

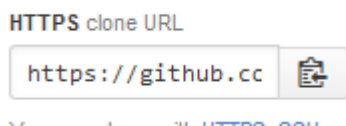
NPI - Guia de uso do Git e GitHub com Eclipse

Primeiramente será necessário clonar o projeto do Github para o computador, seguindo os passos abaixo:

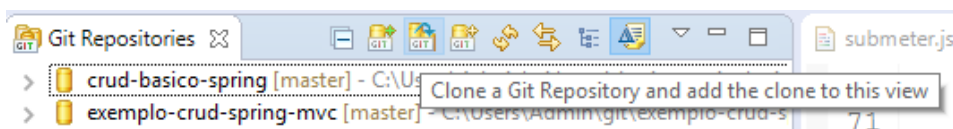
1. Abra a perspectiva do **Git** no eclipse.



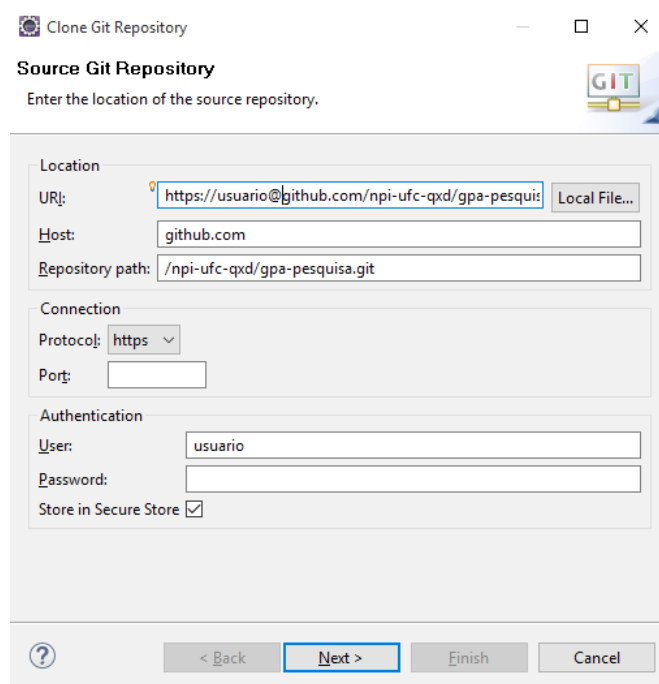
2. Na página do projeto no Github, copie o link do projeto.



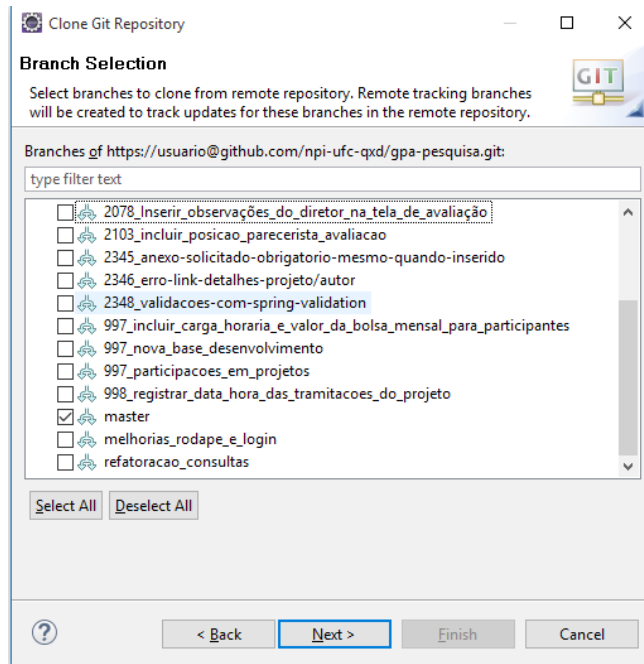
3. Na aba **Git Repositories** do Eclipse, clique no botão **Clonar**.



