

Conceitos importantes do Git/GitHub

Repository: Local onde fica todos os arquivos do projeto, inclusive o histórico de versões.

Commit: Coleção e controle das alterações realizadas. É como se fosse um “checkpoint” do projeto. Sempre que for necessário é possível retroceder um commit.

Revert: Reverte (retrocede) um commit efetuado.

Branch: É uma ramificação do seu projeto. Cada branch representa uma versão do seu projeto. Podemos seguir uma linha de desenvolvimento a partir de cada branch.

Fork: Basicamente é uma bifurcação, uma cópia de um projeto existente para seguir em uma nova direção.

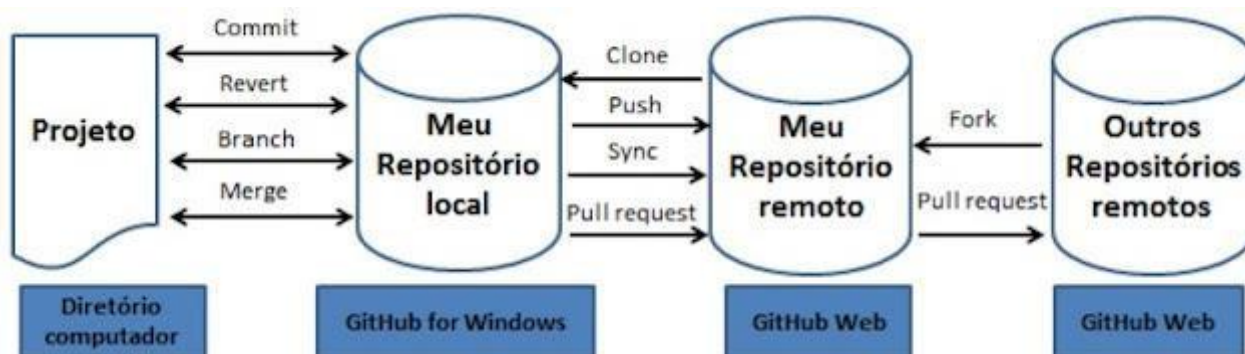
Push (Publish): Envia os commits dos arquivos locais para um repositório remoto.

Clone: Obtém uma cópia de um repositório remoto. Quando desejar colaborar com algum projeto de software livre utilize esta opção.

Pull: Atualiza as alterações locais efetuadas com o repositório remoto.

Merge: É a capacidade de incorporar alterações do Git, onde acontece uma junção dos branches.

Segue abaixo diagrama das interações das funcionalidades do Git/GitHub.



Utilizando o GitHub

É possível utilizar as funções mais básicas do GitHub sem instalar nenhum programa. Criar um repositório (uma pasta com seus arquivos), colocar seus scripts online (copiando e colando manualmente), e criar uma cópia de um repositório de outra pessoa. Todas essas opções podem ser feitas diretamente no site.

Criando uma conta no GitHub

Para criar sua conta no GitHub Web acesse o site: <https://github.com/>

Em seguida clique no botão **Sign up**. Na tela da Figura 1 preencha os campos abaixo:

Username: Informe o nome do seu usuário.

E-mail Address: Informe seu endereço de e-mail.

Password: Informe sua senha.

Após informar corretamente os campos clique no botão **Create an account**. Após isso sua conta será criada no GitHub Web.

Join GitHub

The best way to design, build, and ship software.

Step 1:
Set up a personal account

Step 2:
Choose your plan

Step 3:
Go to your dashboard

Create your personal account

Username

This will be your username — you can enter your organization's username here.

Email Address

You will occasionally receive account-related emails. We promise not to share your email with anyone.

Password

Use at least one lowercase letter, one numeral, and seven characters.

You'll love GitHub

Unlimited collaborators

Unlimited public repositories

✓ Great communication

✓ Friction-less development

✓ Open source community

Figura 1 – Criando uma conta no GitHub Web

Acessando o GitHub

Para usar o GitHub Web acesse o site: <https://github.com>

Em seguida clique no botão **Sign in**. Informe seu usuário e senha e clique no botão **Sign in**.

Criando um repositório no GitHub


Após criar sua conta no GitHub e acessá-la, o próximo passo é criar um repositório. Para criar um repositório é só clicar em no **+** no canto superior direito (Figura 2), e escolher a opção **New repository**.



Ao clicar na opção **New repository** será exibida a tela da Figura 3 abaixo. No campo **Repository name** informe o nome que desejar para o repositório (geralmente separados por hífen, como por exemplo, “meu-primeiro-repositorio”). No campo opcional **Description** coloque uma descrição se assim desejar e para finalizar clique em **Create repository**.

Create a new repository

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

Owner:  carlaoliveira / Repository name:

Great repository names are short and memorable. Need inspiration? How about studios-waffle

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

Figura 3 – Criando um repositório

Fazendo um fork de um repositório no GitHub

Para fazer um fork de um repositório de outra pessoa para a sua conta também é bem fácil. Para isso pesquise o projeto que desejado. Quando localizar o projeto clique no link desse projeto como mostra a Figura 4.



Figura 4 – Pesquisando um projeto

Ao clicar no link será exibida uma tela similar à tela da Figura 5. Em seguida clique em **Fork** no canto superior direito. Após isso o repositório desejado estará na sua lista de repositórios.

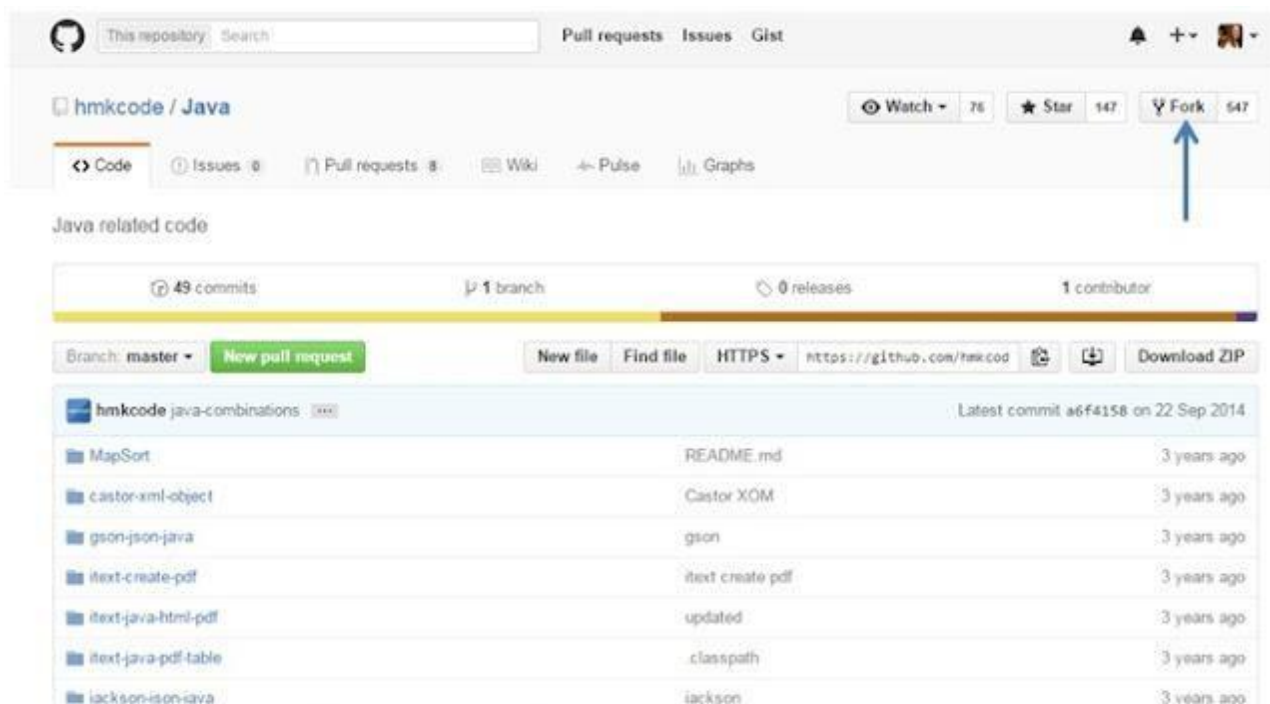


Figura 5 – Clonando (copiando) o repositório de outra pessoa

Efetuando commit no GitHub

Para adicionar ou modificar arquivos, você precisa fazer um *commit*. Vamos ver um exemplo. Escolha o projeto e em seguida clique no arquivo que deseja alterar conforme mostra a Figura 6.

