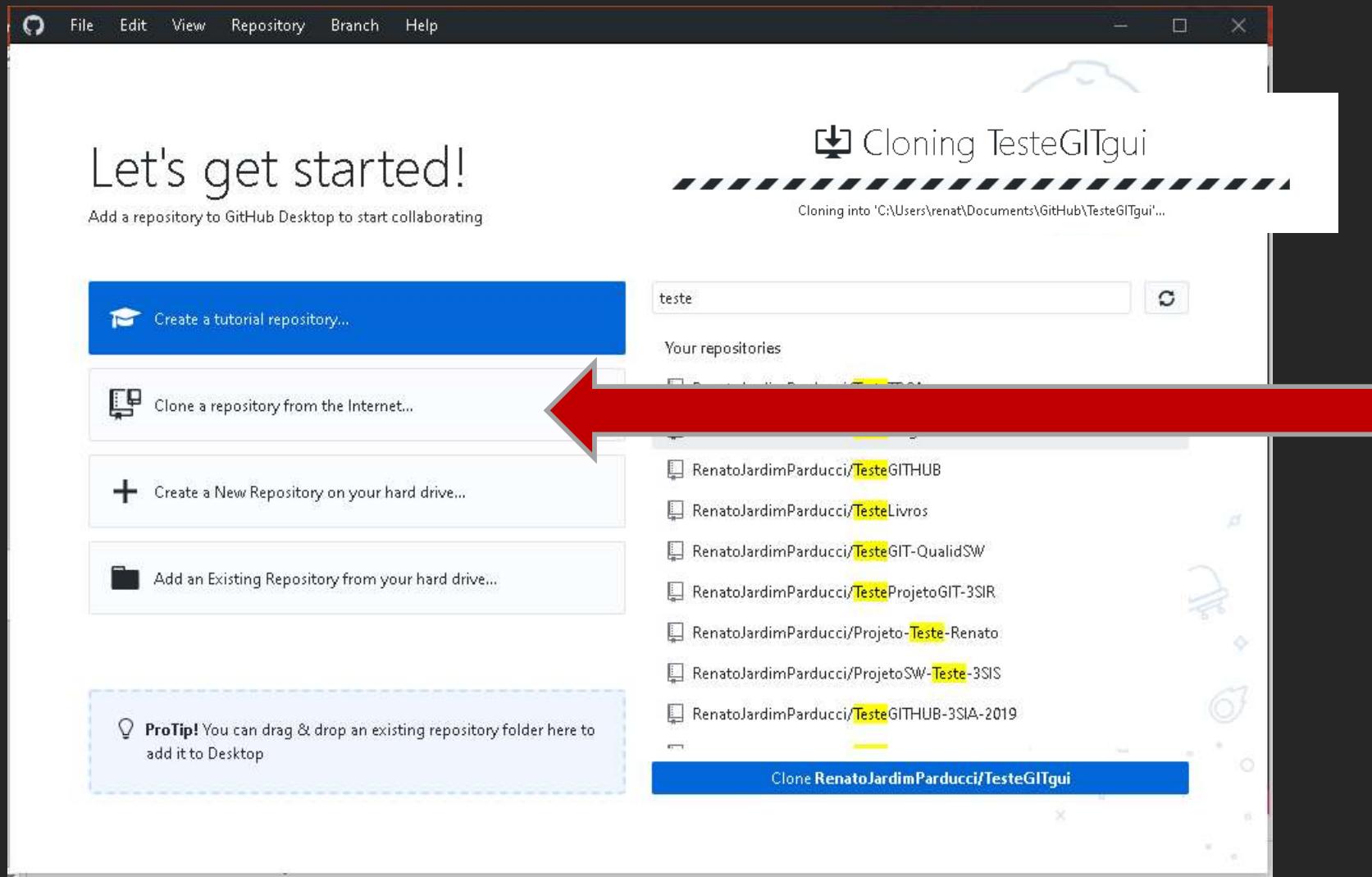
Esses aplicativos client permitem que você publique e puxe arquivos para manutenção usando somente o seu computador pessoal, sem necessidade de usar o navegador, como nós fizemos no nosso exemplo.



Vamos explorar agora o GitHub Desktop que pode ser obtido no próprio portal do GITHUB.

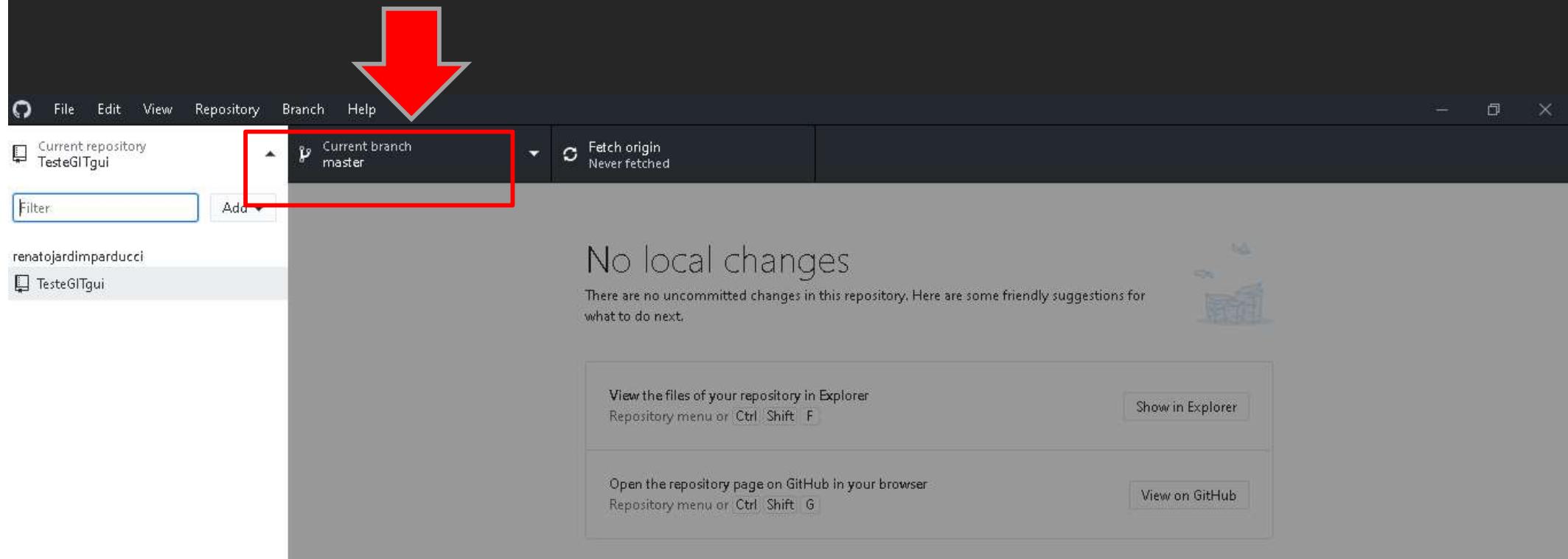
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Você pode sincronizar os repositórios em nuvem com o computador pessoal.



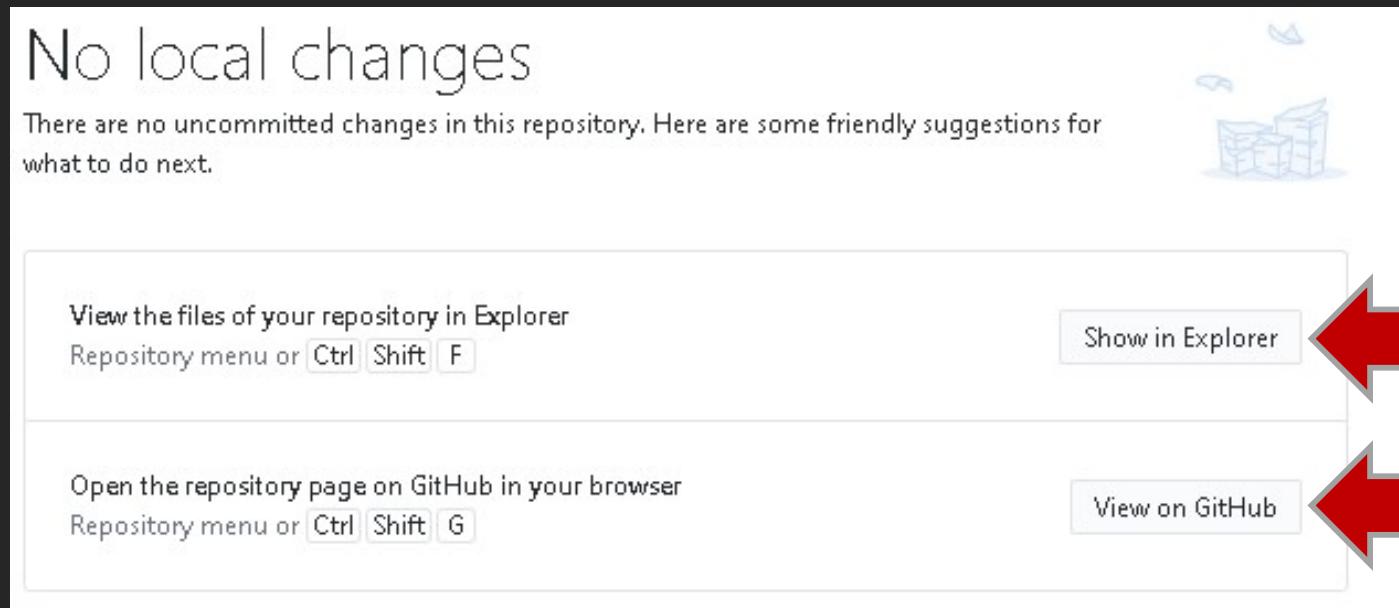
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Com o GitGUI, por exemplo, você pode acessar as áreas Master, Branches e pastas pelos menus o seu PC..



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Você pode acessar o clone local ou a pasta remota do GITHUB por esse software cliente.



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

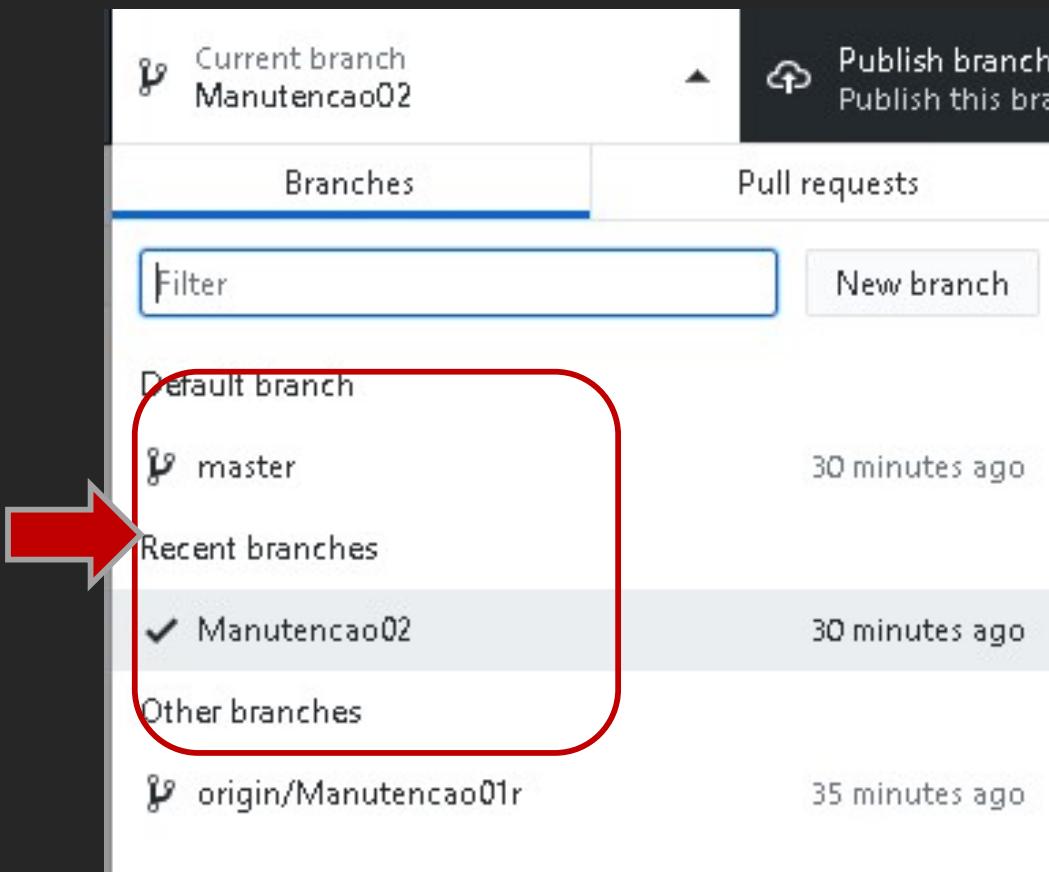
Pode ainda criar branch...

The screenshot shows a GitHub repository interface. At the top left, it says "Current branch master". To the right, there's a "Fetch origin" status: "Last fetched 3 minutes ago". Below this, there are two tabs: "Branches" (which is selected) and "Pull requests". A red arrow points to the "New branch" button. Under the "Branches" tab, there's a "Filter" input field and a "Default branch" section. The "master" branch is listed with a checkmark, last updated "28 minutes ago". Below that is an "Other branches" section, which lists "origin/Manutencao01r" last updated "33 minutes ago". The background shows a blurred "Commit history" section.