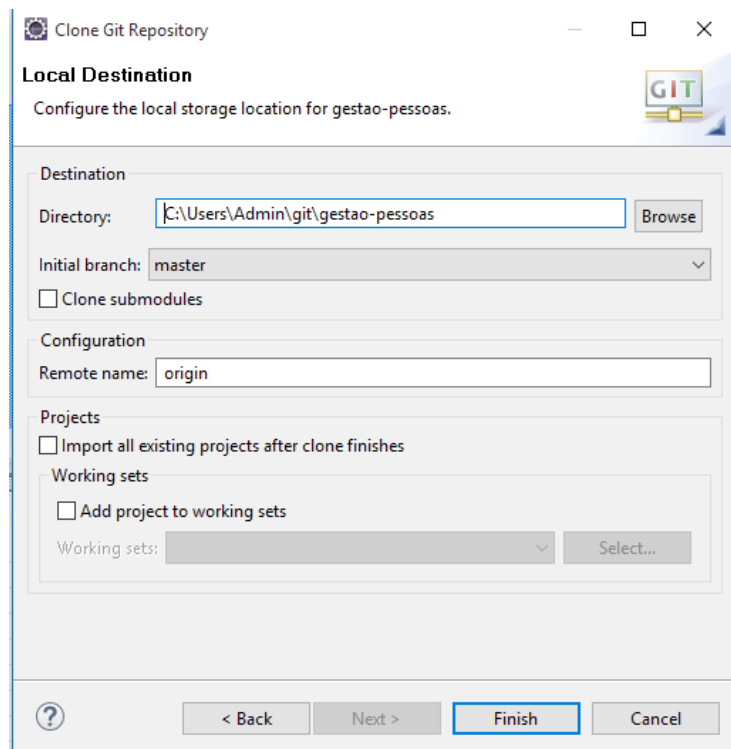
4. No campo URI, após **https://** adicione seu usuário do Github seguido de **@** para que não seja necessário digitá-lo todas as vezes que realizar uma ação do git posteriormente. Digite sua senha no campo correspondente e avance.



