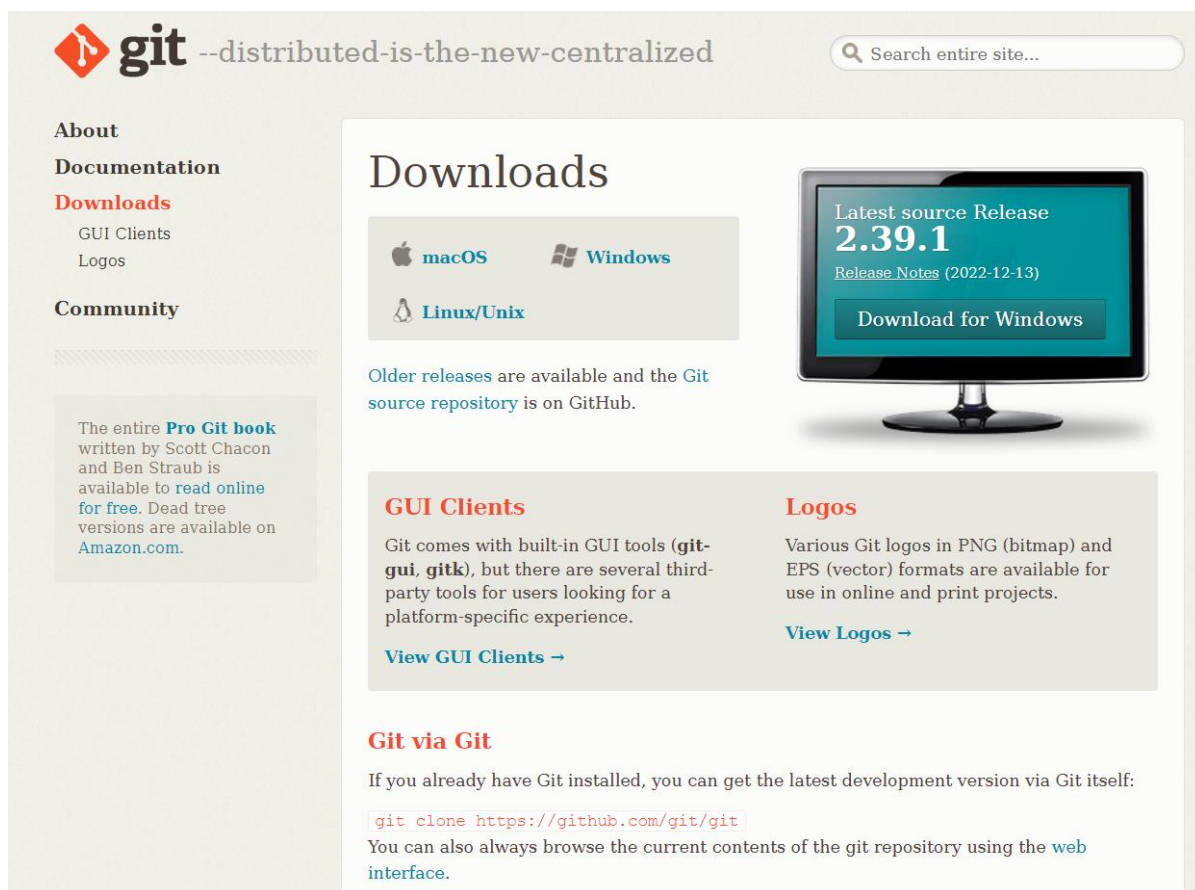
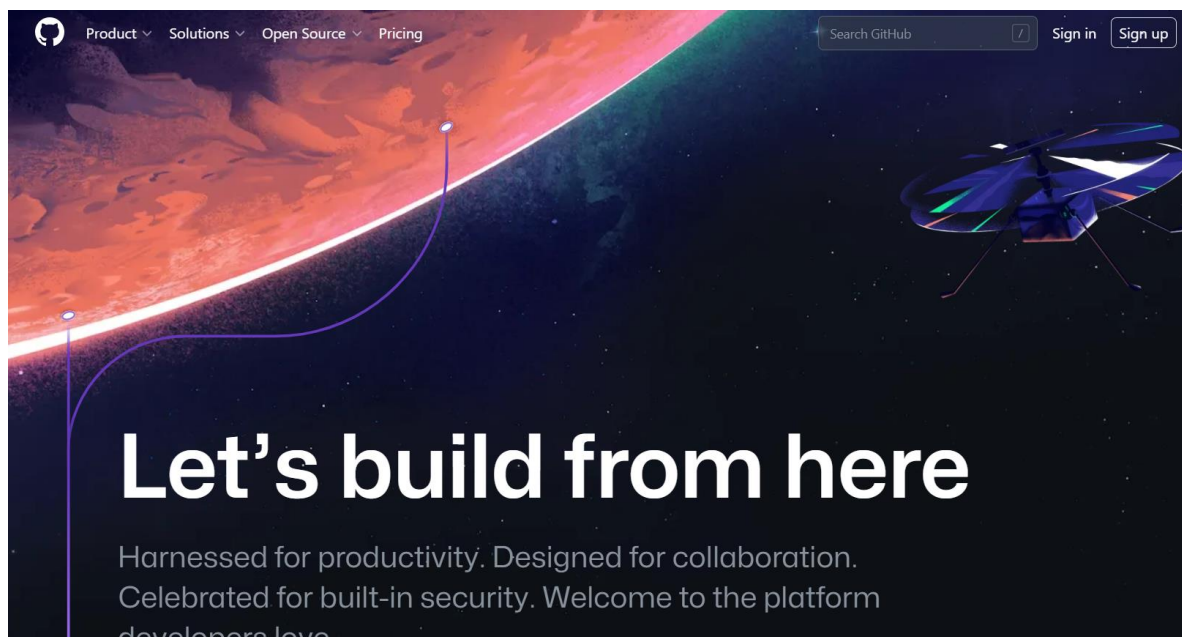


