



Guia de Deploy do Backend PricePro

Problema

O backend FastAPI está configurado para rodar em `localhost:8000`, o que funciona apenas para acesso local (via VNC). Para acessar a aplicação via URL de preview remota (`https://xxxxx.preview.abacusai.app`), o backend precisa estar publicamente acessível.

Soluções

Opção 1: Deploy em Railway (Recomendado - GRATUITO)

Railway oferece deploy gratuito com 5\$ de crédito mensal (suficiente para projetos pequenos).

Passo a Passo:

1. Criar conta no Railway:

- Acesse: <https://railway.app>
- Faça login com GitHub

2. Preparar o projeto:

```
bash
cd /home/ubuntu/pricepro_backend
```

3. Criar `requirements.txt`:

```
bash
pip freeze > requirements.txt
```

4. Criar `railway.json`:

```
json
{
  "$schema": "https://railway.app/railway.schema.json",
  "build": {
    "builder": "NIXPACKS"
  },
  "deploy": {
    "startCommand": "python main.py",
    "restartPolicyType": "ON_FAILURE",
    "restartPolicyMaxRetries": 10
  }
}
```

5. Deploy:

- No Railway, clique em “New Project”
- Selecione “Deploy from GitHub repo”
- Ou use Railway CLI:

```
bash
npm install -g @railway/cli
railway login
```

```
railway init
railway up
```

6. Configurar variáveis de ambiente no Railway:

- DATABASE_URL (Railway fornece PostgreSQL gratuito)
- SECRET_KEY
- REDIS_URL (Railway fornece Redis gratuito)

7. Obter a URL pública:

- Railway gerará uma URL como: `https://pricepro-backend-production.up.railway.app`

8. Atualizar frontend:

```
bash
```

```
# No arquivo .env do frontend:
NEXT_PUBLIC_API_URL=https://pricepro-backend-production.up.railway.app
```

Opção 2: Deploy em Render (Alternativa GRATUITA)

Render oferece tier gratuito com algumas limitações (sleep após inatividade).

Passo a Passo:

1. Criar conta no Render:

- Acesse: <https://render.com>
- Faça login com GitHub

2. Criar Web Service:

- Clique em “New” → “Web Service”
- Conecte ao repositório do backend

3. Configurar:

- **Build Command:** `pip install -r requirements.txt`
- **Start Command:** `python main.py`
- **Environment:** Python 3

4. Adicionar PostgreSQL:

- Em “Dashboard”, clique em “New” → “PostgreSQL”
- Copie a `DATABASE_URL` interna

5. Configurar variáveis de ambiente:

- DATABASE_URL
- SECRET_KEY
- REDIS_URL (use Upstash Redis gratuito)

6. Obter URL:

- Render gerará: `https://pricepro-backend.onrender.com`

7. Atualizar frontend:

```
bash
```

```
NEXT_PUBLIC_API_URL=https://pricepro-backend.onrender.com
```

Opção 3: Túnel Local (Desenvolvimento Rápido)

Use **ngrok** ou **Cloudflare Tunnel** para expor o backend local publicamente.

3.1 - Usando ngrok:

1. Instalar ngrok:

```
bash
# Ubuntu/Debian
curl -s https://ngrok-agent.s3.amazonaws.com/ngrok.asc | sudo tee /etc/apt/trusted.gpg.d/ngrok.asc >/dev/null
echo "deb https://ngrok-agent.s3.amazonaws.com buster main" | sudo tee /etc/apt/sources.list.d/ngrok.list
sudo apt update && sudo apt install ngrok
```

2. Criar conta gratuita:

- Acesse: <https://dashboard.ngrok.com/signup>
- Copie o authtoken

3. Autenticar:

```
bash
ngrok authtoken YOUR_AUTH_TOKEN
```

4. Iniciar túnel:

```
bash
# Em um terminal separado
ngrok http 8000
```

5. Copiar a URL pública:

- ngrok mostrará algo como: <https://abcd1234.ngrok.io>

6. Atualizar frontend:

```
bash
NEXT_PUBLIC_API_URL=https://abcd1234.ngrok.io
```

3.2 - Usando Cloudflare Tunnel:

1. Instalar cloudflared:

```
bash
wget https://github.com.cloudflare/cloudflared/releases/latest/download/cloudflared-
linux-amd64.deb
sudo dpkg -i cloudflared-linux-amd64.deb
```

2. Criar túnel:

```
bash
cloudflared tunnel --url http://localhost:8000
```

3. Copiar a URL exibida no terminal

4. Atualizar frontend:

```
bash
NEXT_PUBLIC_API_URL=https://xxxxx.trycloudflare.com
```

Opção 4: Heroku (Pago após Nov 2022)

Heroku removeu o tier gratuito, mas ainda é uma opção paga confiável.



Checklist Final

Depois de configurar o backend público:

1.  Backend está rodando em 0.0.0.0:8000
2.  CORS configurado para aceitar *.preview.abacusai.app
3.  Variável NEXT_PUBLIC_API_URL configurada no frontend
4.  Frontend reconstruído:

```
bash
cd /home/ubuntu/pricopro_frontend/nextjs_space
yarn build
```

5.  Testar acesso via URL de preview
-



Segurança

Produção:

- Use HTTPS sempre
- Configure CORS apenas para origens específicas
- Use variáveis de ambiente para secrets
- Habilite rate limiting
- Configure autenticação adequadamente

Desenvolvimento:

- Túneis (ngrok/cloudflare) são seguros para testes
 - Não exponha credenciais em repositórios públicos
-



Suporte

Se encontrar problemas:

1. Verifique logs do backend
 2. Verifique console do navegador
 3. Teste com curl direto no backend
 4. Verifique configurações de CORS
-



Comparação das Opções

Opção	Custo	Facilidade	Recomendado Para
Railway	Gratuito (5\$/mês)	★★★★★	Produção
Render	Gratuito (sleep)	★★★★★	Produção
ngrok	Gratuito (limitado)	★★★★★	Desenvolvimento
Cloudflare	Gratuito	★★★★★	Desenvolvimento
Heroku	Pago (\$7+/mês)	★★★★★	Produção

Recomendação: Use **Railway** para produção e **ngrok** para testes rápidos.