5. Selecione apenas o item **master** e avance.

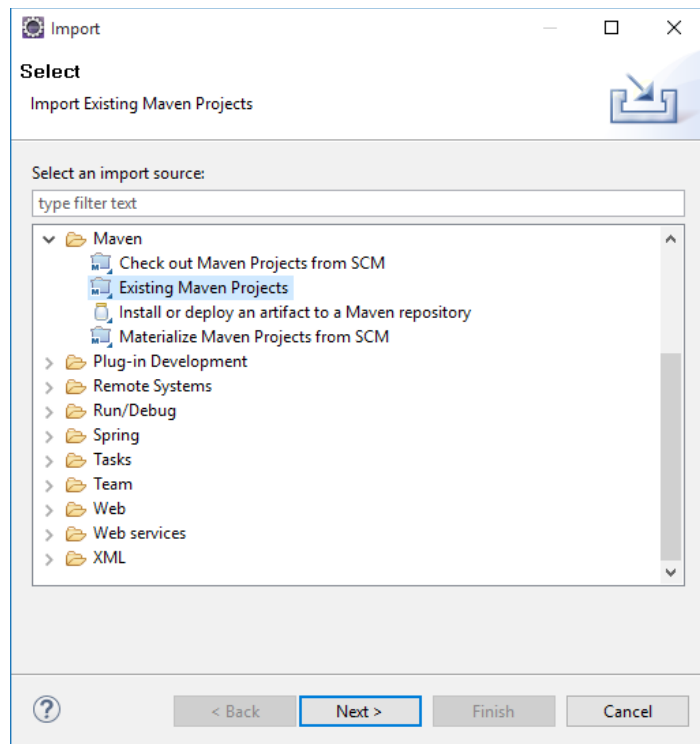


6. Selecione o diretório do git de sua preferência e finalize. Em seguida, será feito o download do projeto.

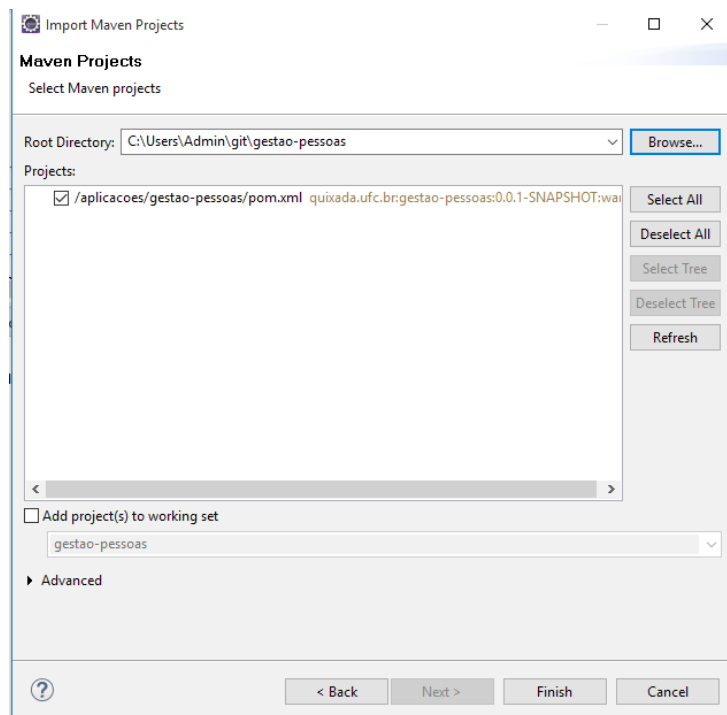


Após finalizar o download, será necessário importar o projeto para o eclipse para que seja possível executá-lo.