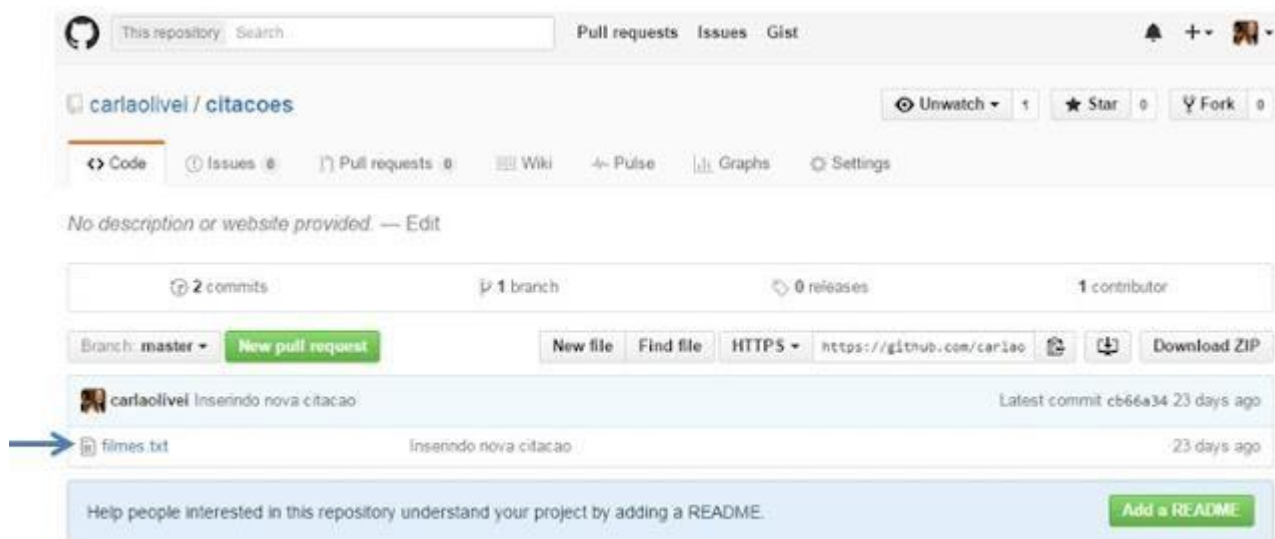


Figura 6 – Escolha do arquivo que será alterado

Quando abrir a tela da Figura 7 clique no ícone em forma de lápis, disponível à direita para habilitar a edição do arquivo.

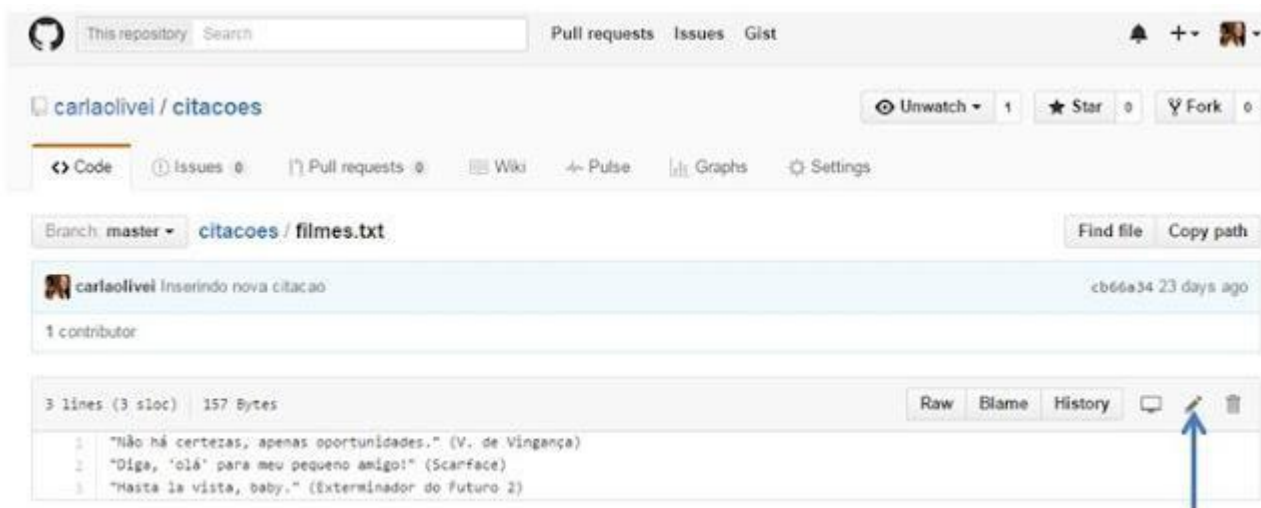


Figura 7 – Modo de edição do arquivo que será alterado

Edite o arquivo conforme desejar e na parte inferior da tela (Figura 8) efetue o commit. Para isso informe o texto do commit, uma descrição opcional e em seguida clique no botão **Commit changes** conforme Figura 8. Após isso as alterações efetuadas no seu arquivo serão executadas.

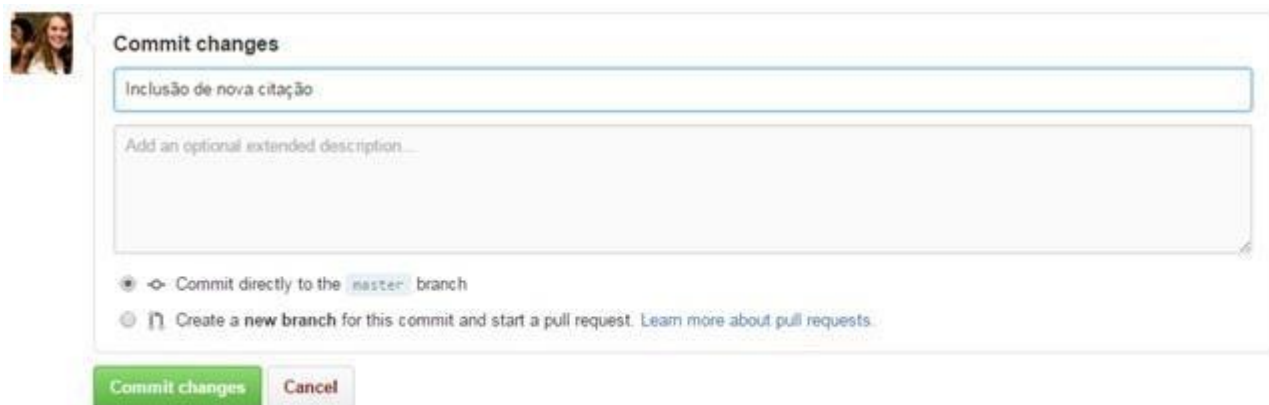


Figura 8 – Efetuado commit das alterações efetuadas no arquivo

Com isso você já pode usar o GitHub Web. No entanto, para usar o todo o potencial do Git é necessário instalar o programa no seu computador.

O Git tem versões para Windows, MacOS e Linux, então não há problema quanto à compatibilidade.

Para facilitar a vida do usuário a equipe do GitHub desenvolveu uma aplicação do GitHub para o Windows (GitHub for Windows) descrita abaixo.

GitHub for Windows

Instalando o GitHub for Windows

Para instalar o GitHub For Windows, precisamos baixar a aplicação. Para isso acesse o site: <http://windows.github.com> conforme mostra a Figura 9.

Em seguida clique no botão **Download GitHub Desktop**.

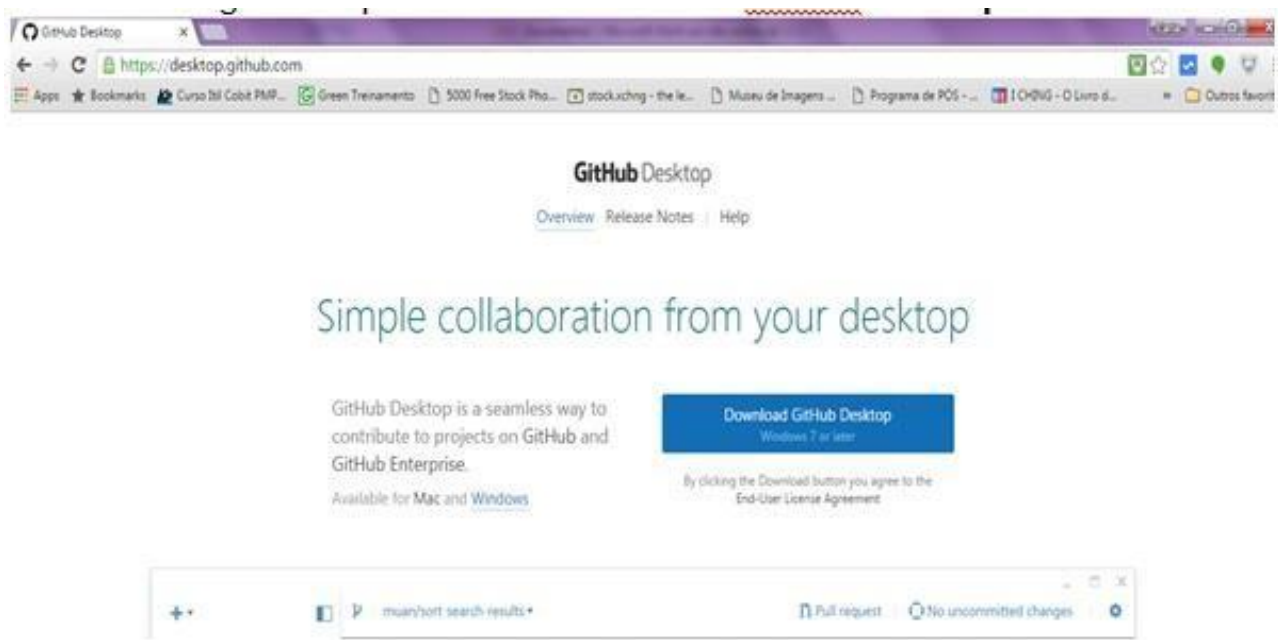


Figura 9 – Tela de Download do GitHub for Windows

NOTA: VERSÕES SUPORTADAS: O GitHub for Windows funciona apenas no Windows Vista, Windows 7 e Windows 8, Windows 8.1, Windows 10 e Windows 11. Existem versões para Mac e Linux, também.

Após a conclusão do download do arquivo “**GitHubSetup.exe**”, devemos executá-lo clicando duas vezes sobre ele.

Ao clicar duas vezes no arquivo “**GitHubSetup.exe**” será exibida a tela de instalação.

Após a conclusão da instalação será exibida uma tela onde deve ser informado o usuário e senha cadastrados no **GitHub Web**. Após informar o **usuário** e **senha** clicar no botão **Log in**.