Branch	Last Updated
master	28 minutes ago
origin/Manutencao01r	33 minutes ago

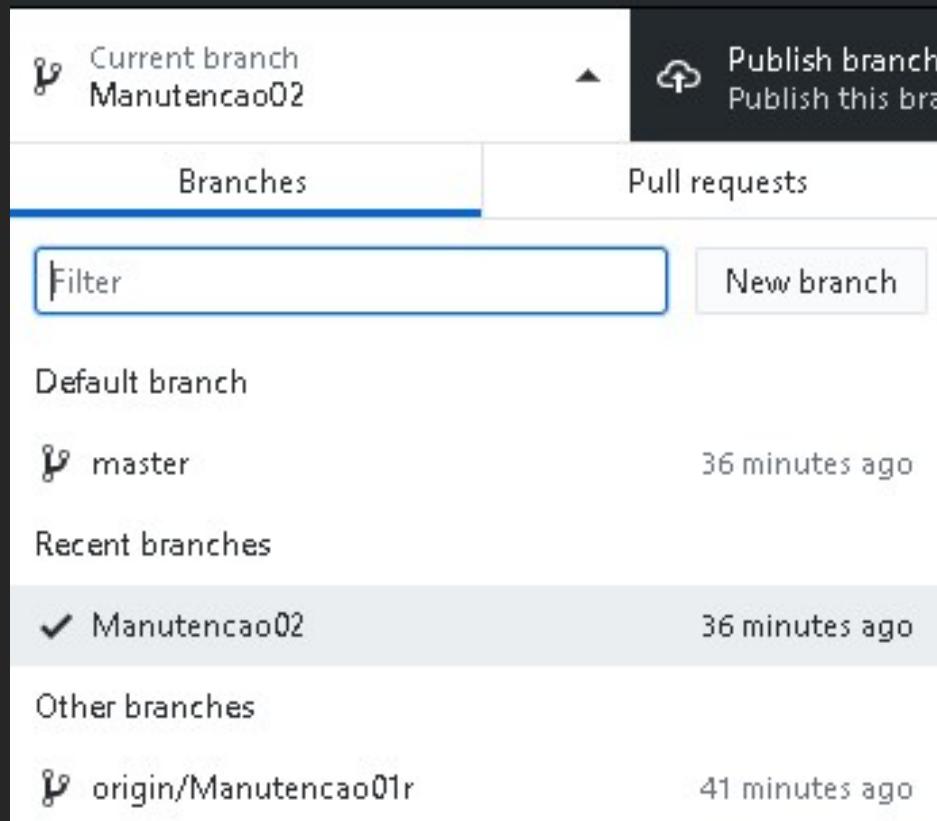
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

E essa Branch quando criada localmente, pode ser visualizada no GitHub client.



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

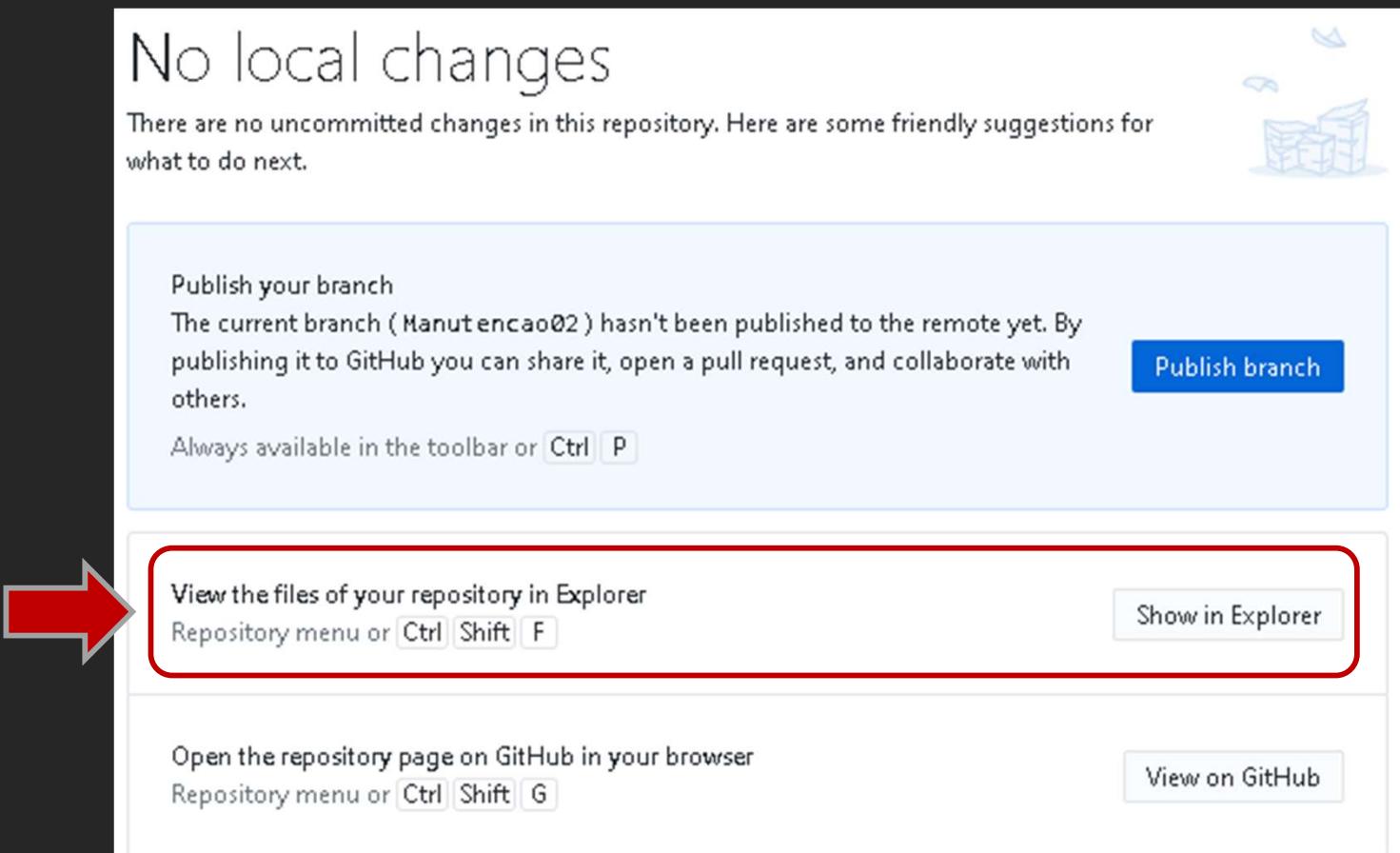
Para editar arquivos na Branch, faça um Checkout!



Selecione a branch no menu de navegação da ferramenta!

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Após criada a Branch, a navegação de acesso aos fontes é via menu da aplicação GUI.



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

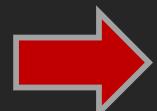
Assim que alterado o arquivo ou criado um arquivo novo na pasta do GIT local, o GitHub Client vai mostrar as diferenças entre a Master e a Branch.

The screenshot shows the GitHub Desktop application interface. The top navigation bar includes File, Edit, View, Repository, Branch, and Help. The repository dropdown shows "Current repository: TesteGITgui" and the branch dropdown shows "Current branch: Manutencao02". A "Publish branch" button is also present. The main window displays a "Changes" tab with 1 changed file, "README.md". The diff view shows three lines of code: line 1 is "# TesteGITgui", line 2 is "-# Texto NOVO", and line 3 is "+# Texto NOVO MESMO!!!".

Line	Content
1	# TesteGITgui
2	-# Texto NOVO
3	+# Texto NOVO MESMO!!!

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Se você desejar publicar a Branch criada no seu PC no GitHub.com, primeiro, faça um Commit no Git local

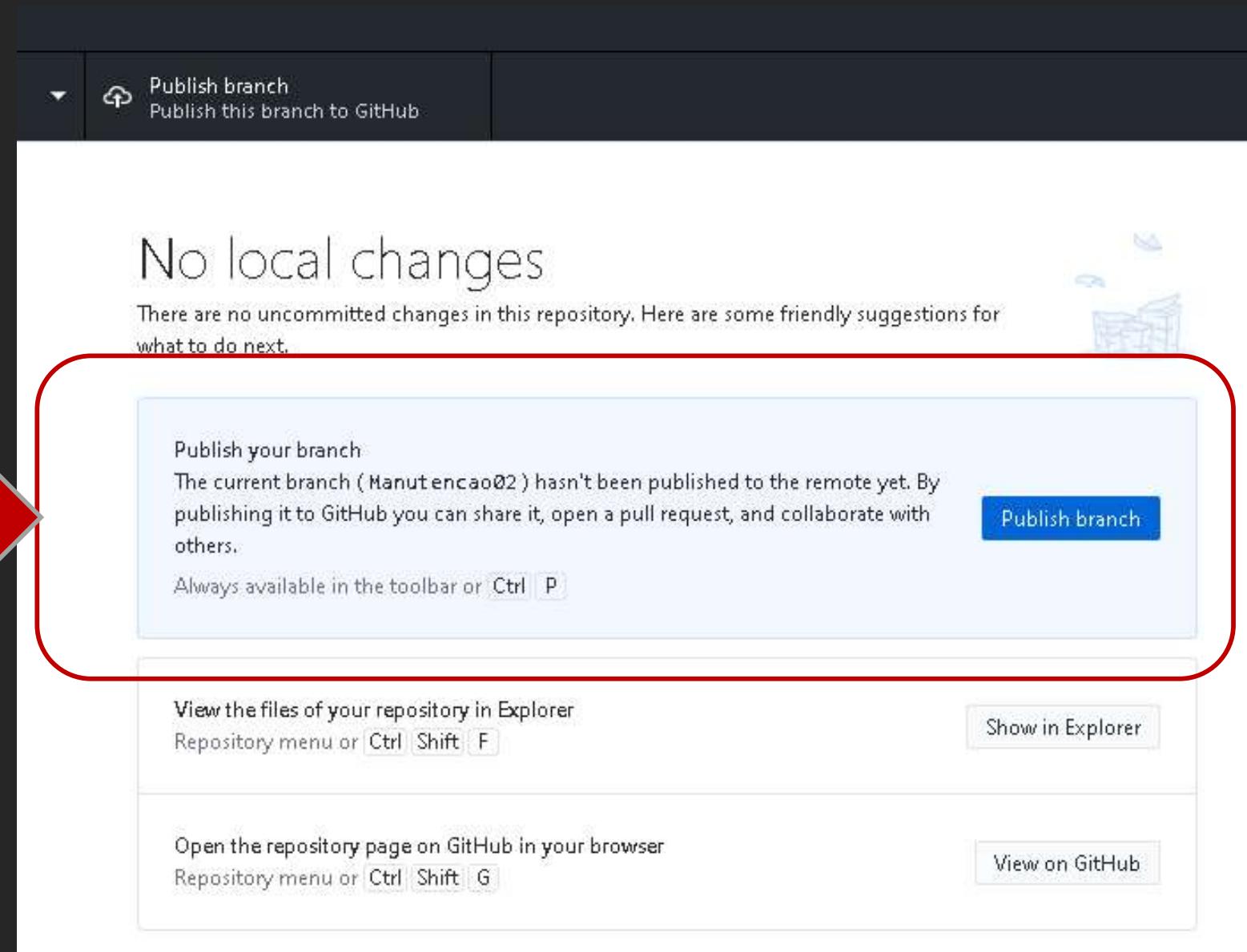


The screenshot shows a Git commit interface with the following details:

- Current repository:** TesteGITgui
- Current branch:** Manutencao02
- Changes:** 1 (1 changed file)
- History:** README.md
- Detailed Changes:**
 - Line 1: @@ -1,2 +1,2 @@
 - Line 2: # TesteGITgui
 - Line 3: -# Texto NOVO @@
 - Line 4: +# Texto NOVO MESMO!!! @@
- Action Buttons:**
 - Update README.md
 - Description: (empty)
 - Commit to Manutencao02

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Faça o PUSH no
GITHUB remoto
(WEB).



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

As edições são refletidas do GitGui para o GitHub!

Crie a Pull Request remota via GITHUB Client para atualizar a Master remota.

The screenshot shows the GitHub desktop application interface. At the top, it says "Fetch origin" and "Last fetched 2 minutes ago". Below this, the main area displays the message "No local changes". It includes a friendly suggestion: "Create a Pull Request from your current branch". It notes that the current branch ("Manutencao02") is already published to GitHub and encourages creating a pull request to propose and collaborate on changes. There are three buttons: "Create Pull Request", "Show in Explorer", and "View on GitHub". The "Create Pull Request" button is highlighted in blue. To the right of the main area, there are small icons for a file, a branch, and a person.

Fetch origin
Last fetched 2 minutes ago

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Create a Pull Request from your current branch
The current branch (Manutencao02) is already published to GitHub. Create a pull request to propose and collaborate on your changes.

Create Pull Request

Branch menu or Ctrl R

Show in Explorer

View the files of your repository in Explorer
Repository menu or Ctrl Shift F

View on GitHub

Open the repository page on GitHub in your browser
Repository menu or Ctrl Shift G

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Ao solicitar a Pull Request, o Git Hub é acionado. Você fará Login e confirmará o Merge/Pull Request!

