

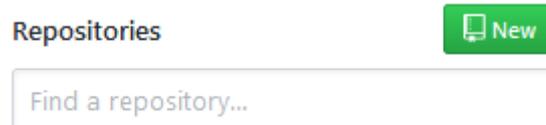
GITHUB: Instalação e configuração

[JLE-08, 09, 14 e 15/10/2019]

O Github é um repositório para desenvolvimento colaborativo de artefatos, inclusive ontologias. Os passos a seguir orientam a configuração dos ambientes web e desktop do Github para uso do código produzido de forma colaborativa na construção da ontologia.

I- Instalação e configuração:

- 1) Crie conta no **Github web**, endereço => <https://github.com/>
- 2) Use a barra de pesquisa de repositórios para localizar o repositório <https://github.com/JLouize/ontomcpse>



- 3) Ao localizá-lo, clique no ícone "**Fork**" localizado no canto superior direito da página, de forma a criar uma cópia do repositório original do projeto, para a sua conta particular no **Github web**. Assim é possível modificar livremente uma versão particular sem afetar a versão original.



- 4) Baixar o **Github desktop**, no endereço => <https://desktop.github.com/>
- 5) Localizar na pasta de downloads do computador local o arquivo **GitHubDesktopSetup.exe** e clicar 2x para iniciar a instalação.
- 6) Ao abrir o **Github Desktop** pode-se efetuar login na conta do **Github Web** (criada no passo 1), para que seja possível a sincronização dos ambientes. Esta configuração será necessária para permitir a realização de *downloads* e *uploads* de versões dos arquivos das ontologias.

OBS. **O nome do usuário e email deverão ser os mesmos cadastrados no Github web.**

Esta opção também pode ser acessada a partir do Menu "File", "Options", e em seguida, "Accounts" e "Sign in". Veja o roteiro na Figura 1:

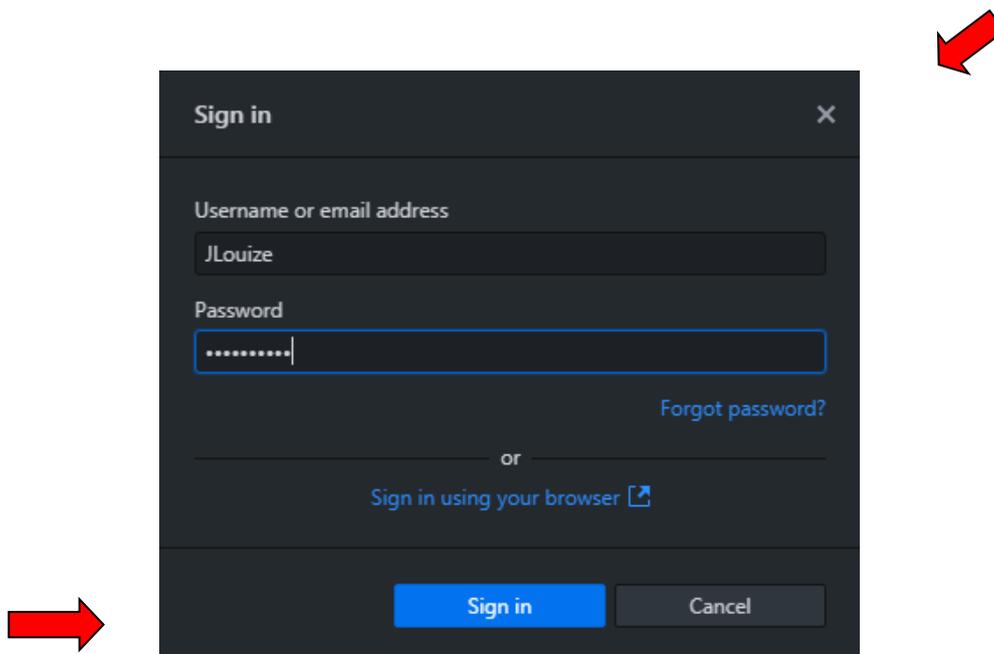
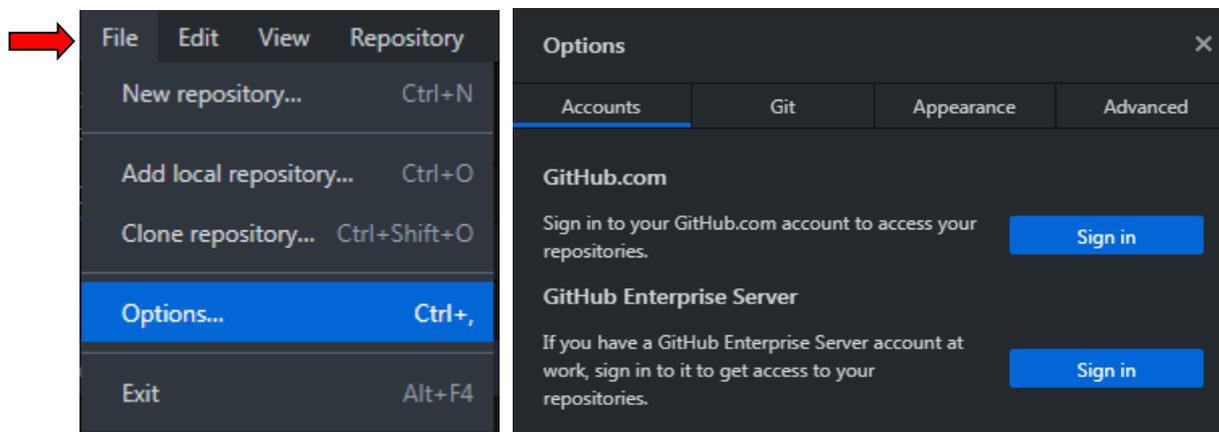