Na próxima tela informar o **Nome** e o **E-mail** do usuário Git. Em seguida clicar no botão **continue**.

Será exibida uma tela onde serão listados os repositórios Git encontrados no computador. Caso não seja encontrado nenhum repositório, será exibida uma mensagem informando que não foram encontrados repositórios. Neste caso, clicar no botão **Skip**, para adicionar os repositórios posteriormente.

Após clicar no botão **Skip**, será direcionado para a tela principal do GitHub for Windows.

Até aqui a aplicação foi instalada. Se verificar na área de trabalho do Windows foram adicionados dois novos atalhos, sendo um chamado **GitHub**, que serve para executar o GitHub for Windows, e o outro chamado **Git Shell**, para executar o Git via prompt de comando.

IMPORTANTE: O GitHub for Windows criará um novo diretório (pasta) chamado **GitHub**, localizado na pasta **Documentos** do usuário (C:\Users\nome\Documents\GitHub). Esse é o diretório padrão onde os novos repositórios serão criados.

Criando um novo repositório

Agora que o GitHub já foi instalado e configurado, podemos criar um novo repositório Git. Para isto, basta clicar no botão **+** localizado na tela principal da aplicação conforme mostra a Figura 10.

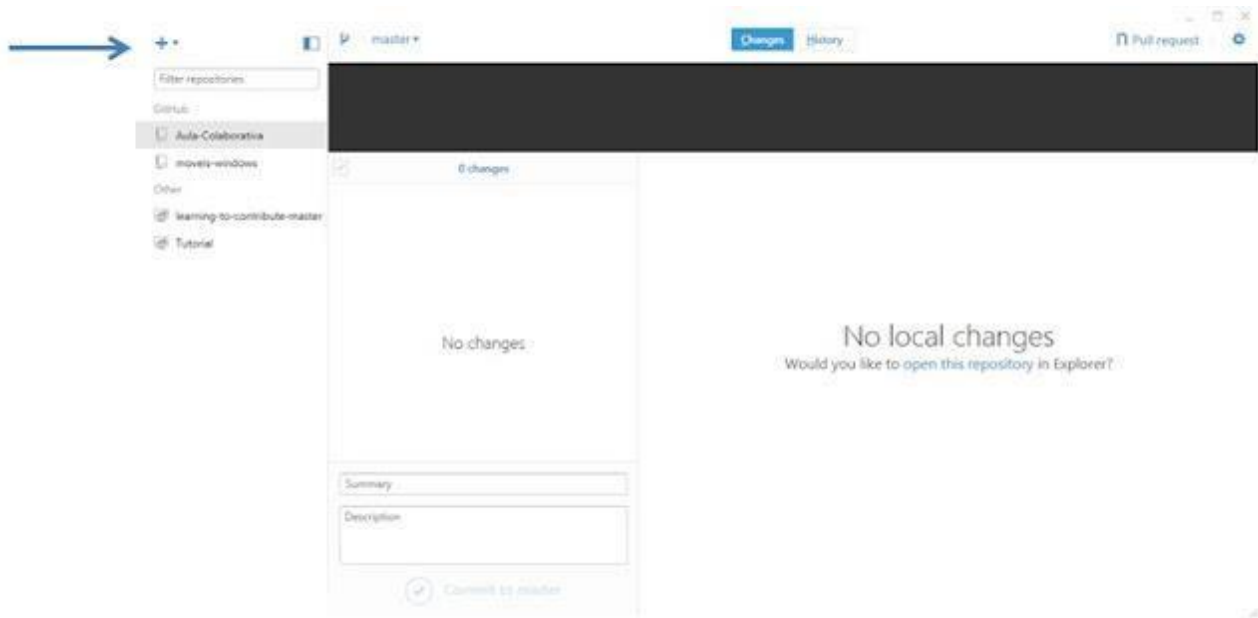


Figura 10 – Novo repositório no GitHub for Windows

Ao clicar no botão + será exibida a tela da Figura 11.

No campo **Nome** preencher com o nome do repositório, no nosso caso preencher com **Laboratorio**. No campo **Local path** escolher o diretório onde o repositório deverá ser criado. Neste caso, a sugestão é deixar preenchido com o diretório padrão.

Observe que há também o campo chamado **Git ignore**, onde é possível escolher a linguagem de programação utilizada no projeto, dentre as opções disponíveis, e com isto o arquivo **.gitignore** será criado automaticamente. No nosso caso selecione a opção **None**.

Após preencher os campos, clicar no botão **Create repository**. Aguardar a criação do repositório e então o mesmo será listado na tela.

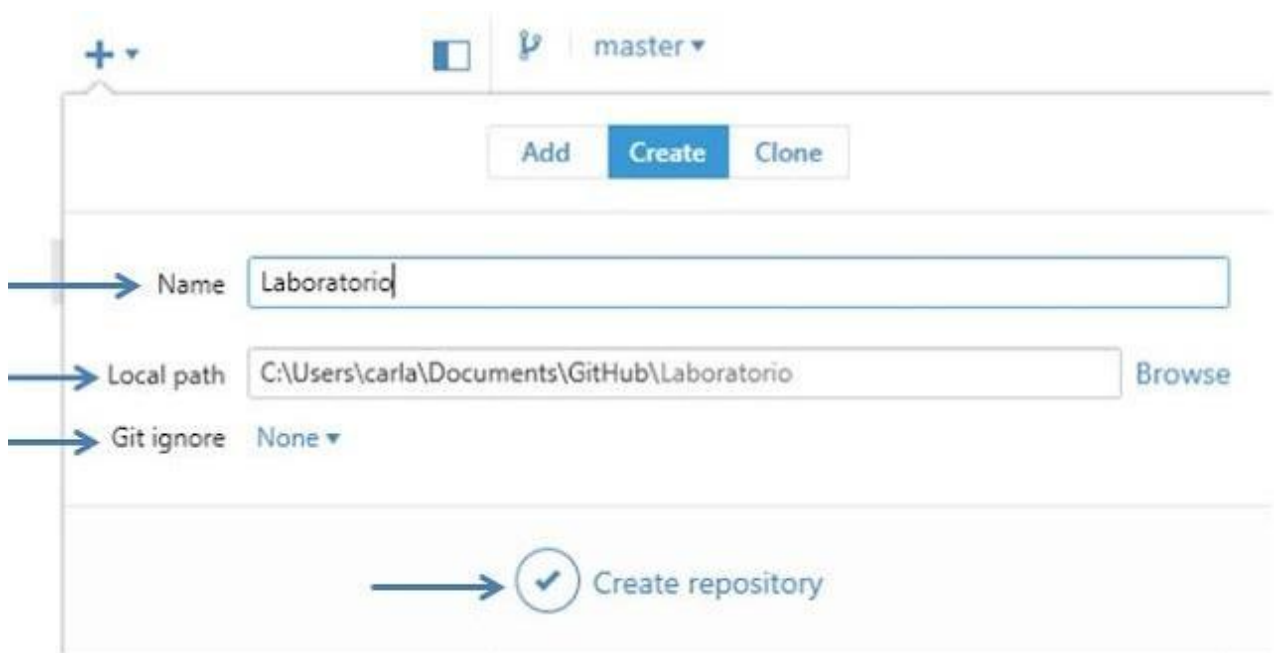


Figura 11 – Criando um novo repositório

Após o repositório ser criado repare que a tela está dividida em três colunas conforme mostra a Figura 12. Na primeira são listados os repositórios Git. Na segunda os commits do repositório juntamente com um formulário para efetuar um novo commit. Na última coluna são exibidos os arquivos do repositório.

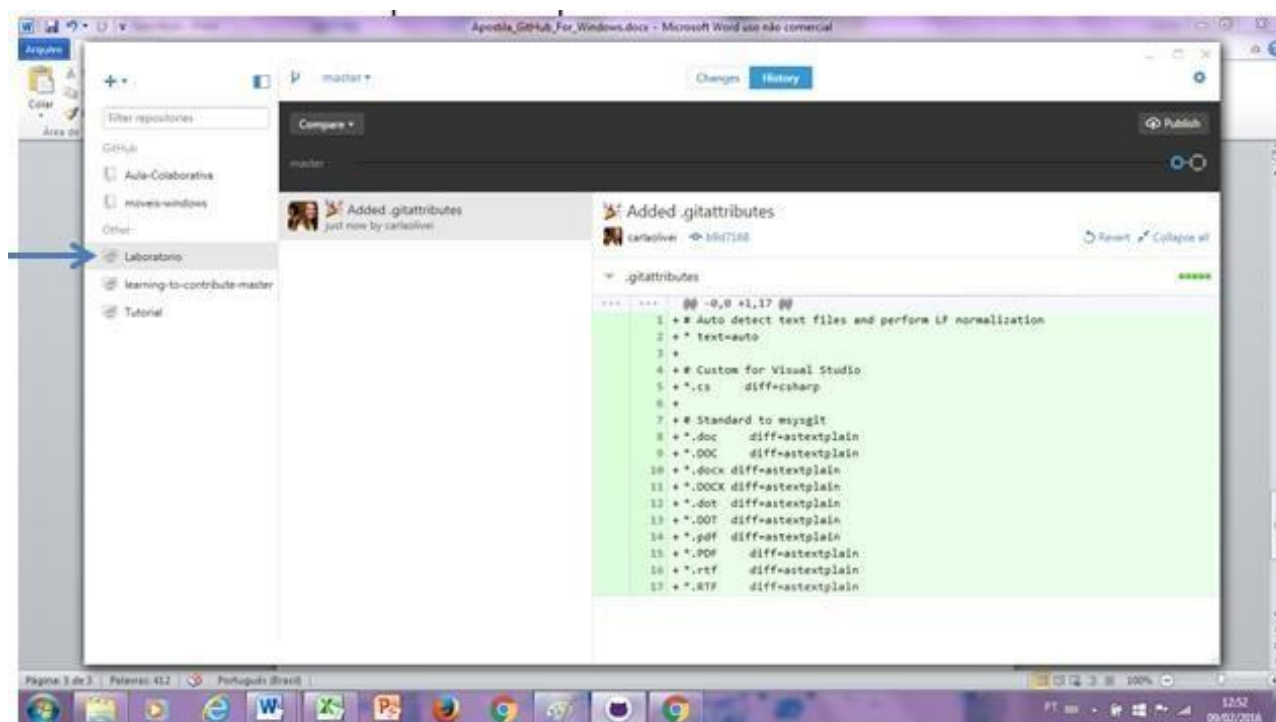


Figura 12 – Tela do novo repositório criado

Efetuando commits no repositório

Agora vamos criar um novo arquivo no repositório **Laboratorio**. Em seguida efetuar um commit.