The screenshot shows a GitHub pull request for a file named 'Update README.md' from branch '#2'. The pull request is from user 'RenatoJardimParducci' to 'Manutencao02'. It has 1 commit and 1 file changed. A comment from 'RenatoJardim Parducci' states: 'No description provided.' Below the commit, there is a green checkmark icon indicating 'Continuous integration has not been set up'. Another section shows a green checkmark and the text: 'This branch has no conflicts with the base branch'. At the bottom, there is a 'Merge pull request' button and a note: 'You can also open this in GitHub Desktop or view command line instructions.' A red arrow points to the 'Merge pull request' button.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

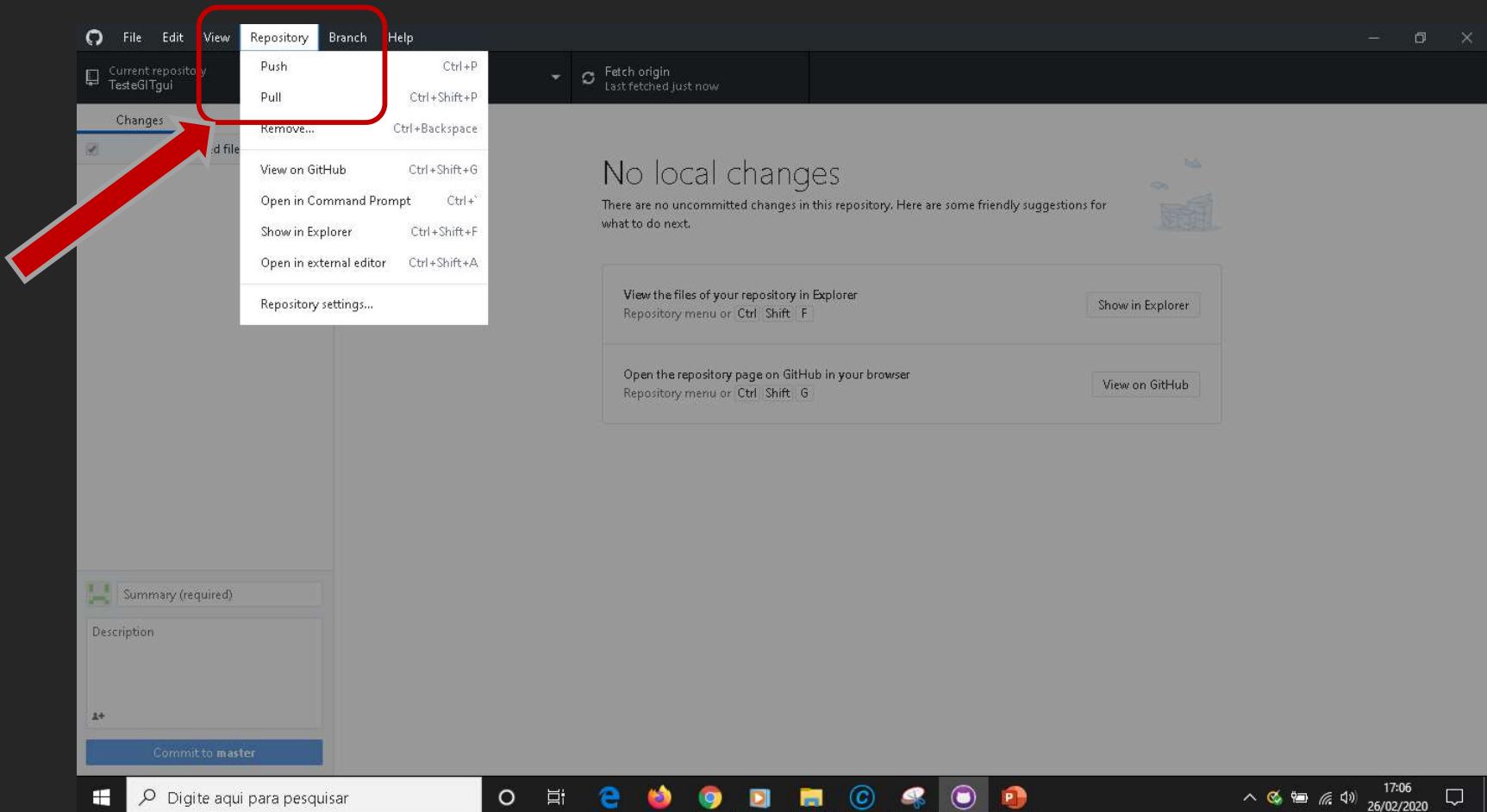
Com esses procedimentos, você consegue fazer Check out e Check in (Push) , mantendo sincronia entre o ambiente remoto dos fontes e o seu ambiente local de trabalho.

Após os Pushes, confira sempre se o GitHub.com foi atualizado corretamente.

The screenshot shows a GitHub repository interface. At the top, there are statistics: 5 commits, 3 branches, 0 packages, 0 releases, and 1 contributor. Below this, a navigation bar includes 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find file', and 'Clone or download'. A list of recent activity shows a merge pull request from 'RenatoJardimParducci' to 'RenatoJardimParducci/Manutencao02' and an update to 'README.md' 9 minutes ago. The 'README.md' file content is displayed below, showing the text 'TesteGITgui' and 'Texto NOVO MESMMO!!!'.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

IMPORTANTE: o fato de você ter atualizado a Branch e Master no GITHUB não implica que a Master Local tenha sido atualizada.
Para atualizar a Master Local, faça um Pull do Repositório.



ASSISTA OS VÍDEOS NO CANAL DO PROFESSOR



https://youtu.be/ps_ZTFrasAg

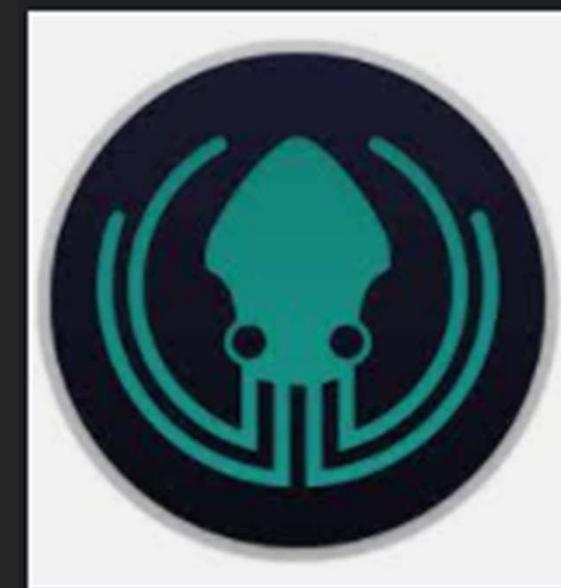
A screenshot showing a GitHub repository page for "RenatoJardimParducci / ExemploGitDesktop". The repository has 1 commit, 1 branch, 0 packages, and 0 releases. It contains files like "README.md" and "ExemploGitDesktop". To the right of the repository page is a video player window showing a man speaking, identified as Prof. Renato Jardim Parducci.



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Agora, conheça mais um cliente para trabalhar com GIT: o KRAKEN!

Além de gerenciar repositório GIT, ele permite gerenciar as tarefas da equipe de projeto com KANBAN e um roadmap de entregas.

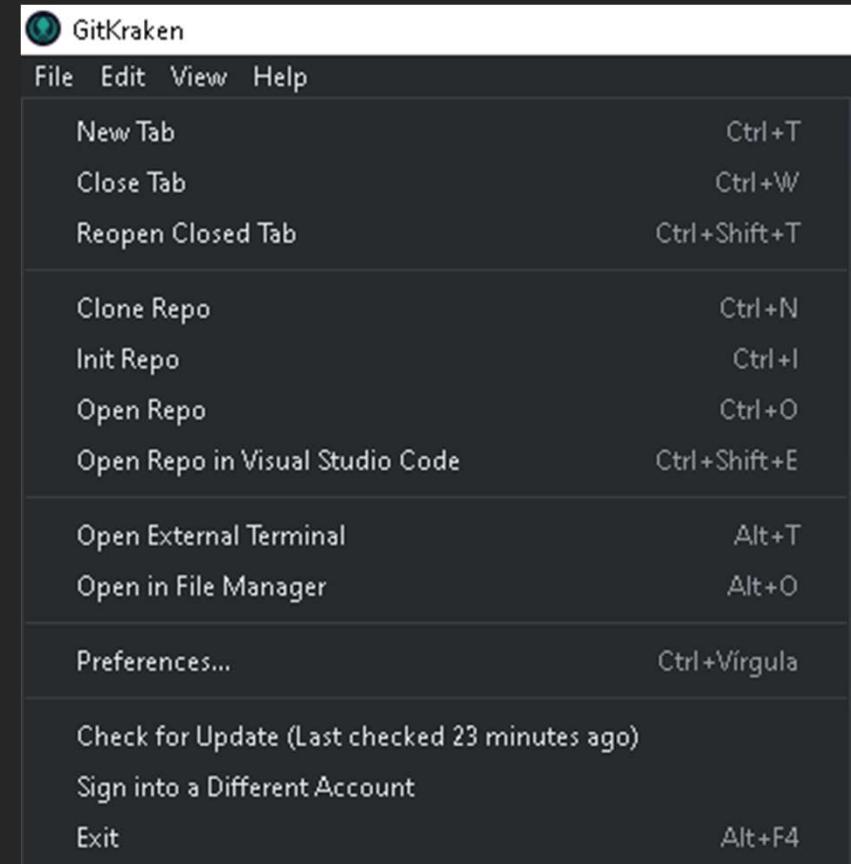


GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Ao logar no KRAKEN, você precisará definir com qual produto vai integrar:

- Trabalhar repositório do próprio Kraken ou
- Ligar com um repositório GIT HUB.

Associe o Kraken com sua conta GIT para poder trabalhar nos repositórios do GIT HUB com a interface do Kraken.



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

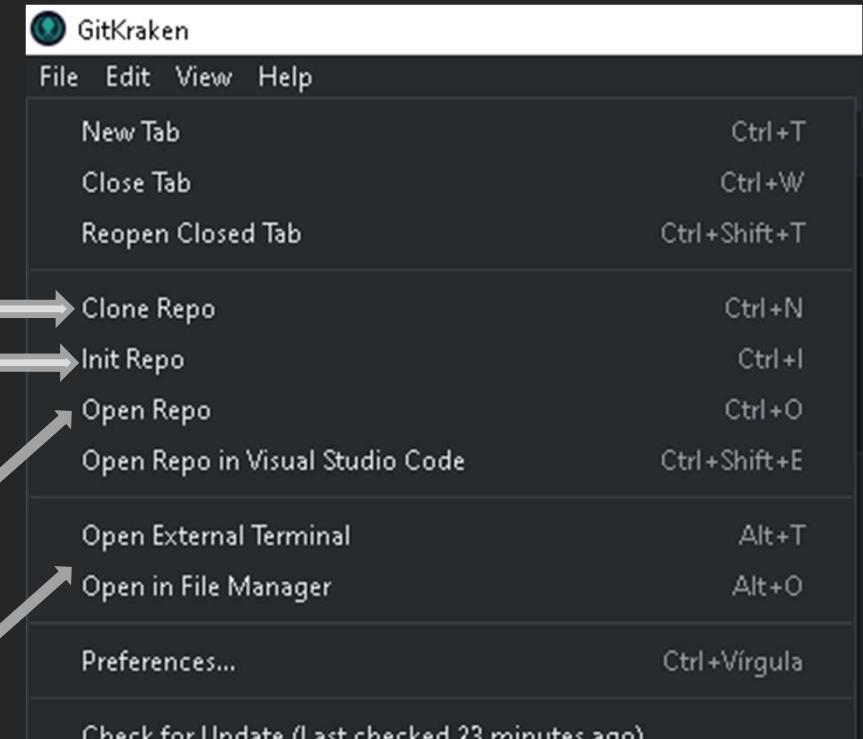
As operações do cliente você já conhece pois, são equivalentes às dos outros clientes que estudou (Github Desktop e Git Gui):

Clonar repositório do HUB, criando uma pasta concontrole de fontes no seu computador

Criar um repositóri no seu computador

Abrir um repositório clonado

Abrir a pasta de arquivos remota ou local, via file explorer



The screenshot shows the 'Repository Management' window in GitKraken. The 'Clone' tab is selected in the sidebar. The main area contains fields for 'Where to clone to' and 'URL', and a 'Clone the repo!' button. A dropdown menu titled 'Clone a Repo' lists various cloning options: GitHub.com, GitHub Enterprise, GitLab.com, GitLab (Self-Managed), Bitbucket.org, Bitbucket Server, and Azure DevOps. The 'GitHub Enterprise' option is currently highlighted.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

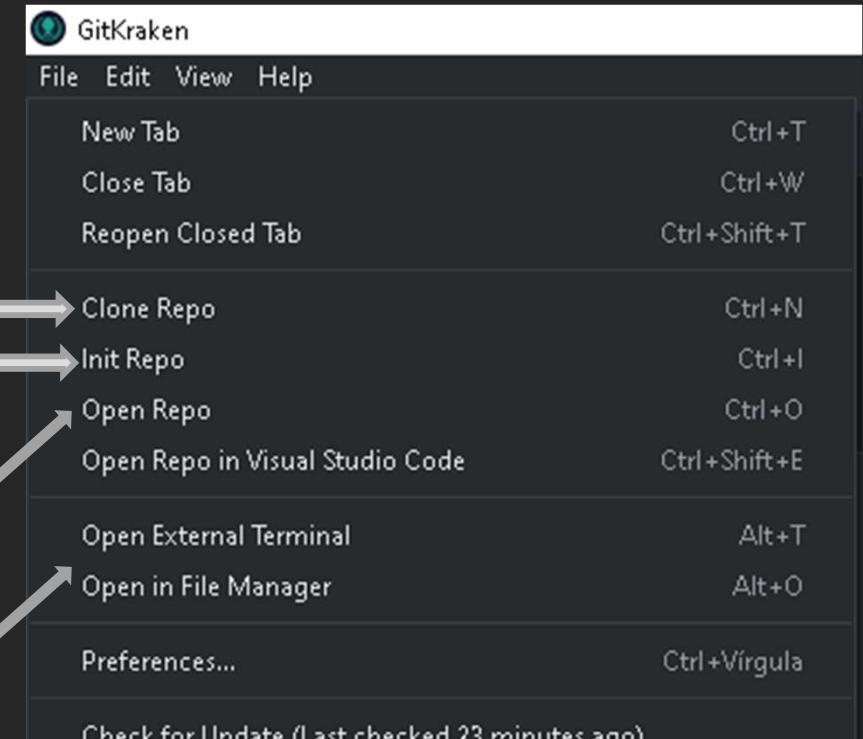
As operações do cliente você já conhece pois, são equivalentes às dos outros clientes que estudou (Github Desktop e Git Gui):

Clonar repositório do HUB, criando uma pasta concontrole de fontes no seu computador

Criar um repositóri no seu computador

Abrir um repositório clonado

Abrir a pasta de arquivos remota ou local, via file explorer



The screenshot shows the 'Repository Management' section of GitKraken. The 'Clone' tab is active. The interface includes:

- A sidebar with tabs: Open, Clone (selected), and Init.
- A 'Clone a Repo' form with fields for 'Where to clone to' (with a 'Browse' button) and 'URL'.
- A 'Clone the repo!' button.
- A dropdown menu for cloning from GitHub Enterprise, listing options like GitHub.com, GitHub Enterprise, GitLab.com, GitLab (Self-Managed), Bitbucket.org, Bitbucket Server, and Azure DevOps.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Faça 1º o clone do repositório do HUB:



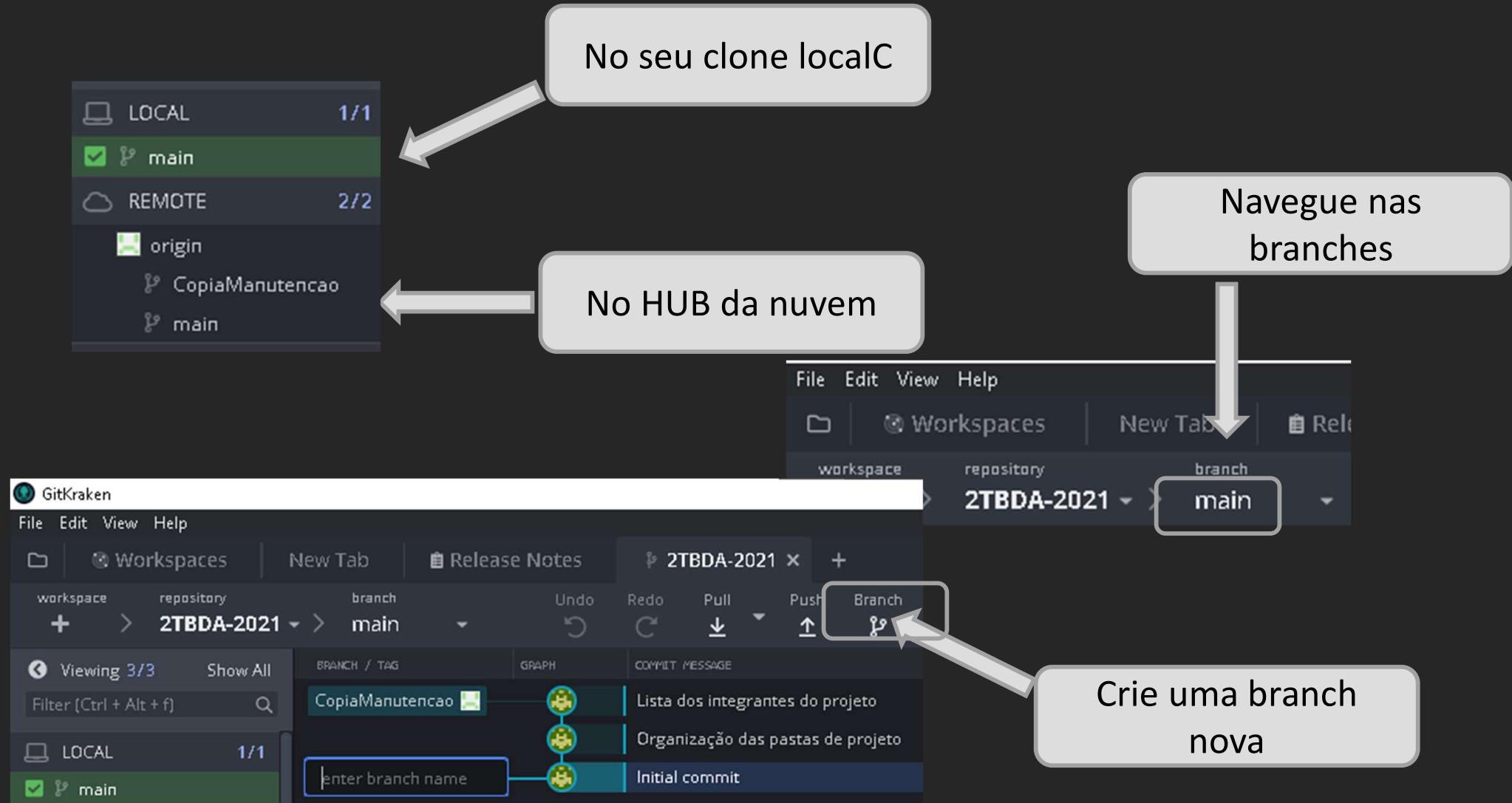
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Observe as branches:

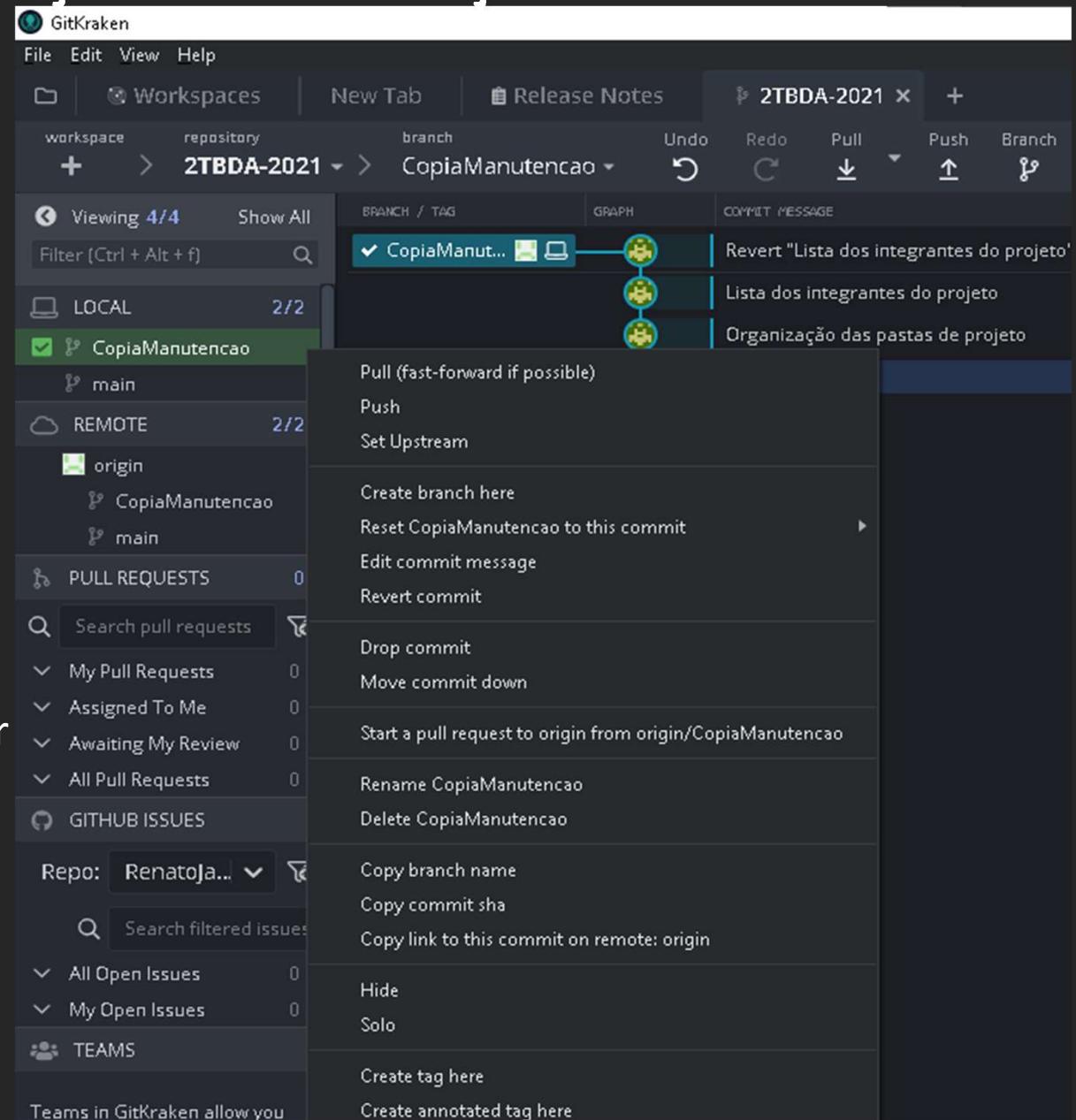


GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Observe as branches:



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE



Clique em ...
Ao lado de uma branch
Para abrir o menu de operações

Você pode fazer:

- Fetch e Push (e trazer atualizações do hub para o computador local)
- Commit e Pull (para atualizar o HUB com as mudanças locais)
- Pode deletar branch se se arrepender (CUIDADO AO FAZER ISSO)
- Renomear a branch

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

The screenshot shows the GitKraken application interface. The main window displays a commit history for the branch 'Manut-Parducci' of the repository '2TBDA-2021'. The commits are:

- Revert "Lista dos integrantes do projeto"
- Lista dos integrantes do projeto
- Organização das pastas de projeto
- Initial commit

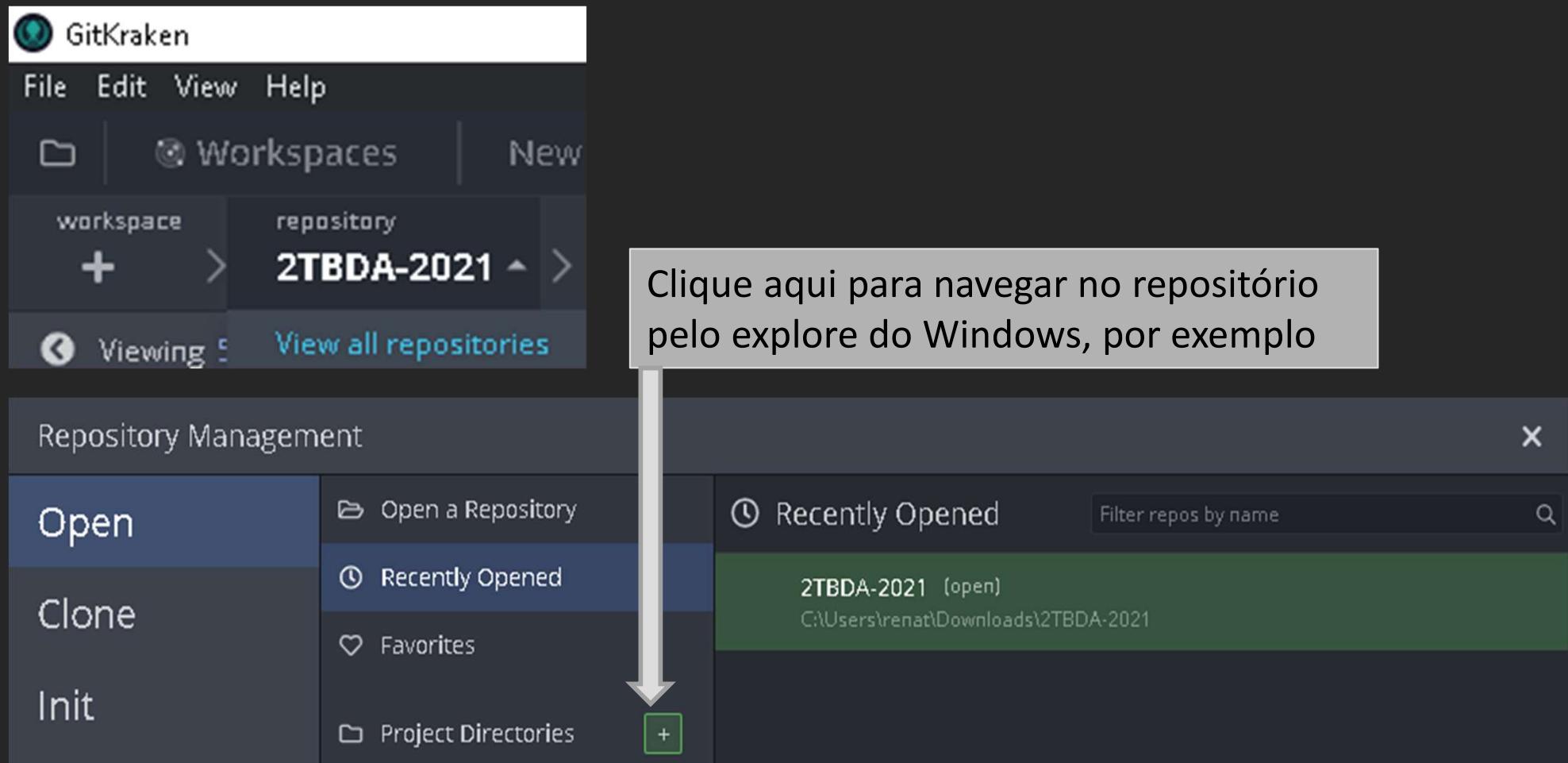
A callout box with the text "Navegue dos arquivos de uma branch..." points to the file viewer on the right, which is currently displaying the 'README.md' file. The file content is:

```
1 # 2TBDA-2021
2 # Projeto teste de uso de Branch de manutenção no GIT com KRAKEN
```

The sidebar on the left shows workspace, repository, and branch navigation, along with sections for Pull Requests and GitHub Issues.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Você pode clicar no nome do seu clone e ver todos os repositórios, podendo navegar neles com seu explorador de arquivos, do sistema operacional local!

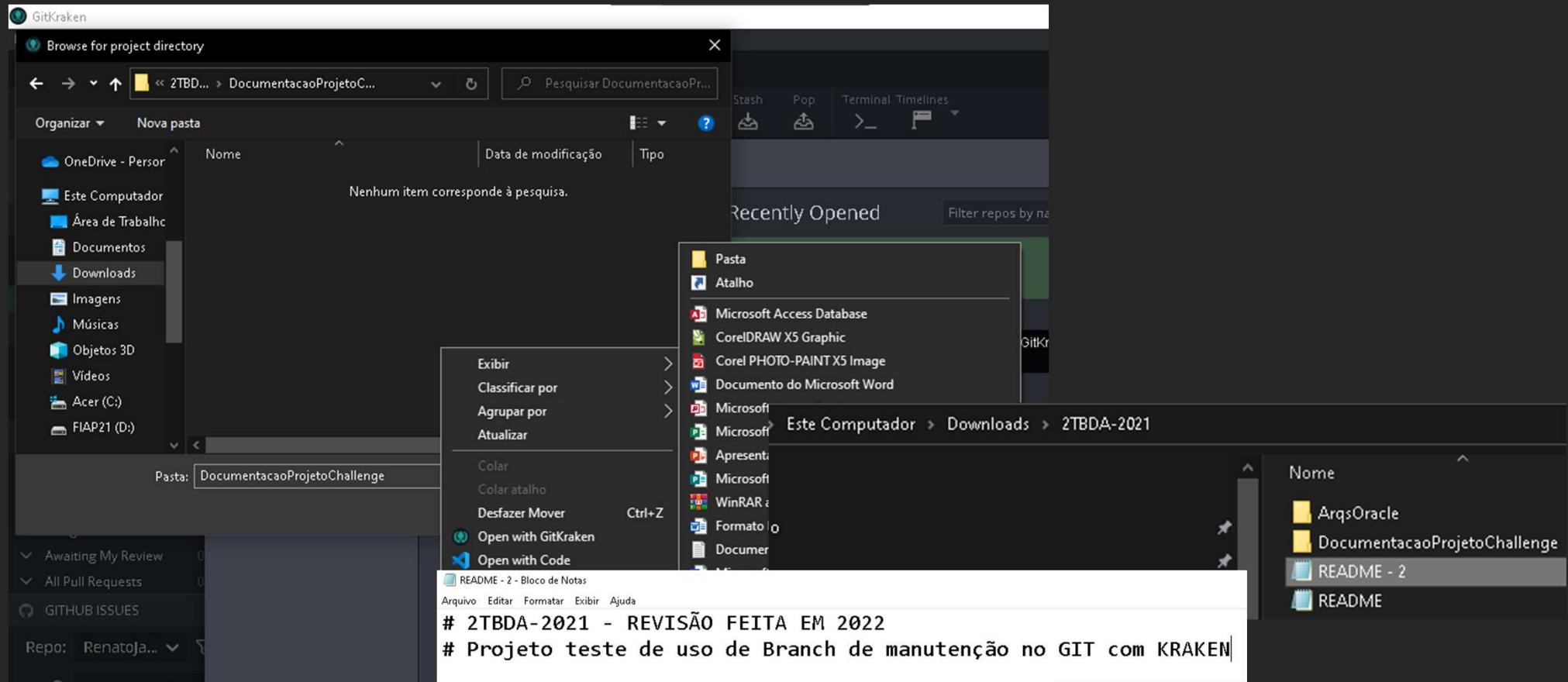


GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Crie, altere e exclua arquivos à vontade no repositório GIT local!

