



# Guia Completo: Deploy no Render.com

Este guia vai te ensinar passo a passo como colocar o DonShop007 no ar usando o Render.com (hospedagem gratuita).



## Pré-requisitos

- Código publicado no GitHub (veja `GUIA_GITHUB.md`)
- Repositório público no GitHub
- URL do repositório anotada
- 20-30 minutos de tempo

## 🎯 Passo 1: Criar Conta no Render.com

### 1.1 Acessar o Render

1. Abra seu navegador
2. Acesse: <https://render.com>
3. Clique no botão “Get Started” ou “Sign Up”

### 1.2 Cadastrar com GitHub (RECOMENDADO)

1. Clique em “Sign up with GitHub”
2. Faça login no GitHub se solicitado
3. Autorize o Render a acessar sua conta
4. Preencha informações adicionais se solicitado

OU

### 1.3 Cadastrar com Email

1. Clique em “Sign up with Email”
2. Preencha:
  - **Email:** Seu melhor email
  - **Password:** Senha forte
  - **Name:** Seu nome
3. Clique em “Sign Up”
4. Verifique seu email e clique no link de confirmação

### 1.4 Completar Perfil

1. Escolha um nome para sua equipe (pode ser seu nome)
2. Clique em “Continue”
3. Você será levado ao Dashboard do Render

Conta criada com sucesso!

## Passo 2: Conectar Render ao GitHub

### 2.1 Conectar Conta GitHub (se não fez no cadastro)

1. No Dashboard do Render, clique no seu avatar (canto superior direito)
2. Clique em “**Account Settings**”
3. Na aba “**GitHub**”, clique em “**Connect GitHub Account**”
4. Autorize o Render a acessar seus repositórios
5. Selecione “**All repositories**” ou apenas o `donshop007-ecommerce`

 **GitHub conectado!**

## Passo 3: Criar Banco de Dados PostgreSQL

**IMPORTANTE:** Crie o banco ANTES do Web Service!

### 3.1 Criar Novo PostgreSQL

1. No Dashboard do Render, clique em “**New +**” (canto superior direito)
2. Selecione “**PostgreSQL**”

### 3.2 Configurar o Banco de Dados

Preencha os campos:

Campo	Valor	Descrição
<b>Name</b>	<code>donshop007-db</code>	Nome do banco
<b>Database</b>	<code>donshop007</code>	Nome do database
<b>User</b>	<code>donshop007</code>	Usuário (gerado automaticamente)
<b>Region</b>	Oregon (US West)	Escolha a região mais próxima
<b>PostgreSQL Version</b>	16	Versão mais recente
<b>Instance Type</b>	<b>Free</b>	Plano gratuito

### 3.3 Criar o Banco

1. Role até o final da página
2. Clique no botão azul “**Create Database**”
3. Aguarde 2-3 minutos enquanto o banco é provisionado

### 3.4 Copiar URL de Conexão

Quando o banco estiver pronto (status “Available”):

1. Na página do banco, procure a seção “**Connections**”
2. Você verá várias URLs. Procure por “**Internal Database URL**”

3. Clique no ícone de **copiar** ao lado da URL

A URL será parecida com:

```
postgresql://donshop007:SENHA_GERADA@dpq-xxxxx.oregon-postgres.render.com/donshop007
```

**⚠ IMPORTANTE:** Copie essa URL e cole em um bloco de notas. Vamos usar ela no próximo passo!

**✓ Banco de dados criado!**

## 🎯 Passo 4: Criar Web Service (Aplicação)

### 4.1 Criar Novo Web Service

1. Volte ao Dashboard do Render
2. Clique em “**New +**” (canto superior direito)
3. Selecione “**Web Service**”

### 4.2 Conectar Repositório GitHub

Você verá uma lista dos seus repositórios:

1. Procure por **donshop007-ecommerce**
2. Clique no botão “**Connect**” ao lado dele

**Se não aparecer o repositório:**

- Clique em “**Configure account**”
- Autorize o Render a acessar o repositório específico
- Volte e tente novamente

### 4.3 Configurar o Web Service

Preencha os campos conforme abaixo:

Campo	Valor	Descrição
<b>Name</b>	donshop007	Nome único (será parte da URL)
<b>Region</b>	Oregon (US West)	Mesma região do banco
<b>Branch</b>	main	Branch principal do Git
<b>Root Directory</b>	(deixe vazio)	Raiz do projeto
<b>Runtime</b>	Python 3	Ambiente Python
<b>Build Command</b>	pip install -r requirements.txt	Comando de instalação
<b>Start Command</b>	gunicorn app:app	Comando para iniciar
<b>Instance Type</b>	Free	Plano gratuito

## 4.4 Adicionar Variáveis de Ambiente

Role até a seção “Environment Variables” e clique em “Add Environment Variable”.