Figura 1: entrando no Github

Após login correto na conta, será possível ver uma tela semelhante a da Figura 2.

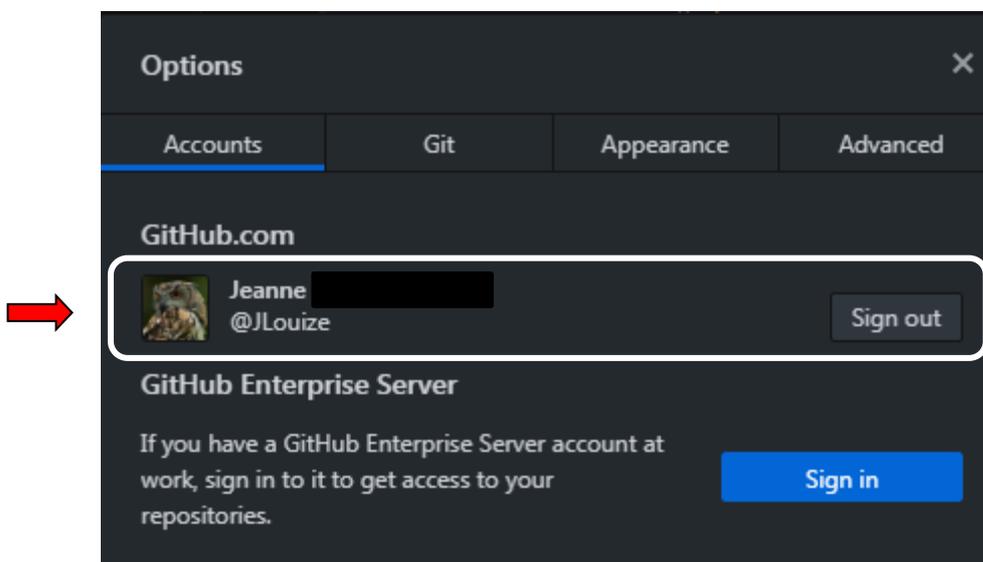


Figura 2: entrando no GtiHub

7) Selecionar as opções de menu "File", "Clone repository" (Figura 3).

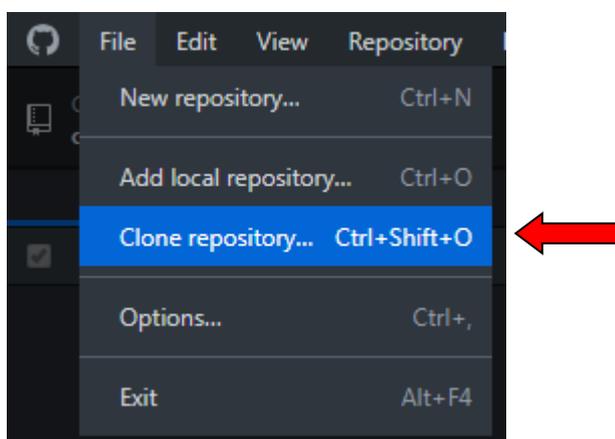


Figura 3: clonado repositório

8) Serão exibidos os repositórios criados ou copiados (quando se faz um *fork*) na respectiva conta do Github Web. Este passo é necessário para trazer os arquivos da web para computador local. Selecionar o repositório "<seu usuário>/ontomcpse". Clicar em "Clone".

