

CHECKLIST: Download e Deploy do DonShop007

Guia Passo a Passo com Checkboxes

Use este checklist para acompanhar seu progresso no processo completo de deploy do projeto DonShop007.

FASE 1: DOWNLOAD DOS ARQUIVOS DO DEEPAGENT

Pré-requisitos:

- [] Tenho acesso à interface do DeepAgent
- [] Sei onde está o botão “Files”
- [] Tenho espaço suficiente no meu computador (mínimo 100 MB)

Download:

- [] Cliquei no botão “Files” no DeepAgent
- [] Naveguei até `/home/ubuntu/donshop007/`
- [] Visualizei os arquivos do projeto
- [] Baixei a pasta completa como ZIP
- [] Salvei o arquivo ZIP no meu computador

Extração:

- [] Extraí o arquivo ZIP
- [] Salvei em local apropriado:
 - Windows: `C:\Users\[MeuNome]\donshop007`
 - Mac: `~/Documents/donshop007`
 - Linux: `~/donshop007`
- [] Abri a pasta extraída
- [] Verifiquei que todos os arquivos estão presentes

Verificação dos Arquivos:

- [] `server.js` está presente
- [] `package.json` está presente
- [] Pasta `public/` está presente
- [] Pasta `views/` está presente
- [] Pasta `data/` está presente
- [] `README.md` está presente
- [] Todos os guias de documentação estão presentes

 **Fase 1 Completa!** Arquivos baixados com sucesso.

FASE 2: CONFIGURAÇÃO DO AMBIENTE LOCAL

Instalação de Ferramentas:

Git:

- [] Git está instalado (`git --version`)
- [] OU baixei Git de: <https://git-scm.com/downloads>
- [] OU instalei via gerenciador de pacotes
- [] Fechei e reabri o terminal após instalação
- [] Testei novamente: `git --version`

Node.js (se necessário para testar):

- [] Node.js está instalado (`node --version`)
- [] OU baixei Node.js de: <https://nodejs.org/>
- [] OU instalei via gerenciador de pacotes
- [] npm está instalado (`npm --version`)

Configuração do Git:

- [] Configurei meu nome: `git config --global user.name "Meu Nome"`
- [] Configurei meu email: `git config --global user.email "meu@email.com"`
- [] Verifiquei configuração: `git config --list`
- [] Email é o mesmo da conta GitHub

Navegação no Terminal:

- [] Abri o terminal/PowerShell/CMD
- [] Naveguei até a pasta do projeto
- [] Confirmei que estou na pasta correta (`pwd` ou `cd`)
- [] Listei os arquivos (`dir` ou `ls`)

 **Fase 2 Completa!** Ambiente configurado.



FASE 3: INICIALIZAÇÃO DO REPOSITÓRIO GIT LOCAL

Inicialização:

- [] Executei: `git init`
- [] Vi mensagem: “Initialized empty Git repository”
- [] Executei: `git status`
- [] Vi lista de arquivos “Untracked”

Adicionar Arquivos:

- [] Executei: `git add .`
- [] Executei: `git status` novamente
- [] Vi arquivos “Changes to be committed”

Primeiro Commit:

- [] Executei: `git commit -m "Initial commit - DonShop007 E-commerce"`
- [] Vi confirmação do commit

- [] Não vi mensagens de erro

 **Fase 3 Completa!** Repositório local inicializado.

FASE 4: CRIAÇÃO DO REPOSITÓRIO NO GITHUB

Conta GitHub:

- [] Tenho conta no GitHub
- [] OU criei conta em: <https://github.com/join>
- [] Fiz login no GitHub

Criar Repositório:

- [] Acessei: <https://github.com/new>
- [] OU cliquei no “+” → “New repository”
- [] Escolhi nome: `donshop007-ecommerce` (ou outro)
- [] Adicionei descrição (opcional)
- [] Escolhi visibilidade (Public ou Private)
- [] **NÃO marquei** “Add a README file”
- [] **NÃO marquei** “Add .gitignore”
- [] **NÃO marquei** “Choose a license”
- [] Cliquei em “Create repository”

Anotações Importantes:

Nome do meu repositório: _____

URL do repositório: https://github.com/_/

Visibilidade: [] Public [] Private

 **Fase 4 Completa!** Repositório GitHub criado.

FASE 5: CONECTAR REPOSITÓRIO LOCAL AO GITHUB

Adicionar Remote:

- [] Copiei a URL do meu repositório GitHub
- [] Executei: `git remote add origin [URL_DO_MEU_REPO]`
- [] Executei: `git remote -v`
- [] Vi “origin” listado com a URL correta

Renomear Branch:

- [] Executei: `git branch -M main`
- [] Confirmei que estou na branch “main”

Primeiro Push:

- [] Executei: `git push -u origin main`
- [] Autentiquei quando solicitado:

- [] Username: meu usuário GitHub
- [] Password: token de acesso pessoal (PAT)
- [] Vi progresso do upload
- [] Vi confirmação de sucesso

