Tutorial de início rápido

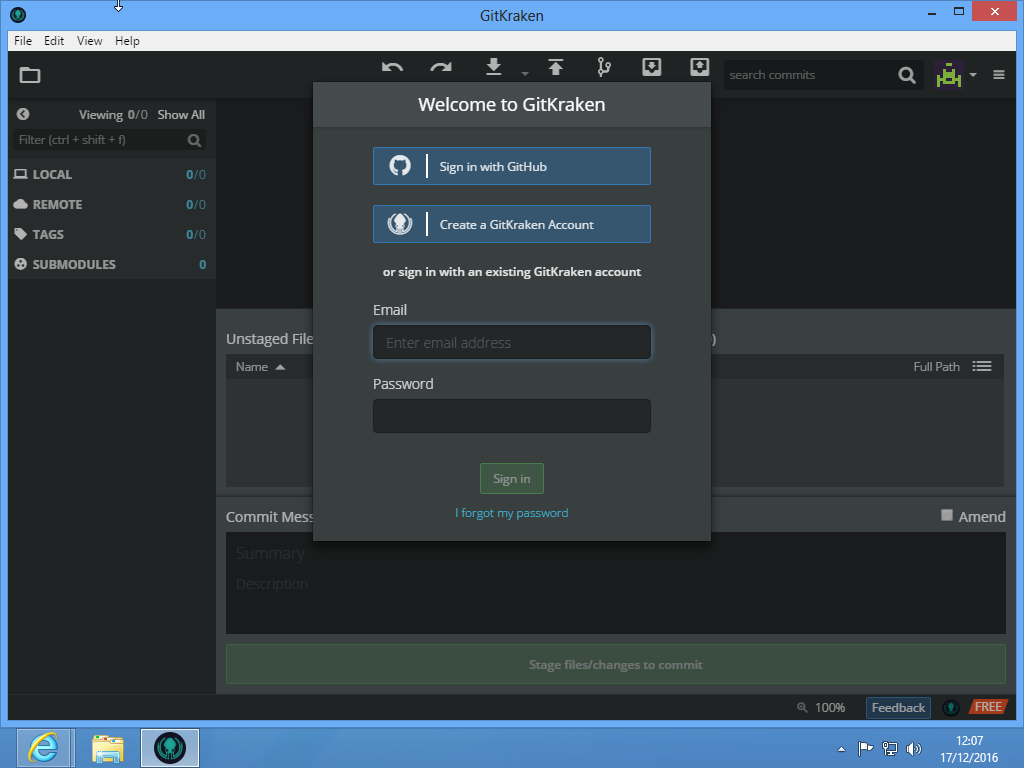
GitHub e GitKraken

## Passo 1 - Baixar e instalar o GitKraken

Escolha a versão para seu sistema operacional em <https://www.gitkraken.com/>, baixe e instale o software, seguindo as instruções do instalador de acordo com o seu SO. O GitKraken oferece interface gráfica para repositórios Git como o GitHub e o BitBucket.

## Passo 2 - Configurar seu acesso

O Próprio GitKraken gerencia seus repositórios Git. Para isso, conecte-se ao GitHub pelo GitKraken, assim que você baixar o arquivo e iniciá-lo, já é aberta uma tela de login onde você pode se conectar pelo GitHub.

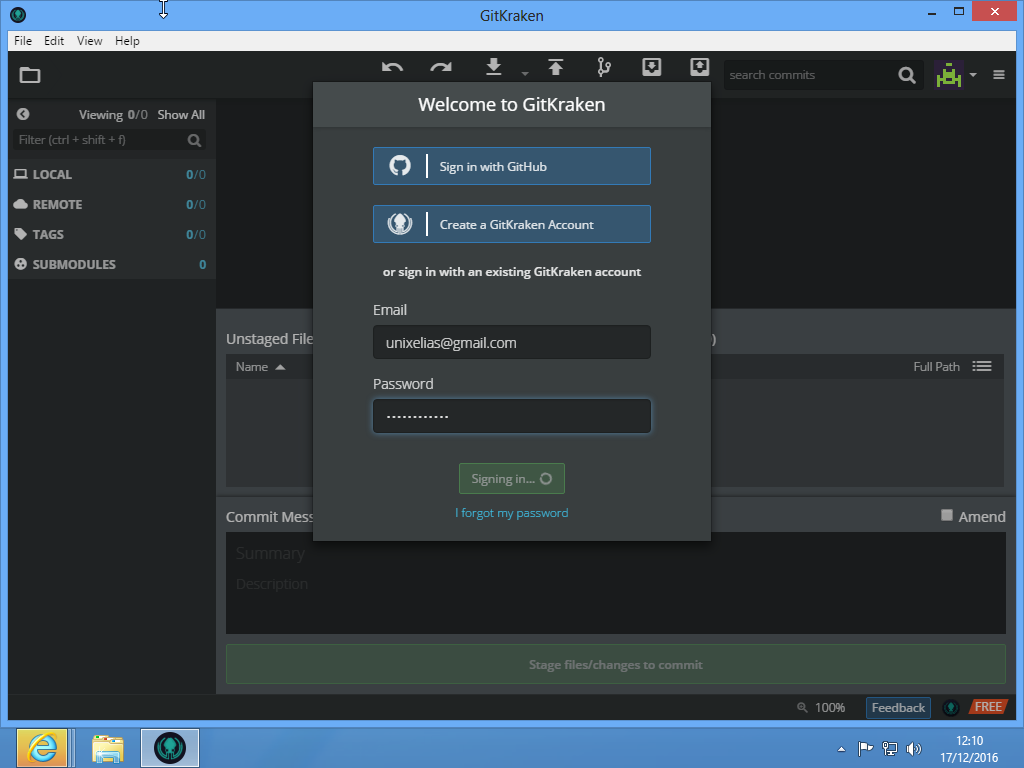
  
Figura 1: Fazendo Login

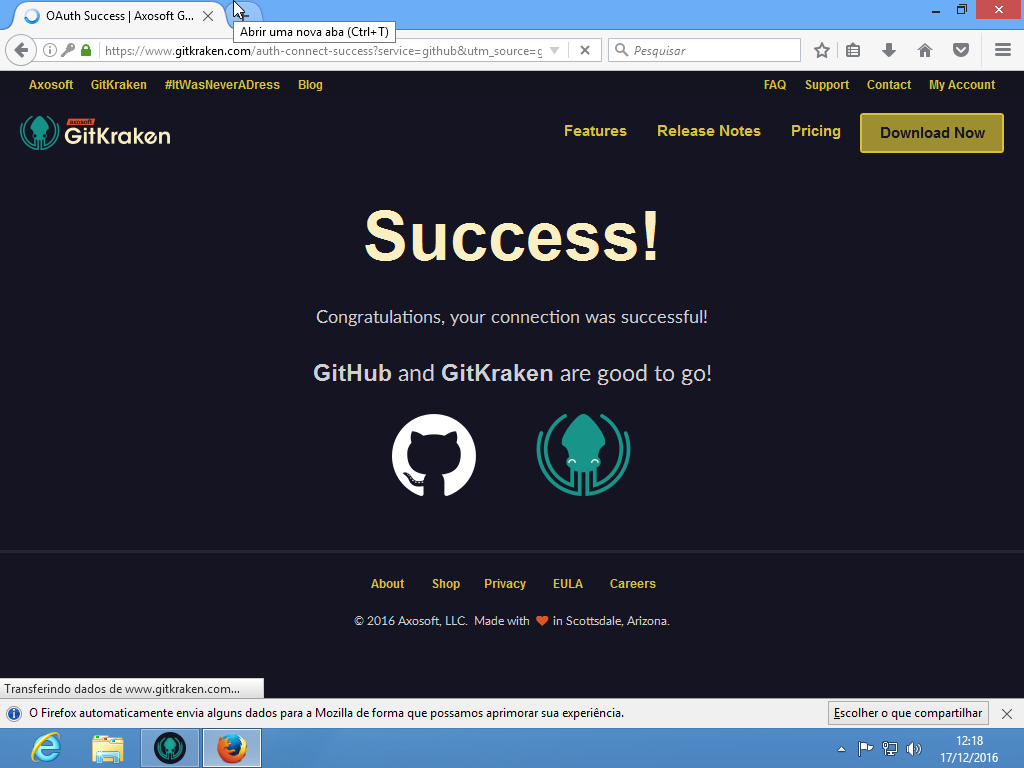
### Passo 2.1-a Registre-se

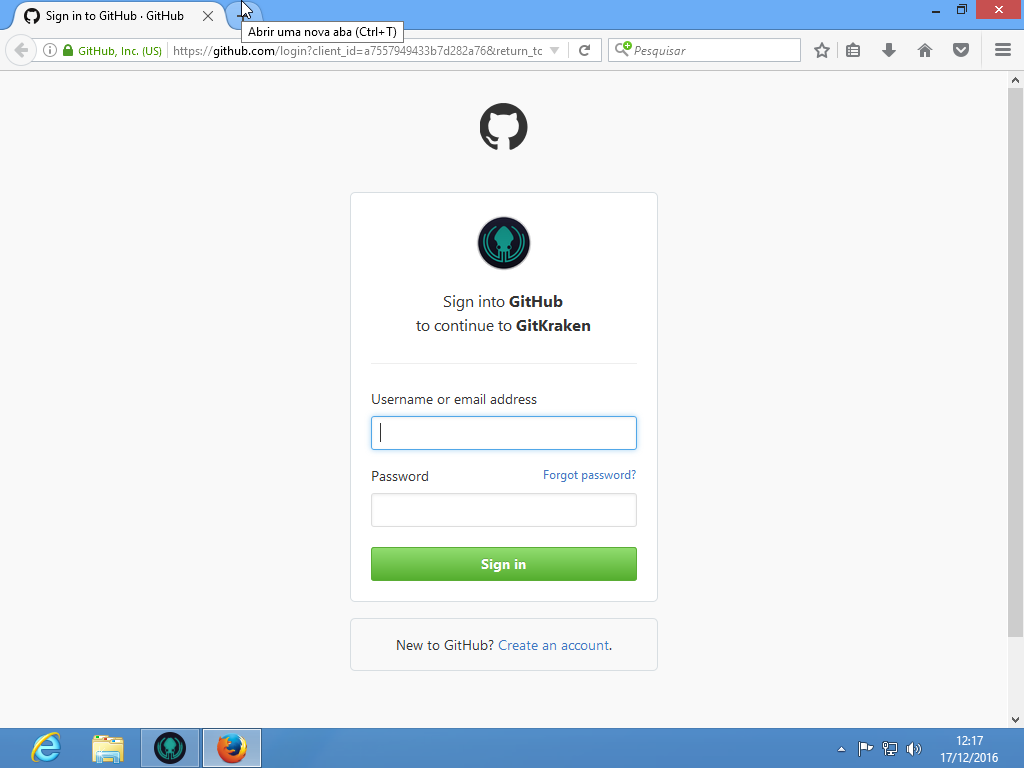
Ele pede um registro para uso, mas é gratuito, não se preocupe e se registre preferencialmente com o mesmo endereço de e-mail usado no GitHub.

