



SOLUÇÕES PARA PROBLEMAS COMUNS

Guia de Troubleshooting - DonShop007 Deploy

Este documento lista os problemas mais comuns ao fazer deploy e suas soluções.



PROBLEMAS COM ARQUIVOS

✗ Erro: “No such file or directory”

Mensagem: `cd: no such file or directory: /home/ubuntu/donshop007`

Causa: Você está tentando acessar o diretório no servidor DeepAgent, mas está no seu computador local.

Solução:

1. **Baixe os arquivos do DeepAgent primeiro!**
2. No DeepAgent, clique em “Files” (canto superior direito)
3. Navegue até `/home/ubuntu/donshop007/`
4. Baixe toda a pasta como ZIP
5. Extraia no seu computador
6. Navegue até a pasta extraída:

```
```bash
```

```
Windows
```

```
cd C:\Users\SeuNome\donshop007
```

```
Mac/Linux
```

```
cd ~/donshop007
```

```
```
```

✗ Erro: “Cannot find module” ou arquivos faltando

Mensagem: Algum arquivo importante não está presente.

Solução:

1. Verifique se baixou TODOS os arquivos do DeepAgent
2. Liste os arquivos:

```
```bash
```

```
Windows
```

```
dir
```

```
Mac/Linux
```

```
ls -la
```

```
``
```

3. Arquivos essenciais que devem estar presentes:

```
- server.js - package.json - public/ (pasta)
```

```
- views/ (pasta)
```

- data/` (pasta)

4. Se faltar algum, **baixe novamente do DeepAgent**

---

## PROBLEMAS COM GIT

---

### Erro: “git: command not found”

**Mensagem:** bash: git: command not found ou 'git' is not recognized

**Causa:** Git não está instalado no seu computador.

**Solução:**

**Windows:**

```
Opção 1: Baixar instalador
Acesse: https://git-scm.com/download/win
Baixe e instale o Git for Windows

Opção 2: Usar winget (Windows 10/11)
winget install --id Git.Git -e --source winget
```

**Mac:**

```
Com Homebrew (recomendado)
brew install git

Ou baixe de: https://git-scm.com/download/mac
```

**Linux (Ubuntu/Debian):**

```
sudo apt update
sudo apt install git -y
```

**Linux (Fedora/CentOS):**

```
sudo yum install git -y
```

**Depois de instalar, feche e abra o terminal novamente e teste:**

```
git --version
```

---

### Erro: “Please tell me who you are”

**Mensagem:**

```
*** Please tell me who you are.
Run: git config --global user.email "you@example.com"
```

**Causa:** Git precisa saber quem você é antes do primeiro commit.

**Solução:**

```
git config --global user.name "Seu Nome Completo"
git config --global user.email "seu.email@example.com"
```

⚠ Use o mesmo email da sua conta GitHub!

---

## ✗ Erro: “remote origin already exists”

**Mensagem:** fatal: remote origin already exists

**Causa:** Você já adicionou o remote ‘origin’ anteriormente.

**Solução:**

**Opção 1 - Remover e adicionar novamente:**

```
git remote remove origin
git remote add origin https://github.com/SeuUsuario/donshop007-ecommerce.git
```

**Opção 2 - Apenas atualizar a URL:**

```
git remote set-url origin https://github.com/SeuUsuario/donshop007-ecommerce.git
```

**Verificar:**

```
git remote -v
```

---

## ✗ Erro: “Permission denied (publickey)”

**Mensagem:** Permission denied (publickey) ou fatal: Authentication failed

**Causa:** Problema com autenticação SSH ou credenciais.

**Solução 1 - Usar HTTPS ao invés de SSH:**

```
Remova o remote atual
git remote remove origin

Adicione usando HTTPS (mais fácil)
git remote add origin https://github.com/SeuUsuario/donshop007-ecommerce.git

Faça push
git push -u origin main
```

**Solução 2 - Criar Token de Acesso Pessoal (PAT):**

1. Acesse: <https://github.com/settings/tokens>

2. Clique em “Generate new token” → “Generate new token (classic)”
3. Marque: `repo` (Full control of private repositories)
4. Clique em “Generate token”
5. **COPIE O TOKEN** (você só verá uma vez!)
6. Quando o Git pedir senha, use o **token** ao invés da senha da conta

### Solução 3 - Usar GitHub CLI:

```
Instale o GitHub CLI
Windows: winget install --id GitHub.cli
Mac: brew install gh
Linux: sudo apt install gh