Para isso acesse o diretório através do seguinte caminho: C:\Users\nome\Documents\GitHub\Laboratorio.

Crie um novo arquivo chamado **lab1.txt** com o seguinte conteúdo:

GitHub for Windows

Seja bem vindo!

Primeira Laboratorio.

Agora ao voltar para o GitHub for Windows veremos o novo arquivo sendo listado. Para isso clique no botão **Changes**, como mostra a Figura 13 abaixo.

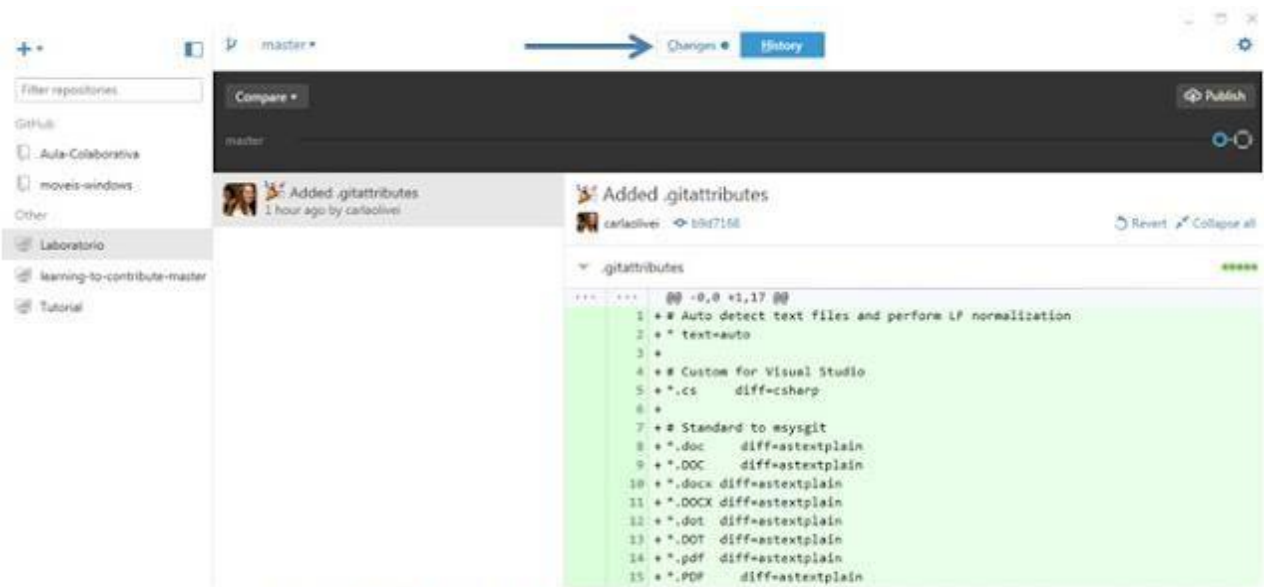


Figura 13 – Exibição do novo arquivo criado

Ao clicar no botão **Changes**, o novo arquivo criado será listado conforme mostra a Figura 14.



Figura 14 – Novo arquivo criado

Para efetuar o commit do arquivo, basta digitar a mensagem do commit no campo **Summary**, e opcionalmente, preencher uma descrição mais detalhada no campo **Description**. Após isso efetuar o commit clicando no botão **Commit to master** conforme mostra a Figura 15.

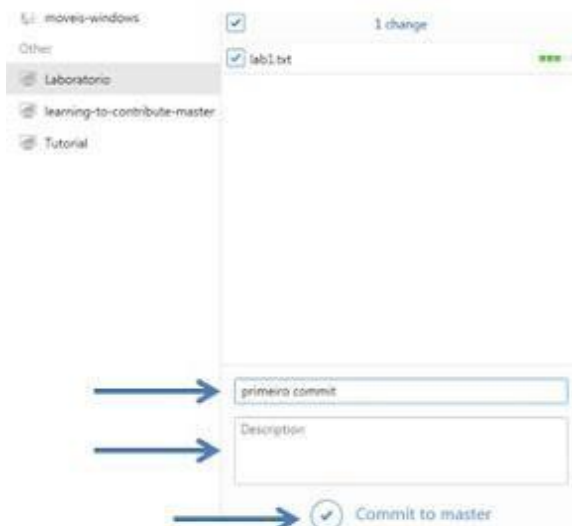


Figura 15 – Efetuando commit de um arquivo

Para visualizar o commit efetuado clique no botão **History** conforme mostra a Figura 16.

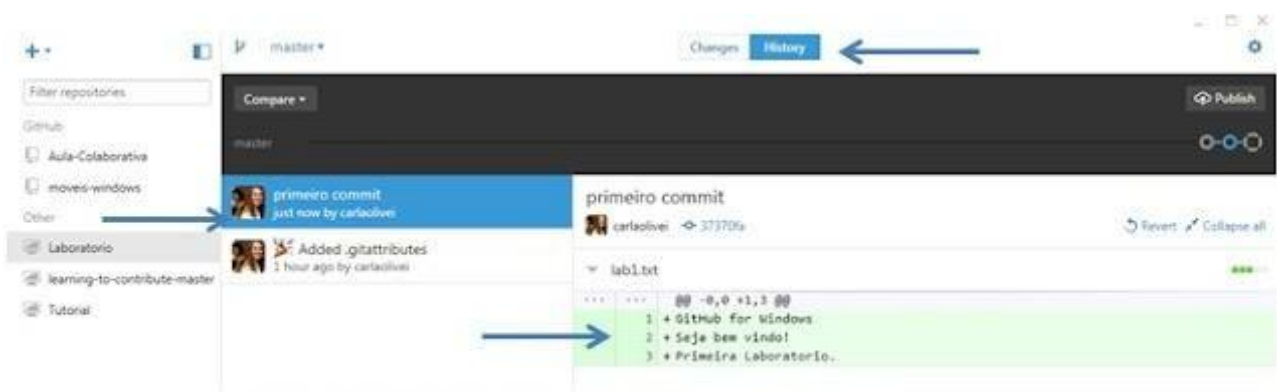


Figura 16 – Visualizando o commit de um arquivo

Detalhando os commits efetuados

Vamos efetuar um novo commit no repositório. Para isso, altere o arquivo **lab1.txt** incluindo o trecho abaixo:

GitHub for Windows

Seja bem vindo!

Primeiro Laboratorio.

Alterando o arquivo para o segundo commit.

Agora ao voltar para o GitHub for Windows e visualizar a alteração efetuada clique no botão **Changes**. Ao clicar no botão **Changes** a alteração será exibida. Após isto registre a alteração com um novo commit conforme mostra a Figura 17.

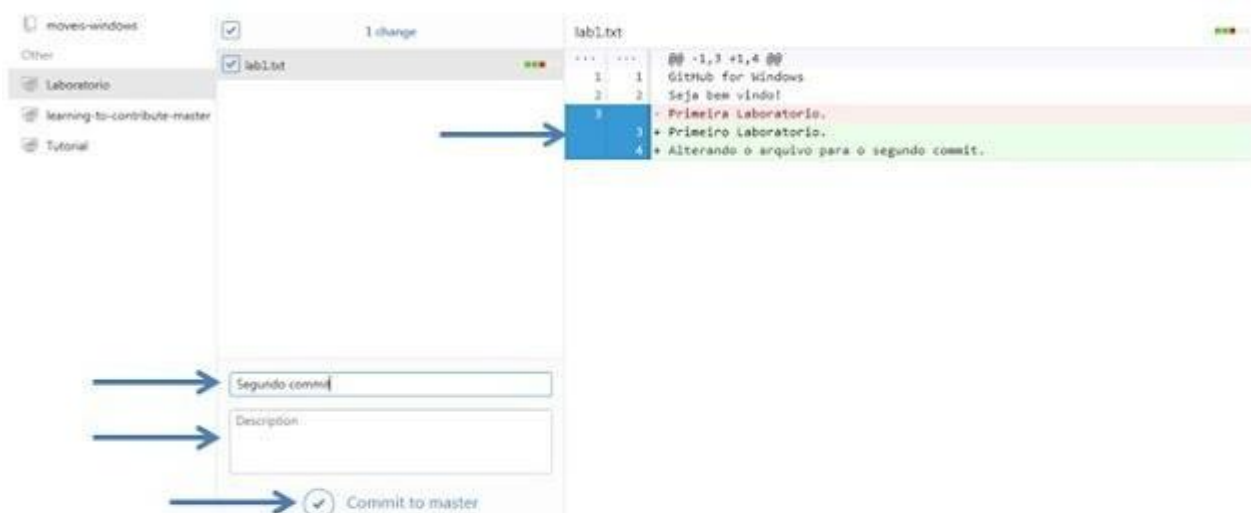


Figura 17 – Registrando a nova alteração no arquivo com o commit

Repare na Figura 18 que o conteúdo já existente que foi alterado é exibido com o sinal de -. O novo conteúdo inserido é mostrado com sinal de +.

