

# Configurando o GIT.

O GIT guarda suas informações em três lugares.

GIT Config do sistema (System).

```
git config --system "Use system config file"
```

GIT Config do usuário (Global).

```
"use global config file"
```

GIT Config do repositório/projeto (Local).

```
git config --local "use repository config file"
```

## 1 - Configurações do usuário (Global).

1.1 - Configurar o User Name.

```
git config --global user.name "Leonardo Almeida"
```

1.2 - Configurar o E-mail.

```
git config --global user.email "almeidass.leonardo@gmail.com"
```

1.3 - Configurar editor de texto padrão.

```
git config --global core.editor sublime
```

1.4 - Lista configurações.

Listar User Name.

```
git config user.name
```

Lista E-mail

```
git config user.email
```

Listar todas as configurações.

```
git config --list
```

Listar configurações do sistema (System).

```
git config --system --list
```

Lista configurações do usuário (Global).

```
git config --global --list
```

Lista configurações repositório/projeto

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

Referencia: <https://git-scm.com/book/pt-br/v2/Primeiros-passos-Configura%C3%A7%C3%A3o-Inicial-do-Git>

## 2 - Obtendo um Repositório Git

É possível obter um projeto Git utilizando duas formas principais. A primeira faz uso de **um projeto ou diretório existente e o importa para o Git**. A segunda clona um repositório Git existente a partir de outro servidor **"git clone"**.

### 2.1 - Criando um novo repositório.

Criando um novo repositório via linha de comando

```
echo "# helloworld" >> README.md
git init
git add README.md
git commit -m "first commit"
git remote add origin https://github.com/almeidass/helloworld.git
git push -u origin master
```

Push de um repositório existente via linha de comando

```
git remote add origin https://github.com/almeidass/helloworld.git
git push -u origin master
```

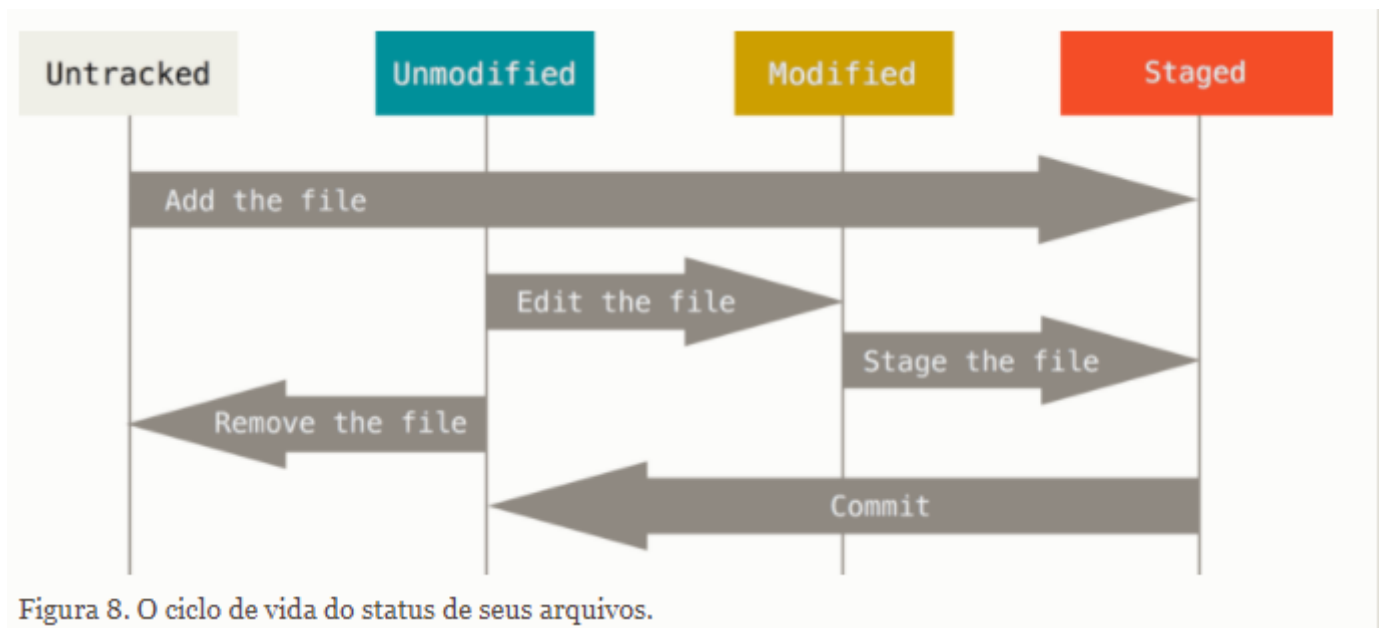
Comandos:

- git init: Cria um repositório Git vazio ou reinicializa um existente.
- git add: Adicionar conteúdo do arquivo ao índice, (staged area).

- `git commit -m`: Grava as alterações no repositório, a opção "-m" é da mensagem/descrição do commit.
- `git remote`: Gerencia os repositórios remotos, a opção "add origin" adiciona o caminho do repositório "https://github.com/almeidass/helloworld.git" com alias origin
- `git push`: Enviar arquivos/diretórios para o repositório remoto.
  - O primeiro **push** de um repositório deve conter o nome do repositório remoto e o branch (`git push -u origin master`).
  - Os demais **pushes** não precisam dessa informação (`git push`).

Referencia: <https://git-scm.com/docs>

### 3 - O ciclo de vida dos status dos arquivos



#### 3.1 - Status dos arquivos

##### Untracked "Não marcado"

Quando o arquivo é adicionado no repositório, mas ainda não é monitorado/visto pelo GIT, não existe nenhuma versão deste arquivo no repositório. Ao adicionar o arquivo no GIT "`git add`" ele passa para o status "Staged"

##### Unmodified

Não modificado/alterado desde a ultima versão criada.

##### Modified

Quando um arquivo já pertencente do repositório é modificado/alterado

##### Staged

Staged Area ou preparado, quando os arquivos estão prontos para serem commitados.

### 3.1 - Comandos para verificar o status e o histórico dos arquivos.

Verifica o estado dos arquivos/diretórios

```
git status
```

Exibir o histórico

```
git log
```