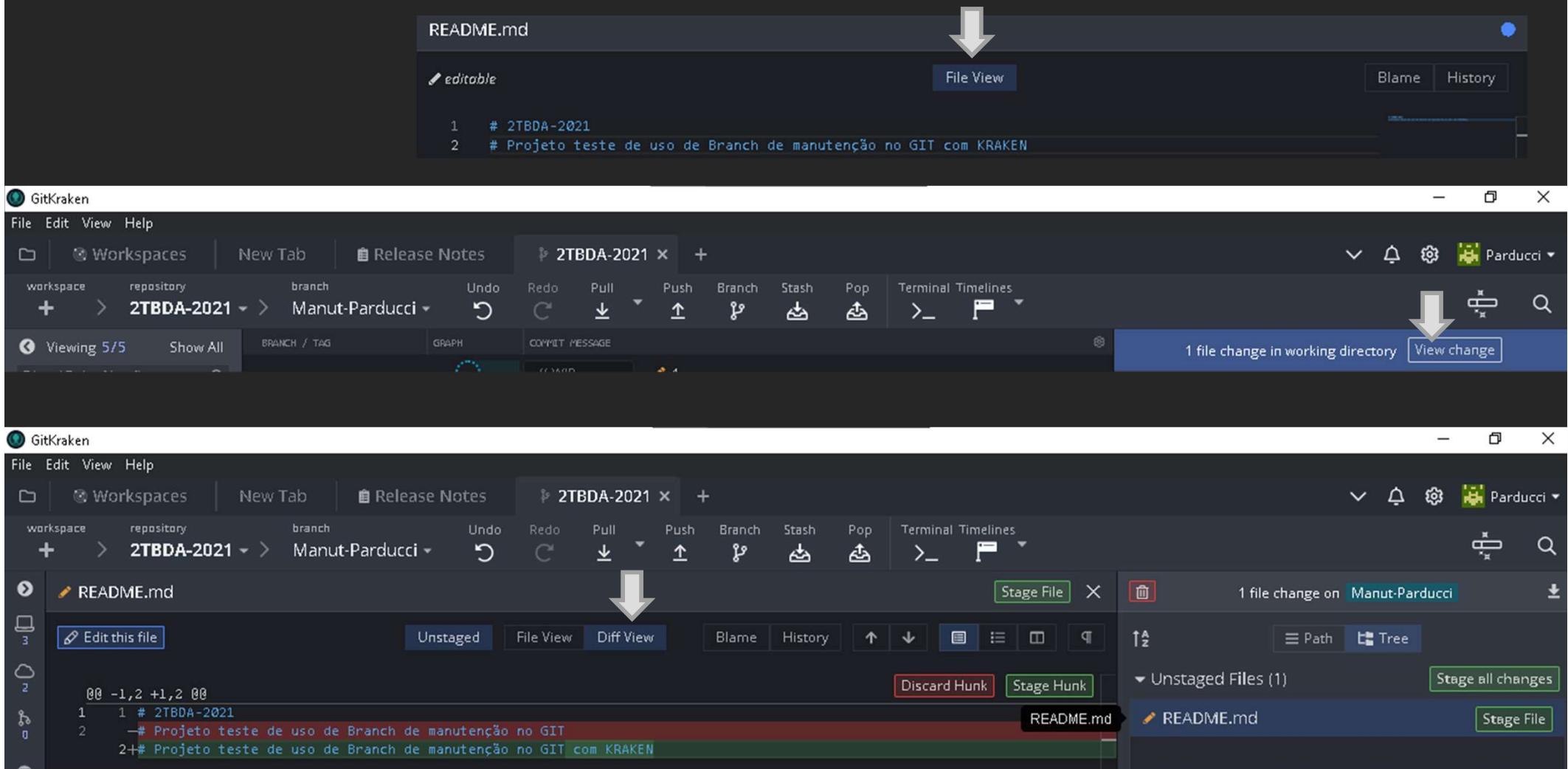
O Kraken reconecerá as mudanças!

APENAS GARANTA QUE VOCÊ ESTÁ NA BRANCH CORRETA, ANTES DE ALTERAR ARQUIVOS!



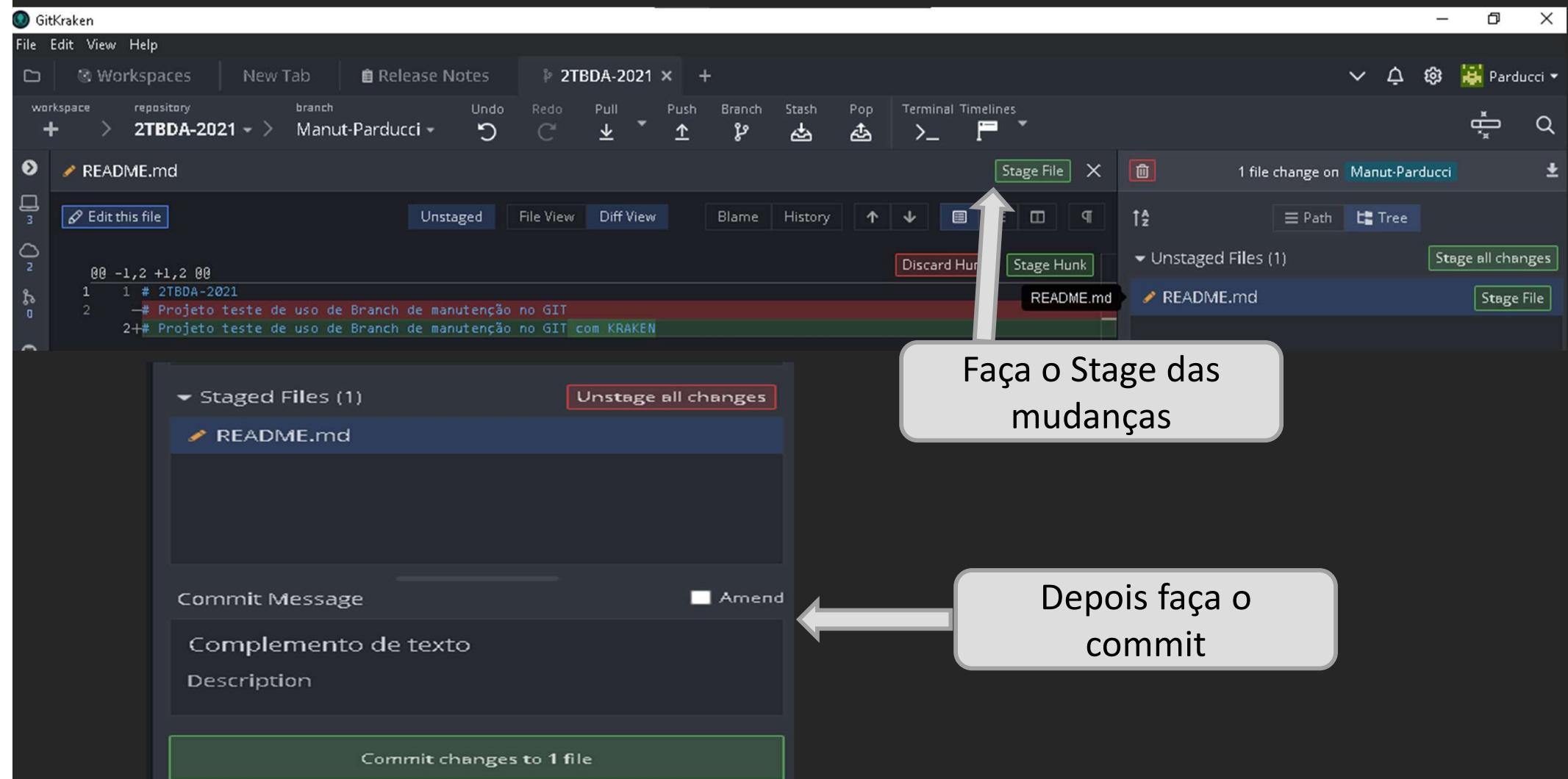
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Arquivos alterados são sinalizados na branch e basta clicar neles para ver as diferenças em relação à versão anterior.



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

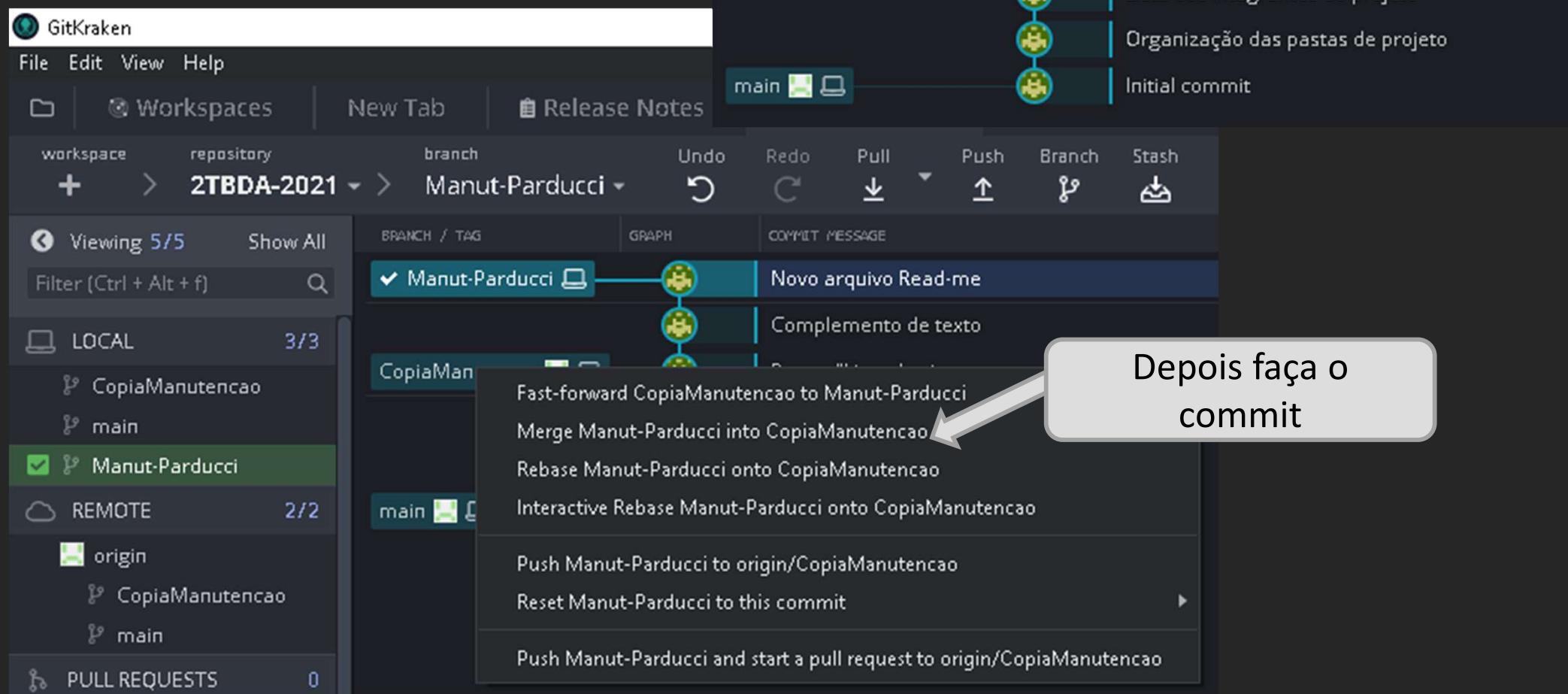
Arquivos alterados são sinalizados na branch e basta clicar neles para ver as diferenças em relação à versão anterior.



