PROGRAMAÇÃO BACK END II

Maurício de Oliveira Saraiva

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS

Implantação de um sistema de controle de versões distribuído

Objetivos de aprendizagem

Ao final deste texto, você deve apresentar os seguintes aprendizados:

- Criar repositórios Git e GitHub.
- Relacionar o repositório local com o repositório remoto.
- Transferir o repositório local para o GitHub.

Introdução

Com a evolução da tecnologia da informação, os sistemas se tornaram cada vez maiores e mais complexos, exigindo, em muitos casos, a colaboração entre diversos desenvolvedores de uma ou mais equipes, de modo que as tarefas possam ser divididas em grupos especializados.

A partir desse processo colaborativo, identificou-se a necessidade de controlar as versões dos arquivos dos projetos em repositórios locais e remotos, uma vez que muitos arquivos são modificados isoladamente pelos desenvolvedores.

Neste capítulo, você irá estudar sobre a criação de repositórios Git e GitHub para o controle de versão de arquivos, o relacionamento entre o repositório local e o repositório remoto e a transferência do repositório local para o GitHub.

Repositórios Git e Github

Normalmente utilizados por programadores para gerenciar as versões de seus códigos-fonte, Git e GitHub são repositórios *open source* que implementam o versionamento eficiente de arquivos. O gerenciamento ocorre pelo controle das alterações realizadas nos arquivos utilizados em projetos de qualquer natureza (GIT..., 2019b; GITHUB..., 2019).

As versões são gerenciadas pelos usuários, em âmbito local (Git) ou remoto (GitHub), e estes decidem o momento em que elas serão registradas/atualizadas. Isso significa que todas as versões que foram definidas podem ser restauradas, permitindo, assim, um bom controle de versão dos arquivos.

Git

Criado por Linus Torvalds, o Git é um sistema distribuído de controle de versão que tem como objetivo gerenciar as modificações realizadas nos arquivos de um determinado projeto no âmbito local. Embora tenha sido criado com o objetivo de controlar as versões de sistemas em desenvolvimento, pode gerenciar o histórico de modificações em arquivos de qualquer tipo de projeto (GIT..., 2019a).

O Git trabalha com uma estrutura em forma de árvore, cujas modificações realizadas podem ser organizadas em ramificações (*branches*) para facilitar a administração do conteúdo. O controle de versão implementado pelo Git facilita o trabalho colaborativo, registrando dados das modificações realizadas nos arquivos, bem como seus responsáveis (GIT..., 2019c).

Este sistema pode ser baixado em seu *site* oficial, que contém *links* para sua documentação e fórum de discussão, assim como para os *downloads*, que estão disponíveis para diversas plataformas, como Windows, Linux/Unix e Mac OS X.

Link

Acesse o *link* a seguir para fazer o *download* do Git.

https://qrgo.page.link/bCvB1

A instalação do Git é simples e não requer nenhuma configuração adicional: basta clicar no botão *next* das janelas que aparecem durante a execução do assistente de instalação.

Após concluir a instalação, pode-se criar um repositório local por meio de sua interface gráfica, *Graphical User Interface* (GUI), ou por linha de comando (Git Bash). Neste capítulo, apresentaremos a criação dos repositórios por meio de sua interface gráfica.

Repositório Git

Criar um repositório local é o primeiro passo para começar a trabalhar com o Git. Para isso, clique com o botão direito do mouse em qualquer pasta do gerenciador de arquivos de seu computador e selecione *Git GUI Here*, conforme ilustra a Figura 1.

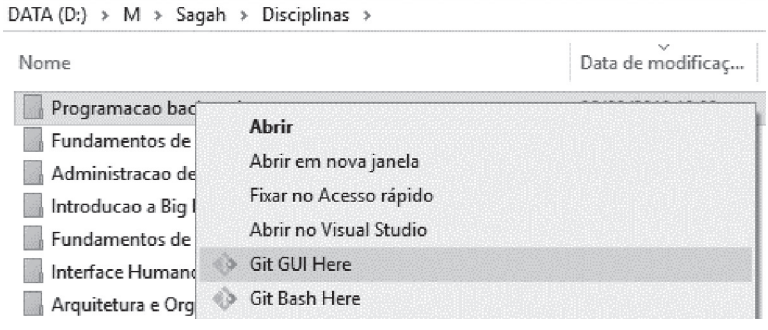

Figura 1. Acessando o Git GUI.

Uma janela do Git GUI que apresenta a opção criar um novo repositório (*Create New Repository*) será aberta, conforme ilustrado na Figura 2.

