



# Exercícios - Git - Github

1. O git é um sistema de controle de versão distribuído e utilizado amplamente pela comunidade de desenvolvimento de software. Esse sistema possui um conjunto de comandos utilizados para o versionamento de código. Dito isso, qual o comando utilizado para enviar as alterações do repositório local para o repositório remoto?
  - a. git commit
  - b. git push
  - c. git add
  - d. git pull
  - e. git send
2. O Git é um sistema de controle de versões. Uma vez que o Git esteja instalado e em condições ideais, um Técnico digita um comando que aciona o Git no seu diretório atual, criando a *branch* *main* ou *master* e permitindo a ele executar os demais comandos do Git. Este comando é:
  - a. git add
  - b. git initialize
  - c. git commit
  - d. git init
  - e. git master

## Práticas:

### Proposta - Nível Médio

Praticar a adição, commit e push para o GitHub.

# Passos para realização:

## 1. Conteúdo do projeto

1. Crie uma pasta com o nome do projeto (a sua escolha) e um arquivo chamado `index.html`
2. Abra-o com o bloco de notas e adicione o seguinte conteúdo ao arquivo:

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <title>Título da página</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Aqui vai um título</h1>
  </body>
</html>
```

## 2. Subindo seu projeto para o GitHub

1. Crie um repositório em seu GitHub
2. Abra o seu terminal de comando e adicione a origem remota e conecte seu projeto local com o GitHub
3. Verifique as alterações do seu projeto e adicione ao fluxo de versionamento
4. Faça um commit com uma mensagem útil e na sequência, um push para o repositório remoto (GitHub)

## 3. Lidando com alterações

Abra a pasta do seu projeto, crie uma pasta chamada `imagens`, procure no Google por uma imagem de gato e salve dentro dessa nova pasta. Feito isso, abra seu arquivo `index.txt` com o bloco de notas e adicione as seguintes modificações e salve o arquivo:

```
<!DOCTYPE html>
<html lang="pt-br">
  <head>
    <title>Fanpage de Gatinhos</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Perfil #catsoninstagram</h1>
    
  </body>
</html>
```

Agora, repita os passos 2 e 3 do **item 2**.

## Proposta - Nível Difícil

Em vários casos, você encontrará arquivos que não pertencem (ou não deveriam) ao repositório. Exemplos típicos incluem:

Credenciais (exemplo: senhas, usuários, keys)

Cache de arquivos (exemplo: .pyc)

Ferramentas e bibliotecas complementares para execução do projeto (exemplo: node\_modules)

O Git permite que os usuários adicionem um arquivo `.gitignore` ao seu projeto, na raiz da pasta, para ignorar esses arquivos específicos.

## Passos para realização:

### Criando um arquivo não rastreável

1. Acesse a pasta do projeto que você criou no exercício 1 e crie um arquivo chamado `minhas_senhas_secretas.txt`
2. Abra o terminal e veja se o arquivo está aparecendo como não rastreado e que pode ser adicionado (dica)

### Gerando o .gitignore

1. Crie o arquivo `.gitignore` e salve-o na raiz do projeto (ele deverá aparecer no mesmo nível dos arquivos `index.html` e da pasta `imagens`)
2. Abra o arquivo `.gitignore`, digite `minhas_senhas_secretas.txt` e salve-o
3. No terminal, digite novamente o comando para verificar o status dos arquivos e veja que o `minhas_senhas_secretas.txt` não aparece mais!

### Publicando seu .gitignore

No seu terminal, rode os comandos:

1. `git add` para adicionar o arquivo
2. `git commit` para commitar o novo arquivo, sem esquecer de adicionar uma mensagem útil e informativa sobre o que será publicado
3. `git push` para enviar para o repositório no GitHub