GERENCIAMENTO DA PRODUÇÃO

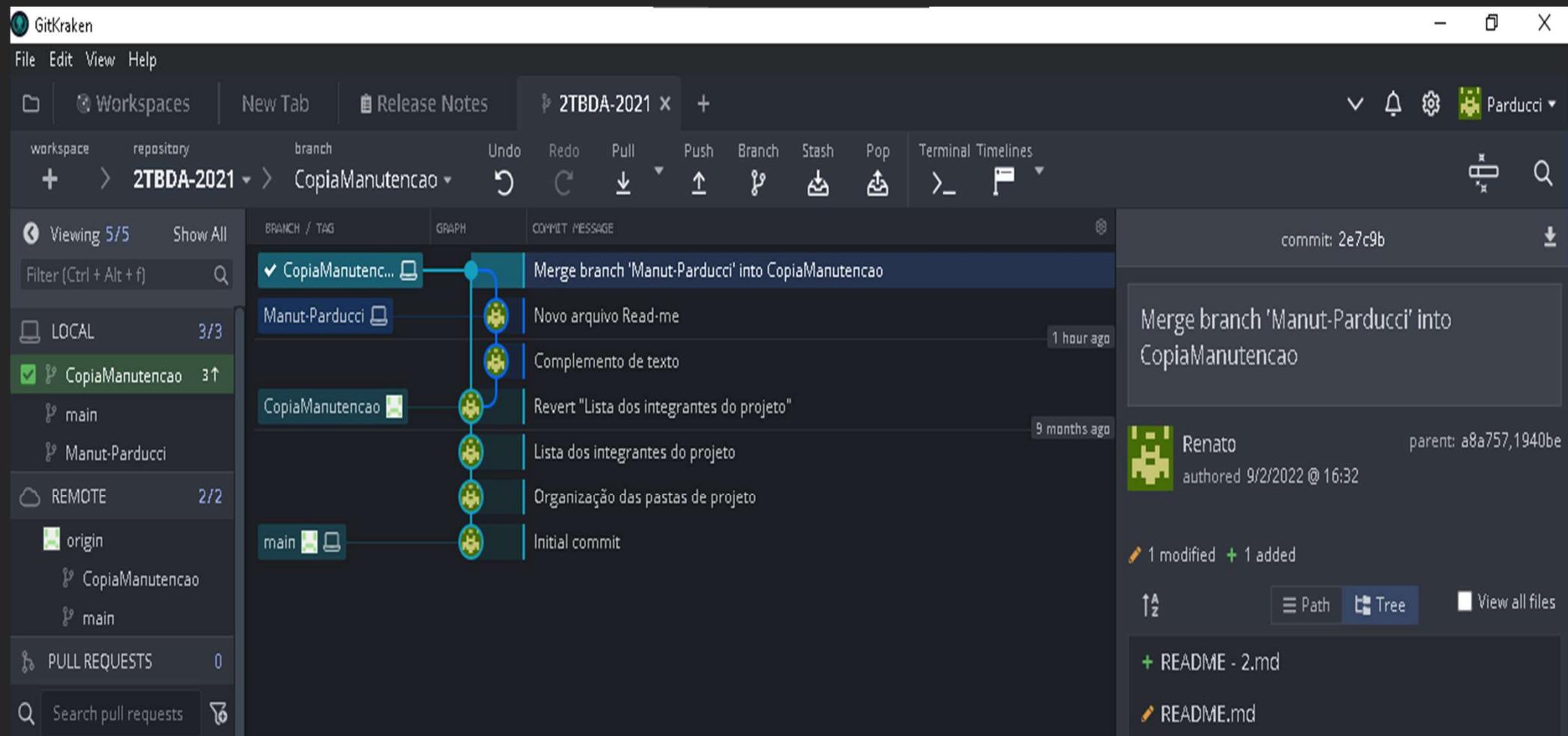
Para atualizar uma branch com as mudanças de uma outra branch com os novos arquivos e arraste sobre a branch que quer ser atualizada.

Depois, escolha o tipo de operação que quer realizar.



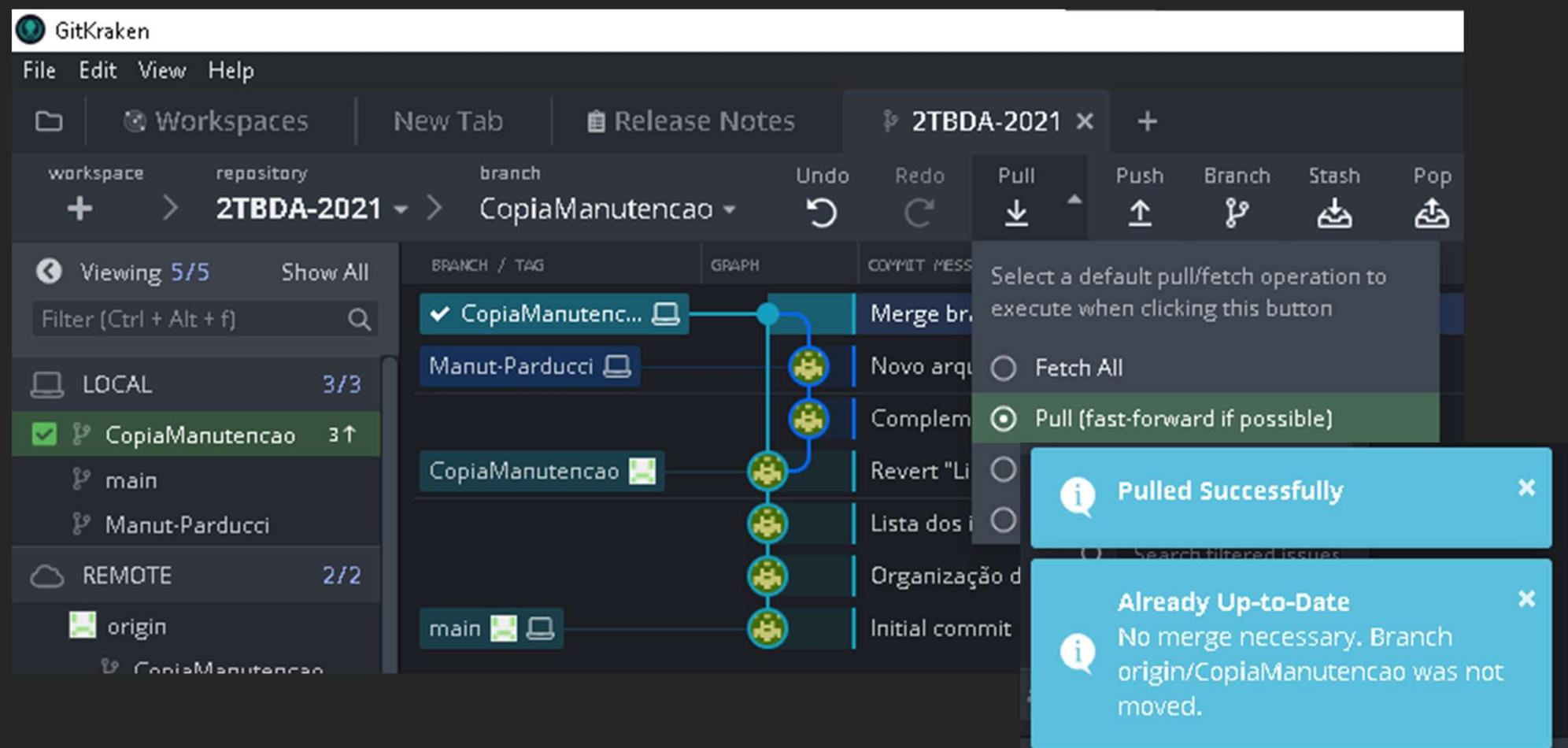
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Você deve atualizar a branch local que foi base para a sua branch de manutenção e depois, atualizar a branch remota, através de pull request!



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Depois, crie a Pull request para atualizar a branch na origem remota.



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Verifique o resultado no GITHUB (origem remota):

The screenshot shows a GitHub repository page. At the top, there is a purple header bar with the URL <https://github.com/RenatoJardimParducci/2TBDA-2021>. Below the header, the repository name 'RenatoJardimParducci / 2TBDA-2021' is displayed, along with a 'Public' badge, a 'Pin' button, and an 'Unstar' button. The main navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A prominent green notification bubble in the center says 'CopiaManutencao had recent pushes less than a minute ago' with a 'Compare & pull request' button. Below this, there are buttons for 'main' (with a dropdown arrow), '2 branches', '0 tags', 'Go to file', 'Add file', and 'Code'. The repository's history section shows a single commit by 'RenatoJardimParducci' titled 'Initial commit' with hash 'eb8c95d' and timestamp 'on 29 Apr 2021'. A file named 'README.md' is listed at the bottom.

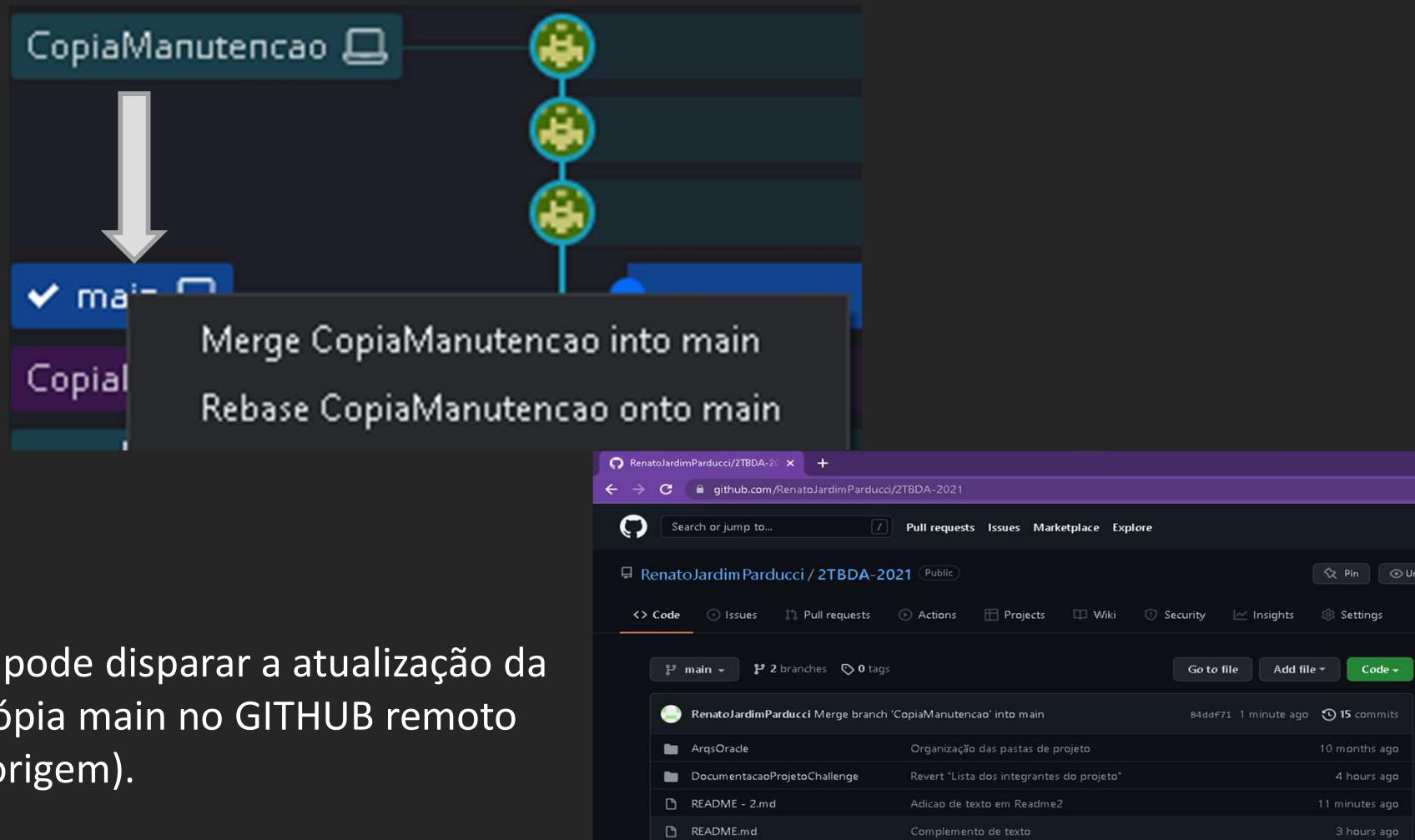
GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Atualize a Master do GITHUB (origem remota) quando estiver com os fontes terminados:

The screenshot shows a GitHub interface for creating a pull request. At the top, there's a navigation bar with links: Code, Issues, Pull requests (which is the active tab), Actions, Projects, Wiki, Security, Insights, and Settings. Below the navigation bar, the title "Open a pull request" is displayed. A sub-instruction says "Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#)." Underneath, there are dropdown menus for "base: main" and "compare: CopiaManutencao". A green checkmark icon indicates "Able to merge. These branches can be automatically merged." To the left of the compare dropdown is a small circular profile picture. To the right of the dropdown, a blue-bordered box contains the commit message: "Copia manutencao - alteração de Readme e inclusão de Readme-2".

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Você também pode atualizar a branch main local, arrastando a branch de manutenção para cima dela no navegador do KRAKEN e selecionando Merge ou Rebase “branch” into main.



E pode disparar a atualização da cópia main no GITHUB remoto (origem).

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Outra forma de usar o GIT é na linha de prompt do sistema operacional (opção que muitas vezes é a favorita de alguns desenvolvedores).

Para isso, instale o GIT CMD na sua máquina e ao rodá-lo, a tela de prompt do sistema operacional aparecerá e você poderá executar comandos nesse prompt.

A lista de comandos que podem ser usados via prompt estão relacionados a seguir.



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Lista de comandos do GIT CMD (considerando que o repositório já existe no GIT HUB):

Setar usuário

```
git config --global user.name "Seu nome"
```

Criando um novo branch

```
git branch bug-123
```

Setar email

```
git config --global user.email Seu e-mail (sem aspas)
```

Trocando para um branch existente

```
git checkout bug-123
```

Clonar um repositório remoto já existente

```
git clone Seu URL de Github
```

Apagando um branch

```
git branch -d bug-123
```

Verificar estado dos arquivos/diretórios

```
git status
```

Listar branches

Listar branches

```
git branch
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Lista de comandos do GIT CMD (considerando que o repositório já existe no GIT HUB):

Atualizar os arquivos no branch atual

```
git pull
```

Comitar arquivo/diretório

Comitar um arquivo

```
git commit meu_arquivo.txt
```

Adicionar arquivo/diretório (staged area)

Adicionar um arquivo em específico

```
git add meu_arquivo.txt
```

Comitar vários arquivos

```
git commit meu_arquivo.txt meu_outro_arquivo.txt
```

Adicionar um diretório em específico

```
git add meu_diretorio
```

Comitar informando mensagem

```
git commit meu_arquivo.txt -m "minha mensagem de commit"
```

Adicionar todos os arquivos/diretórios

```
git add .
```

Exibir histórico modificação de um arquivo

```
git log --diff-filter=M -- <caminho_do_arquivo>
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Lista de comandos do GIT CMD (considerando que o repositório já existe no GIT HUB):

Remover arquivo/diretório

Remover arquivo

```
git rm meu_arquivo.txt
```

Desfazendo alteração local (staging area)

Este comando deve ser utilizado quando o arquivo já foi adicionado na **staged area**.

```
git reset HEAD meu_arquivo.txt
```

Remover diretório

```
git rm -r diretorio
```

Se o resultado abaixo for exibido, o comando reset *não* alterou o diretório de trabalho.

```
Unstaged changes after reset:  
M     meu_arquivo.txt
```

Visualizar histórico

Exibir histórico

```
git log
```

Exibir histórico de um arquivo específico

```
git log -- <caminho_do_arquivo>
```

Enviar arquivos/diretórios para o repositório remoto

O primeiro **push** de um repositório deve conter o nome do repositório remoto e o branch.

```
git push -u origin master
```

Coloque o nome da Branch
que você atualizou para ela
ser levada ao GITHUB !!

