



# Checklist de Deploy - DonShop007

---

Use este checklist para acompanhar seu progresso no deploy do e-commerce.

---



## Progresso Geral

---

**Data de Início:** //\_

**Data de Conclusão:** \_//

**Tempo Total:** \_\_\_\_ minutos

---



## FASE 1: Preparação do Código

---

### Verificação Inicial

- ☐ Código do projeto está completo em `/home/ubuntu/donshop007/`
- ☐ Todos os arquivos necessários estão presentes
- ☐ `.gitignore` está configurado corretamente
- ☐ `requirements.txt` está completo
- ☐ `Procfile` existe e está correto
- ☐ `.env.example` foi revisado

### Arquivos Críticos

- ☐ `app.py` - Aplicação principal
  - ☐ `models.py` - Modelos do banco
  - ☐ `init_db.py` - Script de inicialização
  - ☐ `config.py` - Configurações
  - ☐ Pasta `routes/` com todos os blueprints
  - ☐ Pasta `templates/` com todos os HTMLs
  - ☐ Pasta `static/` com CSS, JS e imagens
- 



## FASE 2: GitHub (10-15 minutos)

---

### 2.1 Criar Conta GitHub

- ☐ Acessei <https://github.com>
- ☐ Criei conta (ou já tenho uma)
- ☐ Verifiquei meu email
- ☐ Fiz login com sucesso

**Meu username GitHub:** \_\_\_\_

### 2.2 Criar Repositório

- ☐ Cliquei em "New repository"

- ☐ Nome do repositório: `donshop007-ecommerce`
- ☐ Descrição adicionada
- ☐ Repositório configurado como **PUBLIC** ☒
- ☐ NÃO marquei "Initialize with README"
- ☐ Repositório criado com sucesso

**URL do Repositório:** \_\_\_\_\_

## 2.3 Configurar Git Local

- ☐ Abri o terminal
- ☐ Executei `git config --global user.name "Meu Nome"`
- ☐ Executei `git config --global user.email "meu@email.com"`
- ☐ Verifiquei configuração com `git config --global --list`

## 2.4 Conectar e Enviar Código

- ☐ Naveguei até `/home/ubuntu/donshop007/`
- ☐ Executei `git init` (se necessário)
- ☐ Executei `git remote add origin [URL-DO-REPOSITARIO]`
- ☐ Executei `git add .`
- ☐ Executei `git commit -m "Primeiro commit: E-commerce DonShop007 completo"`
- ☐ Executei `git branch -M main`
- ☐ Executei `git push -u origin main`
- ☐ Push concluído com sucesso ☒

## 2.5 Verificação GitHub

- ☐ Atualizei página do repositório no navegador
- ☐ Todos os arquivos estão visíveis
- ☐ Repositório está marcado como **PUBLIC**
- ☐ README.md está sendo exibido

☒ **FASE 2 CONCLUÍDA!** Código publicado no GitHub.

# **FASE 3: Render.com - Banco de Dados (5-10 minutos)**

## 3.1 Criar Conta Render

- ☐ Acessei `https://render.com`
- ☐ Criei conta (ou fiz login)
- ☐ Usei "Sign up with GitHub" (recomendado)
- ☐ Autorizei Render a acessar GitHub
- ☐ Completei perfil
- ☐ Estou no Dashboard do Render

## 3.2 Conectar GitHub ao Render

- ☐ Conectei minha conta GitHub ao Render
- ☐ Autorizei acesso aos repositórios
- ☐ Repositório `donshop007-ecommerce` está visível

### 3.3 Criar Banco PostgreSQL

- [ ] Cliquei em “New +” → “PostgreSQL”
- [ ] Nome: `donshop007-db`
- [ ] Database: `donshop007`
- [ ] Região: `Oregon (US West)` (ou mais próxima)
- [ ] PostgreSQL Version: `16`
- [ ] Instance Type: **Free** ✓
- [ ] Cliquei em “Create Database”
- [ ] Aguardei provisionamento (2-3 minutos)
- [ ] Status mudou para “Available” ✓

### 3.4 Copiar URL do Banco

- [ ] Acessei página do banco criado
- [ ] Encontrei seção “Connections”
- [ ] Copiei “**Internal Database URL**”
- [ ] Colei URL em lugar seguro (bloco de notas)

**Internal Database URL:**

```
postgresql://donshop007:SENHA@dpg-xxxxx.oregon-postgres.render.com/donshop007
```

✓ **FASE 3 CONCLUÍDA!** Banco de dados criado.