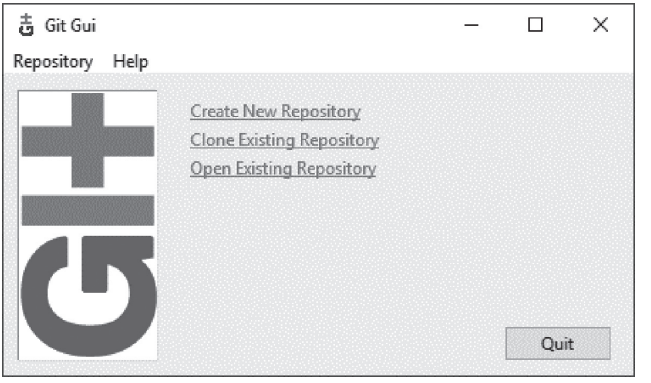

Figura 2. Janela principal do Git GUI.

Clique na opção indicada anteriormente e selecione o local em que o projeto está instalado em seu computador. Em seguida, clique no botão *Create* para confirmar a criação do novo repositório, conforme apresentado na Figura 3.

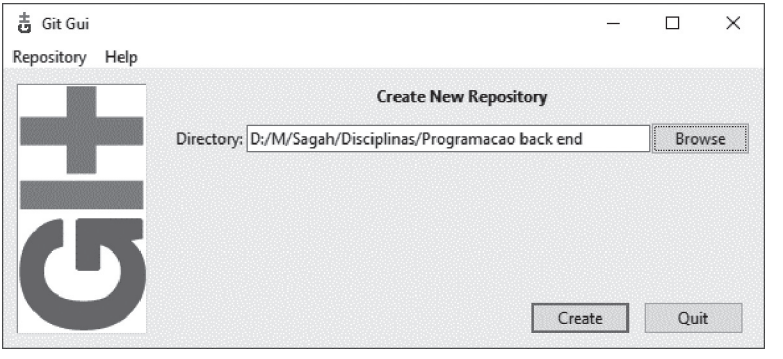

Figura 3. Selecionando a pasta do repositório.

Após a criação do repositório, uma subpasta nomeada **.git** será criada na pasta do projeto. Essa subpasta contém os elementos que farão o gerenciamento das versões dos arquivos do projeto.

A interface gráfica apresentará a lista de arquivos do repositório, cujo *status* da corrente *branch* encontra-se como *Unstaged Changes*, o que significa que as mudanças ainda não foram organizadas. Neste momento é preciso configurar apenas o usuário e o *e-mail*, que futuramente serão utilizados para acessar o repositório remoto (GitHub).

Para isso, acesse o menu *Edit — Options* da interface gráfica e preencha *User Name* e *Email Address* nos dois lados da janela, conforme ilustrado na Figura 4. As demais opções não precisam ser modificadas.

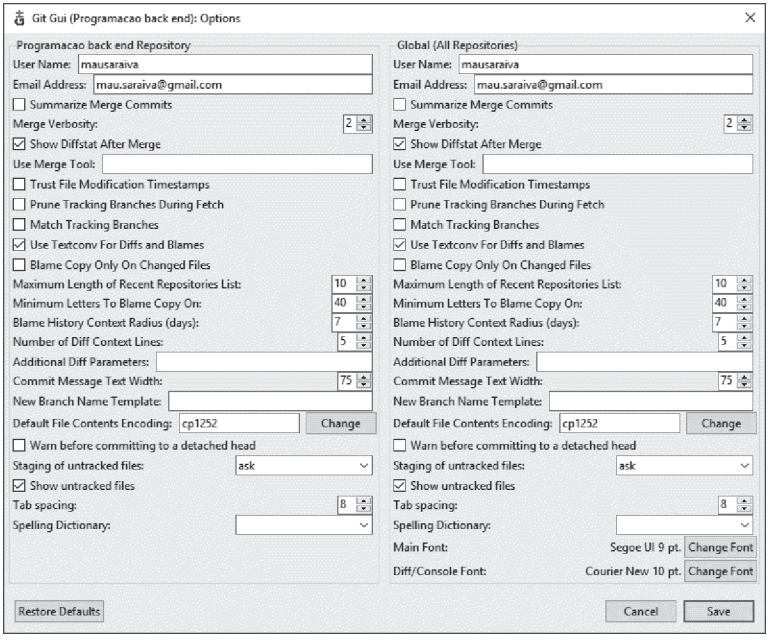

Figura 4. Inserindo o nome do usuário e o *e-mail*.

GitHub

O GitHub é um serviço *open source* de hospedagem de código-fonte que permite o controle de versão de arquivos de projetos de diversos tipos. Os arquivos, que são armazenados em *branches*, podem ser sincronizados com repositórios locais Git, permitindo fazer *uploads* ou *downloads* quando necessário (GITHUB..., 2019).

