

# Guia Completo: Push GitHub + Deploy Vercel

---

## Status do Projeto

-  Código 100% corrigido e testado
  -  Build passando sem erros TypeScript
  -  Todas as variáveis de ambiente configuradas
  -  Git inicializado com 5 commits
  -  .gitignore protegendo arquivos sensíveis
  -  .env.example documentado
  -  README.md completo
- 

## Passo 1: Criar Repositório no GitHub

### 1.1 Criar Repositório

1. Acesse: <https://github.com/new>
2. **Nome do repositório:** recruitai (ou outro nome de sua preferência)
3. **Visibilidade:** Private (recomendado) ou Public
4. **NÃO** marque nenhuma opção de inicialização:
  -  Add a README file
  -  Add .gitignore
  -  Choose a license
5. Clique em “Create repository”

### 1.2 Copiar URL do Re却itório

Após criar, você verá uma URL como:

[https://github.com/SEU\\_USUARIO/recruitai.git](https://github.com/SEU_USUARIO/recruitai.git)

**Guarde essa URL!** Você vai precisar dela no próximo passo.

## Passo 2: Fazer Push para o GitHub

### Opção A: Usando o Script Automatizado (RECOMENDADO)

```
cd /home/ubuntu/ats_platform/nextjs_space
chmod +x push-to-github.sh
./push-to-github.sh
```

Quando solicitado:

1. Digite a URL do seu repositório GitHub
2. Confirme a mensagem de commit (ou pressione Enter para usar a padrão)

## Opção B: Comandos Manuais

```
cd /home/ubuntu/ats_platform

# Adicionar remote do GitHub (substitua pela SUA URL)
git remote add origin https://github.com/SEU_USUARIO/recruitai.git

# Renomear branch para main
git branch -M main

# Push para GitHub
git push -u origin main
```

### **⚠ Se encontrar erro de autenticação:**

1. Vá em: <https://github.com/settings/tokens>
2. Clique em “Generate new token” → “Generate new token (classic)”
3. Dê um nome (ex: “RecruitAI Deploy”)
4. Marque os scopes:
  - repo (todos os sub-items)
  - workflow
5. Clique em “Generate token”
6. **CÓPIE O TOKEN** (você só verá uma vez!)
7. Use como senha quando fizer o push:  
 bash  
 Username: seu\_usuario\_github  
 Password: [cole o token aqui]

## 🌐 Passo 3: Deploy na Vercel

### 3.1 Conectar Vercel ao GitHub

1. Acesse: <https://vercel.com/new>
2. Clique em “Continue with GitHub”
3. Autorize a Vercel a acessar seus repositórios
4. Selecione o repositório `recruitai` (ou o nome que você escolheu)
5. Clique em “Import”

### 3.2 ⚠ CRÍTICO: Configurar Root Directory

**ATENÇÃO:** Este é o passo mais importante!

1. Na seção “Configure Project”
2. Expanda “Build and Output Settings”
3. Em “Root Directory”, clique em “Edit”
4. Digite: `nextjs_space`
5. Clique em “Continue”

### 3.3 Adicionar Variáveis de Ambiente

**NÃO clique em “Deploy” ainda!**

1. Role até a seção “**Environment Variables**”

2. Adicione TODAS as variáveis abaixo (copie os valores do seu arquivo `.env` local):

#### Database

```
DATABASE_URL=postgresql://postgres.jztrqlqrcgljpmxsbwf:Fcm%402025@aws-1-sa-east-1.pooler.supabase.com:6543/postgres?pgbouncer=true
```

#### NextAuth

```
NEXTAUTH_SECRET=cUpDwUY6HSJtYDuZX7eGf1h0Xerdjwjm
NEXTAUTH_URL=https://SEU_DOMINIO.vercel.app
```

**⚠️ IMPORTANTE:** Deixe `NEXTAUTH_URL` em branco por agora. Você vai atualizar depois do primeiro deploy com a URL que a Vercel gerar.