## ● FASE 4: Render.com - Web Service (10-15 minutos)

### 4.1 Criar Web Service

- [ ] Voltei ao Dashboard do Render
- [ ] Cliquei em “New +” → “Web Service”
- [ ] Encontrei repositório `donshop007-ecommerce`
- [ ] Cliquei em “Connect”

### 4.2 Configurar Web Service

- [ ] **Name:** `donshop007` (ou nome único)
- [ ] **Region:** `Oregon (US West)` (mesma do banco)
- [ ] **Branch:** `main`
- [ ] **Root Directory:** (deixei vazio)
- [ ] **Runtime:** `Python 3`
- [ ] **Build Command:** `pip install -r requirements.txt`
- [ ] **Start Command:** `gunicorn app:app`
- [ ] **Instance Type:** **Free** ✓


### 4.3 Adicionar Variáveis de Ambiente

- [ ] Rolei até “Environment Variables”
- [ ] Adicionei variável 1:
- Key: `SECRET_KEY`

- Value: \_\_\_\_\_
- [ ] Adicionei variável 2:
- Key: `DATABASE_URL`
- Value: (colei Internal Database URL)
- [ ] Adicionei variável 3:
- Key: `FLASK_ENV`
- Value: `production`
- [ ] Adicionei variável 4 (opcional):
- Key: `PYTHON_VERSION`
- Value: `3.11.0`

**Minha SECRET\_KEY:** \_\_\_\_\_ (guarde em segredo!)

## 4.4 Criar e Aguardar Deploy

- [ ] Cliquei em “Create Web Service”
- [ ] Fui redirecionado para página de logs
- [ ] Aguardei build (2-5 minutos)
- [ ] Status mudou para “Live” 

**URL do Site:** \_\_\_\_\_




 **FASE 4 CONCLUÍDA!** Site está no ar (mas banco vazio).

## FASE 5: Popular Banco de Dados (5 minutos)

### 5.1 Acessar Shell do Render

- [ ] Na página do Web Service, cliquei em “Shell” (menu lateral)
- [ ] Cliquei em “Launch Shell”
- [ ] Terminal abriu com sucesso

### 5.2 Executar Script de Inicialização


- [ ] Digitei: `python init_db.py`
- [ ] Pressionei Enter
- [ ] Aguardei execução
- [ ] Vi mensagens de sucesso:
- [ ] “ Banco de dados inicializado com sucesso!”
- [ ] “ 12 produtos adicionados”
- [ ] “ 4 categorias criadas”
- [ ] Digitei `exit` para fechar shell

 **FASE 5 CONCLUÍDA!** Banco populado com produtos.

## FASE 6: Testes Finais (5-10 minutos)

### 6.1 Acessar Site

- [ ] Copiei URL do site: `https://donshop007.onrender.com`

- ☐ Abri em navegador
- ☐ Site carregou com sucesso 

## 6.2 Testar Página Inicial

- ☐ Logo aparece corretamente
- ☐ Menu de navegação funciona
- ☐ Produtos estão sendo exibidos
- ☐ Imagens dos produtos carregam
- ☐ Design está correto (cores, fontes)

## 6.3 Testar Cadastro

- ☐ Cliquei em “Cadastrar”
- ☐ Preenchi formulário de cadastro
- ☐ Cadastro foi criado com sucesso
- ☐ Fui redirecionado e logado automaticamente

### Credenciais de Teste Criadas:

- Email: \_\_\_\_
- Senha: \_\_\_\_

## 6.4 Testar Login

- ☐ Fiz logout
- ☐ Cliquei em “Entrar”
- ☐ Fiz login com credenciais criadas
- ☐ Login funcionou corretamente

## 6.5 Testar Produtos

- ☐ Cliquei em um produto
- ☐ Página de detalhes carregou
- ☐ Informações do produto estão corretas
- ☐ Botão “Adicionar ao Carrinho” funciona

