



Guia para Push no GitHub

Seu repositório está configurado e pronto para o push!

Repositório GitHub: <https://github.com/anabmoser/Controle-Financeiro-Abacus>



Status Atual

- ☒ Remote configurado
- ☒ Branch renomeado para `main`
- ☒ Todos os commits prontos (11 commits)
- ☒ `.gitignore` atualizado (protege credenciais)
- ☒ `.env.example` criado (template de configuração)
- ☒ **Pendente:** Autenticação GitHub para push



Opção 1: Push via HTTPS (Personal Access Token)

Passo 1: Criar Personal Access Token (PAT)

1. Acesse: <https://github.com/settings/tokens>
2. Clique em **“Generate new token (classic)”**
3. Configure:
 - **Note:** “Controle Financeiro Deploy”
 - **Expiration:** 90 days (ou conforme preferir)
 - **Scopes:** Marque apenas ☒ **repo** (controle total de repositórios privados)
4. Clique em **“Generate token”**
5. ⚠ **COPIE O TOKEN AGORA** (não poderá ver novamente)

Passo 2: Fazer Push com o Token

```
cd /home/ubuntu/controle_compras_restaurante

# Substitua YOUR_TOKEN_HERE pelo token que você copiou
git push -u origin main
# Quando pedir:
# Username: anabmoser
# Password: cole_seu_token_aqui
```

Ou faça em um comando só:

```
git push https://YOUR_TOKEN_HERE@github.com/anabmoser/Controle-Financeiro-Abacus.git main
```

Opção 2: Push via SSH (Mais Seguro)

Passo 1: Gerar chave SSH (se ainda não tiver)

```
ssh-keygen -t ed25519 -C "seu-email@example.com"
# Pressione Enter para aceitar o local padrão
# Opcionalmente, adicione uma senha

# Copiar a chave pública
cat ~/.ssh/id_ed25519.pub
```

Passo 2: Adicionar chave no GitHub

1. Acesse: <https://github.com/settings/keys>
2. Clique em **"New SSH key"**
3. Cole a chave pública copiada
4. Clique em **"Add SSH key"**

Passo 3: Atualizar Remote para SSH

```
cd /home/ubuntu/controle_compras_restaurante
git remote set-url origin git@github.com:anabmoser/Controle-Financeiro-Abacus.git
git push -u origin main
```

O que será enviado ao GitHub

✓ Código Fonte (Todos os arquivos atualizados)

- `/nextjs_space/app/` - Todas as páginas
- `/nextjs_space/app/api/` - 17 endpoints de API
- `/nextjs_space/components/` - Componentes UI
- `/nextjs_space/lib/` - Utilitários e configurações
- `/nextjs_space/prisma/` - Schema do banco de dados

✓ Arquivos de Configuração





- `package.json` - Dependências
- `next.config.js` - Configuração Next.js
- `tailwind.config.ts` - Estilos
- `tsconfig.json` - TypeScript
- `.env.example` - Template de variáveis de ambiente

✗ NÃO será enviado (protegido por `.gitignore`)

- `.env` - Suas credenciais Supabase e APIs
- `node_modules/` - Dependências (podem ser reinstaladas)
- `.next/` - Build artifacts
- `.logs/` - Logs locais




Depois do Push

Quando o push for concluído com sucesso:

1.  Todo o código estará no GitHub
2.  Você pode clonar em qualquer máquina
3.  Pode colaborar com outros desenvolvedores
4.  Tem backup completo do projeto

Configurar GitHub Actions (Opcional)

Posso criar um workflow de CI/CD para:

-  Deploy automático ao fazer push
-  Testes automatizados
-  Build verification

Resolução de Problemas

Erro: “Authentication failed”

- Verifique se copiou o token corretamente
- Token deve ter permissão **repo**
- Token não pode estar expirado

Erro: “Permission denied (publickey)”

- Verifique se a chave SSH está no GitHub
- Execute: `ssh -T git@github.com` para testar

Erro: “Repository not found”

- Verifique se o repositório existe
- Verifique se o nome está correto
- Verifique se você tem acesso ao repositório

Precisa de Ajuda?

Se encontrar qualquer problema, me avise e eu ajudo a resolver! 😊