#### Stripe

```
STRIPE_SECRET_KEY=sk_live_51SVzwvDtZaVksYfa6t91TF1oVuX0SizCQbkjDFxBihd70821pzf7AFazuQmBw4MIkz5kWUf4Xy
STRIPE_PUBLISHABLE_KEY=pk_live_51SVzwvDtZaVksYfabty7tFEb0LZSCuTv0cFlpAonZ2kw7MIZtMpsPn
QtWpuHfJgx36teTdjn7xobC0LeRMVQbAUV00Z05vWGVw
STRIPE_WEBHOOK_SECRET=[deixe em branco por enquanto]
```

**⚠️ IMPORTANTE:** O `STRIPE_WEBHOOK_SECRET` será configurado no Passo 4.

#### AWS S3

```
AWS_ACCESS_KEY_ID=AKIARJI3AIFWJPOTWNMN
AWS_SECRET_ACCESS_KEY=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
AWS_S3_REGION=us-east-2
AWS_S3_BUCKET_NAME=recruitai-resumes
AWS_S3_FOLDER_PREFIX=resumes/
```

#### Abacus.AI

```
ABACUSA1_API_KEY=5bb8032f287b4b89bfcae4529b50a199
```

#### Email (Zoho SMTP)

```
SMTP_HOST=smtp.zoho.com
SMTP_PORT=587
SMTP_USER=comercial@fcmttech.com.br
SMTP_PASS=xG1PbdchhJYP
SMTP_FROM_NAME=RecruitAI
```

## OAuth

```
GOOGLE_CLIENT_ID=[seu_client_id]
GOOGLE_CLIENT_SECRET=[seu_client_secret]
LINKEDIN_CLIENT_ID=[seu_client_id]
LINKEDIN_CLIENT_SECRET=[seu_client_secret]
```

 **Se você não tiver as credenciais OAuth, deixe em branco.** O login via email/senha continuará funcionando.

## Teste e Manutenção

```
TEST_MODE_EMAIL=teste@fcmtech.com.br
MAINTENANCE_SECRET=[gere um token seguro]
```

Para gerar o `MAINTENANCE_SECRET` :

```
node -e "console.log(require('crypto').randomBytes(32).toString('hex'))"
```

1. Após adicionar TODAS as variáveis, clique em “**Deploy**”

## 3.4 Aguardar o Deploy

- O primeiro deploy leva ~2-5 minutos
- Você verá logs em tempo real
- Aguarde até ver:  “**Build Completed**”

## Passo 4: Configurar Webhook do Stripe

### 4.1 Obter URL da Vercel

Após o deploy bem-sucedido:

1. Copie a URL do seu app (exemplo: <https://recruitai-xyz123.vercel.app>)
2. Anote essa URL

### 4.2 Criar Webhook no Stripe

1. Acesse: <https://dashboard.stripe.com/webhooks>
2. Clique em “**Add endpoint**”
3. **Endpoint URL:** [https://SUA\\_URL.vercel.app/api/webhooks/stripe](https://SUA_URL.vercel.app/api/webhooks/stripe)
4. **Events to send:** Selecione os seguintes eventos:
  - checkout.session.completed
  - customer.subscription.created
  - customer.subscription.updated
  - customer.subscription.deleted
  - invoice.paid
  - invoice.payment\_failed
5. Clique em “**Add endpoint**”
6. Na página do webhook, clique em “**Reveal**” ao lado de “Signing secret”
7. Copie o valor `whsec_...`

## 4.3 Atualizar Variáveis na Vercel

1. Vá para o dashboard do seu projeto na Vercel
  2. Clique em “**Settings**” → “**Environment Variables**”
  3. Adicione/atualize:  

```
STRIPE_WEBHOOK_SECRET=whsec_[valor_copiado_do_stripe]
NEXTAUTH_URL=https://sua-url.vercel.app
```
  4. Clique em “**Save**”
  5. Na aba “**Deployments**”, clique nos 3 pontos do último deploy
  6. Selecione “**Redeploy**” → “**Redeploy**”
- 

## Passo 5: Inicializar Banco de Dados

### Opção A: Localmente (Mais Rápido)

```
cd /home/ubuntu/ats_platform/nextjs_space

# Aplicar schema
yarn prisma db push

# Popular dados iniciais
yarn tsx scripts/seed.ts
```

### Opção B: Via API de Manutenção

```
# Aplicar schema
curl -X POST https://sua-url.vercel.app/api/maintenance/execute \
-H "Authorization: Bearer SEU_MAINTENANCE_SECRET" \
-H "Content-Type: application/json" \
-d '{"action": "prisma_push"}'

# Popular dados
curl -X POST https://sua-url.vercel.app/api/maintenance/execute \
-H "Authorization: Bearer SEU_MAINTENANCE_SECRET" \
-H "Content-Type: application/json" \
-d '{"action": "run_seed"}'
```