Adicione as seguintes variáveis **UMA POR UMA**:

### Variável 1: SECRET\_KEY

- **Key:** SECRET\_KEY
- **Value:** sua-chave-secreta-super-segura-aqui-123456789

(Você pode gerar uma chave aleatória em: <https://randomkeygen.com/>)

### Variável 2: DATABASE\_URL

- **Key:** DATABASE\_URL
- **Value:** (Cole aqui a URL do banco que você copiou no Passo 3.4)

Exemplo:

```
postgresql://donshop007:SENHA@pg-xxxxxx.oregon-postgres.render.com/donshop007
```

### Variável 3: FLASK\_ENV

- **Key:** FLASK\_ENV
- **Value:** production

### Variável 4: PYTHON\_VERSION (opcional)

- **Key:** PYTHON\_VERSION
- **Value:** 3.11.0

**Resumo das variáveis:**

```
SECRET_KEY = sua-chave-secreta-super-segura-aqui-123456789
DATABASE_URL = postgresql://donshop007:SENHA@dpg-xxxxx.oregon-postgres.render.com/donshop007
FLASK_ENV = production
PYTHON_VERSION = 3.11.0
```

## 4.5 Criar o Web Service

1. Role até o final da página
  2. Clique no botão azul “**Create Web Service**”
  3. O Render vai começar o deploy automaticamente
- 

## 🎯 Passo 5: Acompanhar o Deploy

### 5.1 Visualizar Logs

Você será redirecionado para a página do serviço com os logs em tempo real.

#### O que vai acontecer:

1. ⏱️ **Building...** - Instalando dependências (2-5 minutos)
2. ⏱️ **Starting...** - Iniciando aplicação (30 segundos)
3.  **Live** - Aplicação no ar!

### 5.2 Possíveis Erros e Soluções

#### Erro: “Failed to install requirements”

**Causa:** Problema no `requirements.txt`

#### Solução:

- Verifique se o arquivo existe no repositório
- Certifique-se de que está na raiz do projeto

#### Erro: “Application failed to start”

**Causa:** Problema no `Procfile` ou comando de start

#### Solução:

- Verifique se o `Procfile` contém: `web: gunicorn app:app`
- Verifique se o arquivo `app.py` existe

#### Erro: “Database connection failed”

**Causa:** URL do banco incorreta

#### Solução:

- Vá em “**Environment**” no menu lateral
- Verifique se `DATABASE_URL` está correta
- Copie novamente do banco de dados

### 5.3 Aguardar Deploy Completo

Quando você ver “**Live**” em verde no topo da página, o site está no ar! 🎉

---

## Passo 6: Popular o Banco de Dados

**IMPORTANTE:** O banco está vazio! Precisamos criar as tabelas e adicionar produtos.

### 6.1 Acessar Shell do Render

1. Na página do seu Web Service, clique em “**Shell**” no menu lateral esquerdo
2. Clique no botão “**Launch Shell**”
3. Uma janela de terminal vai abrir

### 6.2 Executar Script de Inicialização

No terminal que abriu, digite:

```
python init_db.py
```

Pressione **Enter** e aguarde.

#### O que vai acontecer:

- Criação das tabelas no banco
- Inserção de produtos de exemplo
- Criação de categorias
- Mensagens de sucesso

Você deve ver algo como:

-  Banco de dados inicializado com sucesso!
-  12 produtos adicionados
-  4 categorias criadas

### 6.3 Fechar Shell

Digite `exit` e pressione Enter, ou simplesmente feche a janela.

