

# Guia Passo a Passo: Deploy no Streamlit Community Cloud

Este guia detalha como implantar (fazer o deploy) sua aplicação Streamlit ( `app_streamlit_folium.py` ) na Streamlit Community Cloud, tornando-a acessível publicamente através de um link.

## Pré-requisitos:

- Uma conta no [GitHub](#).
- O Git instalado na sua máquina (recomendado para facilitar o envio dos arquivos para o GitHub, mas você pode fazer upload pela interface web do GitHub também).
- Sua pasta de projeto organizada contendo:
  - `app_streamlit_folium.py` (o script principal).
  - `requirements.txt` (com as dependências: streamlit, pandas, folium, streamlit-folium).
  - `leads_baixada.csv` (seu arquivo de dados, **se você optou por incluí-lo no repositório** e ajustou o script para carregá-lo automaticamente, conforme a "Opção A" do guia de estrutura).

## Passos:

### Parte 1: Preparar e Enviar para o GitHub

#### 1. Criar um Repositório Público no GitHub:

- Vá para [github.com](#) e faça login.
- Clique no botão "+" no canto superior direito e selecione "New repository".
- Dê um nome ao seu repositório (ex: `dashboard-leads-streamlit`).
- **Importante:** Marque a opção "Public". O plano gratuito do Streamlit Cloud só funciona com repositórios públicos.
- Você pode marcar "Add a README file" se quiser.
- Clique em "Create repository".

#### 2. Adicionar Arquivos ao Repositório:

- **Método 1 (Usando Git - Recomendado):**
  - Abra o terminal ou Git Bash na pasta do seu projeto local ( `C:\Users\Valdemir\Projetos\LeadsBaixada` ou onde estiverem os arquivos finais).

- Inicialize o Git (se ainda não fez): `git init`
- Adicione os arquivos essenciais: `git add app_streamlit_folium.py requirements.txt leads_baixada.csv` (adicione o CSV apenas se for usá-lo diretamente do repo).
- Faça o commit: `git commit -m "Versão inicial do dashboard para deploy"`
- Adicione o repositório remoto do GitHub (copie o comando da página do seu repositório no GitHub - geralmente algo como): `git remote add origin https://github.com/SEU_USUARIO/SEU_REPOSITORIO.git`
- Envie os arquivos para o GitHub: `git push -u origin main` (ou `master`, dependendo do nome do seu branch principal).
- **Método 2 (Upload pela Interface Web):**
  - Na página do seu repositório no GitHub, clique em "Add file" > "Upload files".
  - Arraste e solte os arquivos `app_streamlit_folium.py`, `requirements.txt`, e `leads_baixada.csv` (se aplicável) para a área de upload.
  - Adicione uma mensagem de commit (ex: "Adiciona arquivos da aplicação") e clique em "Commit changes".

## Parte 2: Deploy no Streamlit Community Cloud

### 1. Acessar o Streamlit Community Cloud:

- Vá para [share.streamlit.io](https://share.streamlit.io).
- Clique em "Continue with GitHub" e autorize o Streamlit a acessar sua conta GitHub (se for a primeira vez).

### 2. Iniciar um Novo Deploy:

- No seu workspace do Streamlit Cloud, clique no botão "New app".

### 3. Configurar o Deploy:

- **Repository:** Selecione o repositório GitHub que você criou na Parte 1 (ex: `dashboard-leads-streamlit`).
- **Branch:** Selecione o branch principal (geralmente `main` ou `master`).
- **Main file path:** Digite o nome do seu script principal: `app_streamlit_folium.py`.
- **App URL (Opcional):** Você pode customizar parte da URL se desejar.
- **Advanced settings (Opcional):** Normalmente não é necessário mexer aqui para aplicações simples, mas você pode especificar a versão do Python se precisar.

#### 4. Fazer o Deploy:

- Clique no botão "Deploy!".

#### 5. Aguardar o Processo:

- O Streamlit Cloud começará a construir o ambiente para sua aplicação. Você verá logs mostrando a instalação das dependências do seu `requirements.txt`.
- Este processo pode levar alguns minutos, especialmente na primeira vez.
- Se tudo correr bem, você verá uma mensagem de sucesso e sua aplicação será carregada.

#### 6. Acessar sua Aplicação:

- Sua aplicação estará disponível na URL fornecida pelo Streamlit Cloud (ex: `seu-usuario-seu-repo-app_streamlit_folium-abcdef.streamlit.app`).
- Você pode copiar este link e compartilhá-lo com seu cliente!

#### Solução de Problemas:

- **Erro no Deploy:** Se o deploy falhar, verifique os logs exibidos pelo Streamlit Cloud. Erros comuns incluem:
  - Faltando dependências no `requirements.txt`.
  - Erros de sintaxe no seu código Python.
  - Problemas ao carregar o arquivo de dados (se você o incluiu no repo mas o caminho está errado no script).
- **Aplicação Lenta:** O plano gratuito tem recursos limitados. Se sua aplicação for muito complexa ou processar muitos dados, ela pode ficar lenta.

Seguindo estes passos, você deverá conseguir colocar seu dashboard online para fácil compartilhamento.