### Passo 2.1-b Faça o login pelo GitHub

Selecionando o login pelo GitHub, você será redirecionado ao navegador, onde deverá se logar no GitHub e dar permissão ao GitKraken para se conectar. Depois desse processo ele adiciona automaticamente uma chave SSH e você não precisará realizar login novamente.

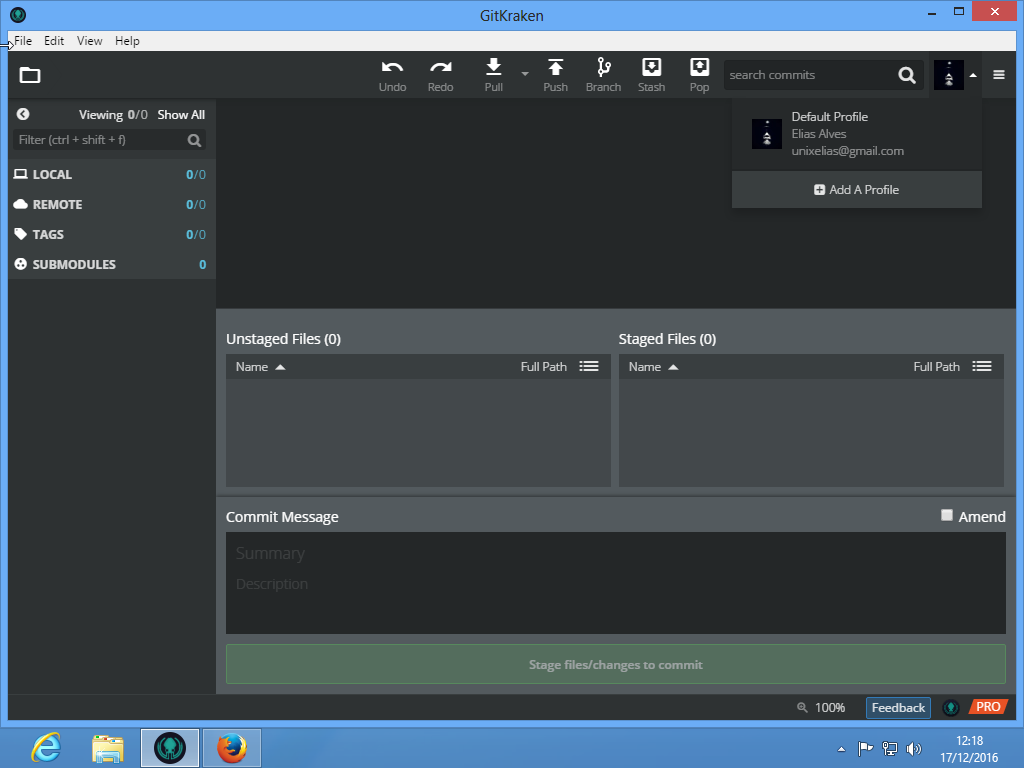
  
Figura 2: Escolhendo conexão pelo GH

  
Figura 3: Sucesso na autenticação pelo GH

Figura 4: Conectar ao GH pelo site

## Passo 3 - Clonar o repositório

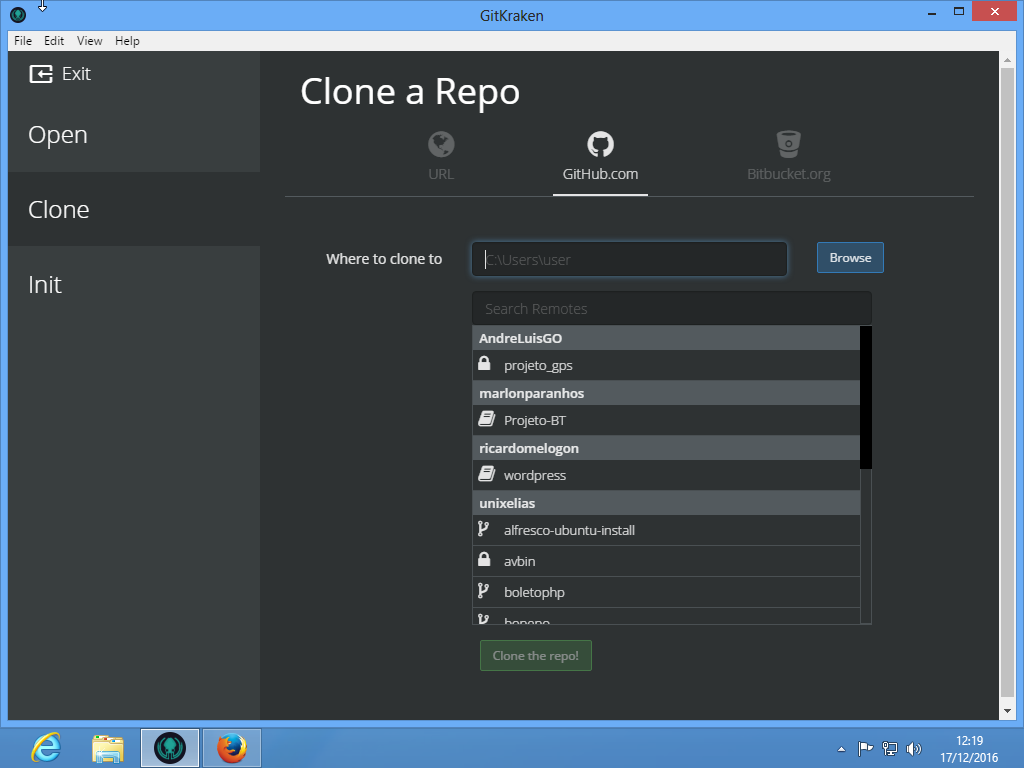
Terminado o processo de vinculo do registro do GitHub com o GitKraken, você terá disponível seu perfil. Para começar a trabalhar em um projeto você precisa agora apenas clonar repositório.

  
Figura 5: Logado ao GitKraken com a conta do GH

### Passo 3.1 - Selecione o repositório

Para adicionar um repositório, clique em "File" -> "Clone Repo" e em seguida no ícone do GitHub. Os repositórios em que você está registrado serão listados e você pode selecionar o desejado para começar a trabalhar com ele.

No nosso caso selecionaremos o Repositório marlonparanhos/Projeto-BT.