 **Banco populado com sucesso!**

## Passo 7: Testar o Site

### 7.1 Acessar URL do Site

1. Na página do Web Service, procure pela URL no topo
2. Será algo como: `https://donshop007.onrender.com`
3. Clique na URL ou copie e cole no navegador

### 7.2 Testar Funcionalidades

#### Página Inicial

-  Deve carregar a página inicial
-  Deve mostrar produtos
-  Deve ter menu de navegação

#### Cadastro de Usuário

1. Clique em “**Cadastrar**”

2. Preencha o formulário
3. Clique em “**Cadastrar**”
4.  Deve criar conta e fazer login

## **Login**

1. Clique em “**Entrar**”
2. Use as credenciais que você criou
3.  Deve fazer login com sucesso

## **Adicionar ao Carrinho**

1. Clique em um produto
2. Clique em “**Adicionar ao Carrinho**”
3.  Deve adicionar e mostrar notificação

## **Carrinho**

1. Clique no ícone do carrinho
2.  Deve mostrar produtos adicionados
3.  Deve calcular total corretamente

## **Finalizar Compra**

1. No carrinho, clique em “**Finalizar Compra**”
2. Preencha dados de entrega
3.  Deve criar pedido com sucesso

## **7.3 Verificar Responsividade**

Teste o site em diferentes dispositivos:

-  Celular
  -  Desktop
  -  Tablet
- 

## **Passo 8: Configurações Adicionais (Opcional)**

### **8.1 Configurar Domínio Personalizado**

Se você tem um domínio próprio:

1. Na página do Web Service, clique em “**Settings**”
2. Role até “**Custom Domain**”
3. Clique em “**Add Custom Domain**”
4. Siga as instruções para configurar DNS

### **8.2 Configurar Auto-Deploy**

O Render já configura auto-deploy por padrão:

- Toda vez que você fizer `git push` no GitHub
- O Render vai fazer deploy automaticamente
- Você pode desabilitar isso em “**Settings**” → “**Auto-Deploy**”

### **8.3 Monitorar Uso**

1. Clique em “**Metrics**” no menu lateral

2. Veja estatísticas de:

- Requisições por minuto
- Tempo de resposta
- Uso de memória
- Uso de CPU

## 8.4 Ver Logs

1. Clique em “**Logs**” no menu lateral
  2. Veja logs em tempo real da aplicação
  3. Útil para debugar problemas
- 

## 🎯 Passo 9: Anotar Informações Importantes

Anote as seguintes informações em um lugar seguro:

### URLs

- **Site:** `https://donshop007.onrender.com` (ou sua URL)
- **Repositório GitHub:** `https://github.com/SEU-USERNAME/donshop007-ecommerce`
- **Dashboard Render:** `https://dashboard.render.com`

### Credenciais do Banco

- **Host:** `dpg-xxxxx.oregon-postgres.render.com`
- **Database:** `donshop007`
- **User:** `donshop007`
- **Password:** (veja no Render)
- **Internal URL:** `postgresql://...`

### Variáveis de Ambiente

- **SECRET\_KEY:** (a que você configurou)
  - **DATABASE\_URL:** (URL do banco)
  - **FLASK\_ENV:** `production`
- 

## ⚠️ Limitações do Plano Gratuito

### Render Free Tier

- **✓ 750 horas/mês** de uso
- **✓ 512 MB RAM**
- **✓ 0.1 CPU**
- **⚠️ Inatividade:** Site “dorme” após 15 minutos sem uso
- **⚠️ Primeiro acesso:** Pode levar 30-60 segundos para “acordar”
- **⚠️ Builds:** Máximo 500 horas/mês

### PostgreSQL Free Tier

- **✓ 1 GB de armazenamento**
- **✓ Backups automáticos** (7 dias)

- **⚠️ Expira em 90 dias** (você pode renovar gratuitamente)
- **⚠️ Conexões limitadas**

## Como Evitar que o Site Durma

Use um serviço de ping gratuito:

- **UptimeRobot**: <https://uptimerobot.com>
- **Cron-job.org**: <https://cron-job.org>

Configure para fazer ping no seu site a cada 10 minutos.

