🚀 GUIA DE DEPLOY - SISTEMA CONVERSOR ZPL PARA PDF

Jalmac Móveis - Versão de Produção

📋 PRÉ-REQUISITOS

```bash
# Python 3.10 ou superior
python --version

# pip atualizado
python -m pip install --upgrade pip
```

📦 INSTALAÇÃO

1. Criar estrutura de pastas

```bash
mkdir conversor-zpl
cd conversor-zpl
mkdir templates
mkdir uploads
mkdir pdfs_temp
mkdir Etiquetas_Impressao
```

2. Instalar dependências

Crie o arquivo `requirements.txt`:

```txt
Flask==3.0.0
Flask-SocketIO==5.3.5
Flask-CORS==4.0.0
eventlet==0.33.3
requests==2.31.0
PyPDF2==3.0.1
python-engineio==4.8.0
python-socketio==5.10.0
```

Instale:

```bash
pip install -r requirements.txt
```

📁 ESTRUTURA DE ARQUIVOS

```
conversor-zpl/
├── app.py                  # Backend Python
├── templates/
│   └── index.html          # Frontend
├── uploads/                # Upload temporário (criado automaticamente)
├── pdfs_temp/              # PDFs temporários (criado automaticamente)
├── Etiquetas_Impressao/    # PDFs finais (criado automaticamente)
├── requirements.txt        # Dependências
└── DEPLOY.md               # Este arquivo
```

🏃 EXECUTAR EM DESENVOLVIMENTO

```bash
# Modo simples (desenvolvimento)
python app.py

# O servidor iniciará em:
# http://localhost:5000
```

Acesse no navegador: `http://localhost:5000`

🔒 EXECUTAR EM PRODUÇÃO

Opção 1: Com Gunicorn + Eventlet (RECOMENDADO)

```bash
# Instalar Gunicorn
pip install gunicorn

# Executar
gunicorn --worker-class eventlet -w 1 --bind 0.0.0.0:5000 app:app
```

Opção 2: Com Eventlet direto

```python
# Já está configurado no app.py
python app.py
```

Opção 3: Com Nginx como proxy reverso

1. Instale o Nginx

```bash
sudo apt install nginx
```

2. Configure o Nginx (`/etc/nginx/sites-available/conversor-zpl`):

```nginx
server {
    listen 80;
    server_name seu-servidor.local;  # ou IP do servidor

    client_max_body_size 50M;

    location / {
        proxy_pass http://127.0.0.1:5000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }

    location /socket.io {
        proxy_pass http://127.0.0.1:5000/socket.io;
        proxy_http_version 1.1;
        proxy_buffering off;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "Upgrade";
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
```

3. Ative a configuração:

```bash
sudo ln -s /etc/nginx/sites-available/conversor-zpl /etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx
```

4. Execute o app Python:

```bash
gunicorn --worker-class eventlet -w 1 --bind 127.0.0.1:5000 app:app
```

🔄 EXECUTAR COMO SERVIÇO (SYSTEMD)

Crie o arquivo `/etc/systemd/system/conversor-zpl.service`:

```ini
[Unit]
Description=Conversor ZPL para PDF - Jalmac Moveis
After=network.target

[Service]
Type=simple
User=seu-usuario
WorkingDirectory=/caminho/para/conversor-zpl
Environment="PATH=/caminho/para/venv/bin"
ExecStart=/caminho/para/venv/bin/gunicorn --worker-class eventlet -w 1 --bind 0.0.0.0:5000 app:app
Restart=always
RestartSec=10

[Install]
WantedBy=multi-user.target
```

Ative o serviço:

```bash
sudo systemctl daemon-reload
sudo systemctl enable conversor-zpl
sudo systemctl start conversor-zpl
sudo systemctl status conversor-zpl
```

🐳 DEPLOY COM DOCKER (OPCIONAL)

Crie o arquivo `Dockerfile`:

dockerfile

```
FROM python:3.11-slim

WORKDIR /app

COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

EXPOSE 5000

CMD ["python", "app.py"]
```

Crie o arquivo `docker-compose.yml`:

yaml

```
version: '3.8'

services:
  conversor-zpl:
    build: .
    ports:
      - "5000:5000"
    volumes:
      - ./Etiquetas_Impressao:/app/Etiquetas_Impressao
      - ./uploads:/app/uploads
    restart: unless-stopped
    environment:
      - PYTHONUNBUFFERED=1
```