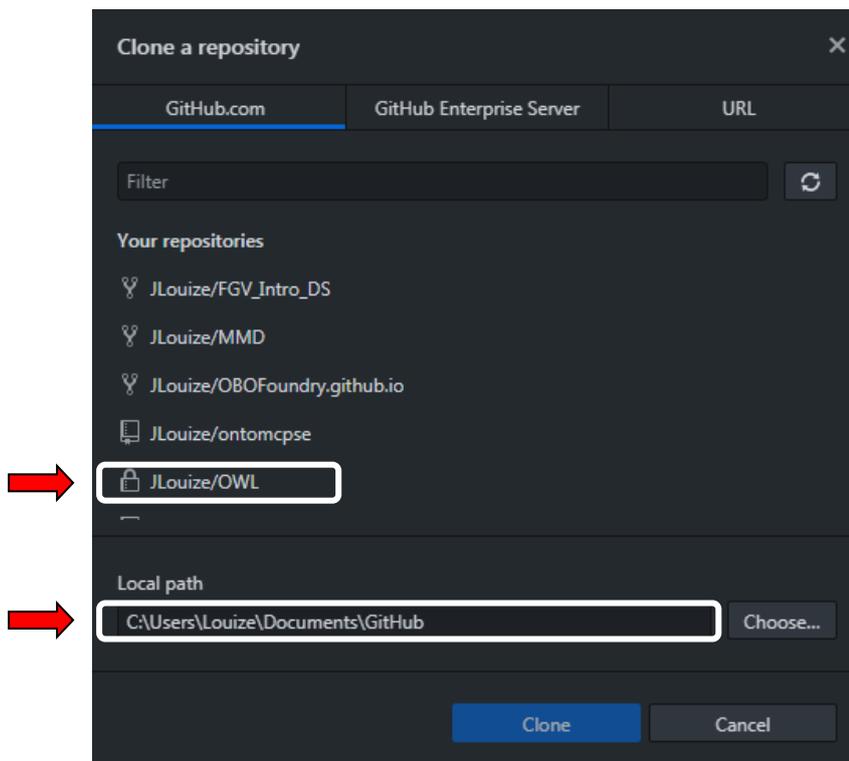


Figura 4: clonando repositório da web para local

Será exibida uma tela conforme Figura 5:

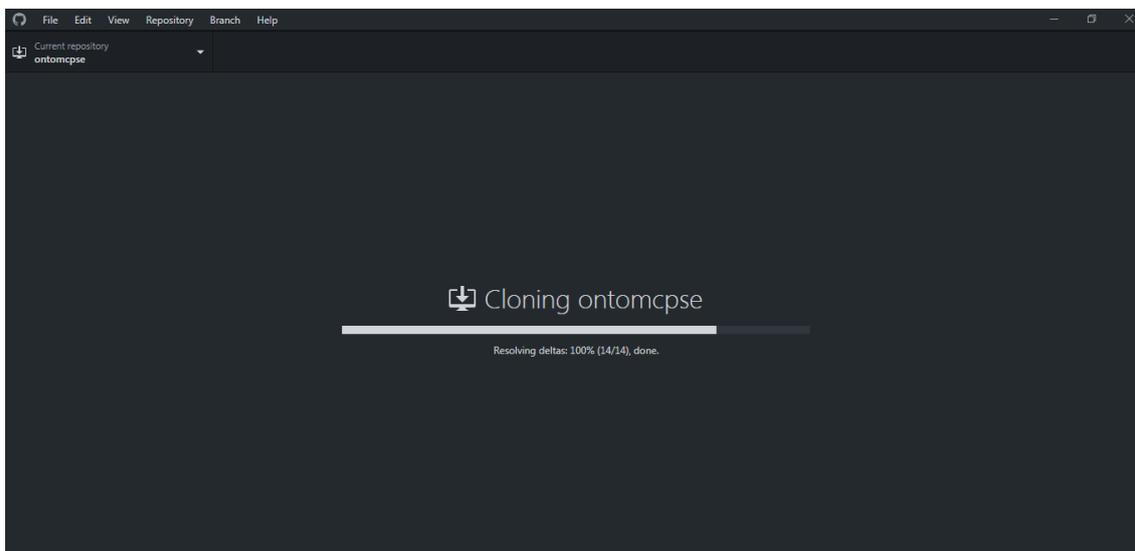
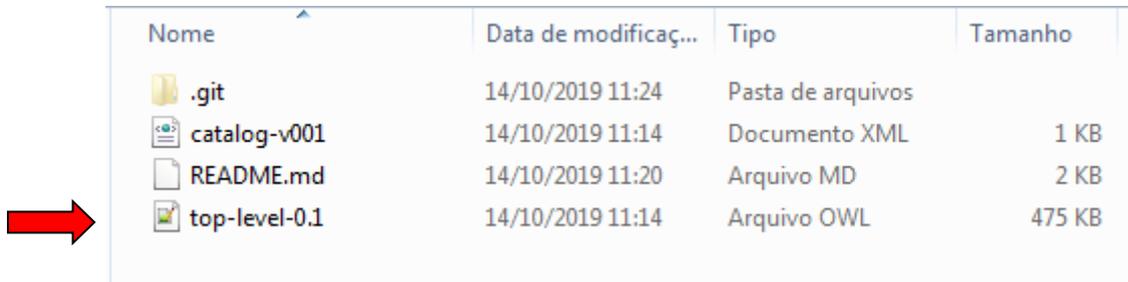