Parte 1 – Trabalhando com o GIT

1. Verifique a disponibilidade do Git CLI em seu notebook, caso necessário você deve efetivar a instalação através do link: <https://git-scm.com/downloads> (use a versão indicada para seu sistema operacional)



2. Para vincular um projeto, você precisa criar uma conta no <https://github.com/> preencha as informações para definir um usuário



Sign in to GitHub

Username or email address

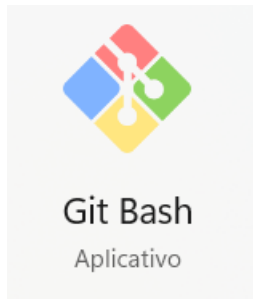
Password

[Forgot password?](#)

Sign in

New to GitHub? [Create an account.](#)

3. Instruções básicas para vincular seu repositório ao diretório do notebook.



Localize o diretório que deseja vincular ao Github.

```
Nisflei Galoni@IEGNTBHPB003 MINGW64 ~/Documents/teste
$ pwd
/c/Users/Nisflei Galoni/Documents/teste
```

```
git init
git config --global user.name "<Seu User>"
git config --global user.email "<Seu Email>"
```

4. Agora basta ir ao Github e criar um repositório.

```
...or create a new repository on the command line
echo "# teste" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M master
git remote add origin https://github.com/Nisflei/teste.git
git push -u origin master

...or push an existing repository from the command line
git remote add origin https://github.com/Nisflei/teste.git
git branch -M master
git push -u origin mast
```

Parte 2 - Praticar a adição, commit e push para o GitHub.

1. Conteúdo do projeto

Crie uma pasta com o nome do projeto (a sua escolha) e um arquivo chamado **index.html**
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

Crie um repositório em seu GitHub

Abra o seu terminal de comando e adicione a origem remota e conecte seu projeto local com o GitHub

Verifique as alterações do seu projeto e adicione ao fluxo de versionamento

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.html** e adicione as seguintes modificações e salve o arquivo.

Atualize esses dados na Git.

4. 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.

Dica: Consulte na web o gitignore.io para mais informações

5. Gerando o .gitignore

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)

Abra o arquivo **.gitignore**, digite minhas_senhas_secretas.txt e salve-o

No terminal, digite novamente o comando para verificar o status dos arquivos e veja que o minhas_senhas_secretas.txt não aparece mais!

6. Publicando seu **.gitignore**

No seu terminal, rode os comandos:

`git add` para adicionar o arquivo

`git commit` para commitar o novo arquivo, sem esquecer de adicionar uma mensagem útil e informativa sobre o que será publicado

`git push` para enviar para o repositório no GitHub

Parte 3 – Trabalhando com Branch.

1. Criando sua nova branch & commit de mudanças

No terminal escreva o comando **`git checkout -b nome-da-sua-branch`** para criar a branch e mudar para ela diretamente

Abra o seu editor e faça suas modificações, não se esqueça de salvar!

2. Voltando para o terminal e dentro da pasta do projeto, use o comando `git add .` para preparar todos os arquivos modificados para o commit

Agora use o comando **`git commit -m "aqui sua mensagem de commit"`** para fazer o commit das suas mudanças na Branch

Para enviar seu commit para seu repositório remoto utilize o comando **`git push origin nome-da-sua-branch`**

3. Fazendo Pull Request (PR) & dando merge

No site do GitHub e na sua página do repositório na aba **Pull requests** clique no botão **New pull request**

Agora ajuste nas opções para qual branch você quer enviar as suas modificações (lado esquerdo) e a sua branch que estará enviando essas modificações (lado direito)

Agora você só precisa fazer o merge, para isso na mesma aba do passo 1, clique na PR que deseja dar merge

Como não há nenhum conflito entre as branches é só clicar no botão **Merge pull request**

Parabéns, você conseguiu! 😊

Quer aprender mais sobre Git... https://learngitbranching.js.org/?locale=pt_BR