Faça login
gh auth login

Siga as instruções interativas
```

## ✗ Erro: “failed to push some refs”

**Mensagem:** `error: failed to push some refs to 'github.com/...'`

**Causa:** O repositório remoto tem commits que você não tem localmente, ou você não tem permissão.

### Solução 1 - Puxar mudanças primeiro:

```
git pull origin main --rebase
git push origin main
```

### Solução 2 - Forçar push (⚠ cuidado!):

```
Use APENAS se tiver certeza que o remoto não tem nada importante
git push -f origin main
```

### Solução 3 - Verificar permissões:

- Certifique-se de que você é dono do repositório
- Ou que tem permissão de escrita (write access)

## ✗ Erro: “Repository not found”

**Mensagem:** `fatal: repository 'https://github.com/...' not found`

**Causa:** URL do repositório incorreta ou repositório não existe.

### Solução:

1. Verifique a URL configurada:

```
bash
```

```
git remote -v
```

2. Confirme que o repositório existe no GitHub
3. Verifique se digitou o nome do usuário e repositório corretamente

4. Atualize a URL se necessário:

```
bash
```

```
git remote set-url origin https://github.com/USUARIO_CORRETO/REPO_CORRETO.git
```

---

## PROBLEMAS COM GITHUB


### Erro: “Support for password authentication was removed”

**Mensagem:**

```
remote: Support for password authentication was removed on August 13, 2021.
remote: Please use a personal access token instead.
```

**Causa:** GitHub não aceita mais senha da conta para autenticação via Git.

**Solução - Criar Token de Acesso Pessoal:**

1. Acesse: <https://github.com/settings/tokens>
2. Clique em “**Generate new token**” → “**Generate new token (classic)**”
3. **Nome:** “DonShop007 Deploy”
4. **Expiração:** 90 dias (ou “No expiration” se preferir)
5. **Selecione o escopo:** Marque `repo` (acesso completo aos repositórios)
6. Clique em “**Generate token**”
7.  **COPIE O TOKEN IMEDIATAMENTE** (ex: `ghp_XXXXXXXXXXXXXXXXXXXX`)
8. Salve em local seguro (você não verá novamente!)
9. Quando o Git pedir senha, **cole o token** ao invés da senha

**Como usar o token:**

```
Quando fizer push, o Git pedirá:
Username: seu-usuario-github
Password: [COLE O TOKEN AQUI, NÃO A SENHA DA CONTA]
```

---

### Não consigo criar repositório

**Problema:** Não encontro onde criar repositório no GitHub.

**Solução:**

1. Faça login em <https://github.com>
2. Clique no + no canto superior direito
3. Selecione “**New repository**”
4. Ou acesse diretamente: <https://github.com/new>

---

## PROBLEMAS COM RENDER.COM

### Build falhou no Render

**Mensagem:** `Build failed` ou `Failed to start service`

## Causas comuns e soluções:

### 1. Faltam dependências:

```
Build Command: npm install
Start Command: node server.js
```

### 2. Arquivo package.json com erro:

- Verifique se o arquivo está correto
- Teste localmente: `npm install`

### 3. Versão do Node.js:

- Adicione no `package.json` :

```
json
"engines": {
 "node": ">=18.0.0"
}
```

### 4. Porta incorreta:

- No Render, use a variável `PORT` fornecida por eles
- Já está configurado no `server.js`

## ✗ Erro: “Application failed to respond”

**Mensagem:** Serviço iniciou mas não responde.

### Solução:

1. Verifique os logs no Render (aba “Logs”)
2. Confirme que o servidor está usando `process.env.PORT` :  

```
javascript
const PORT = process.env.PORT || 3000;
```
3. Verifique se todas as variáveis de ambiente estão configuradas:
  - `PORT` = 10000
  - `NODE_ENV` = production
  - `SESSION_SECRET` = (sua chave secreta)

## ✗ Site carrega mas sem estilo/imagens

**Problema:** Página HTML carrega mas CSS e imagens não aparecem.

### Solução:

1. Verifique se a pasta `public/` foi enviada ao GitHub
2. Confirme que o `server.js` tem:  

```
javascript
app.use(express.static('public'));
```
3. Verifique caminhos relativos no HTML:  

```
```html
```

...

❌ Banco de dados não persiste

Problema: Dados são perdidos após reiniciar o serviço.

Causa: No plano Free do Render, o sistema de arquivos é efêmero.

Solução:

1. **Opção 1:** Upgrade para plano pago (persistência de disco)
2. **Opção 2:** Use banco de dados externo:
 - MongoDB Atlas (gratuito)
 - PostgreSQL no Render
 - Firebase Realtime Database

❌ Deploy automático não funciona

Problema: Fiz push no GitHub mas Render não atualizou.