Figura 4: barra de progresso na clonagem de repositório da web para local

Observe que o "**Local path**" é o caminho onde o **Github Desktop** está instalado no computador local. Será criada uma pasta com o nome do repositório, para armazenagem dos arquivos da ontologia.



| Nome | Data de modificaç... | Tipo | Tamanho |
|---------------|----------------------|-------------------|---------|
| .git | 14/10/2019 11:24 | Pasta de arquivos | |
| catalog-v001 | 14/10/2019 11:14 | Documento XML | 1 KB |
| README.md | 14/10/2019 11:20 | Arquivo MD | 2 KB |
| top-level-0.1 | 14/10/2019 11:14 | Arquivo OWL | 475 KB |

Figura 5: cópia local resultado da clonagem

O arquivo **top-level-0.1.owl** pode ser editado a partir de então, através da ferramenta Protegé.

II- Enviando modificações para o repositório da web:

Quando são realizadas modificações em qualquer arquivo do repositório local, o Github exibe, a partir da aba "**Changes**", os arquivos modificados para que sejam selecionados aqueles que se deseja enviar ao repositório da web (Figura 6).

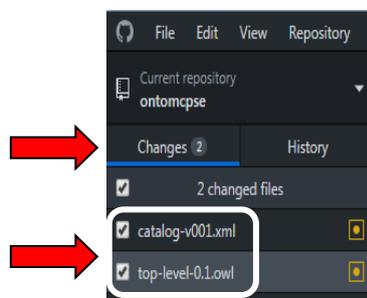


Figura 6: mudanças exibidas na aba "**Changes**"

Será possível ainda, observar as modificações que foram realizadas, a partir da aba onde se encontra o nome da ontologia, no exemplo da Figura 7 consta "**top-level0.1.owl**". As modificações inseridas são precedidas do sinal "+" e exibidas com fundo verde sendo que; as que são excluídas são precedidas do sinal "-" e exibidas com fundo vermelho.

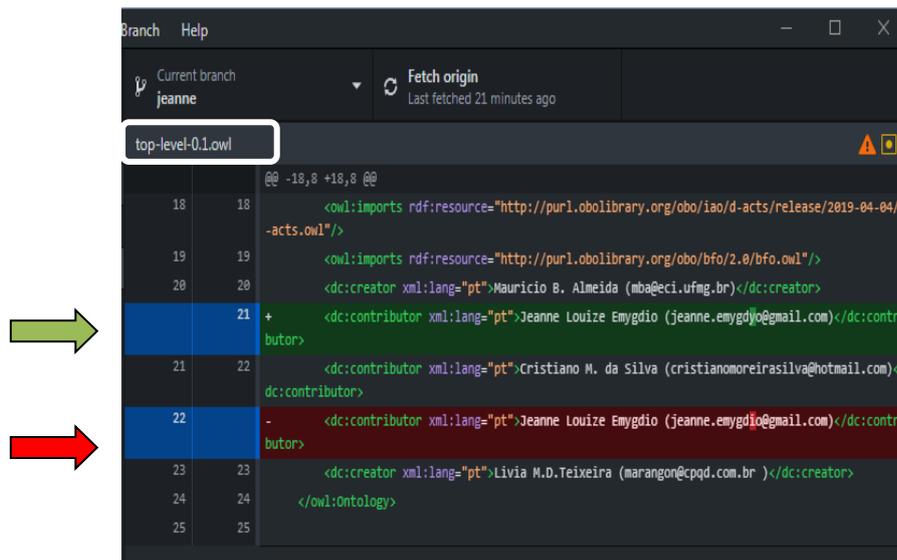


Figura 7: mudanças exibidas

Recomenda-se sempre um *branch* ou *fork* no arquivo atual, em que se está trabalhando, antes de fazer *commits* (entregas) de versões (Figura 8). As versões serão enviadas para o repositório da web.