1. No eclipse, escolha a opção **File > Import**. Depois selecione **Existing Maven Projects** e avance.

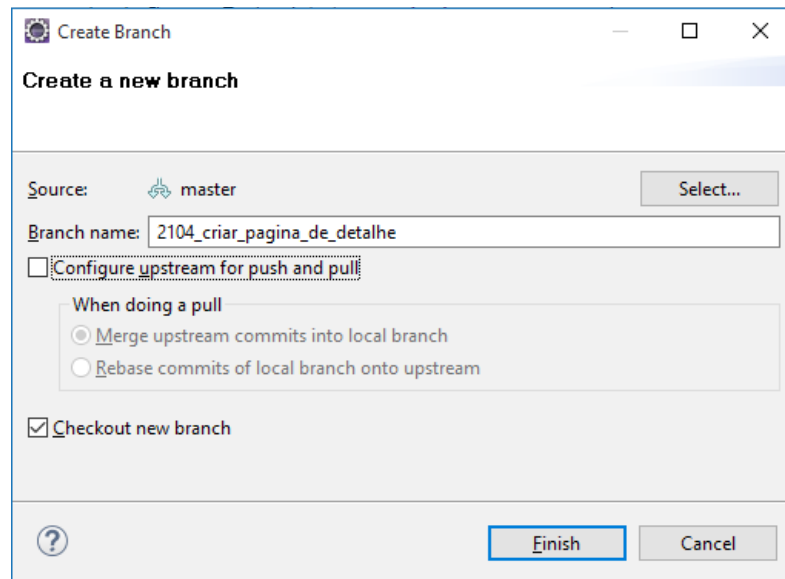


2. Selecione a pasta onde está o projeto e finalize.



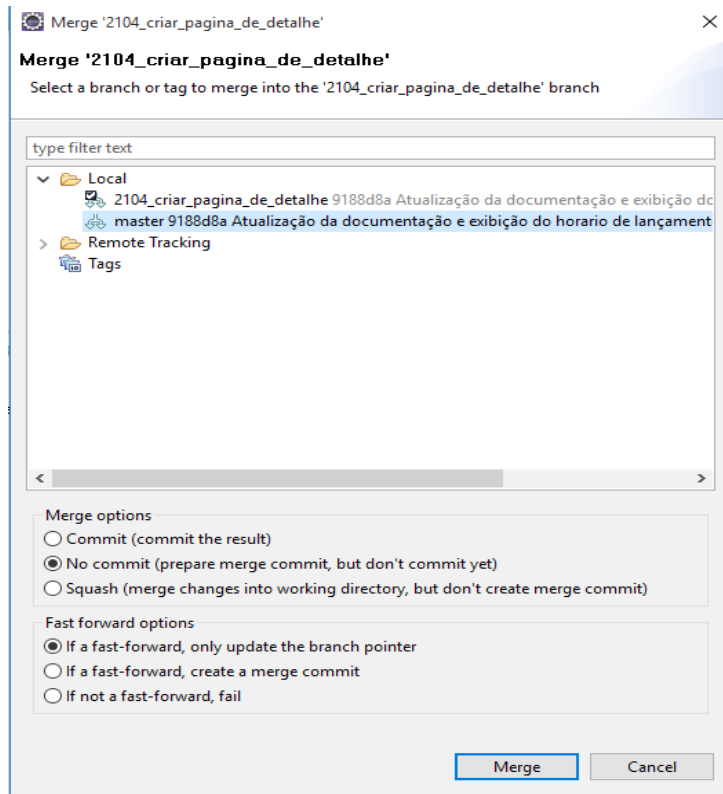
Ao iniciar uma nova atividade, será necessário criar um branch para realizar as alterações necessárias no projeto.

1. Clique com o botão direito no projeto e selecione **Team > Switch to > New Branch....** Em Branch name coloque o identificador da tarefa no redmine seguido de uma breve descrição da atividade e finalize.

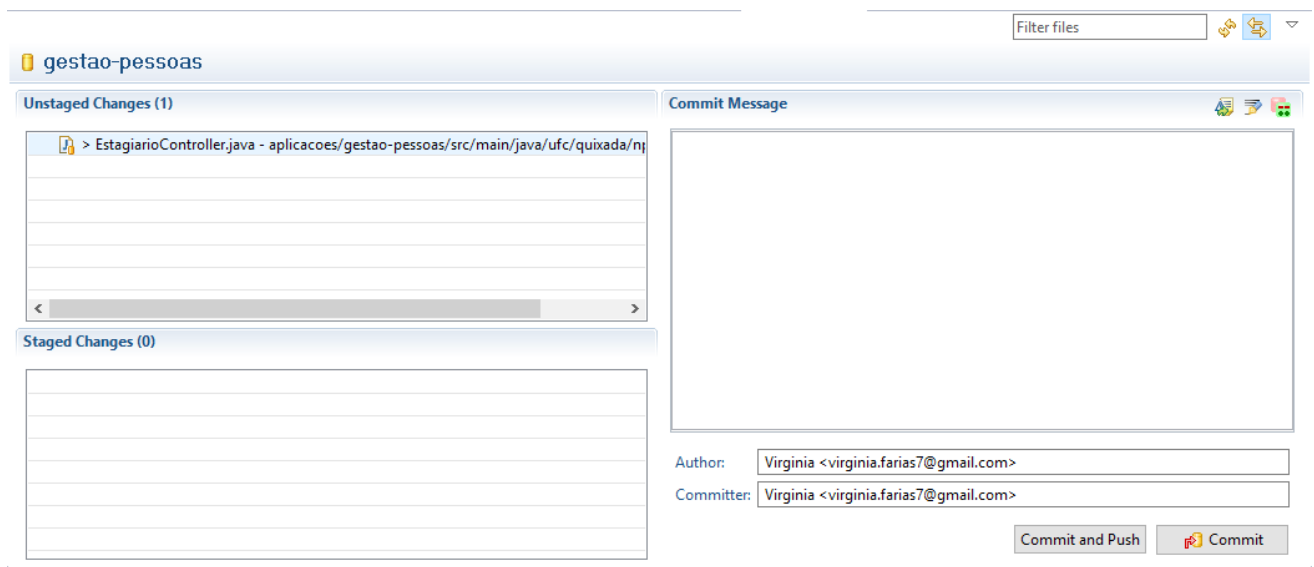


Após o termino da tarefa, será necessário fazer um merge com o master para que outras modificações no projeto reflitam nas atividades desenvolvidas.