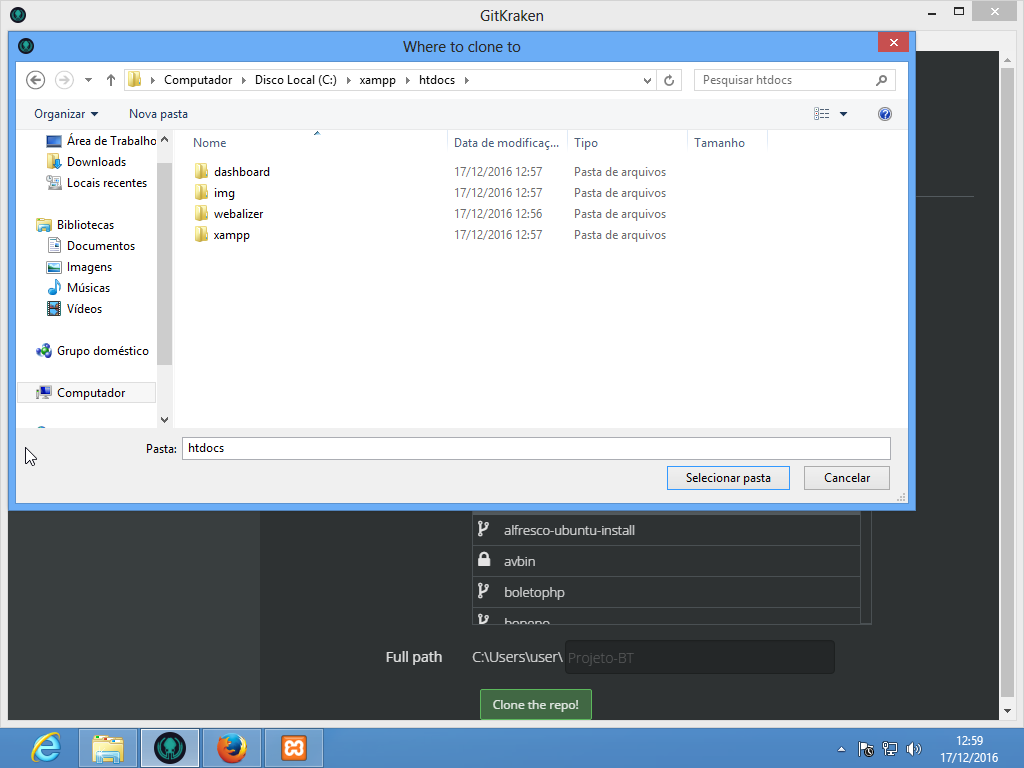
  
Figura 6: Selecionar repositório e pasta

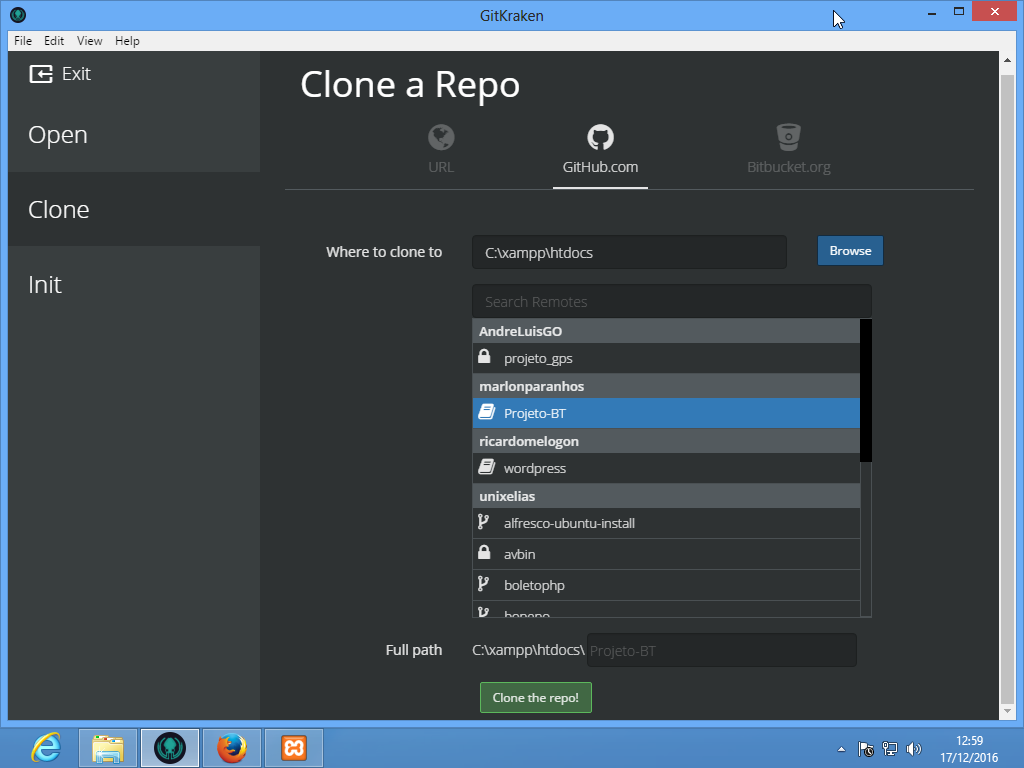
### Passo 3.2 - Selecionar a pasta onde clonar o repositório

Esse é o local onde seu repositório será clonado, e como estamos trabalhando com um projeto Web, teremos que colocar os arquivos dentro das pastas do servidor web.

Se você usa o Xampp, você deve selecionar a pasta [C:/xampp/htdocs](file:///C:/xampp/htdocs). Caso você use linux selecione a pasta /var/www/html. Isso depende de para qual pasta seu servidor Apache está apontando. Prara isso clique em “Browse”

Após seecionar a pasta e o repositório, clique em "Clone Repo" e ele vai baixar o repositório na pasa do seu servidor apache.

  
Figura 7: Selecionar pasta

  
Figura 8: Clique em "Clone Repo"

## Passo 4 - Confira seu projeto

Agora é hora de conferir se está tudo certo, então abra seu navegador para testar se seu projeto está funcionando. Nesse caso, vamos acessar o endereço <http://localhost/Projeto-BT>

Se o seu servidor Apache (Xammp) já estiver rodando, você já deve acessar seu sistema pelo navegador.

Atenção: Nesse passo eu estou considerando que você já tenha configurado o seu servidor Mysql com o banco de dados do projeto e atualizado a sua base de dados, executando a query que está na pasta "BD"

## Passo 5 - Antes de começar

Como estaremos trabalhando no mesmo repositório, para evitar desastres, não daremos commits diretamente em "master".

Para isso iremos trabalhar com Branches. Em caso de dúvidas, eu recomendo a leitura do material nesse link [https://git-scm.com/book/pt-br/v1/Ramifica%C3%A7%C3%A3o-Branching-no-Git-B%C3%A1sico-de-Branch-e-Merge](https://git-scm.com/book/pt-br/v1/Ramificação-Branching-no-Git-Básico-de-Branch-e-Merge)

Em resumo, um Branch é como uma imagem ou snapshot do projeto que você está trabalhando, que você pode criar a partir de qualquer commit. Quando você envia um novo commit, o repositório Git cria um registro do estado dos arquivos. Esse registro pode ser refeito, quando revertemos uma alterção por exemplo.

Antes de começar a trabalhar em alguma função, você deverá criar seu próprio branch e quando sua funcionalidade estiver totalmente implementada você envia um Pull Request

Um Pull Request é um pedido para que seja feito o "Merge" ou junção ao "master", que é a configuração primária do Repositório. Trabalhando em um branch você tem a possibilidade de fazer qualquer alteração, sem que essa seja refletida no tronco principal do projeto. Então, ao final da implementação deverá ser feito um Pull Request para que seu código seja adicionado ao “master”

Os passos são os seguintes:

## Passo 5.1 - Crie um novo branch

Como padrão iremos utilizar o seguinte esquema de nomes ***nomeusuario-função-numeroversão***

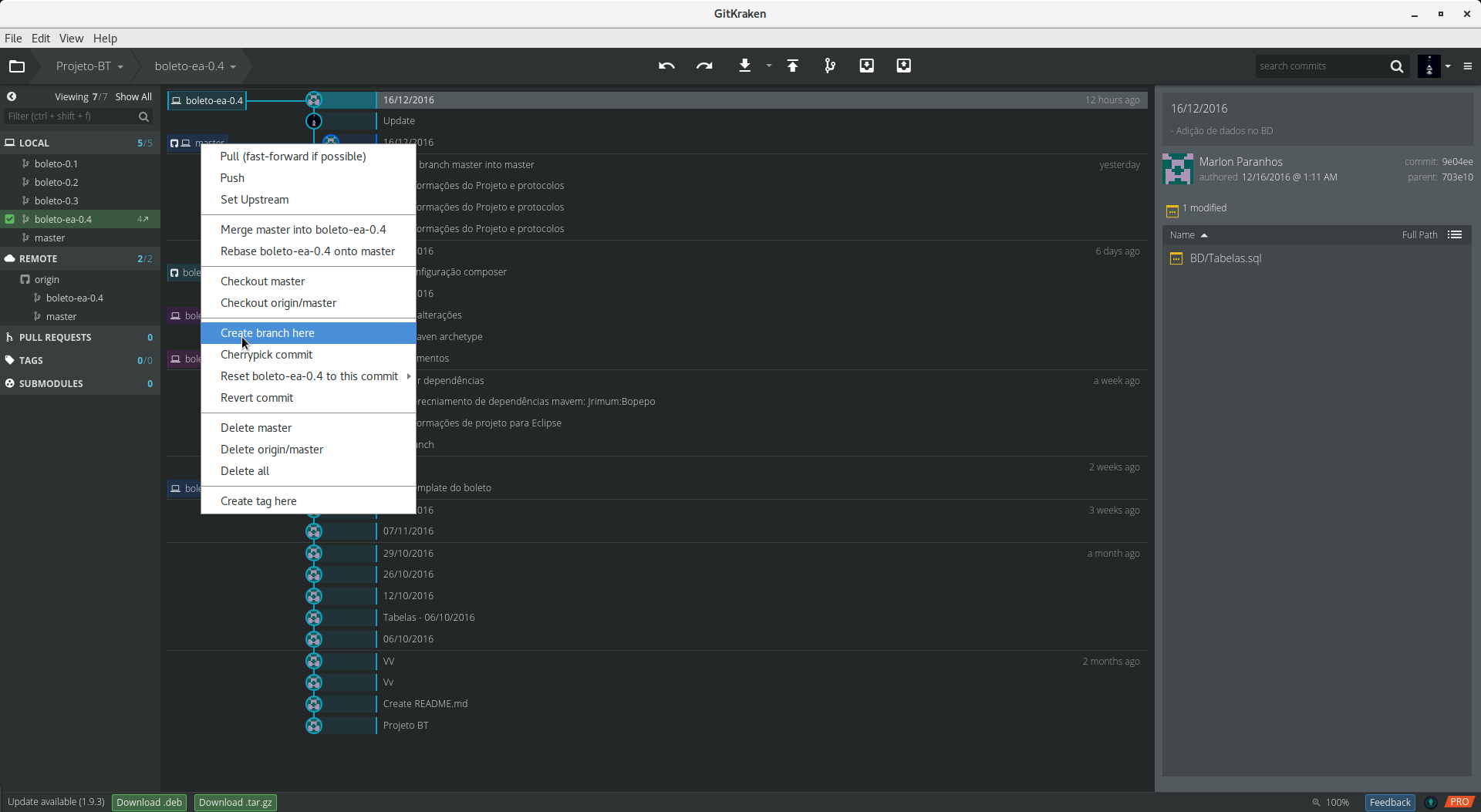
Por exemplo: ***elias-demoteste-0.1***

Cada vez que começar a trabalhar numa função crie um novo branch com esse esquema. A cada Pull Request aceito, se você for continuar trabalhando naquela função, crie um novo branch com o mesmo esquema de nomes e altere o número da versão.

É possível que mais de um usuário esteja trabalhando no mesmo branch, então pode ser registrado o nome da dupla no esquema, mas isso é livre, pode ser um nome ou termo escolhido, é apenas para identificar a equipe e a função.

