



Guia de Deploy - Sítio Recanto da Limeira



Pré-requisitos

- Node.js 18+ instalado
- Yarn ou npm
- Banco PostgreSQL
- Conta SendGrid (gratuita)
- Domínio (opcional)



Deploy na Abacus.AI (Recomendado)

Passo 1: Preparar o projeto

```
# Clone o repositório
git clone https://github.com/seu-usuario/sitio-recanto-limeira.git
cd sitio-recanto-limeira/app

# Instale dependências
yarn install

# Teste localmente
yarn dev
```

Passo 2: Configurar variáveis de ambiente

Crie arquivo `.env` baseado no `.env.example` :

```
DATABASE_URL="sua-url-postgresql"
NEXTAUTH_URL="https://seu-dominio.com"
NEXTAUTH_SECRET="hash-muito-seguro"
SENDGRID_API_KEY="sua-chave-sendgrid"
FROM_EMAIL="seu-email@exemplo.com"
```

Passo 3: Deploy na Abacus.AI

1. Acesse <https://apps.abacus.ai/>
2. Faça upload do projeto
3. Configure as variáveis de ambiente
4. Execute build e deploy
5. Configure domínio personalizado (opcional)



Deploy no Vercel

Passo 1: Instalar Vercel CLI

```
npm install -g vercel
```

Passo 2: Deploy

```
cd app
vercel
```

Passo 3: Configurar variáveis

No dashboard da Vercel:

- Settings → Environment Variables
- Adicione todas as variáveis do `.env`



Deploy com Docker

Dockerfile

```
FROM node:18-alpine

WORKDIR /app
COPY package.json yarn.lock ./
RUN yarn install --frozen-lockfile

COPY . .
RUN yarn build

EXPOSE 3000
CMD ["yarn", "start"]
```

docker-compose.yml

```
version: '3.8'
services:
  app:
    build: .
    ports:
      - "3000:3000"
    environment:
      - DATABASE_URL=postgresql://postgres:password@db:5432/sitio
      - NEXTAUTH_URL=http://localhost:3000
      - NEXTAUTH_SECRET=seu-secret
    depends_on:
      - db
  db:
    image: postgres:15
    environment:
      - POSTGRES_DB=sitio
      - POSTGRES_USER=postgres
      - POSTGRES_PASSWORD=password
    volumes:
      - postgres_data:/var/lib/postgresql/data
volumes:
  postgres_data:
```

Executar:

```
docker-compose up -d
```



Deploy em VPS

Passo 1: Configurar servidor

```
# Ubuntu/Debian
sudo apt update
sudo apt install nodejs npm nginx postgresql

# Instalar PM2
npm install -g pm2
```

Passo 2: Configurar banco

```
sudo -u postgres psql
CREATE DATABASE sitio_recanto_limeira;
CREATE USER sitio_user WITH PASSWORD 'senha_segura';
GRANT ALL PRIVILEGES ON DATABASE sitio_recanto_limeira TO sitio_user;
\q
```

Passo 3: Deploy da aplicação

```
# Clone e configure
git clone https://github.com/seu-usuario/sitio-recanto-limeira.git
cd sitio-recanto-limeira/app
npm install
cp .env.example .env
# Configure o .env com dados reais

# Build e migrate
npm run build
npx prisma migrate deploy
npx prisma db seed

# Iniciar com PM2
pm2 start npm --name "sitio" -- start
pm2 startup
pm2 save
```

Passo 4: Configurar Nginx

```
server {  
    listen 80;  
    server_name seu-dominio.com;  
  
    location / {  
        proxy_pass http://localhost:3000;  
        proxy_http_version 1.1;  
        proxy_set_header Upgrade $http_upgrade;  
        proxy_set_header Connection 'upgrade';  
        proxy_set_header Host $host;  
        proxy_cache_bypass $http_upgrade;  
    }  
}
```



SSL com Let's Encrypt

```
sudo apt install certbot python3-certbot-nginx  
sudo certbot --nginx -d seu-dominio.com
```



Monitoramento

PM2 Monitoring

```
pm2 monit  
pm2 logs sitio
```

Health Check

Configure endpoint `/api/health` para monitoramento.

Deploy Automático (GitHub Actions)

.github/workflows/deploy.yml

```
name: Deploy to Production

on:
  push:
    branches: [main]

jobs:
  deploy:
    runs-on: ubuntu-latest

    steps:
      - uses: actions/checkout@v3

      - name: Setup Node.js
        uses: actions/setup-node@v3
        with:
          node-version: 18

      - name: Install dependencies
        run: cd app && npm install

      - name: Run tests
        run: cd app && npm test

      - name: Build
        run: cd app && npm run build

      - name: Deploy to Vercel
        uses: amondnet/vercel-action@v20
        with:
          vercel-token: ${ secrets.VERCEL_TOKEN }
          vercel-org-id: ${ secrets.ORG_ID }
          vercel-project-id: ${ secrets.PROJECT_ID }
          working-directory: ./app
```

Checklist Pós-Deploy

- ☐ Site carregando corretamente
- ☐ Formulário de reservas funcionando
- ☐ Painel admin acessível
- ☐ Emails sendo enviados
- ☐ SSL configurado
- ☐ Backup automático configurado
- ☐ Monitoramento ativo
- ☐ Domínio apontando corretamente

Troubleshooting

Erro de build:

```
# Limpar cache
rm -rf .next node_modules
npm install
npm run build
```

Erro de banco:

```
# Verificar conexão
npx prisma db pull
npx prisma migrate reset
```

Erro de permissões:

```
# PM2
pm2 delete all
pm2 start npm --name "sitio" -- start
```

 Seu site estará no ar e funcionando perfeitamente!