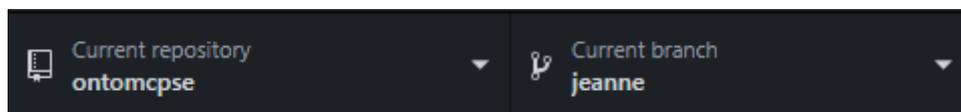


Figura 8: exibição de um *branch* realizado

O envio de versões para o repositório da web é feito em duas etapas: *commit* (entrega) e *push* (envio), conforme ilustrado na Figura 9:

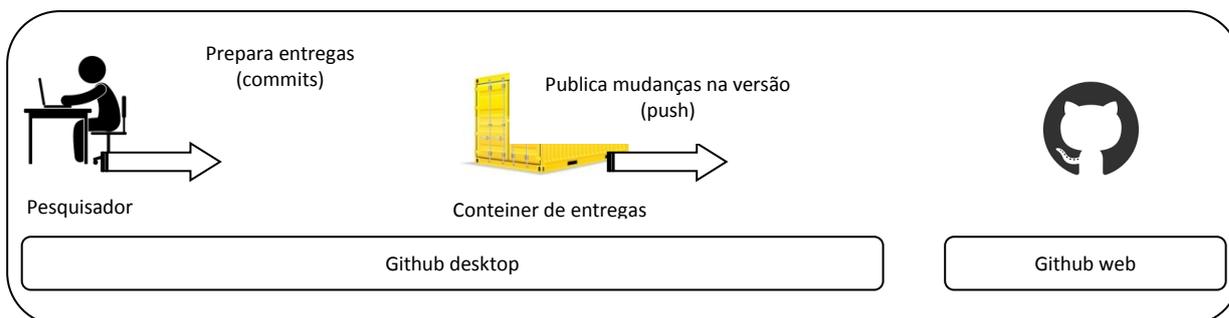


Figura 9: *commit* e *push*

Para realizar um *commit*, insere-se um título e uma descrição das modificações que constam na versão. Em seguida clique no botão "**commit to <nome do repositório>**" (Figura 10).

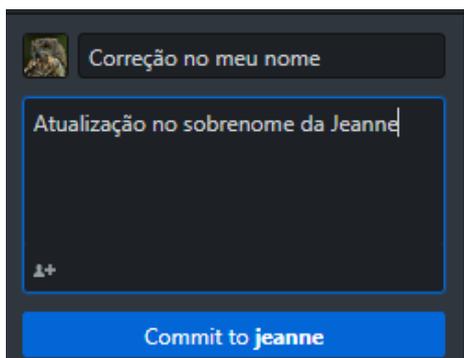


Figura 10: realizando um *commit*

O aplicativo irá informar que há um *commit* local aguardando para ser enviado ao Github. Clique então em "**Push origin**" (Figura 11).

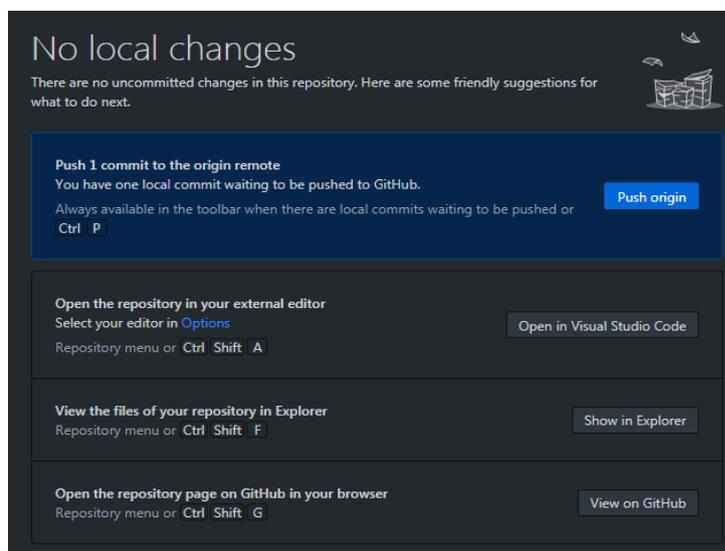


Figura 11: realizando um *push*

As modificações realizadas poderão ser conferidas ao acessar o repositório do **Github web** (Figura 12).

