

Guia: Usando Git e GitHub no VS Code para seu Projeto

Este guia foca em como usar a interface gráfica do Visual Studio Code (VS Code) para gerenciar seu projeto com Git e enviá-lo para o repositório que você criou no GitHub. Isso facilita o versionamento e o processo de deploy no Streamlit Community Cloud.

1. Extensões Recomendadas:

- **Git:** O VS Code já vem com suporte integrado ao Git. Você só precisa ter o [Git](#) instalado no seu sistema operacional para que o VS Code o utilize.
- **GitHub Pull Requests and Issues (Opcional, mas Altamente Recomendado):** Publicada pela própria GitHub, esta extensão integra funcionalidades como visualização de Pull Requests, Issues e facilita a autenticação com sua conta GitHub diretamente no VS Code. Para instalar:
 - Clique no ícone de Extensões na barra lateral esquerda do VS Code (parece um conjunto de blocos).
 - Procure por "GitHub Pull Requests and Issues".
 - Clique em "Instalar".

2. Abrindo seu Projeto e a Aba de Controle de Código Fonte:

- Abra o VS Code.
- Vá em `File > Open Folder...` e selecione a pasta do seu projeto (`C:\Users\Valdemir\Projetos\LeadsBaixada`).
- Clique no ícone de Controle de Código Fonte na barra lateral esquerda (geralmente o terceiro ícone, parece uma bifurcação de galhos: <0xF0><0x9F><0x94><0x84>).

3. Inicializando o Git (Se necessário):

- Se você **ainda não** inicializou o Git na sua pasta (com `git init` no terminal), o VS Code geralmente oferece um botão "Initialize Repository" nesta aba de Controle de Código Fonte. Clique nele.
- Se você **já** inicializou o Git via terminal, o VS Code deve detectar automaticamente o repositório e mostrar os arquivos modificados ou não rastreados.

4. Preparando o Primeiro Commit (Staging):

- Na aba de Controle de Código Fonte, você verá uma seção chamada "Changes". Ela lista todos os arquivos que foram modificados, adicionados ou excluídos desde o último commit (ou todos os arquivos, se for o primeiro commit).
- Para incluir arquivos no seu próximo commit (processo chamado "staging"), passe o mouse sobre cada arquivo que você quer incluir (como `app_streamlit_folium.py`, `requirements.txt`, `leads_baixada.csv`) e clique no ícone `+` (Stage Changes) que aparece ao lado do nome do arquivo.
- Os arquivos movidos para "staging" aparecerão agora na seção "Staged Changes".

5. Fazendo o Primeiro Commit:

- Acima da seção "Staged Changes", há uma caixa de texto com o placeholder "Message". Digite uma mensagem descritiva para este commit. Para o primeiro, algo como "Versão inicial do dashboard com Folium" ou "Adiciona arquivos do projeto" é bom.
- Após digitar a mensagem, clique no botão de "check" (✓ Commit) logo acima da caixa de mensagem.
- **Observação:** Se for a primeira vez que você faz commit no seu sistema, o Git pode pedir para você configurar seu nome de usuário e email. O VS Code geralmente mostra os comandos necessários no terminal (`git config --global user.name "Seu Nome"` e `git config --global user.email "seuemail@example.com"`). Execute-os se solicitado.

6. Conectando ao Repositório Remoto do GitHub:

- Como você já criou o repositório no GitHub, precisamos dizer ao Git local onde ele deve enviar as alterações.
- **Se você já usou `git remote add origin ...` no terminal:** O VS Code já deve saber para onde enviar.
- **Se você ainda não conectou:**
 - O VS Code, após o primeiro commit, pode oferecer um botão "Publish Branch" ou similar na aba de Controle de Código Fonte ou na barra de status inferior.
 - Clicar em "Publish Branch" geralmente abre opções para publicar em um repositório GitHub existente ou criar um novo. Escolha a opção para conectar a um repositório existente.
 - O VS Code pode pedir para você fazer login no GitHub (se tiver a extensão instalada, isso é mais fácil) e depois listar seus repositórios. Selecione o repositório que você criou (`dashboard-leads-streamlit` ou o nome que deu).

- **Alternativa (Via Paleta de Comandos):**

- Pressione `Ctrl+Shift+P` para abrir a Paleta de Comandos.
- Digite `Git: Add Remote` e selecione a opção.
- Cole a URL do seu repositório GitHub (ex: `https://github.com/SEU_USUARIO/SEU_REPOSITORIO.git`).
- Dê um nome ao remoto (o padrão `origin` é recomendado).

7. Enviando as Alterações para o GitHub (Push):

- Após o commit e a conexão com o remoto, você precisa enviar (push) suas alterações locais para o GitHub.
- Procure por um botão com um ícone de nuvem com uma seta para cima (`<0xE2><0x98><0x81>`) ou um botão "Sync Changes" na barra de status inferior (geralmente no canto inferior esquerdo). Ele pode mostrar "0 ↓ 1 ↑", indicando 1 commit para enviar.
- Clicar neste botão (ou usar o menu `...` na aba de Controle de Código Fonte e escolher `Push`) enviará seus commits para o branch `main` (ou `master`) no GitHub.
- Pode ser necessário autenticar com o GitHub novamente.

8. Verificando no GitHub:

- Após o push, vá até a página do seu repositório no GitHub e atualize-a. Você deverá ver seus arquivos (`app_streamlit_folium.py`, `requirements.txt`, etc.) lá.

Gerenciando Alterações Futuras:

- Sempre que você modificar seus arquivos:
 1. Vá para a aba Controle de Código Fonte.
 2. Faça "Stage" das alterações que deseja incluir (clcando no `+`).
 3. Escreva uma mensagem de commit descritiva.
 4. Clique em "✓ Commit".
 5. Clique no botão "Sync Changes" ou use a opção "Push" para enviar as novas alterações ao GitHub.

Usar a interface do VS Code torna o processo de versionamento com Git bem visual e integrado ao seu fluxo de trabalho de desenvolvimento.