

Documentação para Trabalhar com *Git* e *GitHub*

Introdução

Git é um sistema de controle de versão distribuído amplamente utilizado para rastrear alterações em projetos de software e facilitar a colaboração entre desenvolvedores. O *GitHub*, por outro lado, é uma plataforma de hospedagem de código que utiliza o *Git* como seu sistema de controle de versão subjacente. Nesta documentação, abordaremos os conceitos fundamentais do *Git* e como usá-lo em conjunto com o *GitHub* para gerenciar seus projetos de forma eficiente.

Conteúdo

1. Instalação do *Git*
2. Configuração Inicial
3. Criar e Clonar um Repositório
4. Comandos Básicos do *Git*
 - 4.1. *git init*
 - 4.2. *git status*
 - 4.3. *git add*
 - 4.4. *git commit*
 - 4.5. *git log*
5. Trabalhando com Ramificações (*Branches*)
 - 5.1. Criar e Alternar para uma Nova *Branch*
 - 5.2. Mesclar Alterações entre *Branches*
6. Trabalhando com o *GitHub*
 - 6.1. Criar um Repositório no *GitHub*
 - 6.2. Conectar o Repositório Local ao Repositório Remoto
 - 6.3. Enviar Alterações para o Repositório Remoto
 - 6.4. Clonar um Repositório Remoto
 - 6.5. Solicitar *Pull Requests*

7. Visualizando o Histórico de *Commits*

8. Resolvendo Conflitos

1. Instalação do *Git*

Para começar a trabalhar com o *Git*, você precisa instalá-lo no seu sistema operacional. Faça o download da versão adequada para o seu sistema a partir do site oficial do *Git* (<https://git-scm.com/downloads>) e siga as instruções de instalação.

2. Configuração Inicial

Após a instalação, configure seu nome de usuário e endereço de e-mail, que serão usados nas informações de identificação dos *commits* que você criar.

```
git config --global user.name "Seu Nome"
```

```
git config --global user.email "seuemail@example.com"
```

3. Criar e Clonar um Repositório

Para criar um novo repositório *Git* em um diretório existente, utilize o seguinte comando:

```
git init
```

Para clonar um repositório remoto do *GitHub* para o seu computador local, use o comando:

```
git clone URL_do_repositório
```

Substitua "URL_do_repositório" pela URL do repositório que você deseja clonar. Você pode encontrar essa URL no próprio repositório no *GitHub*.

4. Comandos Básicos do *Git*

4.1. *git init*

Inicializa um novo repositório *Git* no diretório atual.

4.2. *git status*

Exibe o estado atual do seu repositório, mostrando quais arquivos foram modificados, adicionados ou removidos.

4.3. *git add*

Adiciona as mudanças no seu diretório de trabalho ao índice (*staging area*) para serem incluídas no próximo *commit*.

```
git add nome_do_arquivo    # Adiciona um arquivo específico
```

```
git add .                  # Adiciona todas as alterações realizadas
```

4.4. *git commit*

Cria um novo *commit* contendo todas as alterações adicionadas anteriormente ao índice.

```
git commit -m "Mensagem do commit"
```

Substitua "Mensagem do *commit*" por uma breve descrição do que foi alterado neste *commit*.

4.5. *git log*

Exibe um histórico de *commits* realizados no repositório, mostrando informações como autor, data e mensagem do *commit*.

```
git log
```

5. Trabalhando com Ramificações (*Branches*)

5.1. Criar e Alternar para uma *Nova Branch*

Para criar uma nova *branch* (ramificação) e alternar para ela, utilize os seguintes comandos:

```
git checkout -b nome_da_branch
```

```
ou git switch -c nome_da_branch # A partir do Git 2.23
```

5.2. Mesclar Alterações entre *Branches*

Para mesclar as alterações de uma *branch* em outra, certifique-se de estar na *branch* de destino e use o comando:

```
git merge nome_da_outra_branch
```

6. Trabalhando com o *GitHub*

6.1. Criar um Repositório no *GitHub*

Acesse o *GitHub* (<https://github.com/>), faça login na sua conta (ou crie uma nova) e clique no botão "New" para criar um novo repositório. Siga as instruções para configurar o repositório.

6.2. Conectar o Repositório Local ao Repositório Remoto

Se você criou um repositório remoto no *GitHub*, conecte-o ao repositório local usando o comando:

```
git remote add origin URL_do_repositório_remoto
```

Substitua "URL_do_repositório_remoto" pela URL fornecida pelo *GitHub* para o seu repositório.

6.3. Enviar Alterações para o Repositório Remoto

Para enviar suas alterações para o repositório remoto, use o comando:

```
git push origin branch
```

Substitua "*branch*" pelo nome da *branch* que você deseja enviar para o repositório remoto (por exemplo, "*master*" ou "*main*").

6.4. Clonar um Repositório Remoto

Clonar um repositório remoto é um processo comum na gestão de projetos de desenvolvimento de software usando sistemas de controle de versão, como *Git*. Clonar um repositório permite que você faça uma cópia completa do projeto em seu ambiente local, onde pode trabalhar nele, fazer alterações e enviar essas mudanças de volta ao repositório remoto.

