⚡ QUICK START - Deploy em 20 Minutos

🎯 Objetivo

Colocar o Telegram Scraper funcionando no N8N o mais rápido possível.

⏱️ PASSO 1: Obter Credenciais (5 min)

1.1 Credenciais do Telegram

1. Acesse: https://my.telegram.org/apps
2. Faça login com seu número
3. Clique em “Create Application”
4. Preencha:
   - **App title**: Telegram Scraper
   - **Short name**: scraper
   - **Platform**: Other
5. Anote:
   `API_ID = _____ (número)`
   `API_HASH = _____ (string longa)`
   `PHONE = +55_____ (seu número com código do país)`

1.2 Gerar Token de API

Execute no terminal:

```
openssl rand -hex 32
```

Anote:

```
API_TOKEN = _____ (string aleatória de 64 caracteres)
```

🚀 PASSO 2: Deploy no Render.com (10 min)

2.1 Preparar Git (se ainda não fez)

```
cd /home/ubuntu/telegram-proxy-service
git init
git add .
git commit -m "Initial commit"
```

2.2 Criar Repositório no GitHub

1. Acesse: https://github.com/new

2. Nome: `telegram-proxy-service`
3. Visibilidade: **Private** (recomendado)
4. NÃO inicialize com README
5. Clique “Create repository”

2.3 Push para GitHub

Copie os comandos exibidos no GitHub e execute:

```
git remote add origin https://github.com/SEU-USUARIO/telegram-proxy-service.git
git branch -M main
git push -u origin main
```

2.4 Deploy no Render

1. Acesse: https://render.com
2. Crie conta (grátis)
3. Clique **“New +” → “Web Service”**
4. Clique **“Connect GitHub”** e autorize
5. Selecione `telegram-proxy-service`
6. Configure:
   - **Name**: `telegram-proxy-service`
   - **Runtime**: Node
   - **Build Command**: `npm install`
   - **Start Command**: `npm start`
   - **Instance Type**: Free
7. Role até **“Environment Variables”**
8. Adicione (clique “+ Add Environment Variable” para cada):

`NODE_ENV = production`
`PORT = 3000`
`TELEGRAM_API_ID = [seu API_ID aqui]`
`TELEGRAM_API_HASH = [seu API_HASH aqui]`
`TELEGRAM_PHONE = [seu telefone aqui, ex: +5511999999999]`
`API_TOKEN = [token gerado no passo 1.2]`

1. Clique **“Create Web Service”**
2. Aguarde deploy (5-10 min)
3. Anote a URL gerada:
   `URL = https://telegram-proxy-service-XXXX.onrender.com`

2.5 Obter SESSION_STRING (IMPORTANTE!)

1. No Render, vá na aba **“Logs”**
2. Aguarde aparecer: `Phone code requested`
3. Abra o Telegram no celular
4. Copie o código recebido (ex: 12345)
5. No Render:
   - Vá em **“Environment”**
   - Adicione variável temporária:
   `TELEGRAM_CODE = 12345`
   - Clique “Save Changes”

6. Serviço vai reiniciar automaticamente
7. Nos **Logs**, procure por: `NEW SESSION STRING - Save this`
8. Copie a string longa que aparece (ex: `1BQAAAAA...`)
9. No Render:
   - Vá em **“Environment”**
   - **ADICIONE** nova variável:
   `TELEGRAM_SESSION = [string copiada]`
   - **REMOVA** a variável `TELEGRAM_CODE`
   - Clique “Save Changes”
10. Serviço vai reiniciar (última vez)

2.6 Verificar Funcionamento

Abra no navegador:

```
https://telegram-proxy-service-XXXX.onrender.com/health
```

Deve retornar:

```
{
  "status": "ok",
  "telegram_connected": true
}
```

✅ Se viu isso, o microserviço está funcionando!