1. Mude o branch para o master. Clique com o botão direito no projeto, depois **Team > Switch to > master.**
2. Atualize o projeto: **Team > Pull.**
3. Mude o branch de volta para o da tarefa.
4. Faça o merge do master com o seu branch: **Team > Merge.** Selecione o **master** e a opção **No commit** em **Merge options** e clique em **Merge.** Isso permitirá que as modificações realizadas sejam visualizadas antes de fazer o commit.



5. Abra a aba Git Staging: **Window > Show View > Git > Git Staging**. Nela é possível visualizar as modificações feitas. Se houver conflitos, será necessário corrigi-los antes de fazer o commit.



6. Selecione os arquivos que deseja realizar o commit e coloque-os em **Staged Changes**. Escreva a mensagem do commit e clique em **Commit and Push**. Na janela que abrirá, apenas avance.
7. Em **Source ref**, selecione o seu branch e clique em **Add Spec**. Avance.

Push to: origin

Push Ref Specifications

Select refs to push.

Add create/update specification

Source ref: Destination ref:

Add delete ref specification

Remote ref to delete:

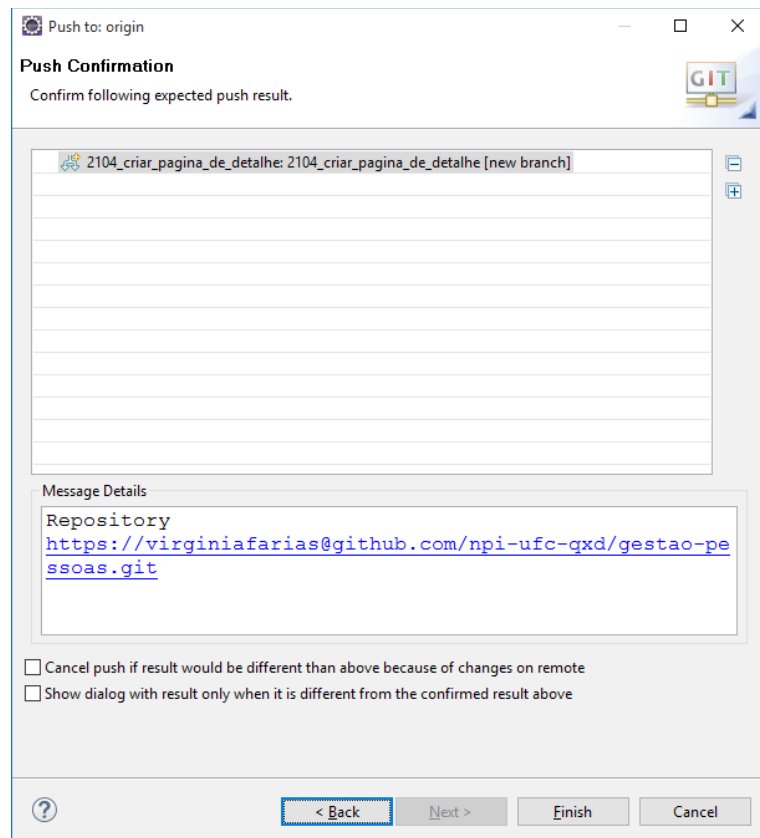
Add predefined specification

Specifications for push

Mode	Source Ref	Destination Ref	Force Update	Remove
Update	refs/heads/2104_criar_...	refs/heads/2104_criar_...	<input type="checkbox"/>	