Para criar um novo Branch, você pode então usar qualquer ponto de commit como referência. Para isso, clique com o botão direiro do mouse sobre o ponto que deseja e em seguida em "Create branch here"

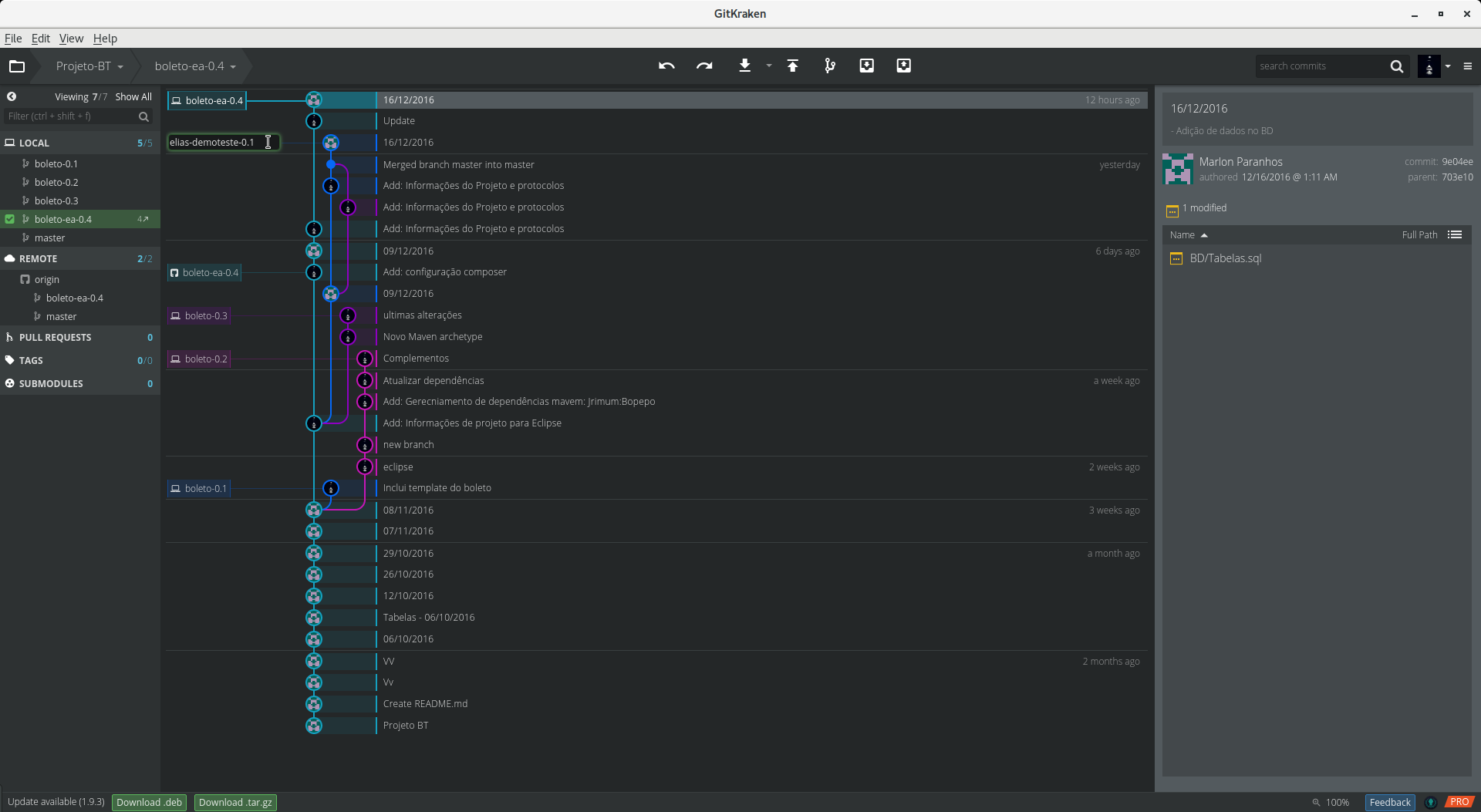
Em seguida insira o nome do seu branch conforme o esquema já especificado.

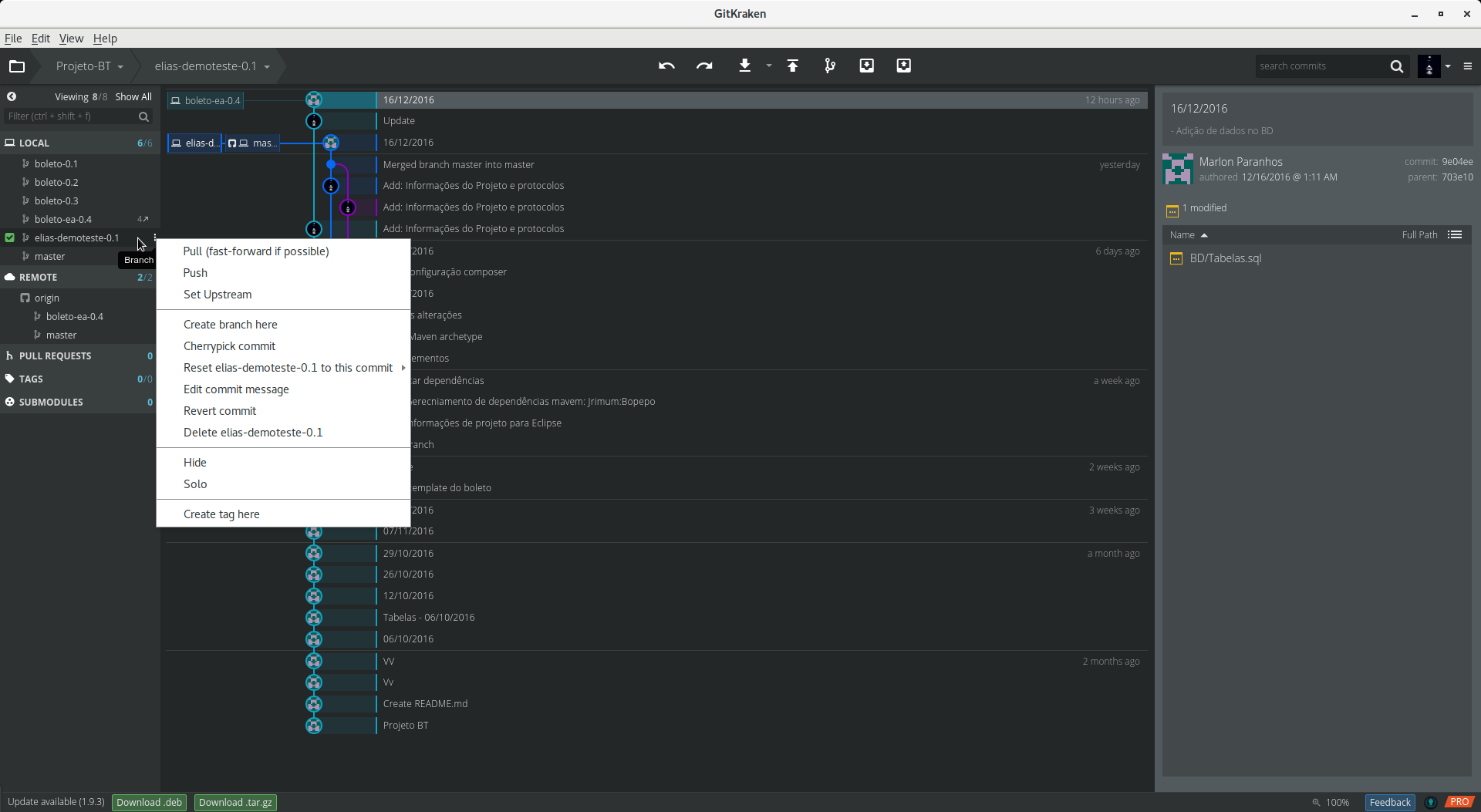
  
Figura 9: Criar novo branch

Eu criei esse branch: elias-demoteste-0.1

Fique atento, porque a janela onde devemos inserir o nome é bem discreta e pode passar despercebida. Em cima do nome do ponto onde desejamos criar esse branch abre um espaço vazio e o que temos que fazer é inserir o nome e digitar Enter.

Certifique-se que você fez chechout (está trabalhando nesse ambiente) no branch criado. Ele fica marcado em verde o espaço superior esquedo. Nesse espaço estão listados todos os Branches e pode também ser usado para realizar algumas tarefas. Você pode clicar com o botão direito para ver.

  
Figura 10: Insere nome no novo branch

  
Figura 11: Branche marcado como verde onde está feito o checkout

## Passo 5.2 - Fazer suas implementações

O passo seguinte não tem nada de novo. Simplesmente implemente o que precisar, faça seus testes e quando estiver pronto envie um novo Pull Request