Verificação:

- [] Acessei meu repositório no GitHub
- [] Vi todos os arquivos listados
- [] Vi o README.md renderizado

 **Fase 5 Completa!** Código está no GitHub.

FASE 6: TOKEN DE ACESSO PESSOAL (SE NECESSÁRIO)

Criar Token:

- [] Acessei: <https://github.com/settings/tokens>
- [] Cliquei em “Generate new token” → “Generate new token (classic)”
- [] Nomeei o token: “DonShop007 Deploy”
- [] Escolhi expiração (90 dias ou No expiration)
- [] Marquei escopo: `repo` (Full control of private repositories)
- [] Cliquei em “Generate token”
- [] **CÓPIEI O TOKEN IMEDIATAMENTE**

Anotação do Token:

 **IMPORTANTE:** Guarde este token em local seguro! Você não verá novamente.

Token: `ghp_`

- [] Salvei o token em local seguro
- [] Usei o token como senha no Git push

 **Fase 6 Completa!** Autenticação configurada.

FASE 7: CRIAR CONTA NO RENDER.COM

Cadastro:

- [] Acessei: <https://render.com>
- [] Cliquei em “Get Started” ou “Sign Up”
- [] Escolhi “Sign up with GitHub” (recomendado)
- [] OU criei conta com email
- [] Autorizei Render a acessar GitHub
- [] Completei o cadastro

Verificação:

- [] Recebi email de confirmação
 - [] Confirmei o email (se aplicável)
 - [] Acessei o dashboard do Render
-  **Fase 7 Completa!** Conta Render criada.
-

FASE 8: CRIAR BANCO DE DADOS POSTGRESQL NO RENDER

Criar Database:

- [] No dashboard Render, cliquei em “New +”
- [] Selecionei “PostgreSQL”
- [] Preenchi os dados:
 - Name: `donshop007-db` (ou outro)
 - Database: `donshop007`
 - User: (gerado automaticamente)
 - Region: `Oregon (US West)` ou mais próximo
 - [] Instance Type: **Free**
 - [] Cliquei em “Create Database”

Aguardar Criação:

- [] Aguardei 2-3 minutos
- [] Status mudou para “Available”
- [] Acessei a aba “Info”

Copiar Credenciais:

- [] Copiei “Internal Database URL” (mais seguro)
- [] OU copiei “External Database URL” (se Internal não funcionar)

Anotações do Banco:

Database Name: _____

Database URL: (copie e guarde com segurança)

`postgres://user:[REDACTED]@host/database`

 **Fase 8 Completa!** Banco de dados criado.

FASE 9: CRIAR WEB SERVICE NO RENDER

Criar Serviço:

- [] No dashboard Render, cliquei em “New +”
- [] Selecionei “Web Service”

- [] Cliquei em “Connect a repository”
- [] Procurei pelo repositório `donshop007-ecommerce`
- [] Cliquei em “Connect”

Configurar Serviço:

Informações Básicas:

- [] Name: `donshop007` (ou outro nome único)
- [] Region: `Oregon (US West)` (ou o mais próximo)
- [] Branch: `main`
- [] Root Directory: (deixei vazio)

Build Settings:

- [] Runtime: `Node`
- [] Build Command: `npm install`
- [] Start Command: `node server.js`

Instância:

- [] Instance Type: **Free**

Variáveis de Ambiente:

Adicionei as seguintes variáveis (botão “Add Environment Variable”):

- [] `PORT = 10000`
- [] `NODE_ENV = production`
- [] `SESSION_SECRET` = (chave aleatória forte)
- [] `DATABASE_URL` = (colei a URL do PostgreSQL do Passo 8)

⚠ Dica para SESSION_SECRET: Use um gerador online ou string aleatória de 32+ caracteres.

Anotações do Web Service:

Service Name: _____

Service URL: https://_.onrender.com

Finalizar:

- [] Roli até o final da página
- [] Cliquei em “Create Web Service”
- [] Vi o build iniciar

✓ Fase 9 Completa! Web Service criado.



FASE 10: AGUARDAR DEPLOY E VERIFICAR LOGS

Acompanhar Build:

- [] Observei os logs em tempo real
- [] Vi “Installing dependencies...” (`npm install`)
- [] Vi “Building...” (se aplicável)
- [] Aguardei até ver “Your service is live” ou “Deploy succeeded”

Tempo estimado: 2-5 minutos

Em caso de erro:

- [] Li a mensagem de erro nos logs
- [] Consultei `SOLUCAO_PROBLEMAS.md`
- [] Corrigi o problema
- [] Cliquei em “Manual Deploy” → “Deploy latest commit”

Verificação:

- [] Status mudou para “Live” (bolinha verde)
- [] Vi a URL do serviço ativa

 **Fase 10 Completa!** Deploy realizado.



FASE 11: POPULAR O BANCO DE DADOS

Acessar Shell do Render:

- [] No dashboard do Web Service, cliquei na aba “Shell”
- [] OU cliquei no botão “Shell” no menu lateral
- [] Terminal interativo abriu

Executar Script de Inicialização:

- [] No Shell, executei: `python init_db.py`
- [] OU: `python3 init_db.py`
- [] Vi mensagens de sucesso:
- “Banco de dados inicializado!”
- “Produtos criados”
- “Usuários criados”
- [] Não vi erros

Verificar:

- [] Acessei o site: <https://meu-app.onrender.com>
- [] Vi produtos na homepage
- [] Conseguí fazer login com credenciais de teste

 **Fase 11 Completa!** Banco de dados populado.



FASE 12: TESTAR O SITE EM PRODUÇÃO

Acesso:

- [] Acessei: <https://meu-app.onrender.com>
- [] Site carregou corretamente
- [] CSS e imagens apareceram

Testes Funcionais:

Homepage:

- [] Produtos estão visíveis

- [] Busca funciona
- [] Filtros funcionam
- [] Menu de navegação funciona

Autenticação:

- [] Conseguí acessar página de cadastro
- [] Conseguí criar uma conta
- [] Conseguí fazer login
- [] Conseguí fazer logout

Produtos:

- [] Conseguí visualizar detalhes de um produto
- [] Conseguí adicionar produto ao carrinho
- [] Conseguí ver o carrinho

Carrinho:

- [] Consigo adicionar/remover produtos
- [] Consigo atualizar quantidades
- [] Total é calculado corretamente

Checkout:

- [] Formulário de checkout carrega
- [] Validação de CEP funciona (API ViaCEP)
- [] Cálculo de frete funciona
- [] Consigo finalizar pedido (pagamento simulado)

Admin (com credenciais admin):

- [] Conseguí acessar /admin
- [] Dashboard carrega
- [] Consigo ver produtos
- [] Consigo ver pedidos

Credenciais de Teste:

Admin:

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

Cliente:

- Email: cliente@teste.com
- Senha: teste123

 **Fase 12 Completa!** Site funcionando em produção!

FASE 13: CONCLUSÃO E PRÓXIMOS PASSOS

Anotações Finais:

 **Projeto implantado com sucesso!**

URLs Importantes:

Serviço	URL
Site em Produção	https://___.onrender.com
Repositório GitHub	https://github.com/_/___
Dashboard Render	https://dashboard.render.com
Banco de Dados Render	(no dashboard Render)

Compartilhar:

- [] Copiei a URL do site
- [] Testei em outro dispositivo/navegador
- [] Compartilhei com amigos/professor
- [] Adicionei ao portfólio

Documentação:

- [] Atualizei o README.md com a URL de produção
- [] Documentei quaisquer problemas encontrados
- [] Anotei soluções para referência futura

Backups:

- [] Código está seguro no GitHub
- [] Tenho cópia local
- [] Anotei credenciais importantes em local seguro

DEPLOY COMPLETO! PARABÉNS!



PRÓXIMOS PASSOS (OPCIONAL)

Melhorias Futuras:

- [] Configurar domínio personalizado
- [] Adicionar mais produtos
- [] Implementar sistema de emails
- [] Adicionar mais formas de pagamento
- [] Melhorar SEO
- [] Adicionar Google Analytics

Manutenção:

- [] Monitorar logs regularmente
- [] Atualizar dependências
- [] Fazer backups periódicos
- [] Responder a feedback de usuários

Upgrades:

- [] Considerar upgrade para plano pago (se necessário)

- Aumentar recursos (RAM, CPU)
 - Adicionar CDN para imagens
 - Configurar cache
-

SOS **SUPORTE E RECURSOS**

Se encontrou problemas:

- Consultei `SOLUCAO_PROBLEMAS.md`
- Li os logs do Render
- Pesquisei o erro no Google
- Consultei documentação oficial

Recursos Úteis:

- Documentação Render: <https://render.com/docs>
 - Documentação Git: <https://git-scm.com/doc>
 - GitHub Guides: <https://guides.github.com/>
 - Stack Overflow: <https://stackoverflow.com/>
-

ESTATÍSTICAS DO DEPLOY

Tempo total gasto: _____ minutos

Principais dificuldades encontradas:

- _____
- _____
- _____

Lições aprendidas:

- _____
- _____
- _____

Avaliação geral: 

✓ RESUMO FINAL

Checklist Geral:

- Baixei arquivos do DeepAgent
- Configurei Git localmente
- Criei repositório GitHub
- Fiz push do código
- Criei conta no Render
- Criei banco PostgreSQL
- Criei Web Service

- [] Configurei variáveis de ambiente
 - [] Deploy realizado com sucesso
 - [] Banco de dados populado
 - [] Site testado e funcionando
 - [] Projeto completo! 🎉
-

Parabéns por concluir o deploy do DonShop007! 🚀✨

Onde estilo encontra excelência - Agora online!