Os demais **pushes** não precisam dessa informação

```
git push
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Lista de comandos do GIT CMD (considerando que o repositório já existe no GIT HUB):

Desconectar do GIT remoto:

```
git config --global --unset user.name  
git config --global --unset user.email
```

Ou:

```
git config --global --unset-all
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

Lista de comandos do GIT CMD (considerando que o repositório já existe no GIT HUB):

Se você desejar desfazer uma versão/atualização específica:

```
git revert 872fa7e
```

ID da versão a ser desfeita

Se você quer voltar para um ponto da história, esquecendo o que aconteceu depois:

```
git reset –hard a1e8fb5
```

ID da versão a ser
reestabelecida/recuperada

ASSISTA OS VÍDEOS NO CANAL DO PROFESSOR



https://youtu.be/aSpdQ_82S9k

A screenshot of a video player window. The video frame shows a man with a shaved head, wearing a blue polo shirt and a black headset with a microphone. He is speaking. The video player has a play button, a progress bar showing 0:03 / 25:30, and a search bar with the placeholder "Digite aqui para pesquisar". Below the video frame is a taskbar with icons for various applications like Google Chrome, Microsoft Word, and File Explorer. In the bottom right corner of the video frame, there is a small YouTube logo.



Conteúdo didático complementar - GIT-CMD

APROVEITE PARA CONTROLAR FONTES DE PROGRAMAS E DOCUMENTOS DE PROJETOS COM GIT/GIT HUB, DAQUI POR DIANTE!



ASSISTA OS VÍDEOS NO CANAL DO PROFESSOR

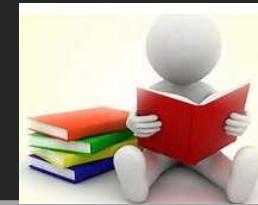


<https://youtu.be/wB6aLlwcb00>



GIT - Simulação completa com HUB, GUI, Desktop e CMD

ESTUDO DE CASO SIMULADO



A GD (Gerência de Desenvolvimento) da empresa de Dilan começou a experimentar o GIT. Algumas equipes preferem operar o GITHIUB diretamente, outras via cliente GitHubDesktop, outras preferem o GitGUI e outras preferem o GitCMD.

Com o uso das ferramentas, surgiu um problema: algumas vezes os desenvolvedores alteravam diretamente a cópia Master, outras vezes, sentiam falta de poder abrir várias Branchs de desenvolvimento à partir de um ambiente/Brench de manutenção geral, de forma a possibilitar testes integrados ao final dos desenvolvimentos de todos programadores para atender a uma Release de versão.

Você descobriu uma forma de trabalhar com o GIT para controlar melhor o fluxo de trabalho, usando o GIT FLOW e vai realizar uma atividade prática para trabalhar com seus colegas e ensiná-los a tecnologia aplicada ao gerenciamento de fontes e documentos de sistemas em desenvolvimento.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

Quando trabalhamos com grandes projetos ou quando temos vários programadores desenvolvendo partes de uma mesma solução que precisa ser integrada ao final, o ideal é trabalhar com um modelo de Branches em cascata (Branch de Branch), , separando os fontes estáveis dos instáveis e daqueles que ainda precisam de uma última bateria de testes.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

O GitFlow é um conjunto de programas embarcados no GIT CMD e permite o controle automático de situações que envolvem múltiplos níveis de Branch.

Ele evita esquecimentos e atualizações erradas. Impede que manutenções que ainda não estão efetivamente validadas sejam movidas para o diretório de fontes que são parte do produto de software estável, o qual é liberado para usuários.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

Vamos trabalhar com as seguintes BRANCHES que são apropriadas para grandes projetos:

- ⇒ ReleaseManufacture (nossa Master onde estão os fontes prontos para liberação para uso);
- ⇒ ReleaseCandidate (contém os fontes liberados pelos desenvolvedores para testes finais). Essa BRANCH acumula as mudanças promovidas por todos programadores no software;
- => Development-NOME DA TAREFA DE DESENVOLVIMENTO (contém os fontes que estão ainda em programação e testes pelos desenvolvedores). Essas BRANCHES devem ser excluídas quando a tarefa de produção do desenvolvedor acabar.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Execute o Git CMD

-Vá até uma pasta onde criaremos uma sub pasta com o nome gitflowExemplo (use os comandos do DOS – MD, CD, DIR para criar e conferir a criação)

ALTERNATIVAMENTE, VOCÊ PODE CLONAR UM REPOSITÓRIO GITHUB OU ADICIONAR DEPOIS ESSE LINK COM O HUB NA PASTA CRIADA.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Execute o comando git flow init para colocar o novo repositório dentro do controle GIT

```
C:\Users\renat\gitflowExemplo>git flow init
Initialized empty Git repository in C:/Users/renat/gitflowExemplo/.git/
No branches exist yet. Base branches must be created now.
```

-Aguarde a criação da pasta e depois, informe o nome da BRANCH que conterá os fontes na versão pronta para gerar a Release e fazer Deploy para liberar o uso (o nome default é Master e vamos renomear para ReleaseManufacture)

```
Branch name for production releases: [master] ReleaseManufacture
```

AGUARDE O PROMPT PEDINDO A DEFINIÇÃO DA BRANCH DE
DESENVOLVIMENTO

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Nomeie em seguida a BRANCH onde estarão os fontes prontos para testes finais pré-liberação
(o nome default é Development e vamos renomear para ReleaseCandidate)

```
Branch name for "next release" development: [develop] ReleaseCandidate
```

AGUARDE O PROMPT PEDINDO A DEFINIÇÃO DOS LABELS DE PREFIXO
DAS BRANCHES DE FEATURE, BUGFIX, HOTFIX, RELEASE, SUPPORT.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Deixe o padrão feature/ como prefixo de BRANCHES de manutenção (já sugerido pelo Git Flow) ou mude, se desejar e aceite o Bugfix (dê ENTER) e não vamos acrescentar TAGs

```
C:\Users\renat\gitflowExemplo>git flow init
Initialized empty Git repository in C:/Users/renat/gitflowExemplo/.git/
No branches exist yet. Base branches must be created now.
Branch name for production releases: [master] ReleaseManufacture
Branch name for "next release" development: [develop] ReleaseCandidate

How to name your supporting branch prefixes?
Feature branches? [feature/] DevelopmentFeatureTesteFlow
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/Users/renat/gitflowExemplo/.git/hooks]
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Ao terminar, o GitFlow vai posicionar você, automaticamente, no diretório de manutenção/feature

```
C:\Users\renat\gitflowExemplo>
```

-Veja as Branches existentes com o comando git branch

```
C:\Users\renat\gitflowExemplo>git branch
* ReleaseCandidate
  ReleaseManufacture
```

Perceba que dentro da sua pasta de tarefa chamada gitflowExemplo, existe a Branch ReleaseCandidate (Master) e a ReleaseManufacture (fontes liberados par os testes finais), as quais você pode usar.

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Crie agora uma feature que corresponde àquilo que você como desenvolvedor precisa realizar para cumprir uma tarefa de programação ou documentação de software que lhe foi designada

```
C:\Users\renat\gitflowExemplo>git flow feature start DocumentacaoProjeto
```

Summary of actions:

- A new branch 'DevelopmentFeatureTesteFlowDocumentacaoProjeto' was created, based on 'ReleaseCandidate'
- You are now on branch 'DevelopmentFeatureTesteFlowDocumentacaoProjeto'

Now, start committing on your feature. When done, use:

```
git flow feature finish DocumentacaoProjeto
```

-Confira a criação da Branch e veja que o GitFlow já posicionou você na Branch de programação/documentação.

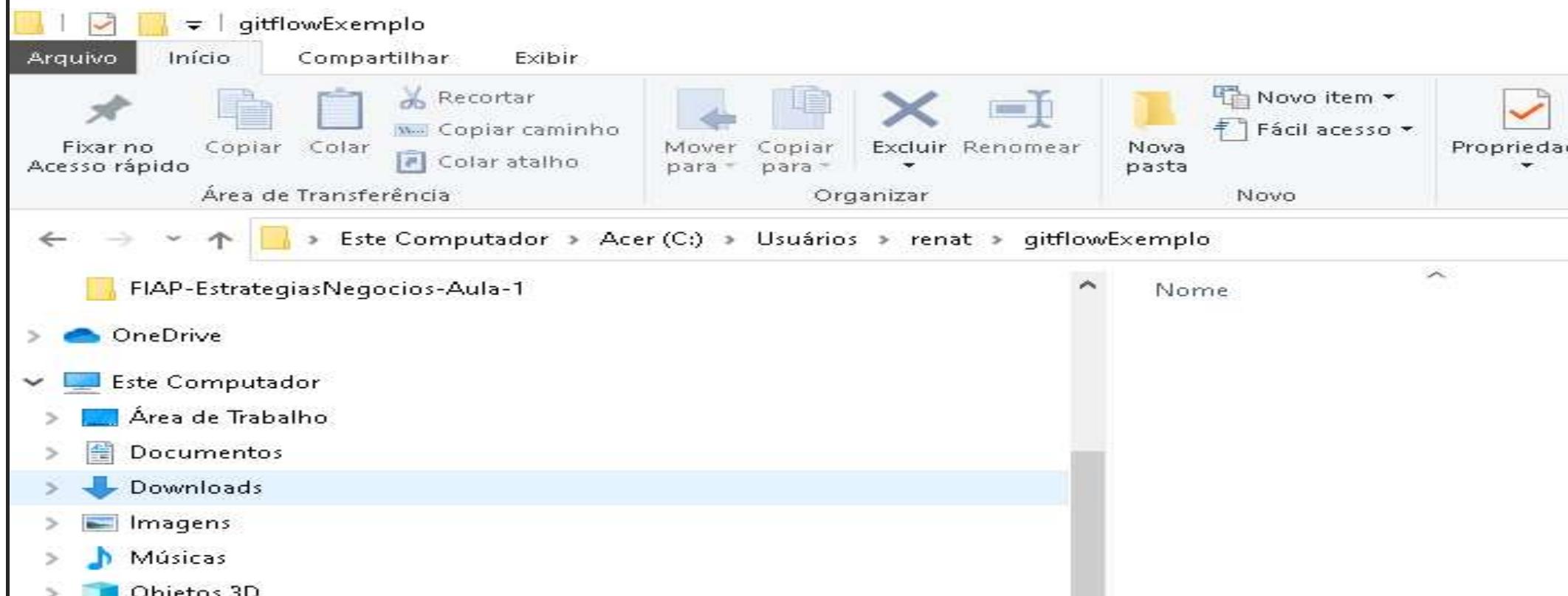
```
C:\Users\renat\gitflowExemplo>git branch
* DevelopmentFeatureTesteFlowDocumentacaoProjeto
  ReleaseCandidate
  ReleaseManufacture
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Agora, você pode usar editores de linguagem ou documentos, navegar via Windows Explorer, por exemplo e copiar, editar, criar arquivos à vontade. O Git vai mapear tudo o que você fizer no diretório do projeto, vinculando com a Branch de tarefa

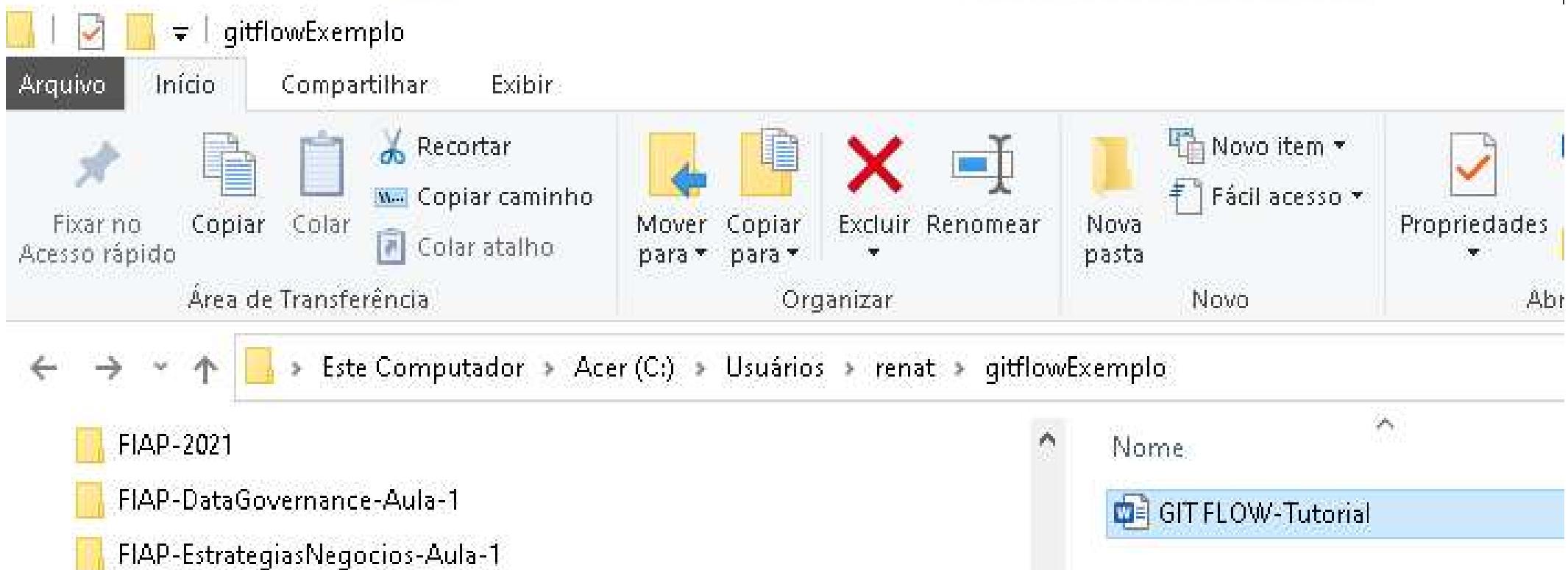
C:\Users\renat\gitflowExemplo>



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Experimente copiar um arquivo qualquer para dentro da pasta e daí para frente, podemos aplicar os comandos do Git Command já estudados (status, add, commit, push, pull, ...)



GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Git status identificará as mudanças em arquivos

```
C:\Users\renat\gitflowExemplo>git status
On branch DevelopmentFeatureTesteFlowDocumentacaoProjeto
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    GIT FLOW-Tutorial.docx

nothing added to commit but untracked files present (use "git add" to track)
```

-Adicione o arquivo para coloca-lo na lista de arquivos prontos para commit, com o Git add

```
C:\Users\renat\gitflowExemplo>git add .
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Confira se o arquivo foi adicionado com Git status

```
C:\Users\renat\gitflowExemplo>git status
On branch DevelopmentFeatureTesteFlowDocumentacaoProjeto
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   GIT FLOW-Tutorial.docx
```

-Agora, faça o commit para que o GitFlow publique o seu novo arquivo/versão nova, na branch de testes finais

```
C:\Users\renat\gitflowExemplo>git commit -m "Adicao do documento de kickoff do projeto"
[DevelopmentFeatureTesteFlowDocumentacaoProjeto 4920569] Adicao do documento de kickoff do projeto
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 GIT FLOW-Tutorial.docx
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

- Informe agora que você finalizou o desenvolvimento da sua tarefa/feature para que o Gitflow leve as atualizações sozinho para a Branch "superior" – a Branch de testes finais chamada ReleaseCandidate. Para isso, execute o comando Git flow feature finish *nome da feature/branch terminada*

```
C:\Users\renat\gitflowExemplo>git flow feature finish DocumentacaoProjeto
Switched to branch 'ReleaseCandidate'
Updating 347b852..4920569
Fast-forward
  GIT FLOW-Tutorial.docx | Bin 0 -> 73308 bytes
    1 file changed, 0 insertions(+), 0 deletions(-)
     create mode 100644 GIT FLOW-Tutorial.docx
Deleted branch DevelopmentFeatureTesteFlowDocumentacaoProjeto (was 4920569).

Summary of actions:
- The feature branch 'DevelopmentFeatureTesteFlowDocumentacaoProjeto' was merged into 'ReleaseCandidate'
- Feature branch 'DevelopmentFeatureTesteFlowDocumentacaoProjeto' has been locally deleted
- You are now on branch 'ReleaseCandidate'
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-o GitFlow já deleta a sua Branch de manutenção/tarefa e atualiza a Branch superior. Ele também posiciona você na Branch superior. Confira as mudanças na Branch superior com Git branch e Git status

```
C:\Users\renat\gitflowExemplo>git branch
* ReleaseCandidate
  ReleaseManufacture
```

```
C:\Users\renat\gitflowExemplo>git status
On branch ReleaseCandidate
nothing to commit, working tree clean
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

Perceba que o arquivo que você alterou/criou/excluiu, já foi modificado e “comitado” na Branch superior. Confira com o comando Dir

```
C:\Users\renat\gitflowExemplo>dir
O volume na unidade C é Acer
O Número de Série do Volume é 2671-A4C5

Pasta de C:\Users\renat\gitflowExemplo

07/03/2021  10:36    <DIR>          .
07/03/2021  10:36    <DIR>          ..
07/03/2021  10:36           73.308 GIT FLOW-Tutorial.docx
                           1 arquivo(s)      73.308 bytes
                           2 pasta(s)   738.099.294.208 bytes disponíveis
```

-Se você executar git flow feature start de novo, o GitFlow vai criar uma nova Branch de manutenção com as últimas versões da ReleaseCandidate

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Quando as manutenções ligadas a uma Release acabarem, você deve disparar a geração da Release com o comando Git flow release start *número da release*

```
C:\Users\renat\gitflowExemplo>git flow release start R01.01
Switched to a new branch 'release/R01.01'
```

Summary of actions:

- A new branch 'release/R01.01' was created, based on 'ReleaseCandidate'
- You are now on branch 'release/R01.01'

Follow-up actions:

- Bump the version number now!
- Start committing last-minute fixes in preparing your release
- When done, run:

```
git flow release finish 'R01.01'
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Nesse momento o GitFlow criou uma Branch de release para que o software receba a carga de testes finais, pré-liberação. Confira com Git branch.

```
C:\Users\renat\gitflowExemplo>git branch
  ReleaseCandidate
  ReleaseManufacture
* release/R01.01
```

Quando você acessar os arquivos da pasta de projeto, agora estará operando sobre a branch da release

```
C:\Users\renat\gitflowExemplo>dir
O volume na unidade C é Acer
O Número de Série do Volume é 2671-A4C5

Pasta de C:\Users\renat\gitflowExemplo

07/03/2021 10:36    <DIR>      .
07/03/2021 10:36    <DIR>      ..
07/03/2021 10:36            73.308 GIT FLOW-Tutorial.docx
                           1 arquivo(s)      73.308 bytes
                           2 pasta(s)   738.100.359.168 bytes disponíveis
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Quando acabar os testes, feche a Release de forma parecida com o que fez com a Feature: Git flow release finish nome da release a fechar -m "mensagem que descreve a release"

```
C:\Users\renat\gitflowExemplo>git flow release finish R01.01 -m "100% testes finalizados da 1a Release"
Switched to branch 'ReleaseManufacture'
Switched to branch 'ReleaseCandidate'
Already up to date!
Merge made by the 'recursive' strategy.
Deleted branch release/R01.01 (was 4920569).

Summary of actions:
- Release branch 'release/R01.01' has been merged into 'ReleaseManufacture'
- The release was tagged 'R01.01'
- Release tag 'R01.01' has been back-merged into 'ReleaseCandidate'
- Release branch 'release/R01.01' has been locally deleted
- You are now on branch 'ReleaseCandidate'
```

-Agora, a Branch da Release é destruída e você vai para a ReleaseCandidate, automaticamente.
Confira com o comando Git branch

```
C:\Users\renat\gitflowExemplo>
C:\Users\renat\gitflowExemplo>git branch
* ReleaseCandidate
  ReleaseManufacture
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-OBS: SE VOCÊ ESQUECER DE COLOCAR O -M “MENSAGEM”, O GIT ABRIRÁ UM EDITOR PARA VOCÊ COLOCAR OS COMENTÁRIOS. ADICIONE O QUE DESEJAR NO TEXTO E DEPOIS DIGITE “ESC” E EM SEGUIDA “:wq!” E ENTER DE NOVO.

-Execute Git tag para verificar que o Git já vinculou a identificação da Release com o conteúdo que se encontra na ReleaseCandidate.

```
C:\Users\renat\gitflowExemplo>git tag  
R01.01
```

```
C:\Users\renat\gitflowExemplo>git branch  
* ReleaseCandidate  
  ReleaseManufacture
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Quando você terminou a Branch de release, o GitFlow TAMBÉM ATUALIZOU A MASTER (ReleaseManufacture). Confira, mudando para a Branch ReleaseManufacture e listando os arquivos com DIR. Você pode também usar o Explorer do Windows para conferir os arquivos publicados na cópia mestre.

```
C:\Users\renat\gitflowExemplo>git checkout ReleaseManufacture
Switched to branch 'ReleaseManufacture'

C:\Users\renat\gitflowExemplo>dir
O volume na unidade C é Acer
O Número de Série do Volume é 2671-A4C5

Pasta de C:\Users\renat\gitflowExemplo

07/03/2021  11:04    <DIR>          .
07/03/2021  11:04    <DIR>          ..
07/03/2021  11:04                73.308 GIT FLOW-Tutorial.docx
                           1 arquivo(s)   73.308 bytes
                           2 pasta(s)   738.099.748.864 bytes disponíveis
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Se você precisar fazer um **HOTFIX** em um fonte que já está em produção (correção rápida de um BUG), você pode criar uma Branch diretamente à partir da Master e publicar diretamente na Master quando terminar.

```
C:\Users\renat\gitflowExemplo>git flow hotfix start CorrecaoKickoff
Switched to a new branch 'hotfix/CorrecaoKickoff'
```

Summary of actions:

- A new branch 'hotfix/CorrecaoKickoff' was created, based on 'ReleaseManufacture'
- You are now on branch 'hotfix/CorrecaoKickoff'

Follow-up actions:

- Start committing your hot fixes
- Bump the version number now!
- When done, run:

```
git flow hotfix finish 'CorrecaoKickoff'
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-O Git vai mover você para o diretório de correção. Confira com Git branch.

```
C:\Users\renat\gitflowExemplo>git branch
  ReleaseCandidate
  ReleaseManufacture
* hotfix/CorrecaoKickoff
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

-Modifique o que precisa (execute `git add` e `commit` para confirmar as mudanças)

```
C:\Users\renat\gitflowExemplo>git status
On branch hotfix/CorrecaoKickoff
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

    modified:   GIT FLOW-Tutorial.docx

no changes added to commit (use "git add" and/or "git commit -a")

C:\Users\renat\gitflowExemplo>git diff
diff --git a/GIT FLOW-Tutorial.docx b/GIT FLOW-Tutorial.docx
index d67755a..064f85b 100644
--- a/GIT FLOW-Tutorial.docx
+++ b/GIT FLOW-Tutorial.docx
@@ -1,4 +1,4 @@
-GIT FLOW
+GIT FLOW - TUTORIAL EXTRA
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

```
C:\Users\renat\gitflowExemplo>git add .

C:\Users\renat\gitflowExemplo>git status
On branch hotfix/CorrecaoKickoff
Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    modified:   GIT FLOW-Tutorial.docx

C:\Users\renat\gitflowExemplo>git commit -m "Adicao de detalhes no documento"
[hotfix/CorrecaoKickoff b82793a] Adicao de detalhes no documento
  1 file changed, 1 insertion(+)
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

- Ao terminar, finalize o Hot fix com Git flow hotfix finish -m "mensagem sobre a manutenção"

```
C:\Users\renat\gitflowExemplo>git flow hotfix finish -m "Texto de kickoff corrigido"
Switched to branch 'ReleaseManufacture'
Merge made by the 'recursive' strategy.
  GIT FLOW-Tutorial.docx | Bin 73308 -> 73133 bytes
    1 file changed, 0 insertions(+), 0 deletions(-)
Switched to branch 'ReleaseCandidate'
Merge made by the 'recursive' strategy.
  GIT FLOW-Tutorial.docx | Bin 73308 -> 73133 bytes
    1 file changed, 0 insertions(+), 0 deletions(-)
Deleted branch hotfix/CorrecaoKickoff (was b82793a).
```

Summary of actions:

- Hotfix branch 'hotfix/CorrecaoKickoff' has been merged into 'ReleaseManufacture'
- The hotfix was tagged 'CorrecaoKickoff'
- Hotfix tag 'CorrecaoKickoff' has been back-merged into 'ReleaseCandidate'
- Hotfix branch 'hotfix/CorrecaoKickoff' has been locally deleted
- You are now on branch 'ReleaseCandidate'

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

Veja de a Branch de hotfix foi excluída automaticamente.

-**Após finalizar o Hot fix, o Git flow atualiza automaticamente a cópia Master (nossa ReleaseManufacture) e a cópia de teste finais (ReleaseCandidate), da mesma forma que acontece quando você finaliza um desenvolvimento normal, usando releases tradicionais (não hot fix). Você pode checar isso, navegando de uma branch para outra e observando o horário de atualização do arquivo corrigido e/ou observando o seu conteúdo.**

```
C:\Users\renat\gitflowExemplo>git branch
* ReleaseCandidate
  ReleaseManufacture

C:\Users\renat\gitflowExemplo>
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

```
C:\Users\renat\gitflowExemplo>dir
O volume na unidade C é Acer
O Número de Série do Volume é 2671-A4C5

Pasta de C:\Users\renat\gitflowExemplo

07/03/2021  11:38    <DIR>          .
07/03/2021  11:38    <DIR>          ..
07/03/2021  11:38            73.133 GIT FLOW-Tutorial.docx
                           1 arquivo(s)   73.133 bytes
                           2 pasta(s)   738.106.187.776 bytes disponíveis

C:\Users\renat\gitflowExemplo>git checkout ReleaseManufacture
Switched to branch 'ReleaseManufacture'

C:\Users\renat\gitflowExemplo>dir
O volume na unidade C é Acer
O Número de Série do Volume é 2671-A4C5

Pasta de C:\Users\renat\gitflowExemplo

07/03/2021  11:38    <DIR>          .
07/03/2021  11:38    <DIR>          ..
07/03/2021  11:38            73.133 GIT FLOW-Tutorial.docx
                           1 arquivo(s)   73.133 bytes
                           2 pasta(s)   738.106.187.776 bytes disponíveis
```

GERENCIAMENTO DA PRODUÇÃO E ATUALIZAÇÃO DE SOFTWARE

GIT FLOW APLICADO

Agora você conhece ferramentas e melhores práticas mais populares e eficientes para gerenciar programas e documentos durante o desenvolvimento de um software



D Ú V I D A S

Referência bibliográficas

BIBLIOGRAFIA:

- **ISACA.** USA, COBIT 5 . 2014 - Disponível para acesso online gratuito em ISACA.org.
- **WEILL**, Peter. ROSS Jeane W. Governança de TI. Makron Books.
- **PRESSMAN**, Roger S.. Engenharia de software. - Uma abordagem profissional, 7^a edição. São Paulo, AMGH.
- **HIRAMA, Kichi.** Engenharia de Software: qualidade e produtividade com tecnologia. Editora Elsevier, Rio de Janeiro.
- **BOEHM, Barry.** Software Engineering Economics. Prentice Hall, USA.

