

Lista de Comandos Git para Desenvolvedores Júnior

Fluxo de Trabalho de Desenvolvimento de Software

Comando	Descrição	Exemplo	Observações
Clonagem de Repositórios			
git clone	Clona um repositório remoto para o seu computador local.	git clone https://github.com/user/repo.git	Cria um novo diretório repo com o conteúdo do repositório remoto.
Modificação e Adição de Arquivos			
git add	Adiciona arquivos modificados à área de staging para serem comitados.	git add README.md src/index.js	Prepara os arquivos para serem incluídos no próximo commit.
git commit	Cria um snapshot permanente das alterações no repositório.	git commit -m "Adicionando nova funcionalidade"	Salva as alterações com uma mensagem descritiva.
git diff	Mostra as diferenças entre o	git diff	Exibe as modificações

Comando	Descrição	Exemplo	Observações
Gerenciamento de Branches			
	estado atual do seu workspace e o último commit.	README.md	feitas no arquivo README.md.
git branch	Lista as branches existentes no repositório.	git branch	Mostra a branch atual e as branches remotas.
git checkout	Troca para a branch especificada.	git checkout feature/nova-funcionalidade	Alterna para a branch feature/nova-funcionalidade.
git merge	Integra as alterações de outra branch na branch atual.	git merge master	Combina as alterações da branch master na branch atual.
Resolução de Conflitos			
git status	Mostra o estado do seu workspace e as alterações não comitadas.	git status	Informa se há arquivos com conflitos de merge.

Comando	Descrição	Exemplo	Observações
git mergetool	Inicia uma ferramenta de merge para resolver conflitos de merge.	git mergetool	Abre a ferramenta de merge para resolver os conflitos.
Operações de Merge e Push			
git push	Envia as alterações locais para o repositório remoto.	git push origin feature/nova-funcionalidade	Envia a branch feature/nova-funcionalidade para o repositório remoto origin.
Comandos Avançados			
git reset	Desfaz alterações no seu workspace ou no histórico de commits.	git reset HEAD~1	Desfaz o último commit.
git rebase	Reorganiza o histórico de commits da sua branch.	git rebase master feature/nova-funcionalidade	Reorganiza a branch feature/nova-funcionalidade para ter o mesmo histórico da branch master.
git tag	Cria uma marcação	git tag v1.0.0	Cria uma tag para a versão 1.0.0 do

Comando	Descrição	Exemplo	Observações
	permanente no histórico de commits.		projeto.
git log	Mostra o histórico de commits do repositório.	git log --all --oneline	Exibe todos os commits em uma única linha.

Observações:

- Esta tabela é apenas um resumo dos comandos git mais usados. Para informações mais detalhadas, consulte a documentação oficial do Git: <https://git-scm.com/doc>.
- A ordem dos comandos segue um fluxo de trabalho típico de desenvolvimento de software, mas você pode adaptá-la às suas necessidades.
- É importante usar o comando git help para obter mais informações sobre cada comando antes de usá-lo.

Dicas para Desenvolvedores Júnior:

- Comece com os comandos básicos e aprenda os mais avançados gradualmente.
- Use o Git com frequência para manter seu código organizado e versionado.
- Faça backups regulares do seu repositório para evitar perdas de dados.
- Aprenda a resolver conflitos de merge o mais rápido possível.
- Use branches para trabalhar em diferentes funcionalidades ao mesmo tempo.

Recursos Adicionais:

- <https://www.w3schools.com/git/>
- <https://education.github.com/git-cheat-sheet-education.pdf>
- <https://git-scm.com/book>

Lembre-se:

- O Git é uma ferramenta poderosa que pode ser usada para melhorar o seu fluxo de trabalho de desenvolvimento de software.
- Investir tempo para aprender o Git pode te tornar um desenvolvedor mais eficiente e produtivo.