---

## Como Atualizar o Site

Quando você fizer alterações no código:

### Método 1: Auto-Deploy (Recomendado)

```
# 1. Fazer alterações no código local
# 2. Commit e push para GitHub
git add .
git commit -m "Descrição das alterações"
git push origin main

# 3. Render detecta e faz deploy automaticamente!
```

### Método 2: Deploy Manual

1. Acesse o Dashboard do Render
  2. Clique no seu Web Service
  3. Clique em “Manual Deploy” no canto superior direito
  4. Selecione “Deploy latest commit”
- 

## ? Problemas Comuns e Soluções

### Site não carrega (Error 503)

**Causa:** Aplicação não iniciou corretamente

**Solução:**

1. Verifique os logs em “Logs”
2. Procure por erros de Python
3. Verifique se `DATABASE_URL` está correta

### Banco de dados vazio

**Causa:** Não executou `init_db.py`

**Solução:**

1. Abra o Shell do Render
2. Execute: `python init_db.py`

## Imagens não aparecem

**Causa:** Render não persiste arquivos de upload

**Solução:**

- Use serviço externo como Cloudinary ou AWS S3
- Ou use URLs de imagens externas

## Site muito lento

**Causa:** Plano gratuito tem recursos limitados

**Solução:**

- Otimize queries do banco
- Use cache
- Considere upgrade para plano pago (\$7/mês)

## Deploy falha

**Causa:** Erro no código ou dependências

**Solução:**

1. Verifique logs de build
  2. Teste localmente antes de fazer push
  3. Verifique `requirements.txt`
- 



## Monitoramento e Manutenção

### Verificações Semanais

- [ ] Site está acessível
- [ ] Produtos carregando corretamente
- [ ] Cadastro e login funcionando
- [ ] Carrinho funcionando
- [ ] Checkout funcionando

### Verificações Mensais

- [ ] Banco de dados não está cheio (< 1 GB)
- [ ] Renovar banco gratuito (a cada 90 dias)
- [ ] Verificar logs de erro
- [ ] Atualizar dependências se necessário

## Backups

O Render faz backup automático do banco, mas é bom ter seu próprio:

1. No Dashboard do banco, clique em “**Backups**”
  2. Clique em “**Create Backup**”
  3. Baixe o backup periodicamente
-

## Checklist Final

---

Antes de considerar o deploy completo:

- [ ] Conta no Render criada
  - [ ] GitHub conectado ao Render
  - [ ] Banco PostgreSQL criado e ativo
  - [ ] Web Service criado e ativo
  - [ ] Variáveis de ambiente configuradas
  - [ ] Deploy concluído com sucesso (status “Live”)
  - [ ] `init_db.py` executado
  - [ ] Site acessível via URL
  - [ ] Cadastro de usuário funcionando
  - [ ] Login funcionando
  - [ ] Produtos aparecendo
  - [ ] Carrinho funcionando
  - [ ] Checkout funcionando
  - [ ] URLs anotadas em lugar seguro
- 

## Parabéns!

Seu e-commerce está no ar! 

### URLs importantes:

-  **Seu site:** <https://donshop007.onrender.com>
  -  **GitHub:** <https://github.com/SEU-USERNAME/donshop007-ecommerce>
  -  **Render Dashboard:** <https://dashboard.render.com>
- 

## Suporte e Recursos

### Documentação Oficial

- **Render Docs:** <https://render.com/docs>
- **Flask Docs:** <https://flask.palletsprojects.com>
- **PostgreSQL Docs:** <https://www.postgresql.org/docs>

### Comunidade

- **Render Community:** <https://community.render.com>
- **Stack Overflow:** <https://stackoverflow.com/questions/tagged/render>

### Supporte Render

- **Email:** [support@render.com](mailto:support@render.com)
  - **Status:** <https://status.render.com>
-



## Próximos Passos

---

Agora que seu site está no ar, você pode:

1. **Personalizar o design** - Edite CSS e templates
  2. **Adicionar mais produtos** - Via admin ou banco de dados
  3. **Integrar pagamento real** - Stripe, PayPal, Mercado Pago
  4. **Adicionar email** - SendGrid, Mailgun
  5. **Melhorar SEO** - Meta tags, sitemap
  6. **Analytics** - Google Analytics
  7. **Domínio próprio** - Compre e configure
  8. **Certificado SSL** - Render fornece automaticamente
- 

**Criado para o projeto DonShop007 | E-commerce com Flask + PostgreSQL**