Link

Este serviço pode ser acessado no *link* a seguir.

http://www.github.com/

Acesse o *site* indicado e faça o *login* (*Sign in*) com sua conta ou crie (*Sign up*) uma conta, caso ainda não tenha. Para o registro do *login* é preciso informar um nome de usuário inédito, assim como um *e-mail* válido e uma senha que possua entre oito e 15 caracteres, contendo números e letras.

Repositório GitHub

Após entrar no GitHub, clique na opção *New*, no canto superior esquerdo da janela, para criar um repositório remoto. Preencha o nome do repositório e sua descrição. Após, selecione se o repositório é público ou privado e clique em *Create repositor*, conforme ilustra a Figura 5.

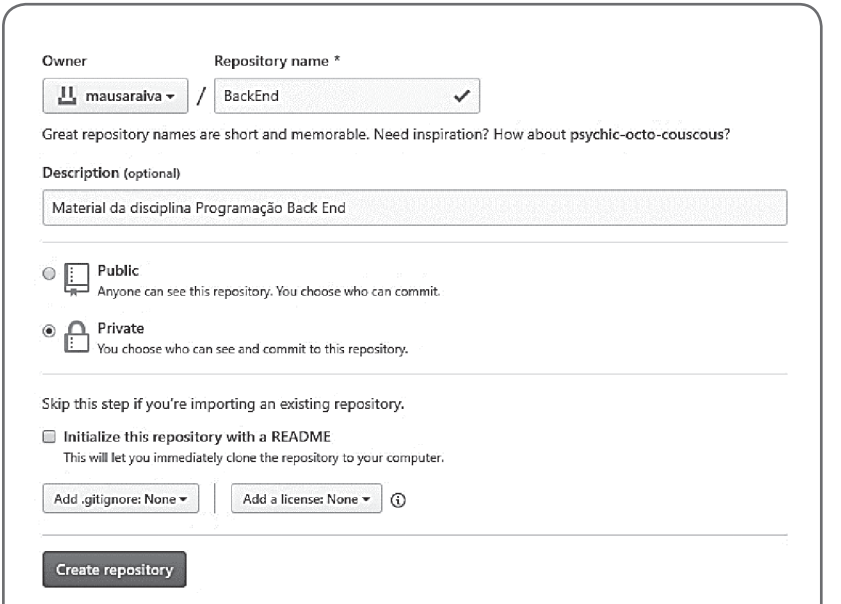

Figura 5. Criando um repositório remoto.

Relacionamento entre o repositório local e repositório remoto

Uma vez que os repositórios local (Git) e remoto (GitHub) tenham sido criados, é possível relacioná-los de modo que os arquivos do repositório local possam ser copiados para o repositório remoto, ou vice-versa.

Para isso, acesse a interface gráfica do Git e selecione a opção *Add* do menu *Remote*, como ilustra a Figura 6.

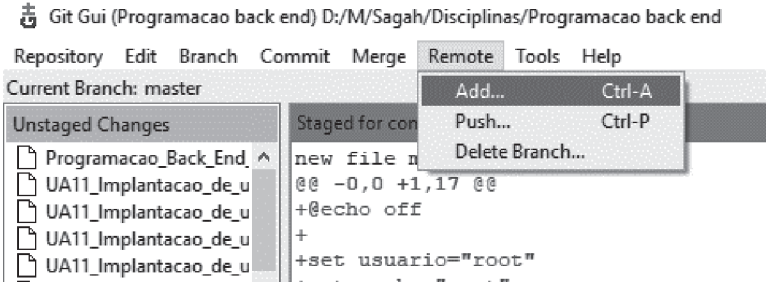

Figura 6. Adicionando um repositório remoto.

Na janela *Add New Remote*, como mostra a Figura 7, preencha o nome do repositório remoto e o *link* desse repositório no serviço GitHub. Neste momento, o propósito é associar os repositórios local e remoto. Sendo assim, selecione a opção *Do Nothing Else Now* para somente deixá-los relacionados.

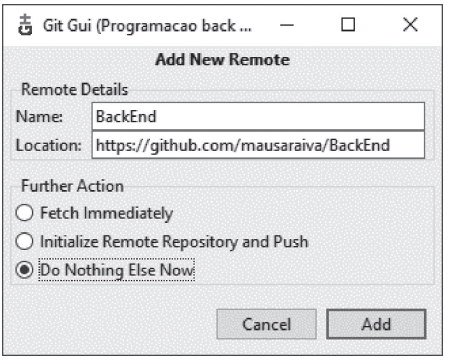

Figura 7. Nome e localização do repositório remoto.