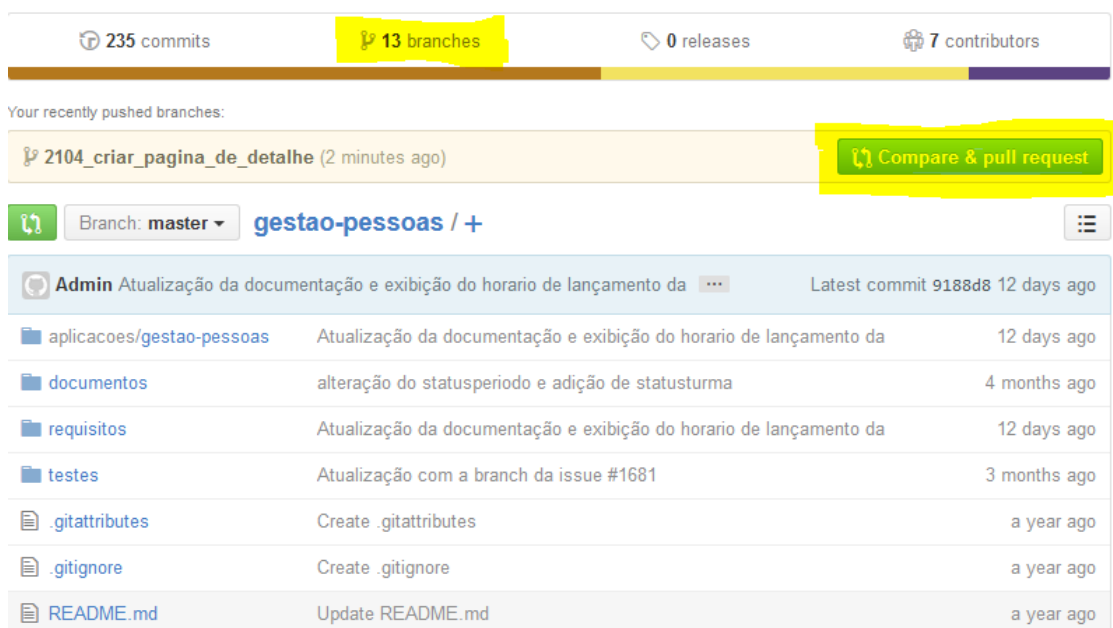
☐ Save specifications in 'origin' configuration

8. Um novo branch será criado remotamente no Github. Finalize.

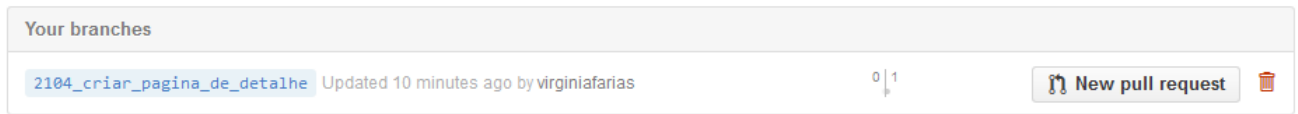


Agora será necessário fazer um **Pull Request** no Github para que o branch criado seja incorporado ao master.

1. Na página do projeto no Github, clique na opção **Branches** ou solicite diretamente o **pull request** se o mesmo estiver disponível.



2. Se escolher a opção **branches** no passo anterior, na página seguinte selecione **New pull request** no branch correspondente.



3. Verifique se o merge poderá ser feito automaticamente pelo Github. Caso não seja, será necessário repetir o processo de merge com o master. Escreva um comentário, se necessário, e clique em **Create pull request**.

Pronto! O processo está finalizado.

Se após a realização do pull request for necessário fazer alterações no branch, basta realizar as modificações no branch local e fazer o commit e push para o branch remoto que as alterações serão refletidas no pull request, não sendo necessário fazer um novo pull request.