



# Git Init Módulo II - Ferramentas e dicas úteis para Git e Github

12/04/2023

---

Alexis Comesana Silvera  
UDESC - Sistemas de Informação  
São Bento do Sul - Santa Catarina

# SUMÁRIO

<b>Módulo II - Ferramentas e dicas úteis para Git e Github</b>	<b>2</b>
Objetivo e Propósito do Módulo	2
Importância da Chave SSH	3
Gerando e Configurando a Chave SSH no GitHub	4
O que é Pull Requests?	6
Criando um Pull Request	7
Como buscar soluções e respostas no Stack Overflow	9
<b>Continue estudando</b>	<b>10</b>
Github Learning Lab	11
Git-it	11
Codecademy	11
Git Exercises	11

## Módulo II - Ferramentas e dicas úteis para Git e Github

### Objetivo e Propósito do Módulo

Nesta parte do curso, vamos explorar como transformar uma pasta comum em um repositório de controle de versão usando o Git, forneceremos uma visão geral do processo de transformação de uma pasta em um repositório Git. Você aprenderá os passos essenciais para iniciar o controle de versão em um projeto existente, permitindo que você acompanhe todas as alterações feitas ao longo do tempo.

Vamos abordar conceitos fundamentais, como inicializar um repositório Git na pasta desejada, adicionar arquivos ao controle de versão, realizar **commits** para registrar as alterações e explorar comandos básicos do Git para navegar e visualizar o histórico de alterações.

Compreender esses conceitos é fundamental para o uso eficiente do Git e a colaboração em projetos de desenvolvimento de software. Vamos explorar cada etapa em detalhes, fornecendo exemplos práticos e orientações passo a passo para garantir uma compreensão clara e uma experiência prática durante o curso.

Lembre-se de que este módulo é projetado para alunos iniciantes em sistemas de informação, com foco na compreensão e aplicação dos conceitos básicos do Git.

## Importância da Chave SSH

A chave SSH (Secure Shell) é um mecanismo de autenticação que permite estabelecer uma conexão segura entre o seu computador e o servidor Git, eliminando a necessidade de inserir suas credenciais (usuário e senha) a cada interação.

Ao utilizar a chave SSH, você terá benefícios significativos, como:

1. Autenticação simplificada: Ao configurar uma chave SSH, você não precisará inserir suas credenciais toda vez que interagir com um repositório Git remoto. Isso economiza tempo e torna o processo de autenticação mais conveniente.
2. Maior segurança: A chave SSH utiliza criptografia de chave pública/privada, o que significa que apenas a sua chave privada, que está armazenada em seu computador, pode descriptografar a conexão. Isso ajuda a proteger suas credenciais de acesso contra possíveis ataques de interceptação.
3. Acesso rápido e eficiente: Com a chave SSH configurada, você pode se conectar aos repositórios Git remotos de forma mais ágil, sem a necessidade de digitar suas credenciais repetidamente. Isso facilita o processo de colaboração e torna o trabalho em equipe mais eficiente.

Nesta seção, iremos guiar você através do processo de criação e configuração de uma chave SSH em seu ambiente de desenvolvimento. Abordaremos os passos necessários, desde a geração da chave até a sua configuração nos serviços de hospedagem do Git.

Ao final desta parte, você terá uma chave SSH configurada e estará pronto para acessar seus repositórios Git de forma rápida e segura.

## Gerando e Configurando a Chave SSH no GitHub

Nesta seção, vamos aprender a gerar e configurar uma chave SSH no GitHub. A chave SSH permitirá que você se autentique de forma segura ao interagir com os repositórios hospedados no GitHub.

Para gerar e configurar sua chave SSH no GitHub, siga os seguintes passos:

1. Verifique se você possui o Git instalado em seu computador. Caso contrário, faça o download e a instalação do Git a partir do site oficial.
2. Abra o Git Bash ou o terminal de sua preferência.
3. No terminal, digite o seguinte comando, substituindo o `email@example.com` pelo seu endereço de e-mail associado à sua conta do GitHub: **`ssh-keygen -t rsa -b 4096 -C "email@example.com"`**  
Esse comando irá gerar uma nova chave **SSH RSA** de **4096** bits.
4. Será solicitado que você escolha o local onde deseja salvar a chave SSH. Pressione Enter para aceitar o local padrão.
5. Em seguida, você será solicitado a inserir uma senha para proteger sua chave SSH. Preencha com uma senha segura ou deixe em branco se preferir não adicionar uma senha extra.