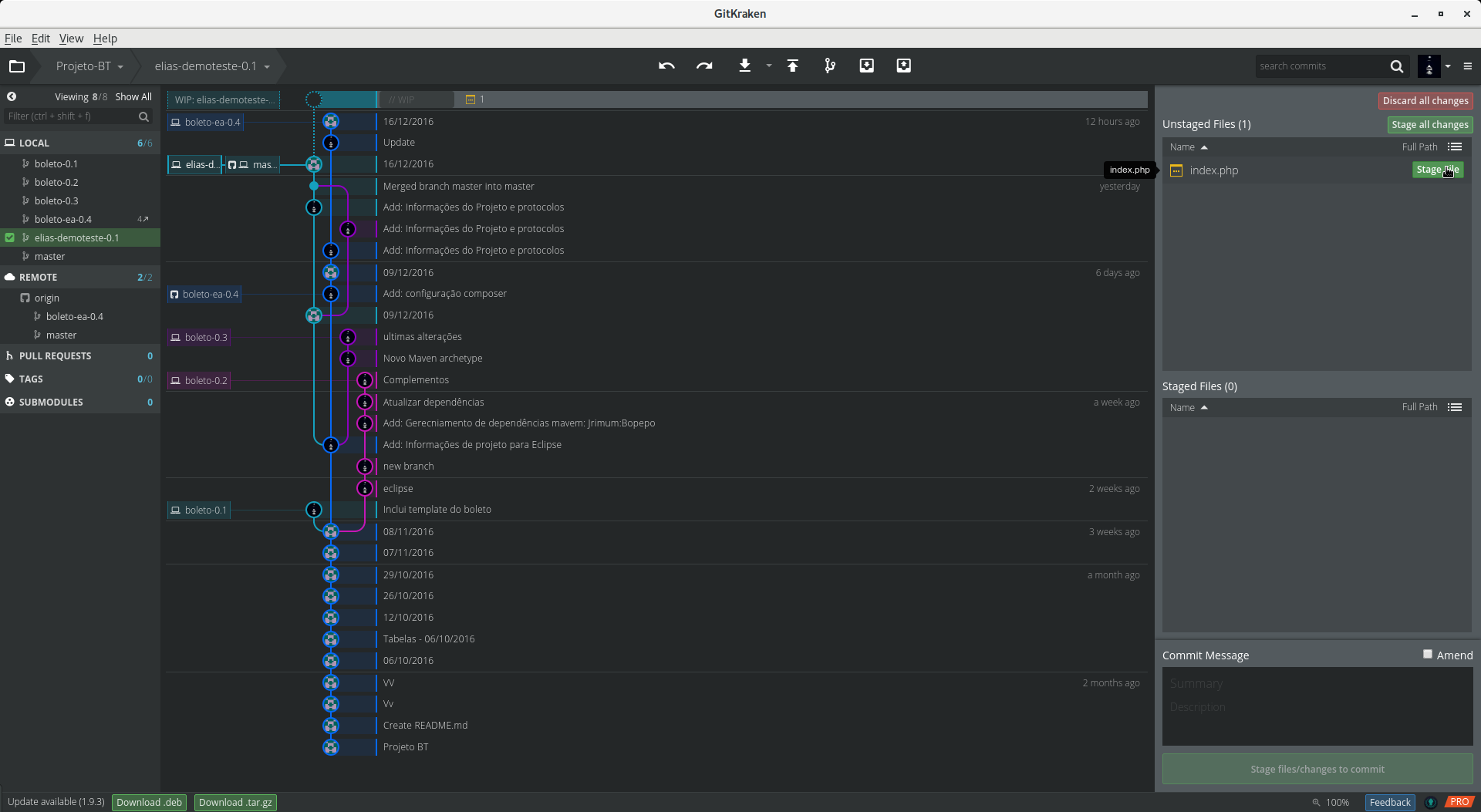
Para essa demonstração o index.php foi modificado no repositório e enviado como Pull Request

## Passo 5.3 - Fazer novo Commit

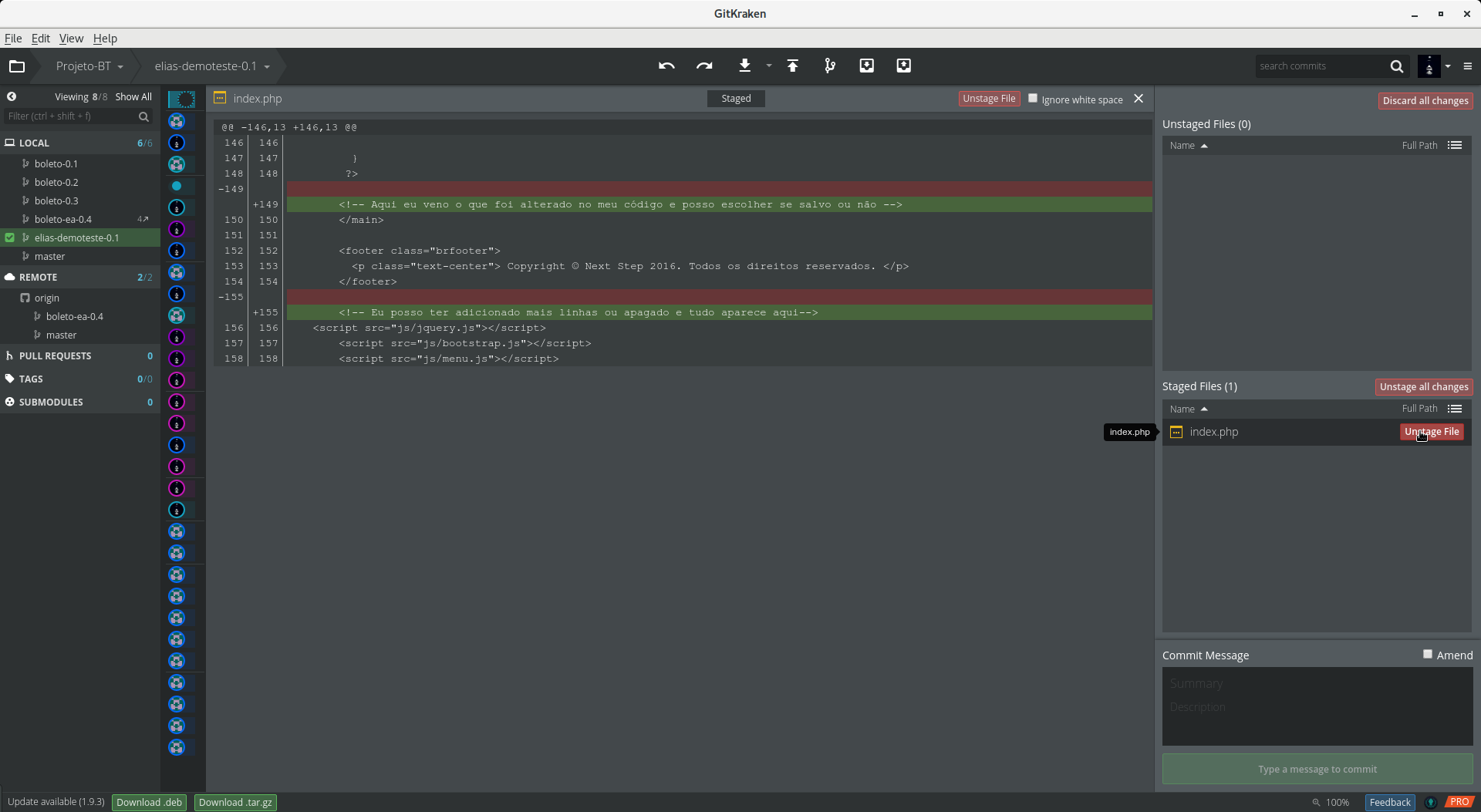
Terminando de trabalhar no fim do dia, envie seu trabalho por commit. ***Não importa se seu código não está finalizado, faça pelo menos um commit por dia.*** Quando você tiver alterado seu código, volte ao GitKraken e veja que já aparecem as suas alterações. Você precisa apenas incluir as alterações que deseja no commit e enviar seus arquivos.

### Passo 5.3.1 - Inclua os arquivos no commit

Clique em "Stage all changes" para incluir todos os arquivos, ou apenas em "Stage File" no arquivo desejado para incluir apenas ele. Note que há duas janelinhas à direita da tela uma em cima e outra embaixo. Quando você inclui o arquivo ele passa para a parte de baixo e será incluído no commit. Não é possível fazer commit sem que haja alguma alteração nos arquivos, então primeiro você deve incluir o nome do arquivo.

  
Figura 12: Incluindo arquivos no commit

Dica: Quando você estiver trabalhando no seu branch use os commits à vontade, pois você pode facilmente reverter uma alteração clicando em "Discard all changes".

  
Figura 13: Verificando o que foi alterado pelo arquivo. Você pode removê-lo do commit clicando no botão vermelho.

Essa opção desfaz qualquer alteração e retorna o repositório para o estado original do último commit registrado.

### Passo 5.3.2 - Incluir informações

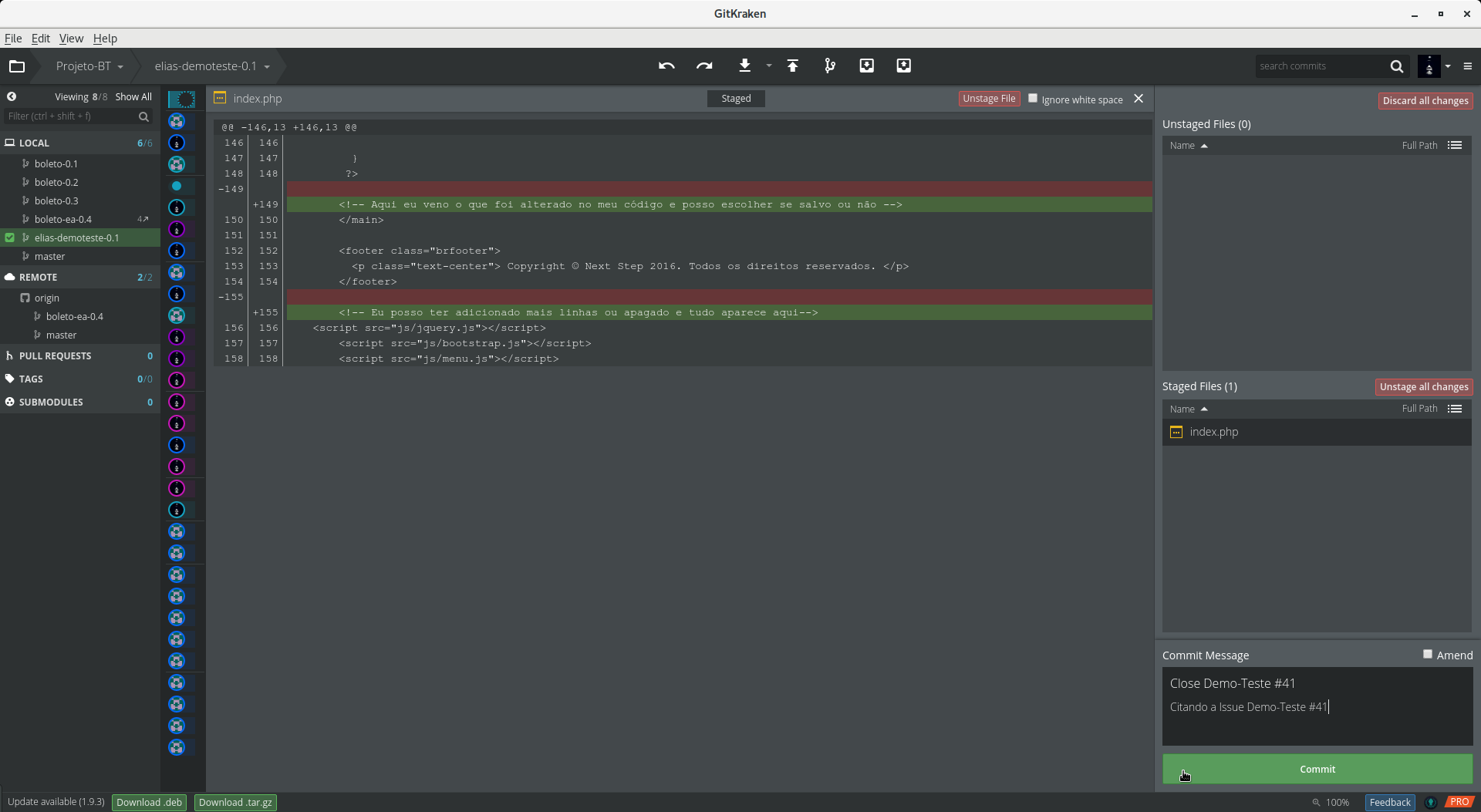
Agora você vai inserir uma descrição para seu commit.