## 6.6 Testar Carrinho

- ☐ Adicionei produto ao carrinho
- ☐ Notificação de sucesso apareceu
- ☐ Cliquei no ícone do carrinho
- ☐ Produto está no carrinho
- ☐ Quantidade pode ser alterada
- ☐ Total está sendo calculado corretamente

## 6.7 Testar Checkout

- ☐ Cliquei em “Finalizar Compra”
- ☐ Preenchi dados de entrega
- ☐ Digitei CEP válido (ex: 01310-100)
- ☐ Endereço foi preenchido automaticamente (ViaCEP)
- ☐ Frete foi calculado
- ☐ Preenchi dados de pagamento (simulado)
- ☐ Pedido foi criado com sucesso

## 6.8 Testar Histórico de Pedidos

- ☐ Acessei “Meus Pedidos”
- ☐ Pedido criado está listado
- ☐ Detalhes do pedido estão corretos

## 6.9 Testar Avaliações

- ☐ Voltei ao produto comprado
- ☐ Deixei uma avaliação (nota + comentário)
- ☐ Avaliação foi salva com sucesso
- ☐ Avaliação aparece na página do produto

## 6.10 Testar Painel Admin

- ☐ Fiz logout
- ☐ Fiz login com credenciais admin:
- Email: `admin@donshop007.com`
- Senha: `admin123`
- ☐ Acessei `/admin`
- ☐ Dashboard carrega com estatísticas
- ☐ Posso ver lista de produtos
- ☐ Posso ver lista de pedidos
- ☐ Posso ver lista de usuários

## 6.11 Testar Responsividade

- ☐ Testei em desktop (tela grande)
- ☐ Testei em tablet (tela média)
- ☐ Testei em celular (tela pequena)
- ☐ Layout se adapta corretamente

✅ **FASE 6 CONCLUÍDA!** Todos os testes passaram.



## Informações Importantes para Guardar

### URLs

Site em Produção: \_\_\_\_\_  
 Repositório GitHub: \_\_\_\_\_  
 Dashboard Render: <https://dashboard.render.com>

### Credenciais Admin

Email: `admin@donshop007.com`  
 Senha: `admin123`

## Credenciais de Teste

Email: \_\_\_\_\_  
 Senha: \_\_\_\_\_

## Variáveis de Ambiente (Render)

SECRET\_KEY: \_\_\_\_\_  
 DATABASE\_URL: postgresql://donshop007:SENHA@dpg-xxxxx.oregon-postgres.render.com/don-shop007  
 FLASK\_ENV: production  
 PYTHON\_VERSION: 3.11.0

## DEPLOY COMPLETO!

### Checklist Final

- ☐ Código no GitHub (público)
- ☐ Banco PostgreSQL criado e ativo
- ☐ Web Service criado e ativo (status “Live”)
- ☐ Variáveis de ambiente configuradas
- ☐ Banco populado com produtos
- ☐ Site acessível via URL
- ☐ Cadastro funcionando
- ☐ Login funcionando
- ☐ Produtos carregando
- ☐ Carrinho funcionando
- ☐ Checkout funcionando
- ☐ Pannel admin funcionando
- ☐ Responsividade OK
- ☐ URLs anotadas em lugar seguro

Status:  **PROJETO PUBLICADO COM SUCESSO!** 

## Estatísticas do Deploy

- **Tempo total gasto:** \_ minutos
- **Problemas encontrados:** \_
- **Soluções aplicadas:** \_

## Próximas Ações

Agora que o site está no ar, você pode:

- ☐ Compartilhar URL com amigos/professores

- [ ] Adicionar URL no README.md do GitHub
- [ ] Adicionar URL no TCC
- [ ] Fazer melhorias no código
- [ ] Adicionar mais produtos
- [ ] Personalizar design
- [ ] Configurar domínio próprio (opcional)
- [ ] Adicionar Google Analytics (opcional)

---

## Suporte

---

Se encontrou problemas:

1. **Consulte os guias:**

- `GUIA_GITHUB.md` - Seção “Problemas Comuns”
- `GUIA_RENDER.md` - Seção “Problemas Comuns e Soluções”

2. **Verifique logs:**

- No Render: Menu “Logs”
- Procure por mensagens de erro

3. **Documentação oficial:**

- Render: <https://render.com/docs>
- Flask: <https://flask.palletsprojects.com>

---

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

**Parabéns pelo deploy!** 🚀