Versão 1

**============================================================**

**Índice**

============================================================

# [Gerenciar usuário](#Gerenciar_usuario)

# [Criar Repositório Git](#Criar_Repositorio_Git)

# [Gerenciar os Commits](#Gerenciar_os_Commits)

# [Ver detalhes dos Commits](#Ver_detalhes_dos_Commits)

# [Mover entre os Commits](#Mover_entre_os_Commits)

# [Desfazer Mudanças nos Arquivos](#Desfazer_Mudancas_nos_Arquivos)

# [Ignorando Mudanças nos Arquivos pelo Git](#Ignorando_Mudancas_Arquivos_Git)

# [Clonando Repositórios Locais](#Clonando_Repositorios_Locais)

**============================================================**

**============================================================**

## Gerenciar usuário

============================================================

**// Criar um usuário**

$ git config --global user.name "rafaelFreitas"

**// configurar o email do usuário**

$ git config --global user.email "rafaelfreitas.lima29@gmail.com"

**// para confirmar se foi criado**

$ git config --global user.name

*rafaelFreitas*

**// Lista todas as caracteristicas do usuário**

$ git config --global --list

*user.name=rafaelFreitas*

*user.email=rafaelfreitas.lima29@gmail.com*

# [Índice](#indice)

**============================================================**

###############################################################

**============================================================**

Criar Repositório Git ============================================================

**// Mostra os arquivos que estão na pasta**

$ dir -a

*. .buildpath .project composer.json index.php views*

*.. .htaccess .settings composer.lock vendor*

**// Para Criar um repositório Git**

rafael@pc-rafael MINGW64 /c/xampp/htdocs

$ git init

*Initialized empty Git repository in C:/xampp/htdocs/.git/*

# [Índice](#indice)

**============================================================**

###############################################################

**============================================================**

Gerenciar os Commits ============================================================

**// Para ver o status dos arquivos, se foram modificados ou se estão sendo rastreados**

$ git status

**// Adiciona o rastreio a todos os arquivos do diretório**

$ git add --all

**// Adiciona o rastreio a um arquivo**

$ git add nomeArquivo.extensão

**// Para commitar as alterações**

$ git commit -m "comentário da versão"

# [Índice](#indice)

**============================================================**

###############################################################

**============================================================**

Ver detalhes dos Commits ============================================================

**// Para ver o log de alterações**

$ git log

**// Para ver o que foi mudado entre commits**

// tem um espaço entre os códigos

$ git diff código\_do\_commit código\_do\_commit\_anterior

**// Para ver com mais detralhes o que foi modificado entre os commits**

$ git log --stat

// Para sai do git log é só apertar no "q"

# [Índice](#indice)

**============================================================**

###############################################################

**============================================================**

Mover entre os Commits ============================================================

MOVER-SE ENTRE OS COMMITS

**// mudar para um commit**

// $ git checkout 2b42496e00be55c37ce895c67a9e2f1324d73bc8

$ git checkout código\_do\_commit

// mudar para o commit master

$ git checkout master

# [Índice](#indice)

**============================================================**

###############################################################

**============================================================**

Desfazer Mudanças nos Arquivos ============================================================

**// Desfazer mudança em um arquivo (não apaga arquivos não rastreados pelo git)**

$ git checkout arquivo.extensão

// **Desfazer mudanças em vários arquivos** **(não apaga arquivos não rastreados pelo git)**

$ git reset --hard

// **Desfazer mudanças em vários arquivos (apaga arquivos não rastreados pelo git)**

// Mostra a lista de arquivos não rastreados e que vão ser apagados

$ git clean -n

// Apaga os arquivos listados

$ git clean -f

// **Desfazer mudanças em vários arquivos selecionados (apaga arquivos não rastreados pelo git)**

// Mostra a lista de arquivos não rastreados e que vão ser apagados

$ git clean -i

// Escolha o "s"

s

// Escolha o número dos arquivos que serão apagados pelo número com e aperte o "enter", após escolher aperte "enter".

// Para executar aperte "c" e após o "enter"

c

# [Índice](#indice)

**============================================================**

###############################################################

**============================================================**

Ignorando Mudanças nos Arquivos pelo Git ============================================================

**// Deve-se criar um arquivo dentro da pasta do projeto com o nome em banco com a extensão ".gitignore" para o linux ou ".gitignore." para o windows, pois ele não aceita nomes de arquivos em branco.**

// **Para ignorar um arquivo ou vário só e necessário colocar o nome do arquivo em um linha dentro do arquivo ".gitignore".**

// **Para ignorar um tipo de arquivo só é necessário colocar o "\*.extesão" , como exemplo " \*.html ", que ignorará todos os arquivos html que não foram rastreados pelo git.**

**// Tirar todos os arquivos do rastreamento do Git,** Pois não se pode ignorar arquivos já rastreados.

$ git rm -r --cached .

# [Índice](#indice)

**============================================================**

###############################################################

**============================================================**

Clonando Repositórios Locais ============================================================

**// Para clonar repositórios locais so é necessário utilizar o comando git clone caminho\_do\_repositorio\_a\_ser\_clonado, você deve estar na pasta onde o clone ficará.**

$ git clone C:/pasta/pasta/repositorio

ou

$ git clone C:/repositorio

// **OBS.: os arquivos ignorados pelo gitignore não serão clonados.**

# [Índice](#indice)

**============================================================**

###############################################################