Após efetuar o commit repare também que o mesmo é exibido na listagem de commits, onde é exibida a mensagem, o autor e a data em que cada commit foi executado (Figura 18).

Ao clicar em algum dos commits serão exibidas as alterações realizadas naquele commit em específico.



Figura 18 – Detalhes do commit

Revertendo commits

É possível reverter os commits facilmente, bastando para isso clicar no commit desejado, e em seguida clicar no botão **revert** conforme mostra a Figura 19.



Figura 19 – Revertendo um commit

Após clicar no botão **revert** um novo commit será efetuado automaticamente, desfazendo as alterações do commit selecionado.

Para confirmar é só abrir o arquivo **lab1.txt**. O conteúdo voltará a ser o que era antes da alteração efetuada.

GitHub for Windows

Seja bem vindo!

Primeira Laboratorio.

Enviando o repositório para GitHub Web

O GitHub for Windows possui integração com o GitHub Web. Permite com isto envio do nosso repositório local para o GitHub Web.

Para enviar o repositório para o GitHub Web basta selecionar o repositório desejado e clicar no botão **Publish** conforme mostra a Figura 20.

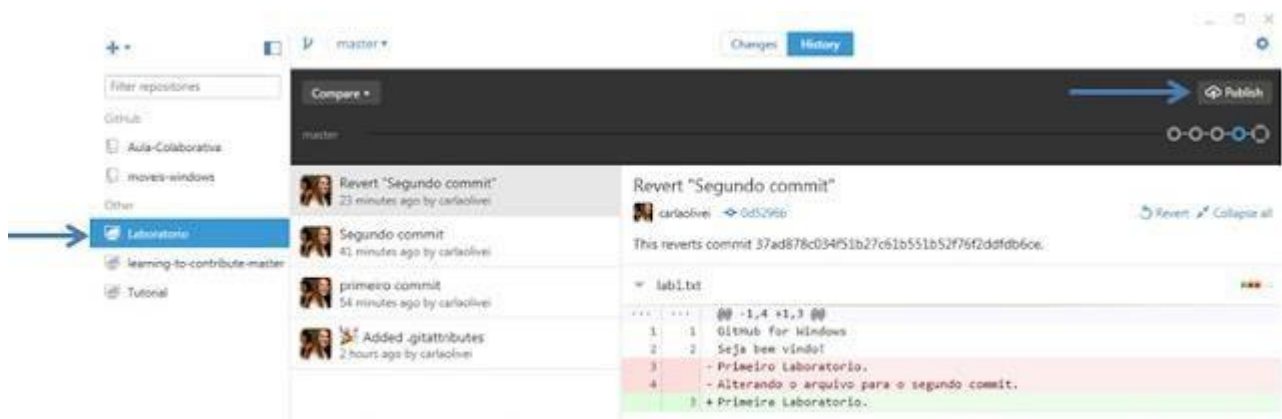


Figura 20 – Enviando o repositório para o GitHub Web

Ao clicar no botão **Publish** será exibida uma tela (Figura 21) onde podemos preencher uma descrição detalhada do repositório, e escolher se o repositório deverá ser privado. Caso o usuário tenha cadastro em algum dos planos pagos do GitHub.

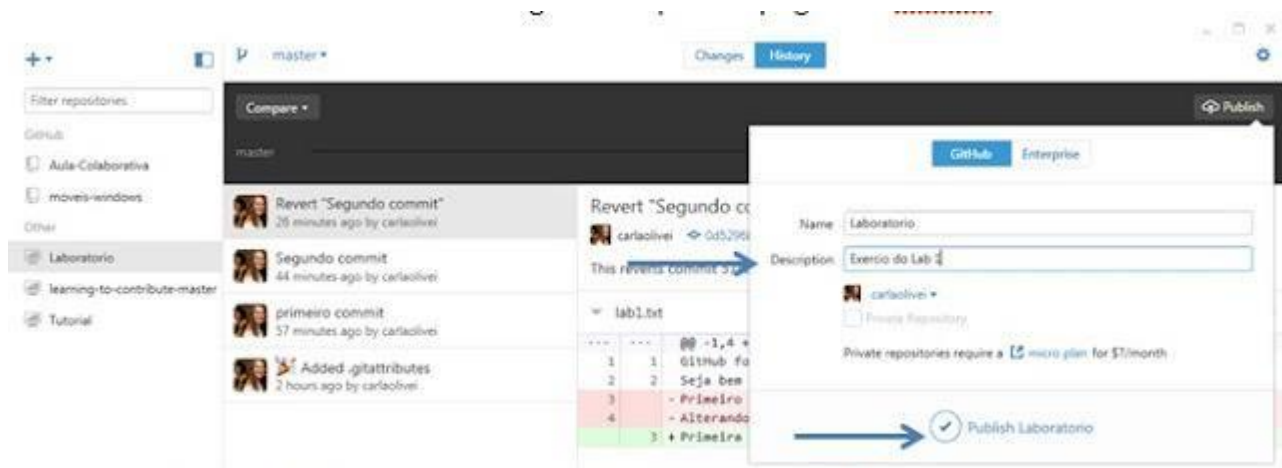


Figura 21 – Tela para enviar o repositório para o GitHub Web

Após preencher as informações basta clicar no botão **Publish Laboratorio** e este repositório será enviado para o GitHub Web.

Se você acessar a página do seu usuário no GitHub Web, verá que o repositório foi enviado corretamente conforme mostra a Figura 22.

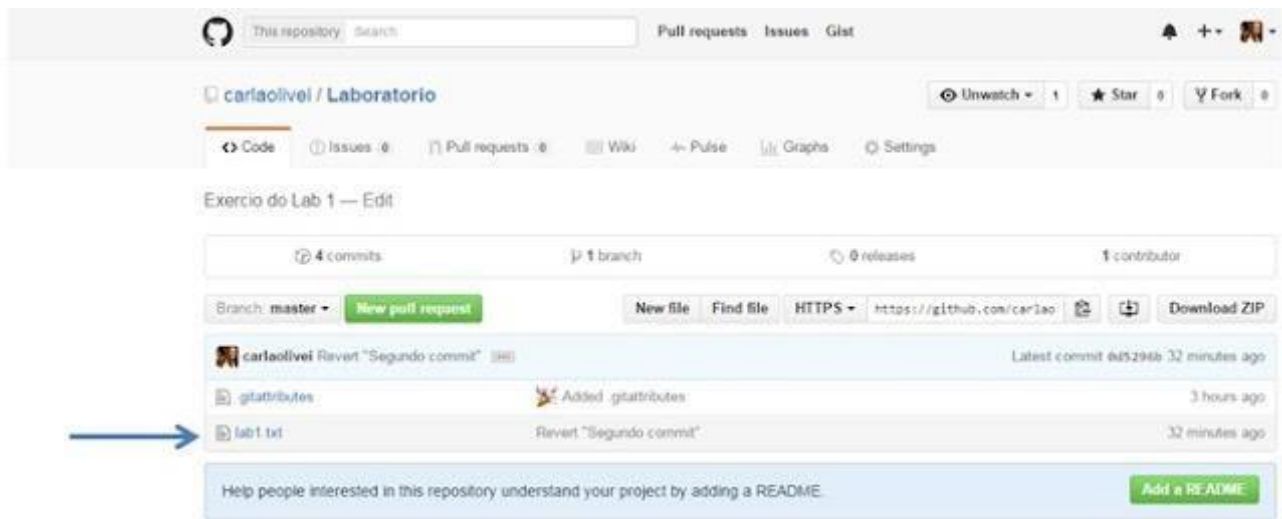


Figura 22 – Repositório enviado com sucesso para o GitHub Web

Após enviar um repositório para o servidor remoto sempre que precisar sincronizar as alterações locais com o repositório remoto no GitHub Web, basta clicar no botão **Sync** conforme mostra a Figura 23.

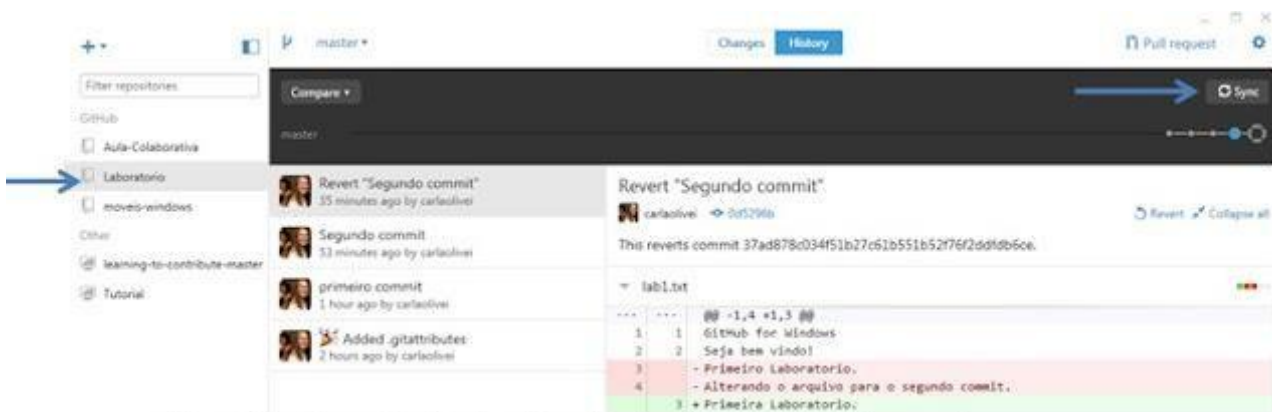


Figura 23 – Sincronização das alterações locais com o servidor remoto

Clonando um repositório do GitHub Web

Para clonar um repositório disponível no GitHub Web primeiro é necessário acessar o GitHub Web e pesquisar o repositório desejado. Neste exemplo vamos pesquisar por “**Manual do GitHub**” conforme mostra a Figura 24.

