

Git - Primeiros passos

1º) Crie uma pasta para o seu projeto

2º) Entre na pasta e inicie o Git:

```
git init
```

3º) Diga quem é você:

```
git config --local user.name "Seu nome aqui"
```

```
git config --local user.email "seu@email.aqui"
```

4º) Para ver se deu certo:

```
git config --list --local
```

Principais comandos

- **git init:** inicializa um repositório no diretório em que o comando for executado. A partir deste comando, o Git poderá gerenciar as modificações realizadas nos arquivos;
- **git status:** analisa o status do nosso repositório;
- **git add:** adiciona as alterações para serem commitadas futuramente.
- **git commit:** Envia as alterações. Parâmetro -m para adicionar uma mensagem.
- **git log -p:** Mostra as alterações detalhadamente. Existem vários outros parâmetros que podem ser adicionados: <https://devhints.io/git-log>
- **git merge:** junta os trabalhos e gera um merge commit.
- **git rebase:** aplica os commits de outra branch na branch atual.
- **git checkout "nome da branch":** altera entre branches;
- **git checkout -- "nome do arquivo":** descarta as alterações realizadas
- **git reset HEAD "nome do arquivo":** desmarca o arquivo que seria commitado
- **git revert "hash":** desfaz um commit a partir do número do hash(git log)
- **git stash:** salva alterações temporariamente. Quando precisamos pausar o desenvolvimento de alguma funcionalidade, ou correção, antes de finalizar, talvez não seja interessante realizar um commit, pois o nosso código pode não estar funcionando ainda. Nesse caso é interessante salvar o trabalho para podermos voltar a ele depois.
- **git stash list:** lista as alterações salvas temporariamente
- **git stash pop:** faz um merge com as alterações temporárias e remove elas da lista.
- **git diff:** ver diferenças entre commits.
- **git tag -a v0.1.0 -m "First version":** adiciona uma tag(um marco) no projeto atual; Cria uma release;

Ignorando arquivos

.gitignore: todas as linhas que estiverem nele serão lidos e ignorados pelo Git. Se temos um arquivo denominado ide-config que queremos que seja ignorado, por exemplo, basta o incluirmos em *.gitignore*, digitando ide-config simplesmente. Da mesma forma, se tivéssemos uma pasta ide, incluiríamos ide/, em uma nova linha.

Conceitos Importantes

- **Branches ("ramos")** são utilizados para desenvolver funcionalidades isoladas umas das outras. A branch master é a branch "padrão" quando você cria um repositório.
- **Qual a diferença entre os comandos rebase e merge?** O merge junta os trabalhos e gera um merge commit. O rebase aplica os commits de outra branch na branch atual.