A partir do momento em que a associação é realizada, o menu *Remote* passa a apresentar as opções que permitem manipular o repositório remoto, conforme ilustrado na Figura 8.

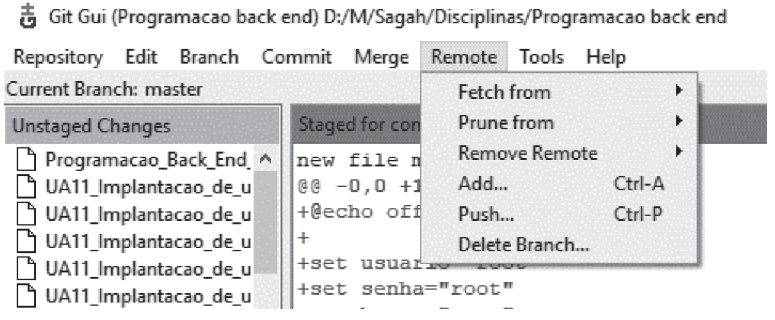

Figura 8. Opções de manipulação do repositório remoto.

Transferência do repositório local para o GitHub

Para transferir os arquivos do repositório local para o Github é preciso adicioná-los na área *Staged*, localizada logo abaixo da área *Unstaged* na interface Git, conforme mostra a Figura 9. Para isso, selecione os arquivos desejados e clique na opção *Stage Changed* e, após, confirme clicando em *Sim*.

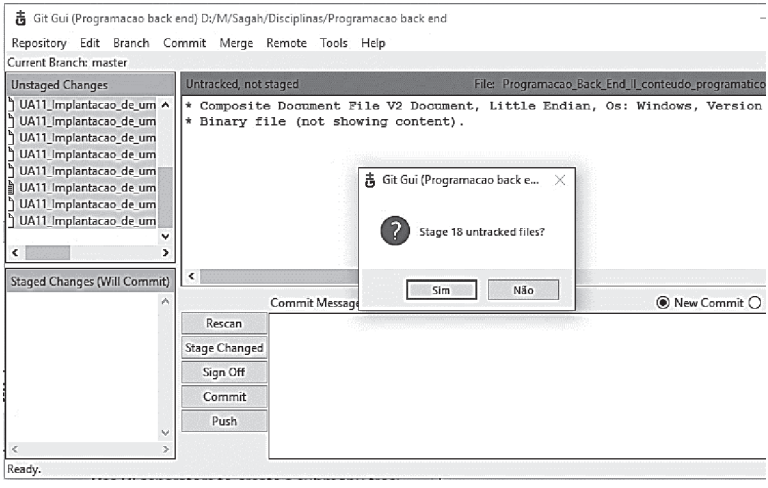

Figura 9. Execução do comando *Commit*.

Esta operação faz com que os arquivos sejam movidos para a área *Staged Changes* da interface gráfica Git. Preencha um comentário em *Commit Message* e clique no botão *Commit* na caixa ao lado. Após, clique no botão *Push*, que fica logo abaixo do botão *Commit*.

Verifique o nome do repositório remoto e confirme no botão *Push* da janela, conforme apresentado na Figura 10.

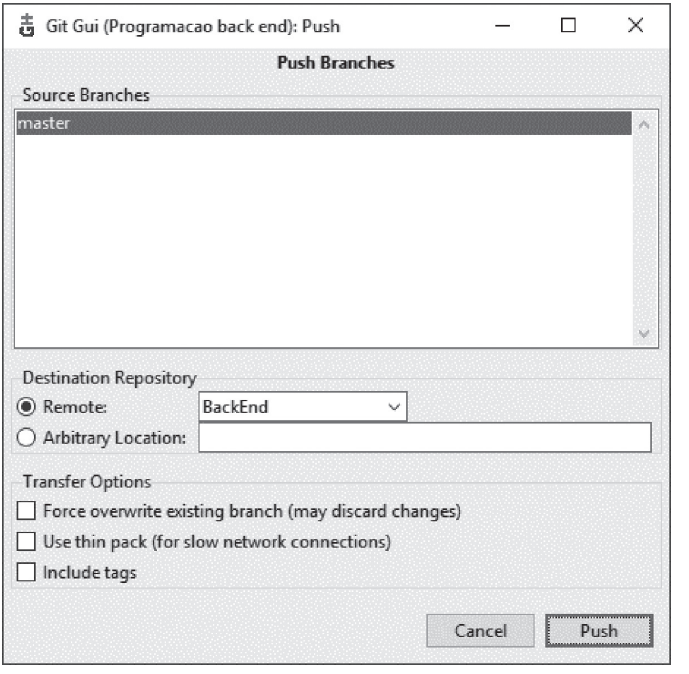

Figura 10. Execução do comando *Push*.