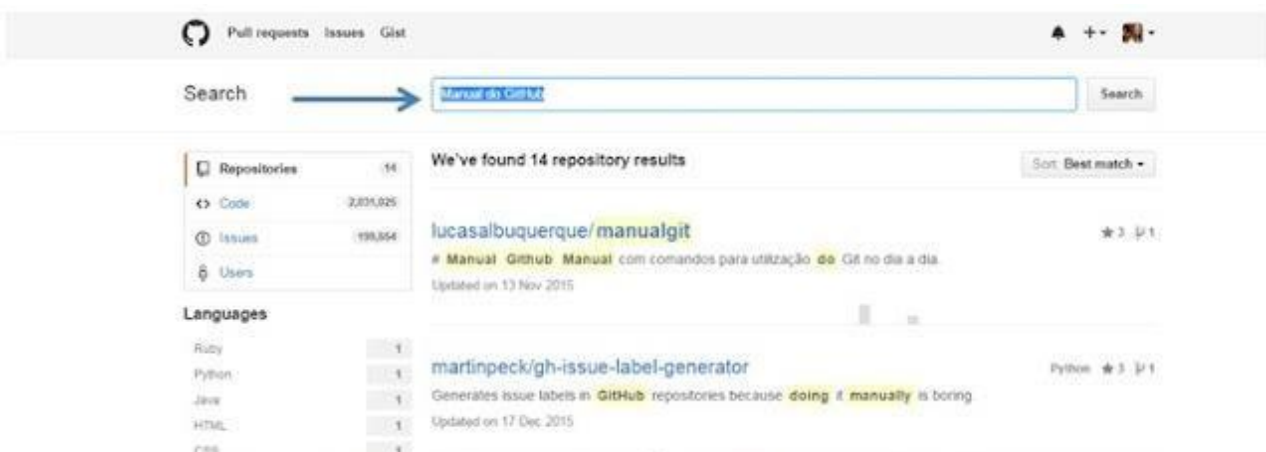


Figura 24 – Pesquisando um projeto no GitHub Web

Em seguida clique no repositório desejado. No nosso caso clique em **lucasalbuquerque/manualgit**. Ao clicar neste repositório será exibida a tela da Figura 25. O próximo passo agora é clicar no botão **Fork**, no canto superior direito, para criar uma cópia desse repositório. Ou clique em Download Zip para baixar o projeto localmente e contribuir com o mesmo.



Figura 25 – Clonando um repositório no GitHub Web

Ao clicar no botão **Fork** será exibida a tela da Figura 26.

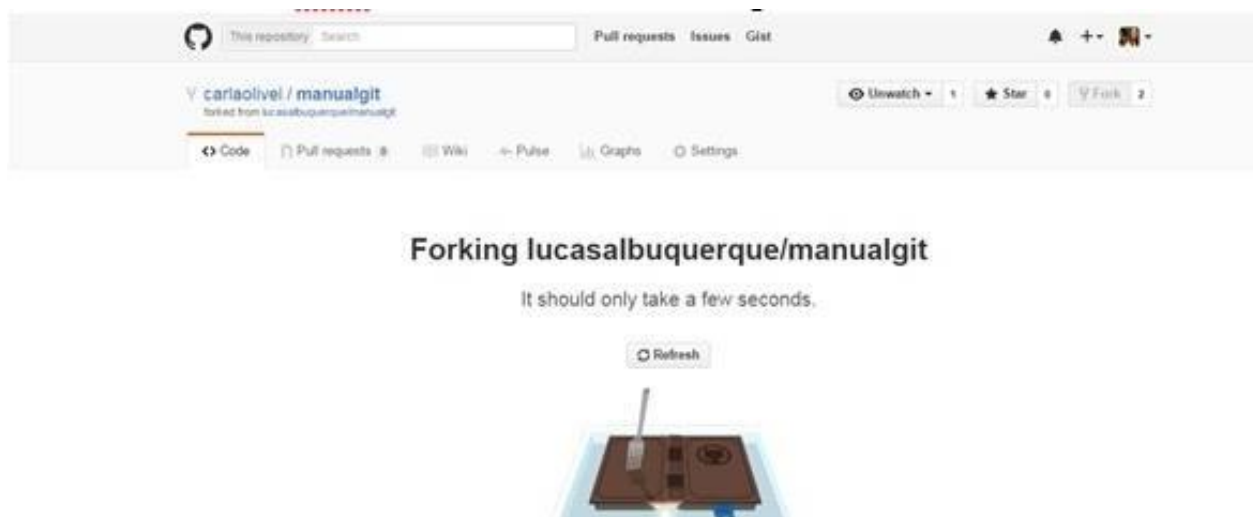


Figura 26 – Andamento do processo de clonagem de um repositório no GitHub Web

Para verificar se o repositório foi copiado corretamente acesse a página do seu usuário no GitHub Web. Filtre pela opção **Fork** e verá o repositório **manualgit** conforme mostra a Figura 27.

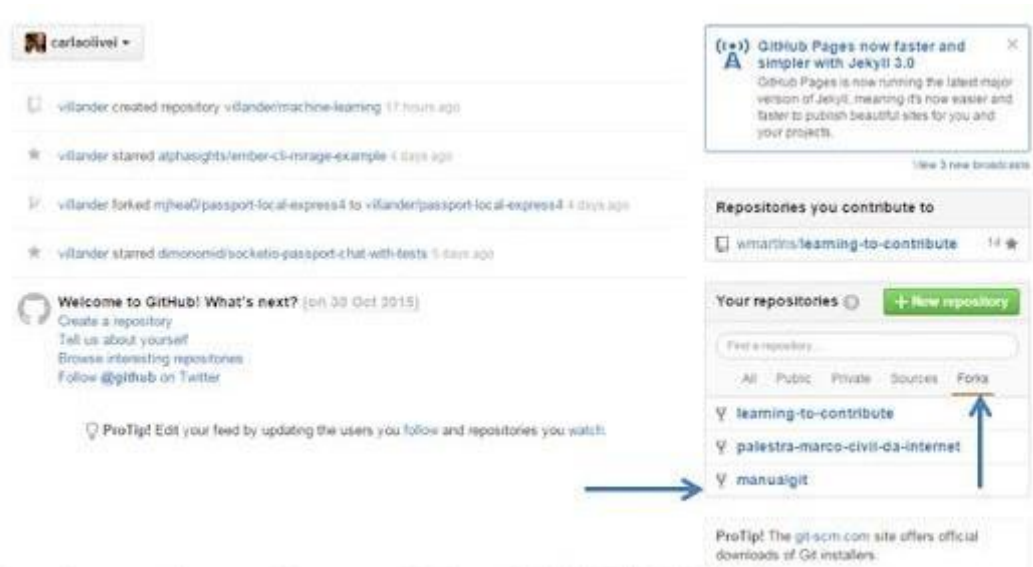


Figura 27 – Confirmando que a clonagem de um repositório no GitHub Web foi efetuada com sucesso

No GitHub for Windows clique no botão **+** e escolha opção **Clone**. Em seguida selecionar o repositório **manualgit** e clique no botão **clone manualgit**. Ao clicar no botão **clone manualgit** será exibida a tela abaixo para seleção do diretório. Mantenha o diretório padrão que GitHub e clique em **OK** conforme a Figura 28.



Figura 28 – Diretório padrão do GitHub for Windows

Após clicar em **OK** será exibida a tela da Figura 29.



Figura 29 – Clonando um repositório do servidor remoto

Quando o processo de clonagem terminar será exibida a tela da Figura 30 com o novo repositório clonado.

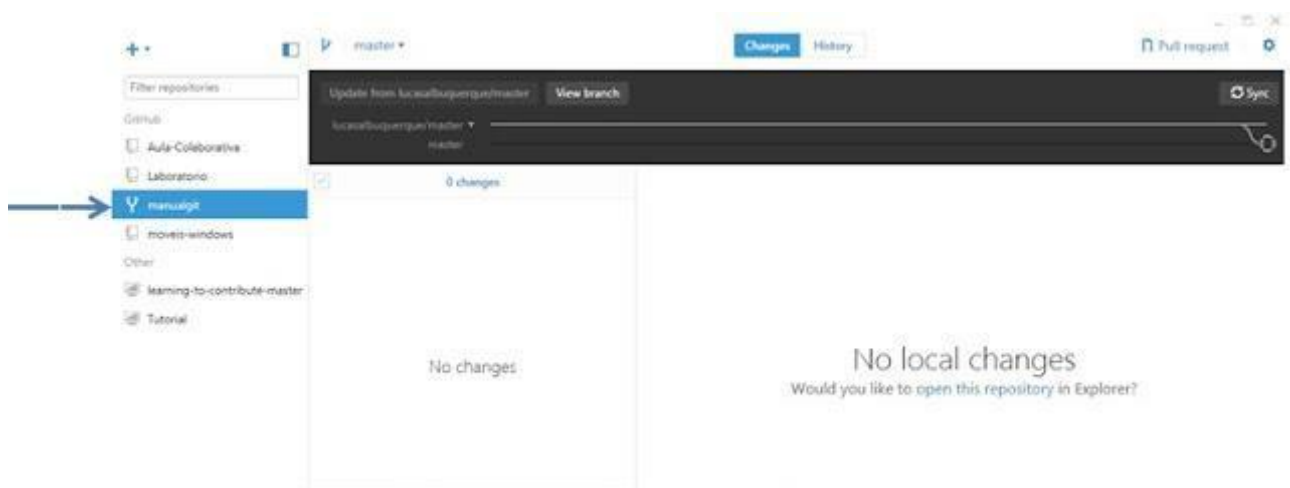


Figura 30 – Novo repositório clonado

Para confirmar que a clonagem foi realmente executada acesse o diretório C:\Users\nome\Documents\GitHub e verifique que foi criada a pasta **manualgit** conforme mostra a Figura 31.