Execute:

bash

```
docker-compose up -d
```

🔧 CONFIGURAÇÕES DE SEGURANÇA

1. Firewall (UFW)

```bash
# Permitir apenas rede interna
sudo ufw allow from 192.168.0.0/16 to any port 5000
```

2. Alterar porta padrão

No `app.py`, linha final:

```python
socketio.run(app, host='0.0.0.0', port=8080)  # Altere a porta aqui
```

3. Limitar acesso por IP

No início do `app.py`:

```python
ALLOWED_IPS = ['192.168.1.0/24', '10.0.0.0/8']

@app.before_request
def limit_remote_addr():
    if request.remote_addr not in ALLOWED_IPS:
        abort(403)
```

📊 MONITORAMENTO

Logs em tempo real

```bash
# Logs do systemd
sudo journalctl -u conversor-zpl -f

# Logs do app
tail -f /var/log/conversor-zpl.log
```

Verificar saúde do sistema

```bash
curl http://localhost:5000/health
```

Resposta esperada:

```json
{
  "status": "online",
  "service": "ZPL to PDF Converter - Jalmac Móveis",
  "version": "1.0.0",
  "timestamp": "2025-01-XX..."
}
```

🐛 TROUBLESHOOTING

Problema: WebSocket não conecta

Solução:

1. Verifique se `eventlet` está instalado
2. Certifique-se que não há firewall bloqueando
3. Teste com `curl http://localhost:5000`

Problema: Erro ao mesclar PDFs

Solução:

1. Verifique se `PyPDF2` está instalado corretamente
2. Confira permissões da pasta `Etiquetas_Impressao`

```bash
chmod 755 Etiquetas_Impressao
```

Problema: API Labelary retorna erro 429

Solução: Aumentar delay entre requests no `app.py`:

```python
RATE_LIMIT_DELAY = 0.6  # Aumentar de 0.4 para 0.6
```

📈 PERFORMANCE

Configurações otimizadas:

```python
```

```
# No app.py - classe Config
RATE_LIMIT_DELAY = 0.4      # Reduzir para mais velocidade (mínimo 0.3)
REQUEST_TIMEOUT = 30        # Aumentar se conexão lenta
MAX_WORKERS = 8             # Não usado atualmente (sequencial)
```

Dicas:

- Use SSD para melhor I/O
- Memória RAM: mínimo 2GB
- CPU: 2+ cores recomendado

🔄 BACKUP E MANUTENÇÃO

Backup automático

Crie script `backup.sh`:

```bash
#!/bin/bash
DATE=$(date +%Y%m%d_%H%M%S)
tar -czf backup_$DATE.tar.gz Etiquetas_Impressao/
find . -name "backup_*.tar.gz" -mtime +30 -delete
```

Adicione ao crontab:

```bash
crontab -e

# Backup diário às 2h da manhã
0 2 * * * /caminho/para/backup.sh
```

Limpeza de arquivos temporários

```bash
# Limpar uploads antigos (mais de 7 dias)
find uploads/ -type f -mtime +7 -delete
find pdfs_temp/ -type f -mtime +7 -delete
```

📞 SUPORTE

Desenvolvedor: Bruno

Empresa: Jalmac Móveis
Versão: 1.0.0 - Produção
Data: 2025

✅ CHECKLIST DE DEPLOY

- [ ] Python 3.10+ instalado
- [ ] Dependências instaladas (`requirements.txt`)
- [ ] Pastas criadas corretamente
- [ ] Arquivo `app.py` no diretório raiz
- [ ] Arquivo `index.html` em `templates/`
- [ ] Servidor rodando em `http://0.0.0.0:5000`
- [ ] WebSocket conectando corretamente
- [ ] Teste de upload funcionando
- [ ] Teste de conversão funcionando
- [ ] Teste de download funcionando
- [ ] Firewall configurado (se necessário)
- [ ] Nginx configurado (se necessário)
- [ ] Serviço systemd ativo (se necessário)
- [ ] Backup configurado

🎉 PRONTO PARA PRODUÇÃO!

O sistema está completo e pronto para uso interno na Jalmac Móveis.