

Guia de Deploy - Serviço em Casa MVP

Como fazer o upload para GitHub

1. Criar repositório no GitHub

1. Acesse [GitHub.com](https://github.com) e faça login
2. Clique em "New repository" (+ no canto superior direito)
3. Nome sugerido: `servico-em-casa-mvp`
4. Descrição: "MVP de marketplace de serviços domésticos"
5. Deixe como **público** (para usar GitHub Pages gratuito)
6. **NÃO** marque "Initialize with README" (já temos um)
7. Clique em "Create repository"

2. Fazer upload do código

Você tem duas opções:

Opção A: Via linha de comando (recomendado)

Bash

```
# No diretório do projeto
git remote add origin https://github.com/SEU_USUARIO/servico-em-casa-mvp.git
git branch -M main
git push -u origin main
```

Opção B: Via interface web do GitHub

1. Na página do repositório criado, clique em "uploading an existing file"
2. Arraste o arquivo ZIP `servico-em-casa-mvp.zip` ou selecione os arquivos
3. Adicione uma mensagem de commit

4. Clique em "Commit changes"

Opções de Deploy Gratuito

1. Frontend (React) - Vercel/Netlify

Vercel (Recomendado)

1. Acesse vercel.com
2. Conecte com sua conta GitHub
3. Importe o repositório `servico-em-casa-mvp`
4. Configure:
 - **Framework Preset:** Vite
 - **Root Directory:** `frontend`
 - **Build Command:** `npm run build`
 - **Output Directory:** `dist`
5. Deploy automático!

Netlify

1. Acesse netlify.com
2. "New site from Git" → GitHub → selecione o repositório
3. Configure:
 - **Base directory:** `frontend`
 - **Build command:** `npm run build`
 - **Publish directory:** `frontend/dist`

2. Backend (Flask) - Railway/Render

Railway (Recomendado)

1. Acesse railway.app
2. "New Project" → "Deploy from GitHub repo"
3. Selecione o repositório
4. Configure:
 - **Root Directory:** `backend`
 - **Start Command:** `python src/main.py`
5. Adicione variável de ambiente: `PORT=5000`

Render

1. Acesse render.com
2. "New" → "Web Service"
3. Conecte o repositório GitHub
4. Configure:
 - **Root Directory:** `backend`
 - **Build Command:** `pip install -r requirements.txt`
 - **Start Command:** `python src/main.py`

3. Banco de Dados - PostgreSQL Gratuito

Supabase (Recomendado)

1. Acesse supabase.com
2. "New project"
3. Anote as credenciais de conexão
4. Atualize o backend para usar PostgreSQL

ElephantSQL

1. Acesse elephantsql.com
2. "Create New Instance"
3. Plano gratuito "Tiny Turtle"
4. Anote a URL de conexão

Configuração de Produção

1. Variáveis de Ambiente

Backend

Plain Text

```
DATABASE_URL=postgresql://user:pass@host:port/db  
SECRET_KEY=sua_chave_secreta_super_forte  
FLASK_ENV=production
```

Frontend

Plain Text

```
VITE_API_URL=https://seu-backend.railway.app
```

2. Atualizações Necessárias

Backend (`backend/src/main.py`)

Python

```
import os  
from flask import Flask  
from flask_cors import CORS  
  
app = Flask(__name__)  
CORS(app, origins=["https://seu-frontend.vercel.app"])
```

```
# Usar porta do ambiente ou 5000
port = int(os.environ.get('PORT', 5000))
app.run(host='0.0.0.0', port=port)
```

Frontend (frontend/src/lib/api.js)

JavaScript

```
const API_BASE_URL = import.meta.env.VITE_API_URL || 'http://localhost:5000';
```

Checklist de Deploy

- ☐ Repositório criado no GitHub
- ☐ Código enviado para o repositório
- ☐ Frontend deployado (Vercel/Netlify)
- ☐ Backend deployado (Railway/Render)
- ☐ Banco de dados configurado
- ☐ Variáveis de ambiente definidas
- ☐ CORS configurado no backend
- ☐ URLs de produção atualizadas
- ☐ Teste completo em produção

Custos Estimados

Gratuito (Tier Free)

- **GitHub:** Repositórios públicos gratuitos
- **Vercel:** 100GB bandwidth/mês

- **Railway:** \$5 crédito inicial + 500h/mês
- **Supabase:** 500MB database + 2GB bandwidth

Pago (se necessário)

- **Vercel Pro:** \$20/mês
- **Railway:** \$5-20/mês conforme uso
- **Supabase Pro:** \$25/mês

Domínio Personalizado (Opcional)

1. **Registro.br:** R\$ 40/ano (.com.br)
2. **Namecheap:** \$10-15/ano (.com)
3. Configure DNS no provedor de hospedagem

Monitoramento

Gratuito

- **Vercel Analytics:** Incluído no plano gratuito
- **Railway Metrics:** Dashboard básico
- **Google Analytics:** Gratuito

Logs e Erros

- **Sentry:** 5k erros/mês gratuito
- **LogRocket:** 1k sessões/mês gratuito

Próximos Passos

1. **Fazer upload para GitHub**
2. **Deploy do frontend na Vercel**
3. **Deploy do backend no Railway**
4. **Configurar banco PostgreSQL**
5. **Testar integração completa**
6. **Configurar domínio (opcional)**
7. **Adicionar monitoramento**

Suporte

Para dúvidas sobre deploy:

- Documentação do Vercel: vercel.com/docs
- Documentação do Railway: docs.railway.app
- Comunidade GitHub: Issues no repositório