## Passo 6: Verificação Final

### 6.1 Testar a Aplicação

1. Acesse sua URL: <https://sua-url.vercel.app>
2. Teste o login com:
3. Verifique:
  -  Login funciona
  -  Dashboard carrega
  -  Upload de arquivos funciona

- Notificações aparecem
- Emails são enviados

## 6.2 Verificar Logs (se necessário)

1. No dashboard da Vercel, clique em “**Logs**”
2. Veja erros em tempo real
3. Use a API de Manutenção para logs detalhados:

```
bash
curl -X GET https://sua-url.vercel.app/api/maintenance/status \
-H "Authorization: Bearer SEU_MAINTENANCE_SECRET"
```

---



## OPCIONAL: Configurar Domínio Customizado

**Se você quiser usar `www.recruitai.com.br`:**

1. No dashboard da Vercel, vá em “**Settings**” → “**Domains**”
  2. Clique em “**Add**”
  3. Digite: `www.recruitai.com.br`
  4. Siga as instruções para configurar DNS no Registro.br:
    - **Tipo:** CNAME
    - **Nome:** www
    - **Valor:** cname.vercel-dns.com
  5. Aguarde propagação (15 min - 48h)
  6. Atualize `NEXTAUTH_URL` para `https://www.recruitai.com.br`
- 



## Atualizações Futuras

### Push de Código

```
cd /home/ubuntu/ats_platform
git add .
git commit -m "Descrição da alteração"
git push origin main
```

A Vercel fará **deploy automático** após cada push!

## Alterações no Schema do Prisma

```
# Local
cd /home/ubuntu/ats_platform/nextjs_space
yarn prisma db push
git add .
git commit -m "Update database schema"
git push origin main

# OU via Maintenance API após deploy
curl -X POST https://sua-url.vercel.app/api/maintenance/execute \
-H "Authorization: Bearer SEU_MAINTENANCE_SECRET" \
-H "Content-Type: application/json" \
-d '{"action": "prisma_push"}'
```

## Problemas Comuns

### Erro: “Root directory not found”

**Solução:** Verifique se configurou `Root Directory` como `nextjs_space`

### Erro: “Cannot find module”

**Solução:** Verifique se todos os arquivos foram committedos:

```
git status
```

### Erro: “Database connection failed”

**Solução:** Verifique se `DATABASE_URL` está correta nas variáveis de ambiente da Vercel

### Erro: “NEXTAUTH\_URL is not defined”

**Solução:** Adicione `NEXTAUTH_URL` com sua URL da Vercel nas variáveis de ambiente

### Erro: “Stripe webhook signature invalid”

**Solução:** Certifique-se de que o `STRIPE_WEBHOOK_SECRET` na Vercel corresponde ao do dashboard do Stripe

## Documentação Adicional

- [Documentação Técnica Completa](#) (`./DOCUMENTACAO_TECNICA.md`)
- [API de Manutenção](#) (`./API_MANUTENCAO.md`)
- [Configuração AWS S3](#) (`./AWS_S3_CONFIG.md`)
- [Sistema de IA Multi-Modelo](#) (`./IA_MULTI_MODELO.md`)
- [Vercel Docs](#) (<https://vercel.com/docs>)

## Checklist Final

---

- [ ] Repositório criado no GitHub
  - [ ] Push bem-sucedido para GitHub
  - [ ] Projeto importado na Vercel
  - [ ] `Root Directory = nextjs_space`
  - [ ] Todas as variáveis de ambiente adicionadas
  - [ ] Deploy concluído com sucesso
  - [ ] Webhook do Stripe configurado
  - [ ] `STRIPE_WEBHOOK_SECRET` atualizado na Vercel
  - [ ] `NEXTAUTH_URL` atualizado na Vercel
  - [ ] Redeploy após atualização das variáveis
  - [ ] Schema do Prisma aplicado
  - [ ] Dados iniciais populados
  - [ ] Login testado e funcionando
  - [ ] Funcionalidades principais verificadas
- 

## Pronto!

Sua aplicação RecruitAI está no ar! 

**URL do Projeto:** [https://\[sua-url\].vercel.app](https://[sua-url].vercel.app)

### **Credenciais de Admin:**

- Email: `admin@recruitai.com`
- Senha: `admin123`

### **Credenciais de Empresa (Teste):**

- Email: `comercial.fcmtech@gmail.com`
- Senha: `fcmtech123`

### **Credenciais de Candidato (Teste):**

- Email: `candidato@teste.com`
  - Senha: `candidato123`
- 

Desenvolvido com  por FCMTech