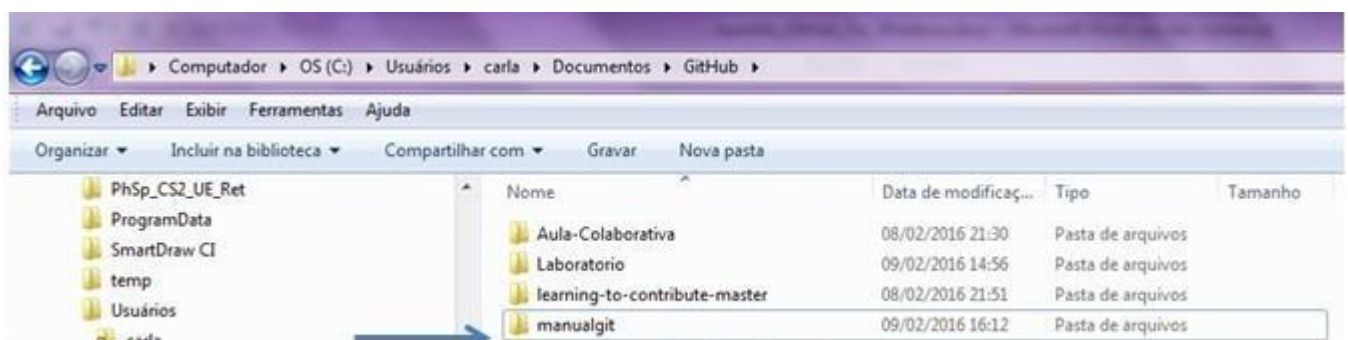


Figura 31 – Confirmando se a clonagem foi efetuada com sucesso

Trabalhando com branches

Também é possível trabalhar com branches no GitHub for Windows. Na tela principal existe um botão onde podemos gerenciar as branches do repositório conforme mostra a Figura 32.

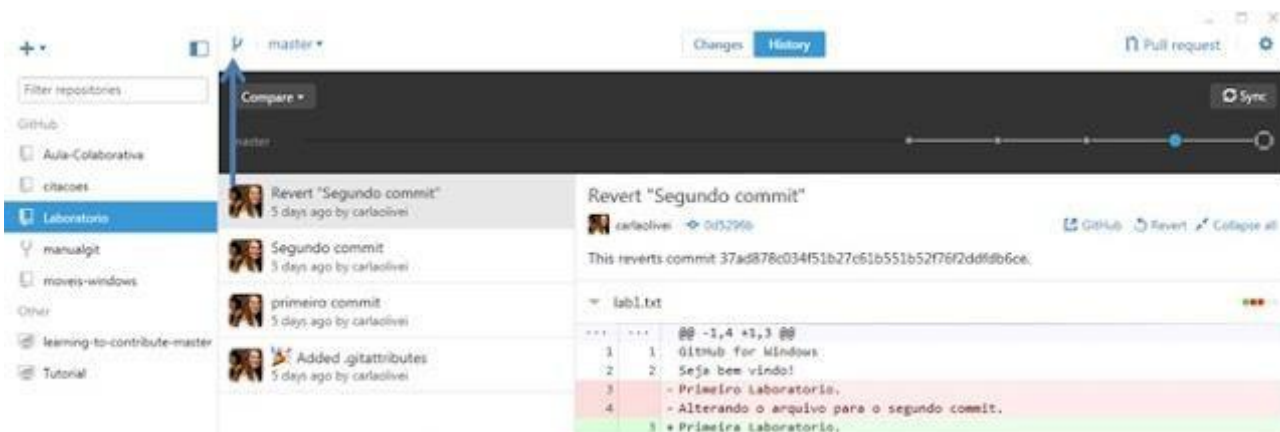


Figura 32 – Gerenciando branches

Vamos criar uma nova branch chamada **teste**, clicando no botão mencionado anteriormente, e em seguida digitar **teste** no campo de texto da tela. Em seguida confirmar clicando em no botão **Create new branch** conforme mostra a Figura 33.

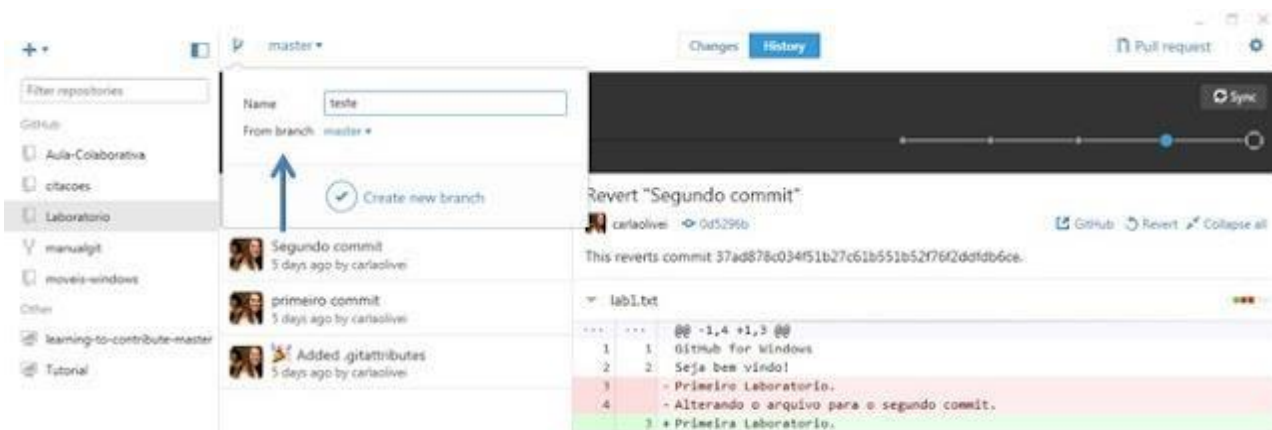


Figura 33 – Criando uma nova branch

Após a nova branch ser criada ela já será selecionada. Se clicar novamente no botão para gerenciar as branches, verá que agora existirão duas branches no repositório conforme mostra a figura Figura 34.

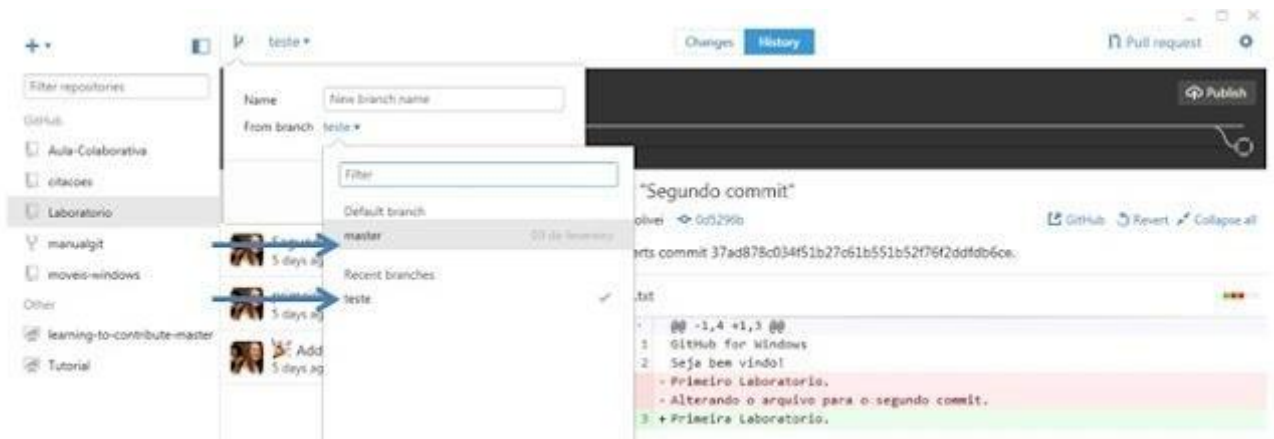


Figura 34 – Branches existentes

Vamos alterar novamente o arquivo **lab1.txt** e em seguida efetuar um novo commit, porém desta vez o commit deverá ser realizado na branch **teste**.

Altere o arquivo **lab1.txt** adicionando o seguinte conteúdo:

GitHub for Windows

Seja bem vindo!

Primeira Laboratorio.

Primeiro teste com branch.

Após efetuar a alteração no arquivo registre a mesma com um novo commit. Certifique-se antes que a branch **teste** está selecionada como mostra a Figura 35.