No futuro padronizaremos mais as mensagens de commit, mas por enquando use uma pequena descrição do que se trata. Você está adicionando ou modificando o que? É uma correção de Bug?

Pela mensagem do commit você pode fechar ou citar uma issue e peço que sempre incluam na descrição a referencia à função que você está trabalhando, então recomendo a leitura desse material <http://blog.da2k.com.br/2015/02/18/segredos-do-github-commits-que-fecham-issues/>

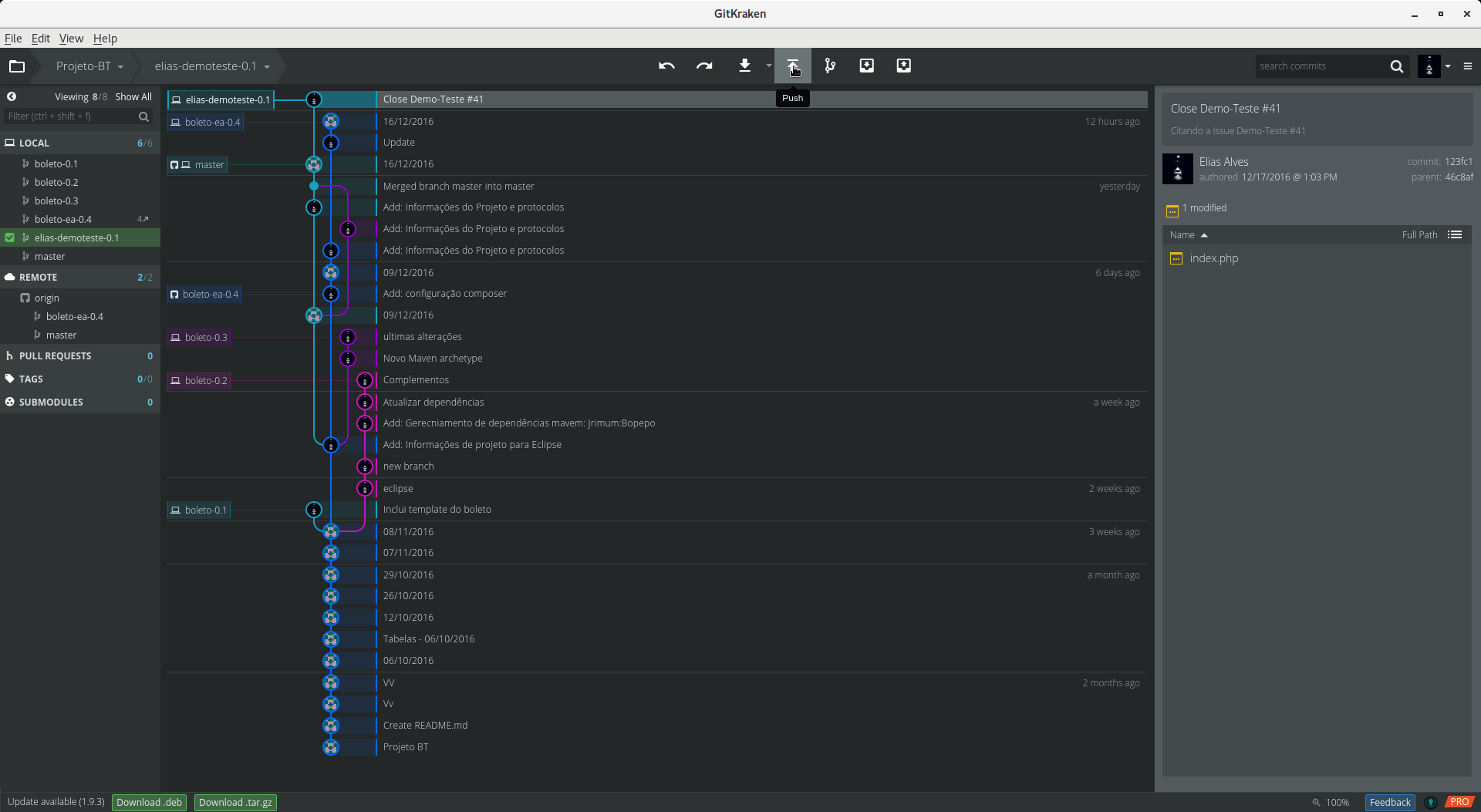
Cada Issue tem um número, no nosso exemplo, foi criada uma de teste Demo-Teste #41

Observe a mensagem do commit na Figura abaixo.

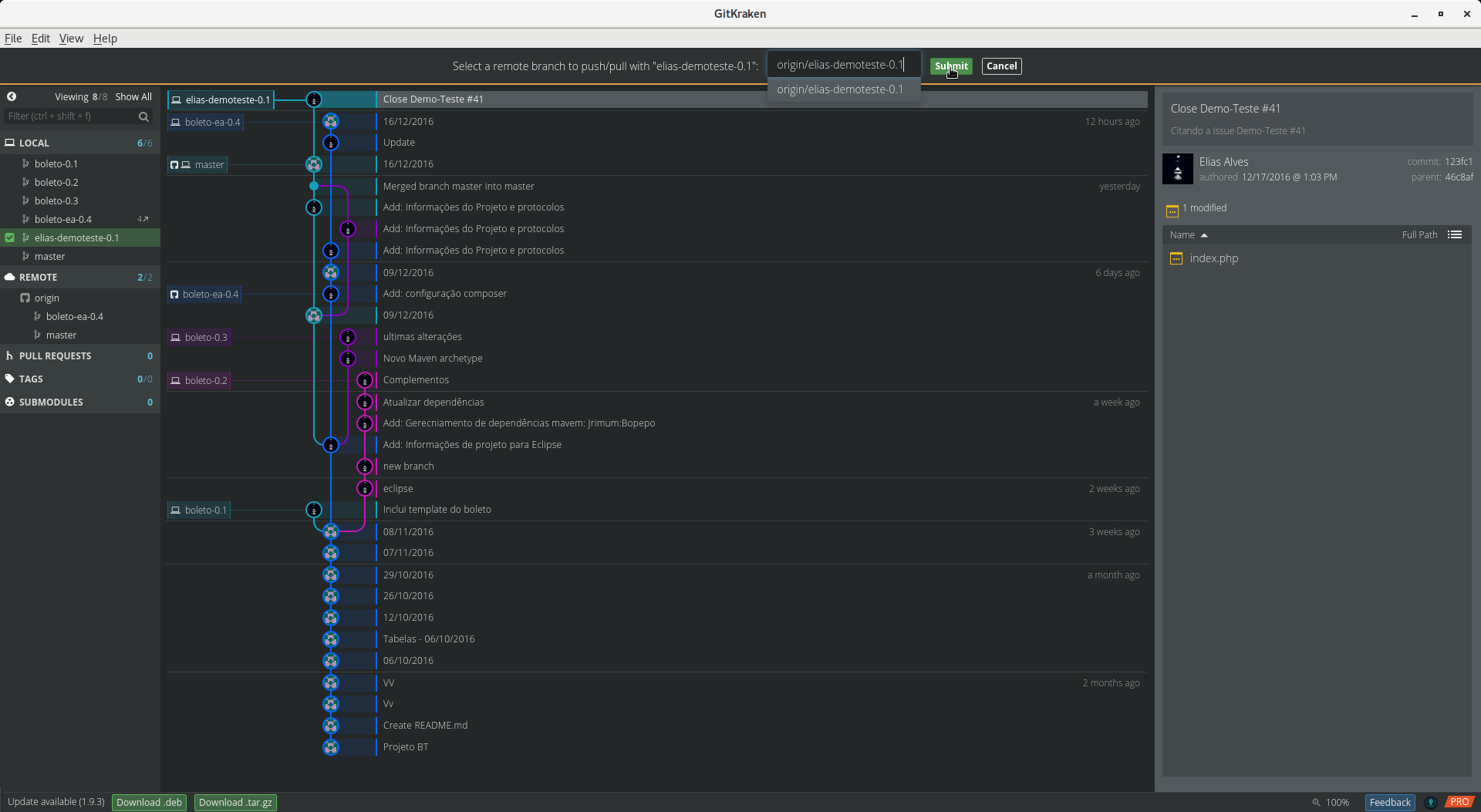
  
Figura 14: Commit pronto, com mensagem

## Passo 5.4 - Enviar seu commit

Agora basta enviar seu códido fazendo o Push. Atenção, caso seu branch seja novo, você será perguntado se deseja criar um remotamente com o mesmo nome. Basta criá-lo e seguir com o processo.