Manage topics

23 commits 2 branches 0 releases 1 contributor

Your recently pushed branches:

jeanne (2 minutes ago) [Compare & pull request](#)

Branch: jeanne [New pull request](#) [Create new file](#) [Upload files](#) [Find file](#) [Clone or download](#)

This branch is 1 commit ahead of master. [Pull request](#) [Compare](#)

JLouize Correção no meu nome Latest commit 532495e 3 minutes ago

| File | Commit | Time |
|-------------------|----------------------|---------------|
| README.md | Update README.md | 1 hour ago |
| catalog-v001.xml | Correção no meu nome | 3 minutes ago |
| top-level-0.1.owl | Correção no meu nome | 3 minutes ago |

README.md

Repositório para uso da equipe do Projeto OntoMCPSE

Branch master: Deverá hospedar a Ontologia de Alto Nível original e integrações futuras
 Autor: Prof. Dr. Maurício Barcellos Almeida
 Versão original: Alfa 0.1, disponibilizada em 13/10/2019 - Construída no Protege 5.00 Build RC1

Ferramenta a ser utilizada pelos pesquisadores:

Figura 12: resultado de alterações

Revertendo um *commit*

É possível reverter um *commit* para desfazer o último trabalho salvo no *branch/fork*. Quando se reverte para um *commit* prévio, um novo *commit* é gerado. O *commit* original também permanece no histórico do repositório.

Se necessária a reversão de múltiplos *commits*, a reversão deve ser realizada do *commit* mais novo para o mais velho. A reversão em ordem diferente gera conflitos de integração.

Para solicitar uma reversão de *commit*, seleciona-se a aba "**History**", clique sobre o *commit* que se deseja reverter com o botão direito do mouse e selecione em seguida "**Revert this commit**" (Figura 13).

MANUAL BÁSICO DE INSTALAÇÃO E USO GITHUB (2019)

Jeanne Louize Emygdio (jeanne.emygdio@gmail.com), PhD candidate, UFMG

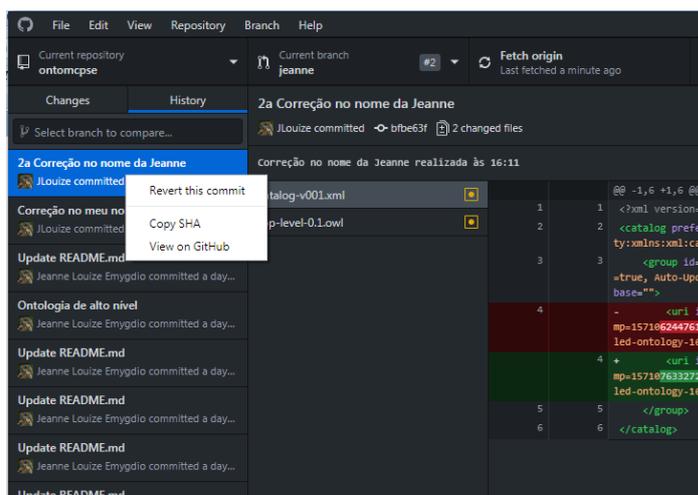
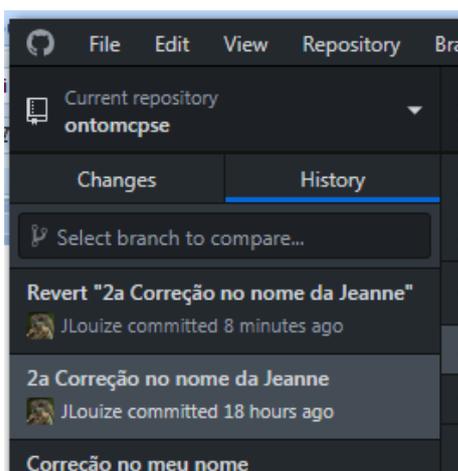


Figura 13: reversão de *commits*

As modificações poderão ser conferidas em ambos os repositórios do Github (Figura 14)

Local:



Web:

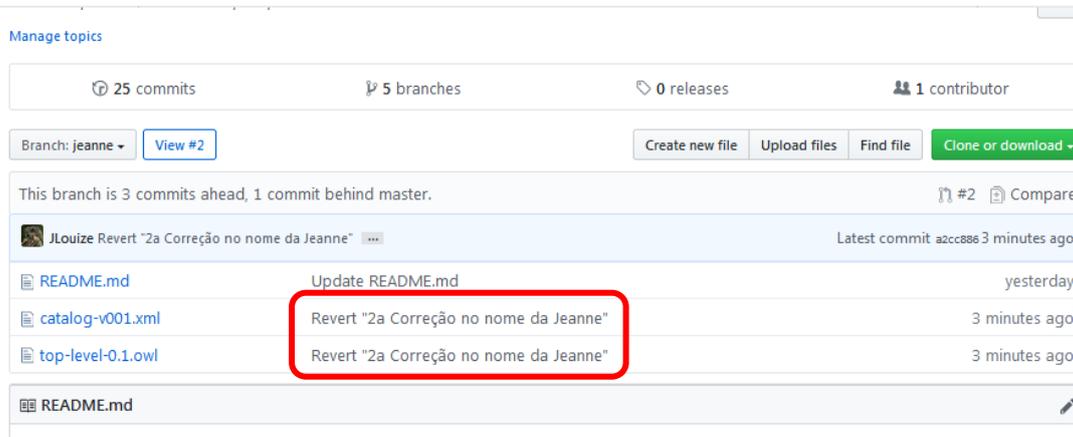


Figura 14: resultados local e web

Problemas comuns de commits:

Falta de sincronização dos repositórios. Deve ser realizado um "**fetch**" para sincronizar:

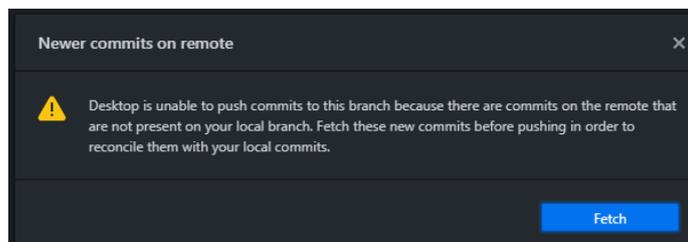


Figura 15: erro de sincronização

Tentativa de *commit* em repositório diferente do repositório do usuário. Deve-se selecionar o *branch/fork* adequado antes do envio.

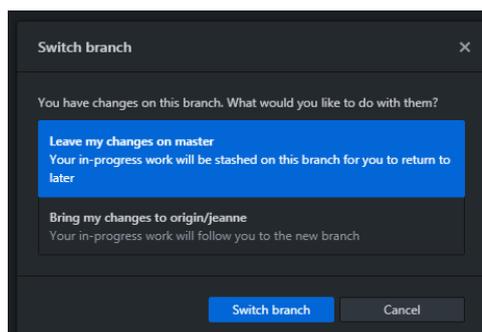


Figura 16: repositório equivocado

III- Criando *pull requests*:

Caso deseje solicitar um *pull request*, clicar em "**Create Pull Request**" na janela que será exibida após a etapa de "**push**" (Figura 17).

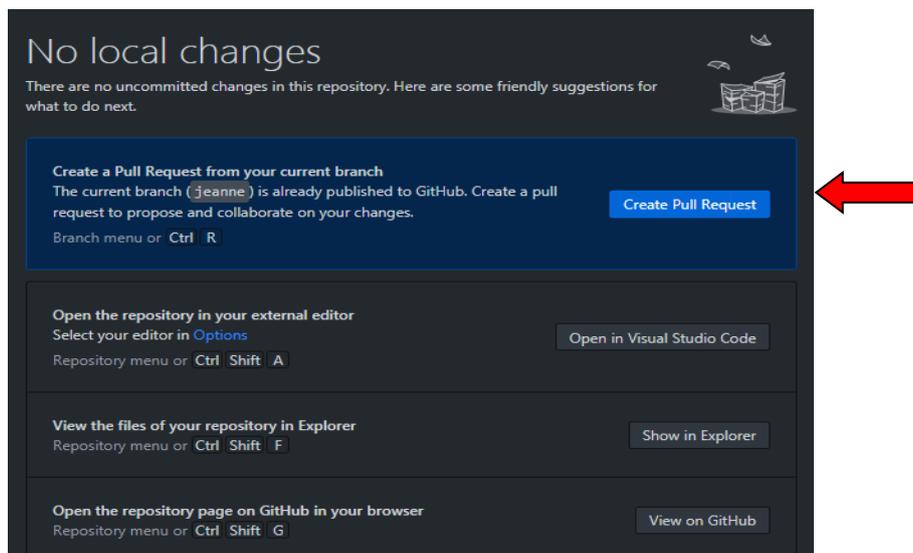


Figura 17: criando um pull-request

Será aberta uma página no navegador web para que o pedido seja registrado. A área de edição disponibilizada a partir da aba "write" permite o envio de mensagens ao administrador do repositório sobre o qual se deseja o pull request.

O Github analisa automaticamente a possibilidade de integração das versões e pode exibir uma mensagem informando que versões podem ser automaticamente integradas. Tal procedimento ocorrerá se não houver regra de segurança habilitada sobre o *branch* em questão. Clicar sobre o botão "Create pull request" (Figura 18):