6. Após a geração da chave SSH, você pode verificar o conteúdo da chave pública com o comando: **cat ~/.ssh/id\_rsa.pub**  
Copie todo o conteúdo da chave SSH exibida no terminal.
7. Acesse o GitHub em seu navegador e faça login em sua conta.
8. No canto superior direito da página, clique na sua foto de perfil e selecione "Settings" (Configurações).
9. Na página de configurações, clique em "SSH and GPG keys" (Chaves SSH e GPG).
10. Clique em "New SSH key" (Nova chave SSH).
11. Forneça um título descritivo para a chave SSH no campo "Title" (Título).
12. Cole o conteúdo da chave pública que você copiou anteriormente no campo "Key" (Chave).
13. Clique em "Add SSH key" (Adicionar chave SSH) para salvar a chave SSH no GitHub.

Agora sua chave SSH está configurada e pronta para uso com o GitHub. Nas próximas seções, veremos como utilizar essa chave SSH para se autenticar de forma rápida e segura ao interagir com seus repositórios Git remotos.

## O que é Pull Requests?

O Pull Request (PR) é uma funcionalidade essencial do Git e do GitHub que permite a colaboração entre diferentes desenvolvedores em um projeto. Ele é usado para propor alterações em um repositório e solicitar que essas alterações sejam revisadas e incorporadas ao projeto principal.

Quando você cria um Pull Request, você está essencialmente propondo uma modificação no código do repositório e solicitando que outras pessoas revisem suas alterações e as incorporem ao projeto. Isso facilita o processo de revisão de código e promove a colaboração entre os membros da equipe.

Ao criar um Pull Request, você pode fornecer informações detalhadas sobre as alterações que fez, incluindo uma descrição do problema que está resolvendo, a motivação por trás das alterações e qualquer informação adicional que possa ser útil para os revisores entenderem suas mudanças.

Os revisores, por sua vez, podem fazer comentários, sugerir modificações ou aprovar diretamente o Pull Request se estiverem satisfeitos com as alterações propostas. O processo de revisão pode envolver discussões e iterações até que as alterações sejam consideradas prontas para serem mescladas ao projeto principal.

Durante o processo de criação de um Pull Request, é importante seguir as melhores práticas de desenvolvimento colaborativo, como manter as alterações pequenas e focadas, fornecer documentação adequada e garantir que os testes sejam executados corretamente.

Nos próximos tópicos, exploraremos um exemplo prático de criação de um Pull Request, onde você terá a oportunidade de enviar suas próprias alterações e solicitar a revisão dos demais participantes do curso. Essa atividade ajudará a solidificar seus conhecimentos sobre o uso do GitHub para colaboração em projetos de desenvolvimento de software.

## Criando um Pull Request

Nesta parte do módulo, vamos realizar um exemplo prático de criação de um Pull Request no seu repositório "GIT-INIT" no GitHub. O objetivo é permitir que vários alunos adicionem seus nomes a um arquivo específico chamado "PullRequest.mb".

O arquivo "PullRequest.mb" já está presente no seu repositório e serve como um espaço para os participantes do curso adicionarem seus nomes. Para participar dessa atividade, siga as etapas abaixo:

1. Acesse o repositório "**GIT-INIT**" no GitHub através do link:  
**<https://github.com/AI3x1sCS/GIT-INIT>**
2. Certifique-se de estar autenticado com sua conta do GitHub. Se necessário, faça login.
3. Navegue até o arquivo "**PullRequest.mb**" no repositório. Você pode encontrá-lo clicando na pasta correspondente.
4. Abra o arquivo "**PullRequest.mb**" clicando nele.



5. No canto superior direito do arquivo, clique no botão "**Edit**" (Editar) para modificar o conteúdo.
6. Adicione seu nome à lista existente no arquivo. Certifique-se de seguir as instruções fornecidas, como o formato de adição do nome.
7. No final da página, adicione uma descrição breve das alterações que você fez. Isso ajudará os revisores a entenderem suas modificações.
8. Depois de adicionar seu nome e fornecer uma descrição, role até a seção "**Propose changes**" (Propor alterações).
9. Selecione a opção "**Create a new branch for this commit and start a pull request**" (Criar um novo **branch** para este **commit** e iniciar um **pull request**).
10. Clique no botão "**Propose changes**" (Propor alterações) para criar um novo **Pull Request**.

Após seguir essas etapas, seu **Pull Request** será criado e os revisores poderão visualizar suas alterações e fornecer feedback. Lembre-se de acompanhar as discussões e fazer as alterações necessárias conforme solicitado pelos revisores.

Essa atividade de criação de **Pull Request** com um exemplo prático permite que você pratique o processo de colaboração e revisão de código no GitHub. É uma ótima oportunidade para aprender e se familiarizar com as práticas de desenvolvimento em equipe.

