

Versionamento - Git E Github Conteúdos

Criar Acesso no GIT - Conta de Usuário

- Configurar o nome de usuário no Git:

```
bashCopy code
git config --global user.name "nome desejado"
```

- Configurar o email do usuário no Git:

```
bashCopy code
git config --global user.email "email desejado"
```

- Visualizar dados configurados:

```
bashCopy code
# Nome do usuário
git config user.name

# Email do usuário
git config user.email
```

Comandos Básicos Git

- **Inicializar Repositório:**

```
bashCopy code
git init
```

- **Gerenciar Repositórios Remotos:**

```
bashCopy code
git remote
```

- **Clonar Repositório Existente:**

- Clonar repositório próprio:

```
bashCopy code
git clone <URL_do_Repositório>
```

- Clonar repositório externo (requer autorização):

```
bashCopy code
git fork <URL_do_Repositório>
```

Adicionar e Comitar Mudanças

- **Adicionar Mudanças no Staged:**

- Adicionar todo conteúdo do arquivo:

```
bashCopy code
git add *
```

- Adicionar a local específico:

```
bashCopy code
git add <local.tipo>
```

- **Salvar as Mudanças na Área de Staged:**

```
bashCopy code
git commit -m "mensagem"
```

- **Adicionar e Comitar em um Comando:**

```
bashCopy code
git commit -am "mensagem"
```

Enviar e Sincronizar com Repositório Remoto

- **Enviar Alterações para Repositório Remoto:**

```
bashCopy code
git push origin remoto/main
```

- **Sincronizar Repositório Local com Remoto:**

```
bashCopy code
git pull
```

Gerenciar Ramificações do Repositório

- **Criar e Listar Branches:**

- Criar uma branch:

```
bashCopy code
git checkout -b nome_desejado
```

- Listar branches:

```
bashCopy code
git branch -l
```

- **Deletar Branch:**

```
bashCopy code
git branch -D nome.tipo
```

- **Navegar Entre Branches:**

```
bashCopy code
git checkout nome_da_branch
```

- **Mesclar Branches:**

```
bashCopy code
git merge nome_da_branch_que_deseja_mesclar
```

Configuração e Visualização do Repositório Remoto

- **Listar Repositório Remoto:**

```
bashCopy code
git remote -v
```

- **Adicionar Repositório Remoto:**

```
bashCopy code
git remote add origin <endereço_URL>
```

Manipulação de Arquivos e Diretórios

- **Criar Arquivo e Visualizar Conteúdo:**

```
bashCopy code
echo "nome" > nome.tipo
cat <arquivo.tipo>
```

- **Listar Conteúdo de uma Pasta:**

```
bashCopy code
ls -lha
```

- **Criar Arquivo/Pasta e Navegar Diretórios:**

```
bashCopy code
touch nome.tipo
mkdir nome.tipo
cd nome.tipo
cd ..
```

Histórico de Commits e Diferenças

- **Ver Histórico de Commit:**

```
bashCopy code
git log
git log --decorate
git log --author="nome"
git shortlog
git shortlog -sn
git log --graph
```

- **Mostrar Diferenças Antes do Commit:**

```
bashCopy code
git diff
git diff --name-only
```

- **Mostrar Autor de Alterações:**

```
bashCopy code
git blame nome.tipo
```

Passo a Passo Local dos Arquivos com GIT

- **Estados dos Arquivos:**

- **Untracked:** Git não conhece a existência do arquivo.
- **Unmodified:** Arquivo existe no Git mas não foi modificado.
- **Modified:** Arquivo modificado no Git.

- **Staged:** Arquivo pronto para commit.
- **Visualizar Mudanças em um Commit:**

```
bashCopy code
git show rash_do_commit
```

Resetar Erros/Alterações

- **Desfazer Alterações Antes de Add:**

```
bashCopy code
git checkout nome.tipo
```

- **Remover do Staged Após Add:**

```
bashCopy code
git reset HEAD nome.tipo
```

- **Voltar Commit e Alterações:**

- Soft - Fica na área de Staged:

```
bashCopy code
git reset --soft rash_do_commit_anterior
```

- Mixed - Volta antes do Staged (Unmodified):

```
bashCopy code
git reset --mixed rash_do_commit_anterior
```

- Hard - Ignora tudo que foi feito (Cuidado):

```
bashCopy code
git reset --hard rash_do_commit_anterior
```

Git Ignore e Stash

- **Ignorar Arquivos:**
 - Criar arquivo `.gitignore` e adicionar padrões dos arquivos a serem ignorados.
- **Stash:**

```
bashCopy code
git stash
git stash apply
git stash list
git stash clear
```

Criar Atalhos e Tags

- **Criar Atalhos para Comandos:**

```
bashCopy code
git config --global alias.s status
```

- **Tags:**

```
bashCopy code
git tag -a versão -m "mensagem_da_tag_de_alteracao"
```

Reverter Modificações

- **Reverter Modificações sem Reset:**

```
bashCopy code
git revert rash_do_commit
```