Git FAQ Documentation

Release dev

Eduardo Klosowski

set. 13, 2021

Sumário

Este texto contem problemas e perguntas sobre git, apresentando algumas dicas de como os mesmos poderiam ser resolvidos ou sugestões de como tratá-los.

Este não é um texto finalizado, recebendo atualizações com novos tópicos conforme as ideias forem surgindo e os tópicos escritos. Caso tenha alguma dica ou sugestão, abra uma issue no repositório do GitHub, mande seu texto como pull request, ou entre em contato pelo grupo Git Brasil do Telegram.

CAPÍTULO 1

Arquivos Versionados

Dicas para os arquivos que devem ser versionados ou ignorados pelo git.

1.1 Ignorar Arquivos

1.1.1 Problema

Podem existir alguns arquivos que ficam dentro do diretório do repositório git, como:

- Arquivos de configuração, que podem conter dados sensíveis como senhas.
- Arquivos compilados, que podem ser gerados automaticamente ao recompilar o projeto.
- Bibliotecas externas, que podem ser baixadas novamente, além de provavelmente serem versionadas externamente.

Desta forma esses arquivos não devem ser compartilhados no repositório git, tanto por questões de segurança (evitando publicar as senhas), otimização do repositório (evitar o aumento do seu tamanho) e redução de conflitos (a compilação tende a gerar arquivos diferentes cada vez que executada, como a inclusão da data de compilação).

1.1.2 Solução

Esses arquivos devem ser ignorados pelo git, de forma que eles não sejam versionados, mesmo que existam dentro do diretório do repositório git.

Esta configuração pode ocorrer em três lugares:

- Arquivo `.gitignore` dentro do repositório, que aplica as regras para o diretório atual e a todos a baixo dele.
- Arquivo `.git/info/exclude`, que aplica as regras para todo o repositório.
- Arquivo `~/.config/git/ignore`, que aplica as regras para todos os repositórios acessados pelo usuário.

O melhor lugar depende de cada caso. A opção mais adotada é um arquivo `.gitignore` na raiz do projeto, que é versionado pelo git, desta forma essas regras serão compartilhadas por todos que usam o repositório, além de ficarem centralizadas. Em casos onde existam muitas regras, regras mais específicas para diretórios, elas podem ser divididas em outros `.gitignore`, podendo existir mais de um `.gitignore` em diferentes diretórios no mesmo repositório.

Porém em casos onde um programador usa um IDE específica que cria arquivos de configuração dentro do diretório do projeto, e se essas regras não forem aceitas por quem controla o repositório, elas podem ser feitas no arquivo `.git/info/exclude` para o projeto específico, ou mesmo em `~/.config/git/ignore` para ser aplicada a todos os repositórios. Mas essas regras não serão compartilhadas com as demais cópias deste repositório, devendo ser refeitas.

1.1.3 Troubleshoot

Caso as regras não estejam sendo aplicadas, como um arquivo continuar aparecendo no comando `git status`, verifique:

- Se o nome do arquivo de configuração está correto.
- Se as regras estão usando a barra normal (`/`) e não barras invertidas (`\`).
- Vá no diretório onde encontra-se o `.gitignore` ou na raiz do repositório e tente listar o arquivo com o comando `ls`, por exemplo, após isso verifique se a regra está de acordo.
- Verifique se o arquivo já não estava sendo versionado antes da criação da regra.

1.1.4 Links Externos

- Manpage: gitignore

1.2 Versionar Diretório

1.2.1 Problema

O projeto necessita de um diretório, por exemplo, para guardar arquivos que foram enviados pelos clientes ou armazenar logs, sendo que esses arquivos devem ser ignorados pelo git (não devem ser versionados), por não fazerem parte do código. Porém ao criar uma regra no `.gitignore` e clonar o repositório em outro local ou computador, este diretório não é criado.

1.2.2 Solução

Uma opção simples seria criar a regra no `.gitignore` e fazer a aplicação tentar criar o diretório caso ele não exista, o que já funcionaria. Porém seria necessário executar a aplicação para que ela crie os diretórios necessários, podendo gerar uma confusão já que esses diretórios não existiriam para alguém que verificou o repositório previamente, como se eles aparecessem magicamente.

Infelizmente o git não versiona diretórios isoladamente, todo diretório deve conter pelo menos um arquivo dentro dele, ou de um subdiretório seu, para ser versionado pelo git e aparecer no repositório. Por este motivo muitos criam um arquivo chamado `.gitkeep` dentro desses diretórios, porém poderia ser qualquer outro nome, esse é apenas o mais usado nos exemplos (existem projetos que nomeiam esses arquivos como `delete.me` ou semelhante). Esse arquivo pode estar em branco ou ter algum conteúdo, não faz diferença para essa abordagem.

Porém para a correta configuração do repositório, este arquivo deve ser versionado, enquanto os demais arquivos do diretório não, isso pode ser feito com regras no `.gitignore`, como no exemplo a baixo, onde apenas o arquivo `.gitkeep` será versionado dentro do diretório `logs`:

```
logs/*
!logs/.gitkeep
```

Outra opção seria criar um arquivo `.gitignore` dentro do diretório:

```
*
!.gitignore
```

1.2.3 Troubleshoot

Caso o diretório não apareça como alterado ao executar um `git status`, primeiramente verifique se não existe nenhuma regra que ignora o diretório onde o arquivo `.gitkeep` está, caso exista, altere a regra para ignorar apenas o que está a baixo dele, adicionado um `/*` no final.

1.2.4 Links Externos

- Manpage: gitignore

CAPÍTULO 2

Extra

Conteúdo extra.

2.1 Materiais

2.1.1 Livros / Documentação

- Pro Git
- Getting Git Right
- Git Notes for Professionals book

2.1.2 Tutoriais

- git - guia prático - sem complicação!
- Try Git
- Katacoda - Git
- Git How To: Tutorial Guiado de Git
- GIT PURR! Git Commands Explained with Cats!

2.1.3 Textos

- Boas práticas para usar o git
- Commit messages guide
- Conventional Commits

2.1.4 Vídeos

- Introdução ao Git
- Git para quem gosta de Git
- Entendendo GIT | (Não é um tutorial!)
- Git na Prática
- Dominando o Git

2.1.5 Ferramentas

- Git Explorer
- Learn Git Branching
- Visualizing Git
- gitvisual

2.1.6 Slides

- Controle de versão com git
- O que é o Git?

2.1.7 Cursos

- Git e Github para iniciantes
- Git e Github

2.1.8 Assuntos Relacionados

Gitflow

- Gitflow Workflow
- cheatsheet do git-flow

CAPÍTULO 3

Licença