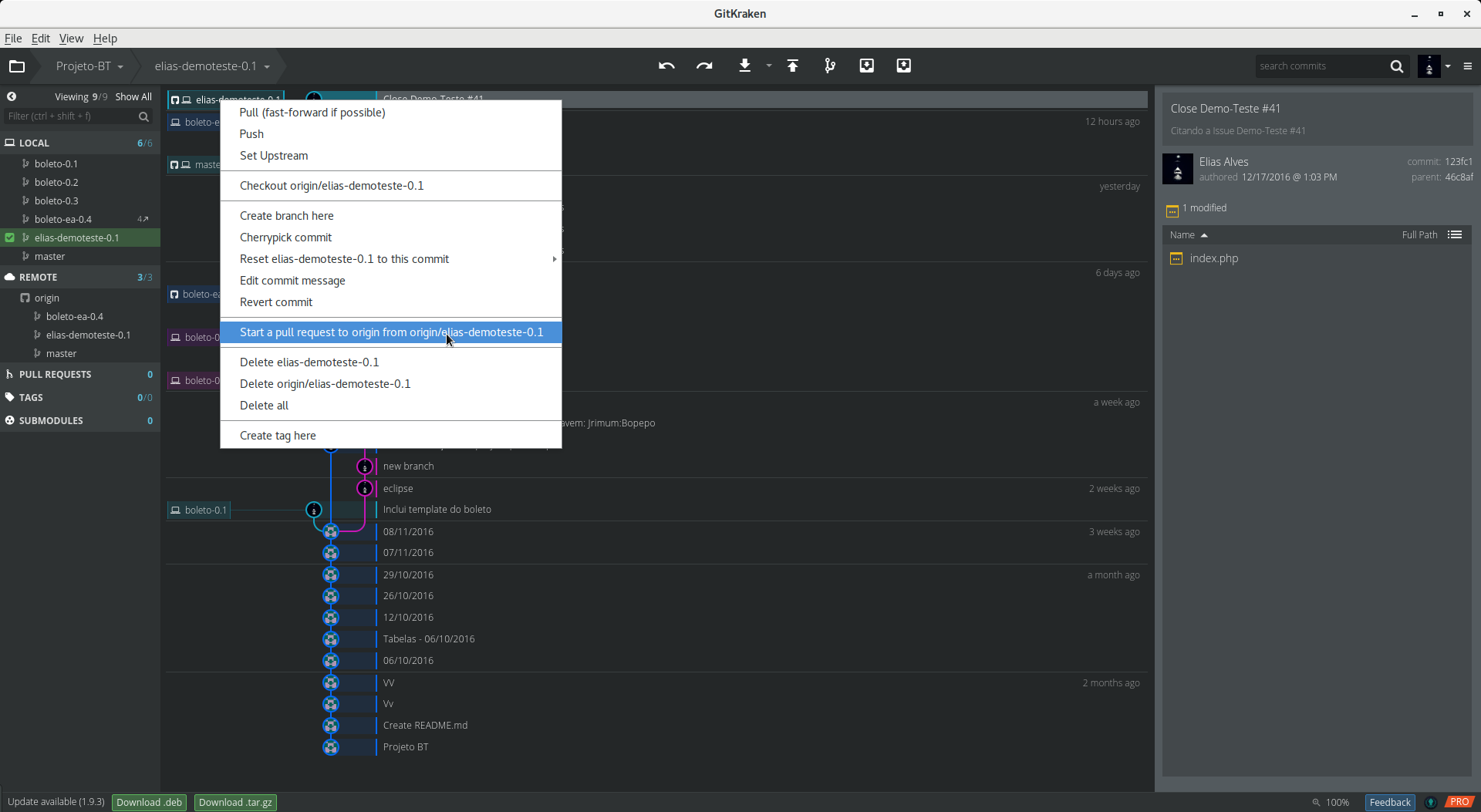
  
Figura 15: Clicando no botão para enviar o código - Push

## Passo 5.5 - Pull Request

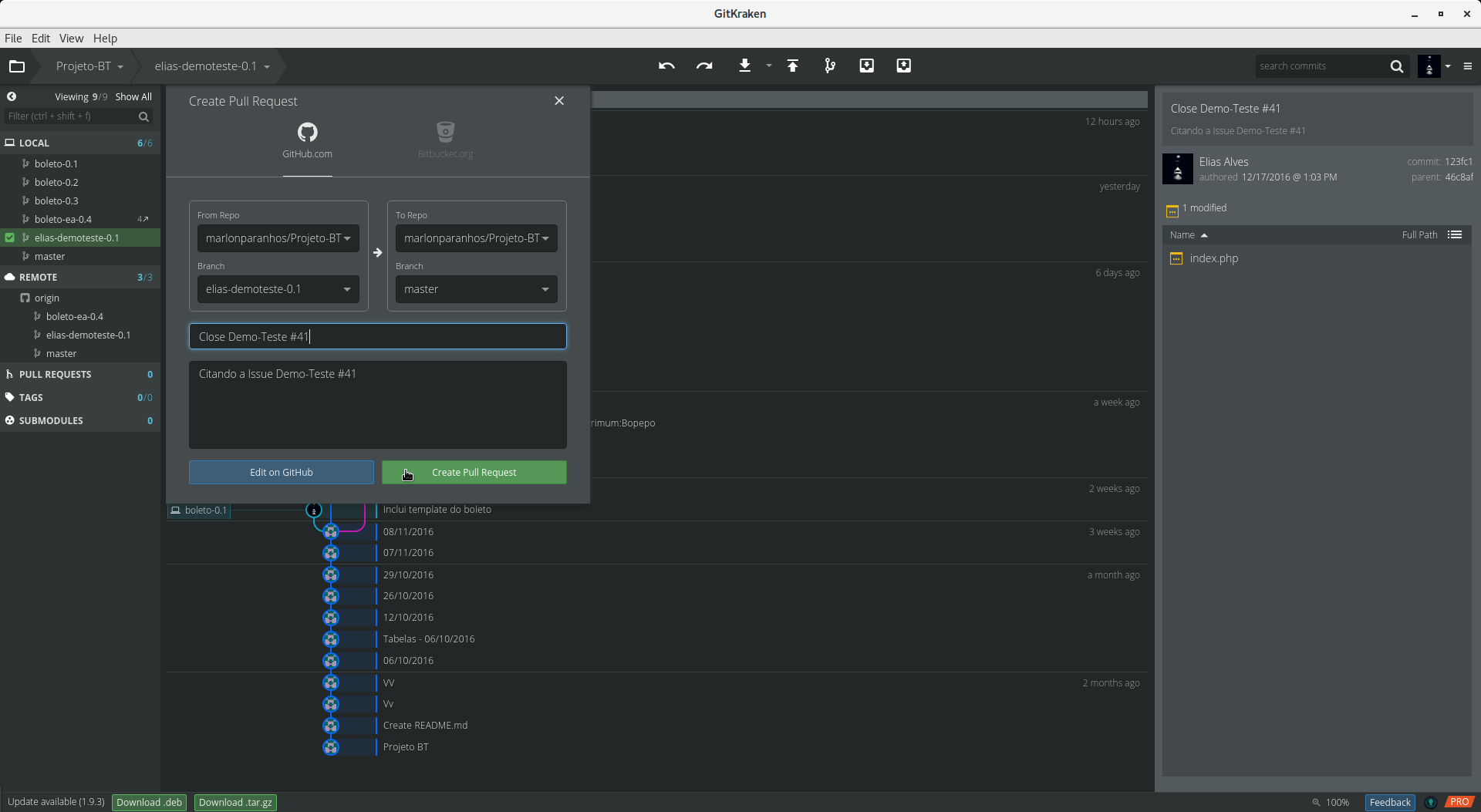
  
Figura 16: Nomeando branch remoto

Seu commit já foi enviado, agora devemos fazer o merge e registrar as suas alterações no tronco principal. Clique com o botão direito do mouse sobre o seu branch e em seguida "Start a pull request to origin..."

Em seguida, clique em "Create Pull Request"

  
Figura 17: Start Pull Request

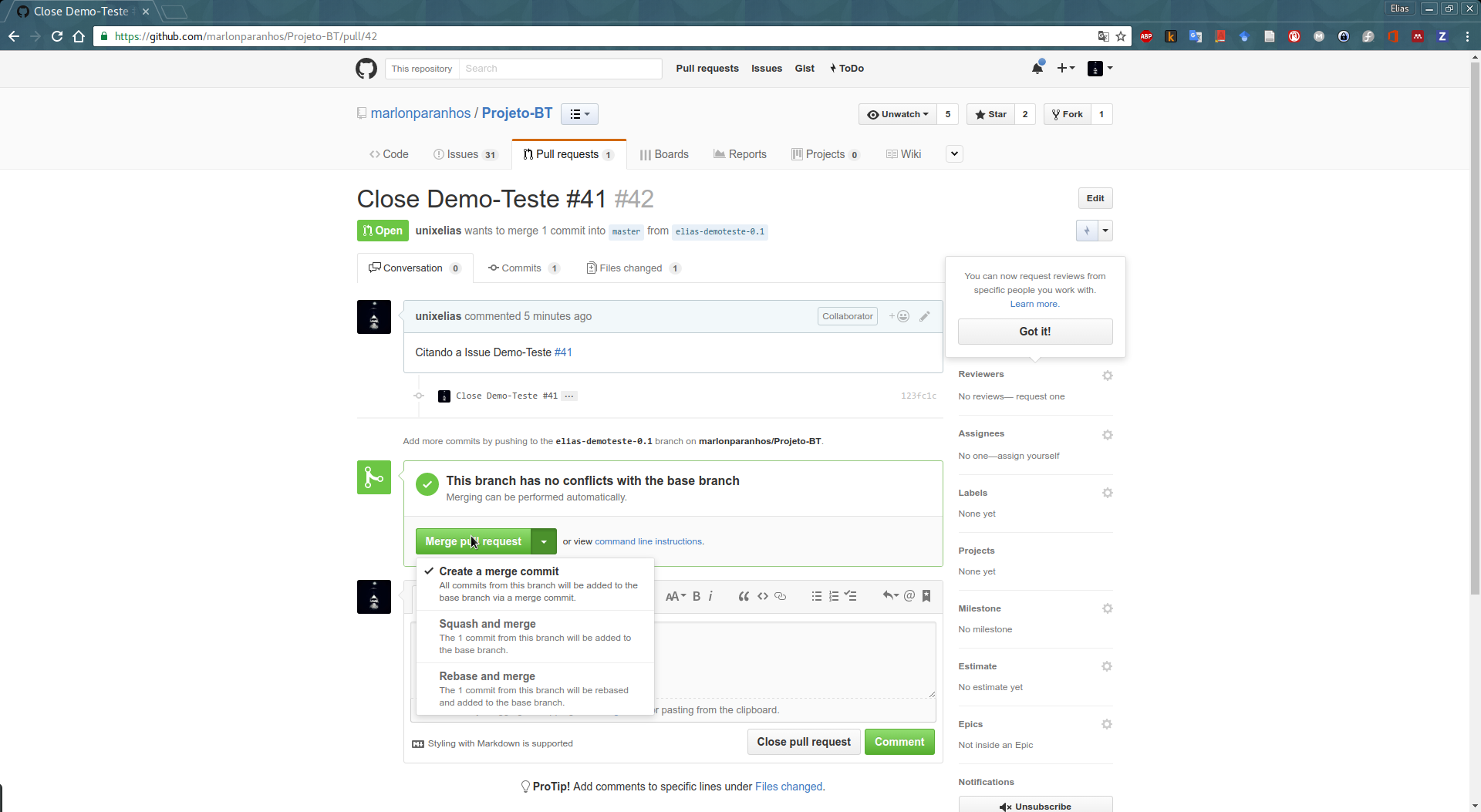
O seu pedido foi enviado ao GitHub e pode ser tratado pelo site agora.

