# GIT

## Comandos:

Criar conta: ssh-keygen -t rsa -C [seu\_email@provedor.com](mailto:seu_email@provedor.com)

Login do usuário: git config --global user.name "YOUR\_USERNAME"

Login do Email: git config --global user.email [your\_email\_address@example.com](mailto:your_email_address@example.com)"

Verificar Configurações: git config --global –list

Baixar projeto: git clone <https://suaurldorepositorio.git>

Verificar as tags do projeto: git tag

Pegar versão especifica: git checkout v0.1

Verificar as diferenças entre versões: git diff v0.1 v0.2

Verificar quem fez as alterações no arquivo: git blame index.html

Verificar o que foi mudado: git status

Colocar uma versão especifica em uma branch: **git reset --hard v1.0.7**

Adicionar no repositório: git add “nomeArquivo”

Adicionar todos arquivos no repositório: git add .

Configurar o repositório remoto: git remote add origin https://github.com/[seu\_nome\_de\_usuario]/curso-git.git

Enviar o commit para o repositório (no lugar do master podemos colocar o branch) : git push origin master

Verificar tudo que foi alterado e as linhas: git whatchanged –p

Verificar todas atualizações do repositório: git fetch origin

Baixar do repositório: git pull --no-commit <remote>

Descartar Commit: git checkout .

## Branchs:

Criar branch local: git branch nomeBranch

Criar branch local e remota: git branch –u nomeBranch

Verificar todas as branchs: git branch

Criar Branch igual origin : git branch develop origin/develop

Escolher branch: git checkout nomeBranch

Compartilhar branch local com repositório: git push origin nomeBranch

Deletar branch: git branch nomeBranch –d

Atualizar uma branch baseado em outra branch (primeiro a branch de base e segundo a branch que vai ser atualizada): git rebase master desenvolvimento

Atualizar Branch baseado em outra branch: git pull origin nomeBranch

Fazer merge com outra branch

git merge --no-ff origin/CAD-514 – alterar o cad-514 pelo nome da branch