## Como buscar soluções e respostas no Stack Overflow

Nesta parte do módulo, vamos explorar como utilizar o Stack Overflow para resolver problemas e obter respostas relacionadas ao Git. O Stack Overflow é uma plataforma online muito popular onde desenvolvedores de todo o mundo compartilham conhecimento e ajudam uns aos outros com suas dúvidas e problemas.

Para ilustrar esse processo, vamos criar um problema comum que pode ser facilmente pesquisado no Stack Overflow e resolveremos utilizando as informações disponíveis na plataforma. Vamos supor o seguinte problema:

Você está trabalhando em um repositório Git local e deseja desfazer as últimas alterações feitas em um arquivo específico. No entanto, você não tem certeza de qual comando usar ou como executá-lo corretamente.

Aqui estão os passos para resolver esse problema utilizando o Stack Overflow:

1. Acesse o site do Stack Overflow em <https://stackoverflow.com>.
2. Na barra de pesquisa, digite uma descrição clara e concisa do seu problema. Por exemplo, você pode pesquisar por "desfazer últimas alterações em arquivo específico no Git".
3. Pressione Enter ou clique no botão de pesquisa para visualizar os resultados relacionados à sua pesquisa.
4. Navegue pelos resultados e leia as perguntas e respostas que foram fornecidas por outros usuários. Procure por respostas que se

encaixem no seu caso e que tenham sido marcadas como corretas ou bem avaliadas pela comunidade.

5. Ao encontrar uma resposta relevante, leia-a atentamente e verifique se contém as informações necessárias para resolver o seu problema.
6. Caso a resposta tenha um código ou comando específico, copie-o e cole-o no seu terminal Git para executá-lo.
7. Teste a solução proposta e verifique se o problema foi resolvido.

Se a solução funcionar para o seu caso, considere marcar a resposta como útil no Stack Overflow para ajudar outros desenvolvedores que possam encontrar o mesmo problema.

Utilizar o Stack Overflow como uma fonte de informações e soluções é uma prática comum no desenvolvimento de software. A comunidade do Stack Overflow é rica em conhecimento e pode fornecer respostas valiosas para problemas específicos. Lembre-se de sempre ler e entender as respostas antes de aplicá-las ao seu próprio código ou projeto.

## Continue estudando

É importante lembrar que este curso é apenas uma introdução ao Git e ao GitHub. O mundo do Git é vasto e existem muitos recursos adicionais disponíveis para você continuar aprendendo e aprimorando suas habilidades. Por isso, encorajamos você a explorar mais além deste curso inicial e se aprofundar nas possibilidades do Git.

Uma ótima maneira de continuar praticando e fortalecendo suas habilidades em Git é utilizar sites dedicados a treinamentos e exercícios interativos. Aqui estão alguns sites recomendados para treinar Git:

## Github Learning Lab

O GitHub Learning Lab oferece uma variedade de cursos interativos para aprender Git e GitHub na prática. Completar esses cursos permitirá que você aprofunde seus conhecimentos e pratique suas habilidades no próprio ambiente do GitHub.

<https://github.com/topics/github-learning-lab>

## Git-it

O Git-it é um tutorial interativo baseado em linha de comando que ensina os principais conceitos e comandos do Git. Ele guiará você através de exercícios práticos que você pode realizar diretamente no seu próprio computador.

<https://github.com/jlord/git-it-electron>

## Codecademy

A plataforma Codecademy oferece um curso online gratuito sobre Git, que abrange desde os conceitos básicos até tópicos avançados. Os exercícios práticos e exemplos reais fornecidos neste curso irão ajudá-lo a aprimorar suas habilidades em Git.

<https://www.codecademy.com/learn/learn-git>

## Git Exercises

O Git Exercises é uma plataforma online que oferece uma variedade de exercícios práticos para fortalecer suas habilidades em Git. Os exercícios são projetados para cobrir diferentes cenários de uso do Git e variam em

dificuldade.

<https://gitexercises.fracz.com/>

Lembre-se de que o próprio site do GitHub também oferece uma série de recursos de treinamento, como guias, documentação detalhada e tutoriais. Explore a seção "Learning Lab" do GitHub para acessar cursos adicionais e aprimorar ainda mais suas habilidades.

Não pare por aqui! O Git tem muito mais a oferecer e o aprendizado contínuo é essencial para se tornar um profissional experiente em controle de versão. Continue explorando, praticando e buscando novos desafios para se tornar um mestre do Git.

Continue empenhado em aprender e aproveite ao máximo os recursos disponíveis para você. O Git é uma ferramenta poderosa e versátil, e dominar suas funcionalidades trará benefícios significativos em sua jornada no desenvolvimento de software.

Boa sorte e bons estudos!