Solução:

1. No Render, vá em **Settings**
2. Verifique se **"Auto-Deploy"** está **Yes**
3. Confira se a branch está correta (geralmente `main`)
4. Ou faça deploy manual: botão **"Manual Deploy"** → **"Deploy latest commit"**



PROBLEMAS LOCAIS

❌ Erro: "npm: command not found"

Mensagem: `bash: npm: command not found`

Causa: Node.js/npm não está instalado.

Solução:

Windows:

```
# Baixe e instale de: https://nodejs.org/  
# Escolha a versão LTS (recomendada)  
  
# Ou use winget:  
winget install OpenJS.NodeJS.LTS
```

Mac:

```
# Com Homebrew:  
brew install node
```

Linux:

```
# Ubuntu/Debian:
sudo apt update
sudo apt install nodejs npm -y

# Fedora/CentOS:
sudo yum install nodejs npm -y
```

✗ Erro: “Port 3000 is already in use”

Mensagem: Error: listen EADDRINUSE: address already in use :::3000

Causa: Outro processo já está usando a porta 3000.

Solução:

Windows:

```
# Encontrar o processo:
netstat -ano | findstr :3000

# Matar o processo (substitua PID pelo número encontrado):
taskkill /PID NUMERO_DO_PID /F
```

Mac/Linux:

```
# Encontrar o processo:
lsof -i :3000

# Matar o processo (substitua PID pelo número encontrado):
kill -9 PID
```

Ou use outra porta:

```
PORT=3001 node server.js
```

✗ Erro ao instalar dependências

Mensagem: npm ERR! durante npm install

Solução:

1. Limpar cache do npm:

```
npm cache clean --force
npm install
```

2. Deletar node_modules e reinstalar:


```
# Windows
rmdir /s node_modules
del package-lock.json

# Mac/Linux
rm -rf node_modules
rm package-lock.json

# Reinstalar
npm install
```

3. Atualizar npm:

```
npm install -g npm@latest
```

PROBLEMAS DE SEGURANÇA

Expus dados sensíveis no GitHub

Problema: Fiz commit de senhas, tokens ou dados sensíveis.

Solução URGENTE:

1. Remover do histórico:

```
# AVISO: Isso reescreve o histórico!
git filter-branch --force --index-filter \
  "git rm --cached --ignore-unmatch caminho/do/arquivo-sensivel" \
  --prune-empty --tag-name-filter cat -- --all

git push origin --force --all
```

2. Invalidar credenciais:

- **Tokens do GitHub:** Revogue em <https://github.com/settings/tokens>
- **API Keys:** Regenere no serviço correspondente
- **Senhas:** Troque imediatamente

3. Usar .gitignore:

Adicione ao arquivo `.gitignore` :

```
# Dados sensíveis
.env
.env.local
config/secrets.json
credentials.txt

# Dados de desenvolvimento
node_modules/
*.log
.DS_Store
```

PRECISA DE MAIS AJUDA?

Recursos úteis:

Documentação oficial:

- Git: <https://git-scm.com/doc>
- GitHub: <https://docs.github.com/>
- Render: <https://render.com/docs>
- Node.js: <https://nodejs.org/docs/>

Tutoriais:

- Git Handbook: <https://guides.github.com/introduction/git-handbook/>
- GitHub Skills: <https://skills.github.com/>
- Render Guides: <https://render.com/docs/deploy-node-express-app>

Comunidades:

- Stack Overflow: <https://stackoverflow.com/>
 - GitHub Community: <https://github.community/>
 - Dev.to: <https://dev.to/>
-

CHECKLIST DE VERIFICAÇÃO

Antes de pedir ajuda, verifique:

- ☐ Baixei todos os arquivos do DeepAgent?
 - ☐ Estou no diretório correto do projeto?
 - ☐ Git está instalado? (`git --version`)
 - ☐ Node.js está instalado? (`node --version`)
 - ☐ Configurei meu usuário Git?
 - ☐ Criei o repositório no GitHub?
 - ☐ A URL do remote está correta? (`git remote -v`)
 - ☐ Testei localmente? (`npm install && node server.js`)
 - ☐ Li as mensagens de erro completamente?
 - ☐ Consultei os logs do Render?
-

Dica: A maioria dos problemas pode ser resolvida seguindo as mensagens de erro cuidadosamente e pesquisando no Google com a mensagem exata do erro entre aspas.

Exemplo: `"fatal: remote origin already exists" git`

Boa sorte! 