**Parte 2: Colocar em Produção para o Cliente Testar**

A maneira mais rápida e profissional de partilhar este protótipo é usando o **Streamlit Community Cloud**, que é gratuito.

**Passo 1: Envie seu Projeto para o GitHub**

1. Crie um novo repositório no [GitHub](https://github.com/).
2. Envie a sua pasta de projeto (contendo app.py, requirements.txt e o perfis\_consultoras\_exemplo.csv) para este repositório.

**Passo 2: Faça o Deploy no Streamlit Community Cloud**

1. Vá para [share.streamlit.io](https://share.streamlit.io/) e faça login com a sua conta do GitHub.
2. Clique em "**New app**".
3. Selecione o repositório do GitHub que acabou de criar.
4. Verifique se o "Main file path" está como app.py.
5. Clique em "**Deploy!**".

O Streamlit irá construir e hospedar a sua aplicação automaticamente. Em poucos minutos, você terá um link público (ex: https-seu-nome-seu-app-aleatorio.streamlit.app) que poderá partilhar diretamente com o cliente. Ele poderá interagir com o protótipo exatamente como você fez localmente.

1. **Git Instalado:** Abra o seu terminal (ou Git Bash no Windows) e digite git --version. Se aparecer uma versão (ex: git version 2.39.2), está tudo certo. Se não, [faça o download e instale o Git aqui](https://git-scm.com/downloads).
2. **Conta no GitHub:** Se ainda não tiver, [crie uma conta gratuita no GitHub](https://github.com/join).
3. **A sua pasta de projeto:** A pasta natura pronta no seu computador.

**Passo a Passo: Enviando seu Projeto para um Novo Repositório**

Este é o cenário mais comum e recomendado. Vamos criar um "espaço" novo e limpo no GitHub e depois enviar a sua pasta para lá.

**Etapa 1: Criar o Repositório no Site do GitHub**

1. Faça login na sua conta do GitHub.
2. No canto superior direito, clique no ícone de + e selecione **"New repository"**.
3. **Repository name:** Dê um nome ao seu repositório. Por exemplo: projeto-natura-crm.
4. **Description (Opcional, mas recomendado):** Escreva uma breve descrição. Ex: "Protótipo do motor de campanhas CRM para a Natura, desenvolvido com Python e Streamlit."
5. **Public ou Private:** Escolha se o seu repositório será visível para todos (Público) ou apenas para você e colaboradores que convidar (Privado). Para um projeto de cliente, **Privado** é geralmente a melhor escolha.
6. **⚠️ Ponto Crítico:** **NÃO** marque as caixas "Add a README file", "Add .gitignore" ou "Choose a license" nesta etapa. Como você já tem uma pasta com arquivos, é mais fácil começar com um repositório completamente vazio para evitar conflitos.
7. Clique no botão verde **"Create repository"**.

O GitHub irá agora mostrar-lhe uma página com algumas instruções de linha de comando. É essa página que usaremos na próxima etapa.

**Etapa 2: Preparar e Enviar a sua Pasta Local via Terminal**

Agora vamos para o terminal (ou Prompt de Comando/Git Bash) no seu computador.

1. **Navegue até a sua pasta de projeto:** Use o comando cd (change directory) para entrar na pasta natura.
   * **Exemplo no Windows:**

Bash

cd C:\Users\SeuUsuario\Documentos\natura

* + **Exemplo no macOS/Linux:**

Bash

cd /home/seu-usuario/documentos/natura

1. **Inicialize o Git na sua pasta:** Este comando cria um subdiretório oculto .git que rastreia todas as alterações.

Bash

git init

1. **Adicione todos os arquivos à "área de preparação" (Staging Area):** O ponto . significa "todos os arquivos e pastas no diretório atual".

Bash

git add .

1. **Crie o "pacote" de alterações (Commit):** O commit é um "snapshot" dos seus arquivos num determinado momento. A mensagem (-m) deve ser descritiva.

Bash

git commit -m "Commit inicial: Adiciona a primeira versão do protótipo Streamlit"

1. **Renomeie a branch principal para main:** Esta é a convenção moderna, substituindo a antiga master.

Bash

git branch -M main

1. **Conecte a sua pasta local ao repositório do GitHub:** Volte à página do seu repositório no GitHub. Copie a URL que aparece lá (deve terminar em .git). O comando abaixo diz ao seu Git local "este é o destino online do seu projeto". origin é o nome padrão para essa conexão.

Bash

# Substitua a URL pela URL do SEU repositório!

git remote add origin https://github.com/seu-usuario/projeto-natura-crm.git

1. **Envie (Push) o seu código para o GitHub:** Este é o comando final que efetivamente transfere os seus arquivos.

Bash

git push -u origin main

* + O Git pode pedir o seu nome de utilizador e senha (ou um Personal Access Token, que é o método mais seguro e moderno) do GitHub para autorizar o envio.

**Verificação Final**

Parabéns! Se tudo correu bem, o seu trabalho está feito. Volte à página do seu repositório no GitHub e **atualize a página (F5)**. Você deverá ver todos os seus arquivos (app.py, requirements.txt, etc.) listados lá.

O seu projeto está agora seguro, versionado e pronto para ser partilhado ou implementado a partir do GitHub.

Para futuras atualizações no seu código, o fluxo de trabalho será muito mais simples:

1. Faça as alterações no seu código.
2. No terminal, dentro da pasta do projeto, execute:

Bash

git add .

git commit -m "Mensagem descrevendo o que você mudou"

git push