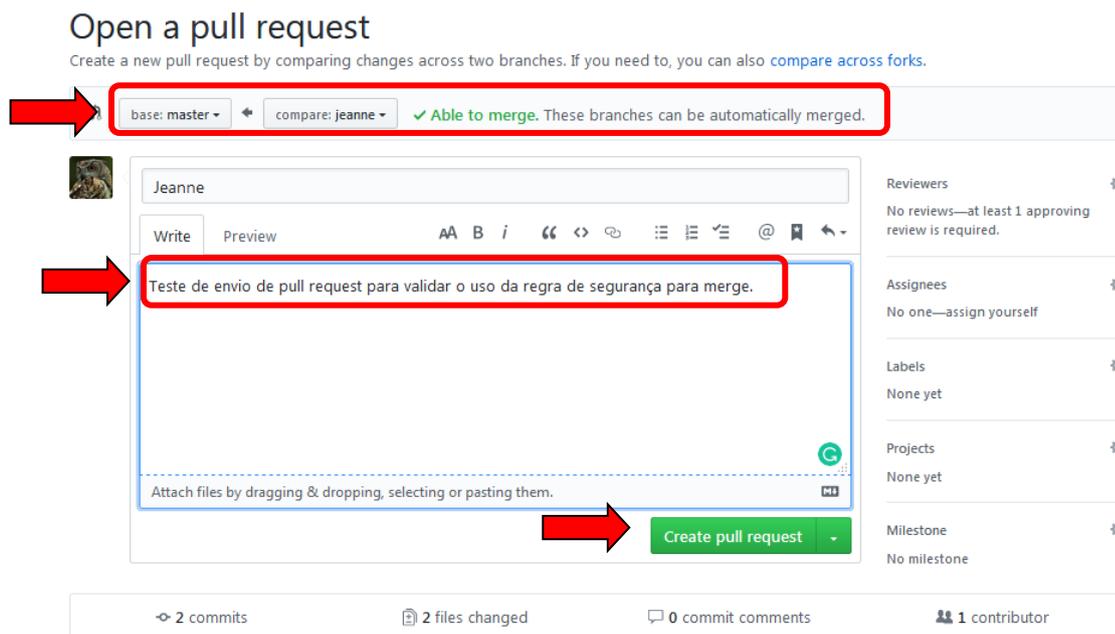


Figura 17: criando um *pull-request*

Será possível consultar os "pulls requests" solicitados para o repositório a partir da aba "Pull requests". Havendo regras de segurança habilitadas para o *branch* desejado, a requisição ficará aguardando revisão.

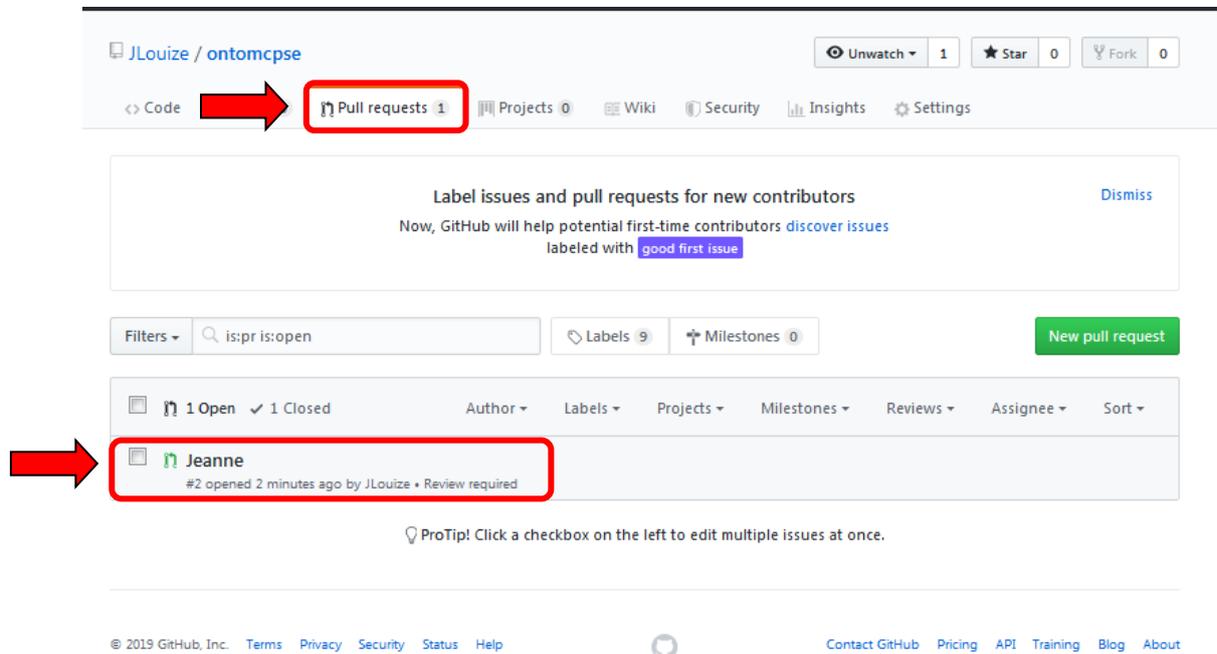


Figura 18: criando um *pull-request*