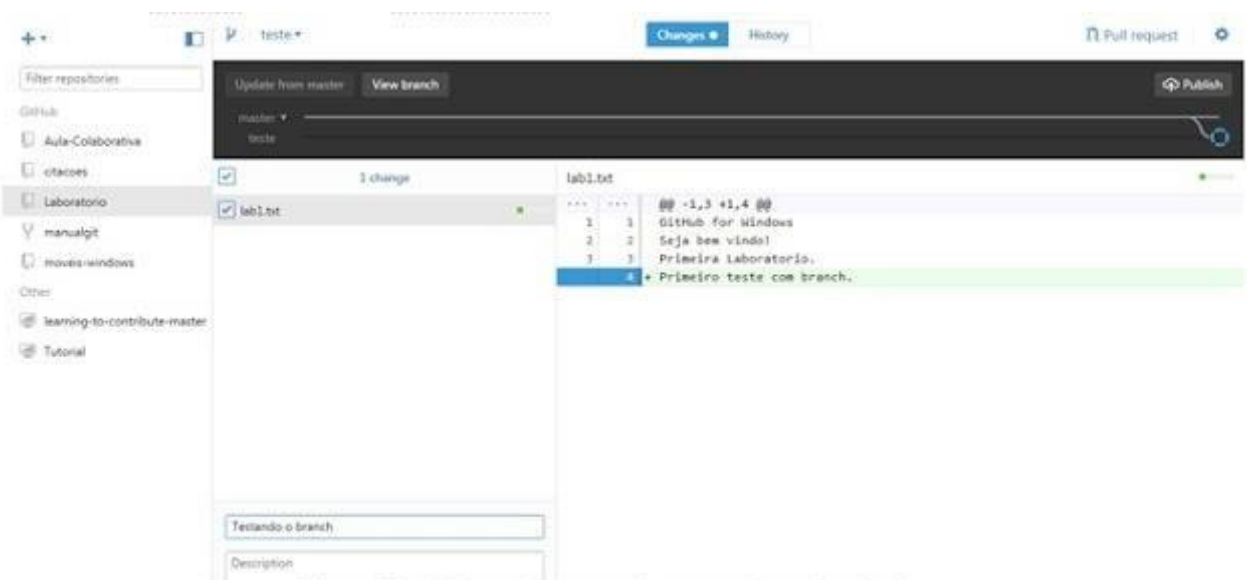


Figura 35 – Efetuando commit na nova branch criada

Observe que após efetuar o commit o mesmo é exibido na listagem de commits da branch **teste** (Figura 36).

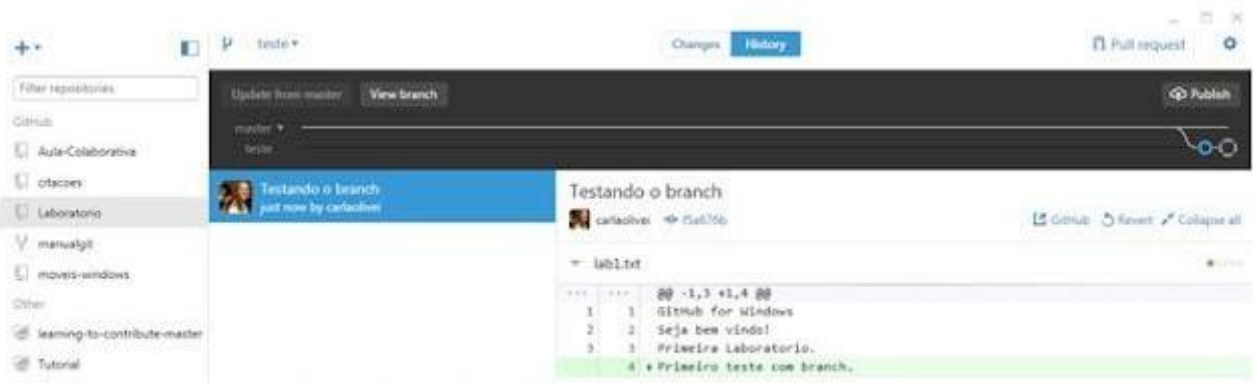


Figura 36 – Commit efetuado na nova branch criada

Porém, se você mudar para a branch **master** (Figura 37) o commit não será exibido, uma vez que ele foi efetuado apenas na branch **teste**.

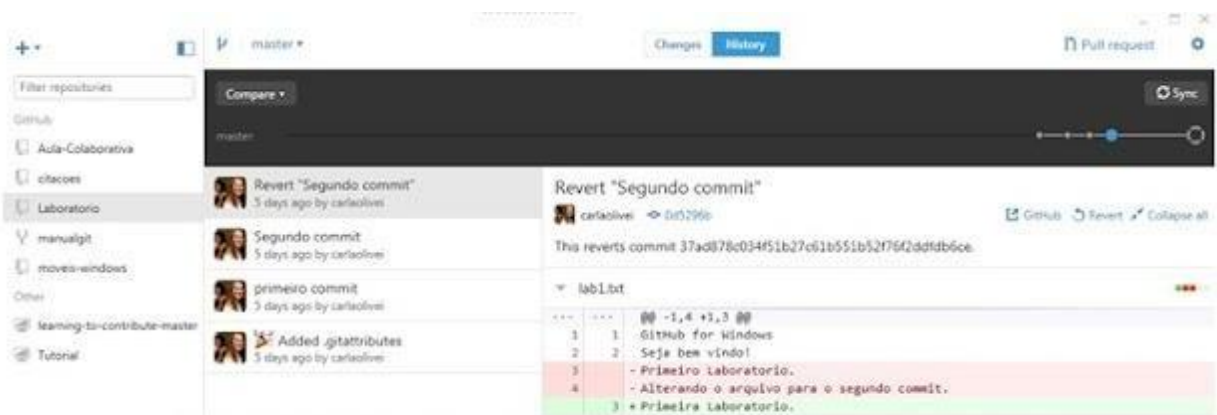


Figura 37 – Confirmando que o commit foi efetuado na nova branch criada

Efetuando o merge dos commits

No GitHub for Windows também é possível efetuar merges, para mesclar as alterações de duas branches diferentes.

Vamos efetuar o merge da **branch** teste na branch **master**. Para isso, no branch Master compare com o branch teste e em seguida devemos clicar no botão **Update from teste**.

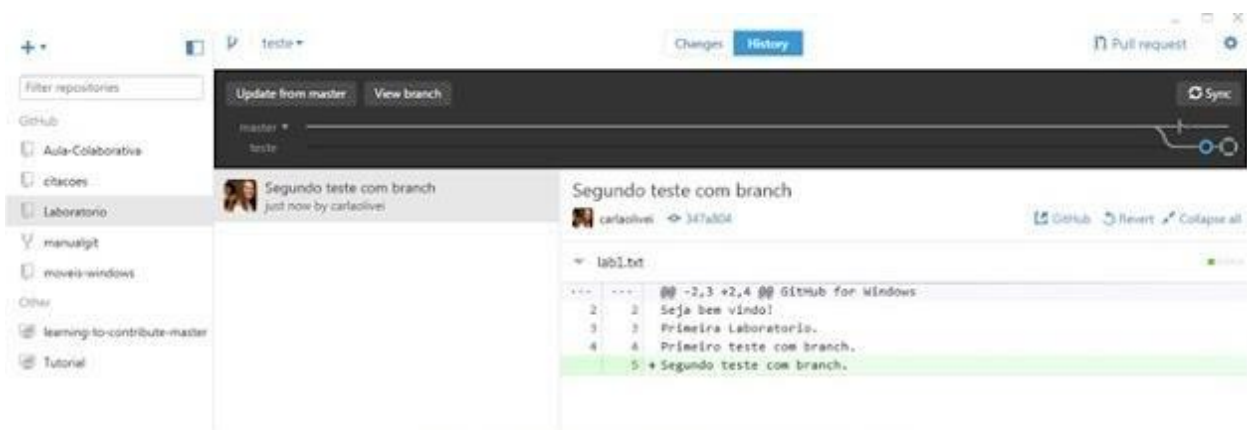


Figura 38 – efetuando merge dos commits

Feito isso o merge será efetuado como mostra a figura abaixo.

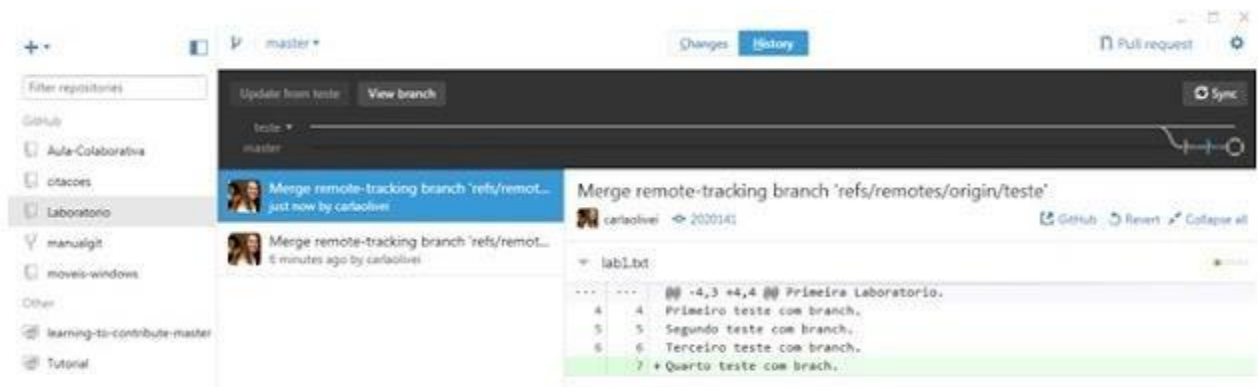


Figura 39 – merge