



# Railway Deploy - Guia Rápido

---



## O Que é Railway?

---

Railway é uma plataforma de deploy moderna que oferece:

- ☒ **\$5 grátis por mês** (suficiente para PricePro)
  - ☒ **PostgreSQL gratuito** (500MB)
  - ☒ **Redis gratuito** (opcional)
  - ☒ **Deploy automático** via Git
  - ☒ **URL permanente** (HTTPS)
  - ☒ **Zero configuração** de servidor
- 



## Deploy em 3 Passos

---

### Passo 1: Criar Conta

1. Acesse: <https://railway.app>
2. Clique em “Start a New Project”
3. Faça login com GitHub

### Passo 2: Deploy Automático

#### Opção A: Via Railway CLI (Recomendado)

```
cd /home/ubuntu/pricepro_backend
chmod +x railway_deploy.sh
./railway_deploy.sh
```

O script fará TUDO automaticamente:

- ☒ Instalar Railway CLI
- ☒ Fazer login
- ☒ Criar projeto
- ☒ Provisionar PostgreSQL
- ☒ Configurar variáveis de ambiente
- ☒ Fazer deploy
- ☒ Gerar URL pública

#### Opção B: Via Dashboard (Manual)

1. **Criar Novo Projeto:**
  - No Railway Dashboard, clique em “New Project”
  - Selecione “Deploy from GitHub repo”
  - Conecte seu repositório do backend
2. **Adicionar PostgreSQL:**
  - Clique em “New” → “Database” → “Add PostgreSQL”
  - Railway configurará `DATABASE_URL` automaticamente

### 3. Configurar Variáveis:

- Clique no serviço
- Vá em "Variables"
- Adicione:

```
SECRET_KEY=seu_secret_key_aqui
APP_NAME=PricePro
DEBUG=False
```

### 4. Deploy:

- Railway detecta automaticamente Python
- Lê `requirements.txt` e `railway.json`
- Faz build e deploy

## Passo 3: Obter URL e Atualizar Frontend

```
# Obter URL pública
cd /home/ubuntu/pricepro_backend
railway domain

# Exemplo de saída:
# https://pricepro-backend-production.up.railway.app
```

Atualize o frontend:

```
# Edite o .env
cd /home/ubuntu/pricepro_frontend/nextjs_space
nano .env
```

Adicione/Atualize:

```
NEXT_PUBLIC_API_URL=https://pricepro-backend-production.up.railway.app
```

Rebuild:

```
yarn build
```

✅ **PRONTO! Tudo funcionando!**



## Monitoramento

### Ver Logs:

```
railway logs
```

### Abrir Dashboard:

```
railway open
```




## Ver Variáveis:

```
railway variables
```

---

## Recursos do Railway

### PostgreSQL (Incluído)

-  500MB de armazenamento
-  DATABASE\_URL configurada automaticamente
-  Backups automáticos

### Redis (Opcional)

```
# Adicionar Redis
railway add redis
```

Railway configurará REDIS\_URL automaticamente.

**NOTA:** PricePro funciona **sem Redis** (usa cache em memória).

---

## Custos

### Tier Gratuito:

- \$5 de crédito por mês
- Renovado automaticamente
- Suficiente para:
- Backend FastAPI
- PostgreSQL (500MB)
- Testes e desenvolvimento

### Se Exceder:

- Cobra apenas o que usar (\$0.000231/GB-hora)
- Pode definir limites de gasto

---

## Segurança

### Variáveis de Ambiente:

- Nunca commite `.env` no Git
- Use o dashboard ou CLI para configurá-las
- Railway criptografa automaticamente

## SECRET\_KEY:

```
# Gerar SECRET_KEY seguro:  
openssl rand -hex 32
```

## HTTPS:

- Railway fornece HTTPS automaticamente
- Certificados gerenciados automaticamente



## Deploy Contínuo

### Auto-Deploy via Git:

1. Conecte ao GitHub:

```
bash
```

```
railway link
```

2. Todo push na branch `main` fará deploy automaticamente!

## ✖ Troubleshooting

### Erro: “Build Failed”

**Causa:** Dependências faltando

**Solução:**

```
pip freeze > requirements.txt  
git add requirements.txt  
git commit -m "Update requirements"  
git push
```

### Erro: “Database connection failed”

**Causa:** `DATABASE_URL` não configurada

**Solução:**

1. Adicione PostgreSQL no dashboard
2. Railway configurará automaticamente

### Erro: “Port already in use”

**Causa:** Railway usa `$PORT` dinâmico

**Solução:** Já configurado em `railway.json` :

```
"startCommand": "uvicorn main:app --host 0.0.0.0 --port $PORT"
```

## Recursos Adicionais

---

- **Documentação Railway:** <https://docs.railway.app>
  - **Status:** <https://railway.statuspage.io>
  - **Suporte:** <https://help.railway.app>
- 

## Checklist Final

---

- ☐ Conta Railway criada
  - ☐ Backend deployado
  - ☐ PostgreSQL provisionado
  - ☐ Variáveis de ambiente configuradas
  - ☐ URL pública obtida
  - ☐ Frontend atualizado com nova URL
  - ☐ Frontend rebuilt
  - ☐ Testado via preview URL
- 

## Próximos Passos

---

Depois do deploy:

1. ☒ Backend sempre disponível (24/7)
2. ☒ Sem necessidade de rodar scripts localmente
3. ☒ Frontend funciona via preview URL
4. ☒ HTTPS automático
5. ☒ Escalabilidade automática

**Você só precisa rodar o frontend!** 🚀