A Figura 11 apresenta o resultado da execução do *upload* dos arquivos, após a execução do *Push*.

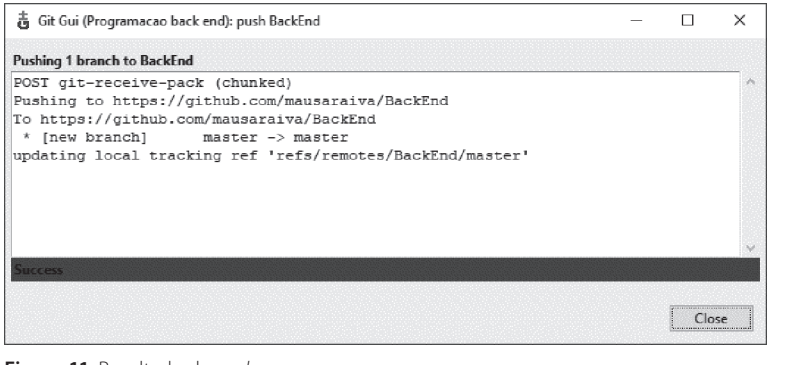

Figura 11. Resultado do *push*.

Agora o repositório remoto (GitHub) apresenta os arquivos do projeto que foram transferidos do repositório local (Git), conforme ilustrado na Figura 12.

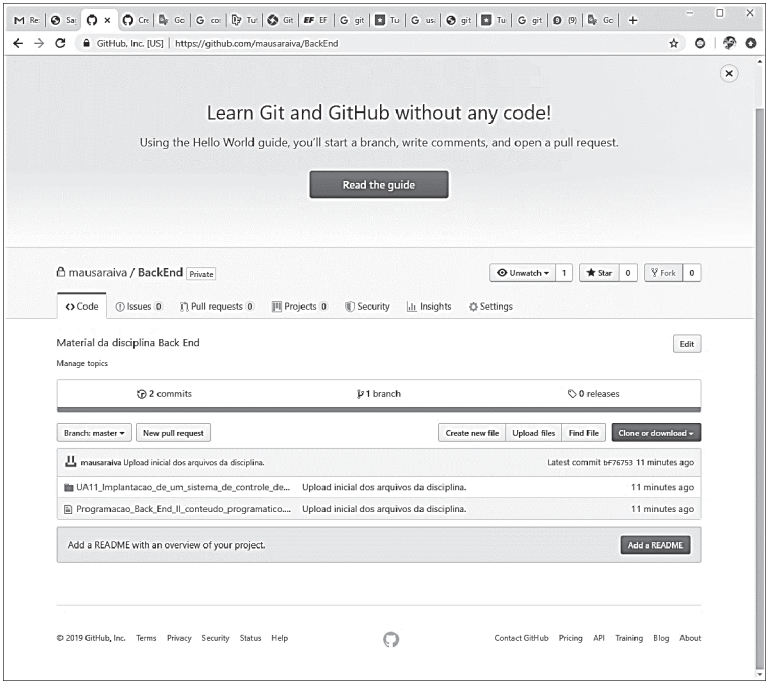

Figura 12. Resultado da execução do comando *push*.

Referências

GIT – a free open source distributed version control system. *Git/Software Freedom Conservancy*, New York, 2019a. Disponível em: https://git-scm.com/. Acesso em: 21 ago. 2019.

GIT – Getting Started – A Short History of Git. *Git/Software Freedom Conservancy*, New York, 2019b. Disponível em: https://git-scm.com/book/en/v2/Getting-Started-A-Short--History-of-Git. Acesso em: 21 ago. 2019.

GIT – Reference. *Git/Software Freedom Conservancy*, New York, 2019c. Disponível em: https://git-scm.com/docs. Acesso em: 21 ago. 2019.

GITHUB – The world's leading software development platform. *GitHub*, San Francisco, 2019. Disponível em: https://github.com/. Acesso em: 21 ago. 2019.

Leitura recomendada

AQUILES, A.; FERREIRA, R. *Controlando versões com Git e GitHub*. São Paulo: Casa do Código, 2014. 220 p.

Encerra aqui o trecho do livro disponibilizado para esta Unidade de Aprendizagem. Na Biblioteca Virtual da Instituição, você encontra a obra na íntegra.

Conteúdo:

SAGAH
SOLUÇÕES EDUCACIONAIS INTEGRADAS