Encontre o URL do repositório remoto: No serviço de hospedagem do seu repositório (como *GitHub*, *GitLab*, *Bitbucket* etc.), vá até o repositório que deseja clonar e copie o URL do repositório. Normalmente, o URL estará na forma "https://github.com/usuario/nome-do-repositorio.git" para repositórios hospedados no *GitHub*, por exemplo.

Abra o terminal (ou prompt de comando): Dependendo do sistema operacional que você está usando, abra o terminal no *macOS* e *Linux* ou o prompt de comando no *Windows*. Navegue até o diretório em que deseja clonar o repositório: Use o comando *cd* (*Change Directory*) para navegar até o diretório no qual você deseja que a cópia do repositório seja armazenada. Por exemplo, para criar a cópia do repositório em sua pasta de documentos, você pode usar:

```
cd ~/Documents
```

Clone o repositório remoto: Use o comando *git clone* seguido do URL do repositório remoto para iniciar o processo de clonagem. Por exemplo:

`git clone https://github.com/usuario/nome-do-repositorio.git`

O *Git* começará a baixar todos os arquivos e histórico de *commits* do repositório remoto para o diretório que você especificou. Autenticação (se necessário):

Dependendo das configurações do repositório remoto, você pode ser solicitado a fornecer suas credenciais (nome de usuário e senha ou chave SSH) para autenticação antes que o processo de clonagem seja concluído.

Após a conclusão do processo de clonagem, você terá uma cópia completa do repositório remoto em seu ambiente local, pronto para que você possa começar a trabalhar nele. Use o comando `git pull` para atualizar sua cópia local com as alterações mais recentes do repositório remoto e use o comando `git push` para enviar suas alterações de volta ao repositório remoto.

6.5. Solicitar *Pull Requests*

Solicitar *pull requests* é um processo pelo qual você pede a outros colaboradores do repositório para revisar e incorporar as alterações que você fez em um ramo (*branch*) específico em relação ao ramo principal (normalmente o ramo "*master*" ou "*main*") do repositório. Essa é uma prática comum quando se trabalha com sistemas de controle de versão, como o *Git*, em projetos colaborativos. Abaixo estão os passos para solicitar um *pull request*:

Certifique-se de que seu código está em um ramo separado: Antes de solicitar um *pull request*, é importante garantir que seu código esteja em um ramo separado do ramo principal do repositório. Isso ajuda a manter o ramo principal limpo e facilita a revisão das alterações.

Faça *commit* das suas alterações:

Verifique se todas as suas alterações estão corretamente "comitadas" (registradas) em seu ramo local. Use o comando `git commit` para adicionar suas alterações ao histórico de *commits*. Atualize seu ramo com o ramo principal (opcional):

É uma boa prática manter seu ramo atualizado com o ramo principal antes de solicitar um *pull request*. Isso pode ser feito usando o comando `git pull` em seu ramo local.

Envie o ramo para o repositório remoto: Use o comando `git push` para enviar seu ramo com as alterações para o repositório remoto. Por exemplo:

`git push origin nome-do-ramo`

Certifique-se de substituir "nome-do-ramo" pelo nome do seu ramo. Acesse o repositório no serviço de hospedagem:

Acesse o repositório no serviço de hospedagem do *Git*, como *GitHub*, *GitLab* ou *Bitbucket*.

- Crie o *pull request*:

No repositório remoto, procure a opção para criar um *pull request*. Essa opção geralmente está disponível na interface do usuário do serviço de hospedagem.

- Selecione o ramo base e o ramo de comparação:

Em seguida, escolha o ramo base (normalmente o ramo principal) e o ramo de comparação (seu ramo com as alterações).

- Adicione uma descrição:

Descreva as alterações que você fez e inclua detalhes relevantes sobre o que está sendo adicionado ou corrigido.

- Crie o *pull request*:

Finalmente, clique no botão ou link para criar o *pull request*. Isso enviará sua solicitação de *pull request* para o repositório.

- Aguarde a revisão:

Os colaboradores do projeto revisarão suas alterações e poderão fazer comentários ou solicitar modificações antes de aceitar o *pull request*. Lembre-se de que o processo exato pode variar ligeiramente dependendo da plataforma de hospedagem e das permissões do repositório. Por exemplo, em alguns projetos, apenas colaboradores com permissões específicas podem criar *pull requests*. Em outros projetos de código aberto, qualquer pessoa pode contribuir com *pull requests*.

Portanto, certifique-se de estar familiarizado com as regras e diretrizes específicas do projeto antes de enviar seu *pull request*.

7. Visualizando o Histórico de Commits:

Veja o histórico de commits em ordem cronológica:

`git log`

8. Resolvendo Conflitos:

Em caso de conflitos durante a mesclagem, edite os arquivos com conflito, resolva as diferenças e faça um novo commit.

Este guia rápido abrange os conceitos essenciais para trabalhar com Git e GitHub. À medida que você se familiariza com essas operações, poderá explorar outras funcionalidades avançadas para otimizar o gerenciamento do seu projeto de software. Consultar a documentação oficial do Git e o guia de ajuda do GitHub também é recomendado para um entendimento mais aprofundado.