  
Figura 18: Criando PR e enviando ao GH

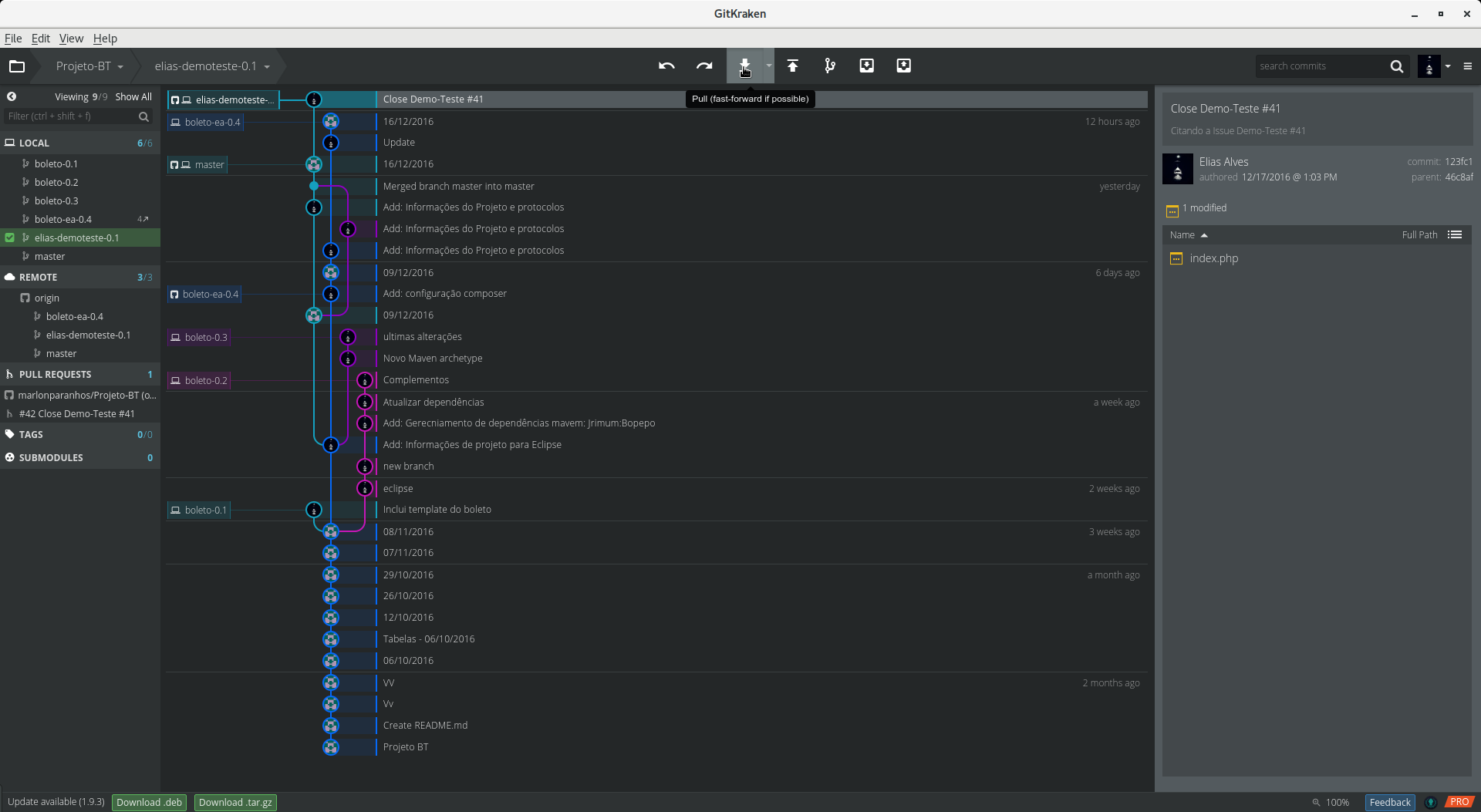
## Passo 5.6 - Fazer o merge (Somente Líderes)

Agora pelo site do GH já é possível ver o novo PR. Ele é registrado como uma nova issue e revebe um número sequencial nós faremos então o merge em "master".

Selecione o PR e em seguida, basta clicar em "Merge Pull Request"

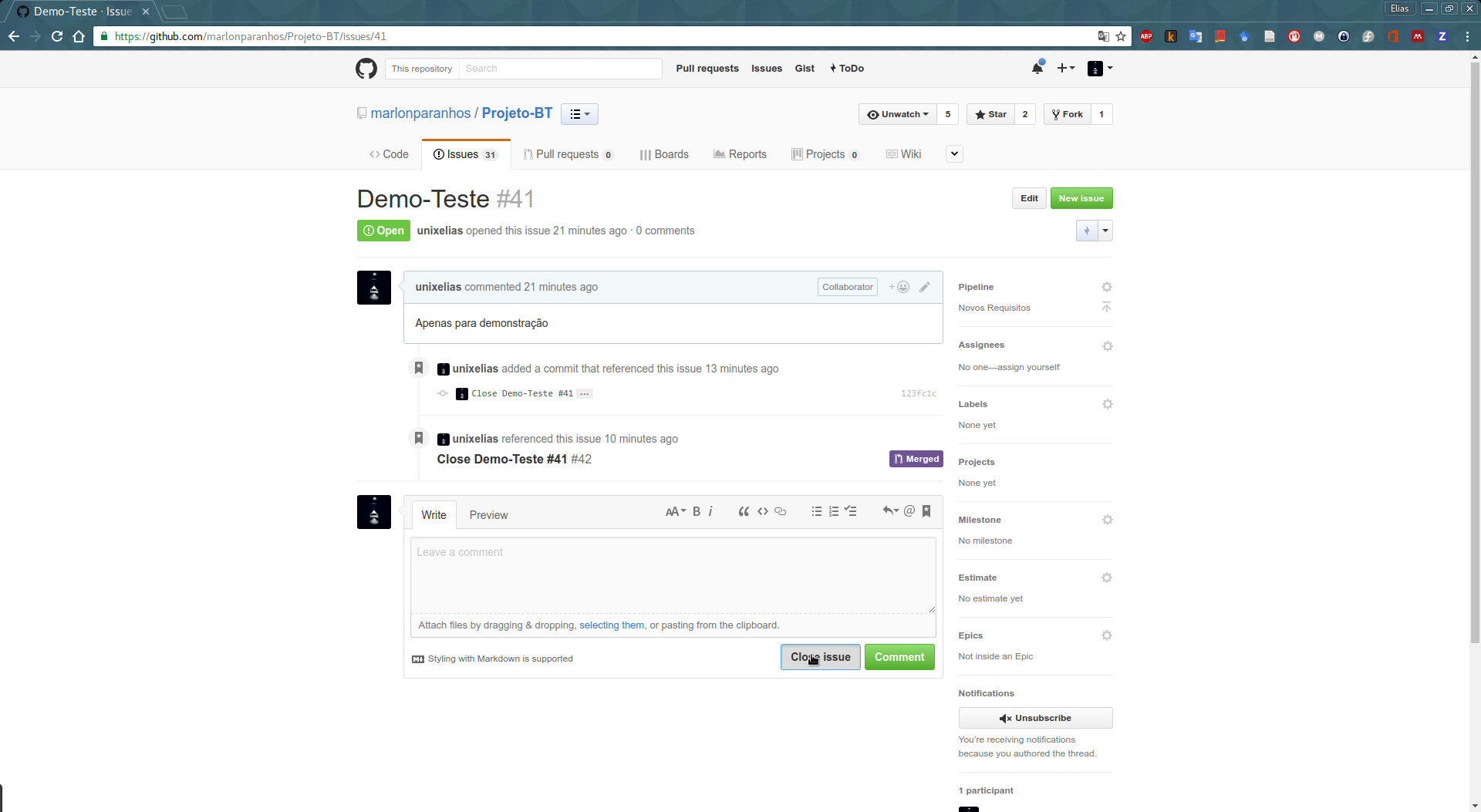
  
Figura 19: Aceitando PRs

Pronto seu PR foi incluido ao tronco principal, como se fosse um commit e agora sua funcionalidade está disponível em “master”, a contribuição será publicada no tronco principal.

  
Figura 20: Vendo o merge realizado como um commit normal

### Passo 5.6.1 - Fechar issue

Agora feche a requisição que originou seu trabalho para que todos saibam que aquela necessidade já foi atendida. Basta selecionar a Issue pelo site e clicar em "Close Issue"

  
Figura 21: Fechando Issue

Para continuar basta iniciar novamente criando um novo branch.

Não se assuste se algum arquivo “sumir”. Quando você faz checkout, o git muda todos os arquivos da pasta para aquele branch.

Bom trabalho!