⚙️ PASSO 3: Configurar N8N (2 min)

3.1 Acessar N8N

Acesse: https://workflows.hospitalarsaude.com.br

3.2 Adicionar Variáveis de Ambiente

1. Clique no **menu** (☰) → **Settings** → **Environment Variables**
2. Adicione as seguintes variáveis:

`TELEGRAM_PROXY_URL = https://telegram-proxy-service-XXXX.onrender.com`
`TELEGRAM_PROXY_TOKEN = [mesmo token do PASSO 1.2]`
`TELEGRAM_CHANNELS = aicommunitybr,chatgptbrasil`
`MESSAGES_PER_CHANNEL = 100`

1. Clique **“Save”**

3.3 Ativar Workflow

1. No menu lateral, clique em **“Workflows”**
2. Procure por: **“Telegram Scraper V2 - Production (FIXED)”**
3. Abra o workflow
4. No canto superior direito, clique em **“Active”** (toggle para ON)

🧪 PASSO 4: Testar (3 min)

4.1 Teste Manual

1. No workflow aberto no N8N
2. Clique em **“Execute Workflow”**
3. Aguarde execução (pode levar 2-5 min)
4. Verifique cada node:
   - ✅ Telegram Scraper API → deve mostrar resposta JSON
   - ✅ Extract Messages → deve mostrar lista de mensagens
   - ✅ Demais nodes processando normalmente

4.2 Verificar Dados

1. Acesse seu Supabase
2. Abra tabela `messages`
3. Verifique se há novas mensagens com timestamp recente

✅ Se viu mensagens novas, está tudo funcionando!

🎉 PRONTO!

Seu Telegram Scraper está funcionando e vai executar automaticamente a cada 6 horas!

🆘 Problemas?

Erro: “Unauthorized” no N8N

Causa: Token incorreto

Solução:
1. Verifique se `TELEGRAM_PROXY_TOKEN` no N8N é **exatamente igual** ao `API_TOKEN` no Render
2. Não deve ter espaços extras
3. É case-sensitive

Erro: “telegram_connected: false” no health check

Causa: SESSION_STRING não configurado ou inválido

Solução:
1. Volte ao PASSO 2.5 e obtenha novo SESSION_STRING
2. Verifique se copiou a string completa (pode ser muito longa)
3. Verifique se as credenciais do Telegram estão corretas

Erro: “No messages returned from API”

Causas possíveis:
1. Canais não existem ou estão com nome errado
2. Canais são privados e você não tem acesso
3. Limite muito baixo

Solução:
1. Verifique nomes dos canais (sem @, ex: `aicommunitybr`)
2. Teste com canais públicos conhecidos
3. Aumente `MESSAGES_PER_CHANNEL` para 100

Serviço no Render trava ou reinicia

Causa: Free tier do Render dorme após 15 min de inatividade

Solução:
1. Configure UptimeRobot para fazer ping no `/health` a cada 5 min
2. Ou faça upgrade para paid tier ($7/mês)

📚 Documentação Completa

Para informações detalhadas, consulte:

- **README.md** → Documentação completa do microserviço
- **TELEGRAM_PROXY_SOLUTION.md** → Solução completa com troubleshooting
- **RESUMO_ENTREGA.md** → Resumo executivo

✅ Checklist Rápido

- [ ] Obteve TELEGRAM_API_ID, TELEGRAM_API_HASH, TELEGRAM_PHONE
- [ ] Gerou API_TOKEN seguro
- [ ] Criou repo no GitHub
- [ ] Deploy no Render com variáveis configuradas
- [ ] Obteve SESSION_STRING e configurou no Render
- [ ] Verificou `/health` retorna `telegram_connected: true`
- [ ] Adicionou variáveis no N8N
- [ ] Ativou workflow no N8N
- [ ] Testou execução manual
- [ ] Verificou dados no Supabase

Tempo total: ~20 minutos ⚡

Resultado: Telegram Scraper funcionando automaticamente a cada 6 horas! 🎊