

Resumo dos Commits - DevSphere.ai

Estado Atual do Repositório

- ✓ 45 commits locais prontos para push
- ✓ Working tree limpo - todas as alterações commitadas
- ✓ Repositório remoto configurado: `git@github.com:daltonpinheiro1/DevSphere.git`

Principais Implementações Commitadas

1. Sistema de Proxies Oxylabs (Último Commit)

- ✓ 6 proxies configurados (BR, US, MX, AR, CO, CL)
- ✓ Rotação automática de proxies
- ✓ Fallback em caso de falha
- ✓ Health checks automáticos

2. Sistema de Cache Redis

- ✓ Cache de conversas (4-6 horas)
- ✓ Redução de consumo de tokens
- ✓ Respostas rápidas para perguntas repetidas

3. ChatGPT Go - Chatbots Personalizados

- ✓ Criação e gestão de chatbots
- ✓ Upload de arquivos de treinamento (TXT, PDF, JSON, CSV, DOCX)
- ✓ Prompts customizados por chatbot
- ✓ Interface completa de gerenciamento

4. Fluxo de Vendas TIM Completo

- ✓ 11 estágios do fluxo de vendas
- ✓ Verificação de viabilidade (CEP + número)
- ✓ Coleta de dados pessoais e endereço
- ✓ Captura de geolocalização
- ✓ Autorização e finalização
- ✓ Botão "Adquira já!" integrado

5. Dashboard de Leads e Vendas

- ✓ Monitoramento em tempo real
- ✓ Estatísticas de conversão
- ✓ Filtros por estágio do fluxo
- ✓ Detalhamento completo de cada lead

6. Sistema Hierárquico de Usuários

- ✓ 4 níveis: ADMIN, MANAGER, ASSISTANT, USER
- ✓ Permissões por nível
- ✓ Interface de gestão de usuários

- ☒ Registro controlado

7. Sistema de Tabulação de Conversas

- ☒ 6 motivos de tabulação
- ☒ Alertas para conversas não tabuladas (+2h)
- ☒ Métricas e estatísticas
- ☒ Interface completa de gestão

8. Correções TypeScript e Schema

- ☒ Alinhamento completo com snake_case do Prisma
- ☒ Geração de UUIDs para todos os registros
- ☒ Campos updated_at em todas as tabelas
- ☒ Relacionamentos corrigidos

9. Integração AWS S3

- ☒ Upload de arquivos de treinamento
- ☒ Upload de mídia para templates
- ☒ Signed URLs para download seguro

10. Documentação Completa

- ☒ GUIA_COMPLETO_SISTEMA_AVANÇADO.md
- ☒ GUIA_SISTEMA_PROXY.md
- ☒ PROBLEMA_QR_CODE_405.md
- ☒ SISTEMA_FALLBACK_AUTOMATICO.md
- ☒ README_MELHORIAS.md

Arquivos Principais Modificados/Criados

Backend (API Routes)

- ☒ /api/whatsapp/chatbots/* - Gestão de chatbots
- ☒ /api/whatsapp/sales-flow/* - Fluxo de vendas TIM
- ☒ /api/whatsapp/instances/* - Gestão de instâncias
- ☒ /api/whatsapp/proxies/* - Gestão de proxies
- ☒ /api/conversations/whatsapp/* - Conversas WhatsApp
- ☒ /api/admin/users/* - Gestão de usuários

Frontend (Componentes)

- ☒ components/whatsapp/chatbots-manager.tsx - Interface chatbots
- ☒ components/whatsapp/sales-leads-manager.tsx - Dashboard leads
- ☒ components/whatsapp/instances-manager.tsx - Gestão instâncias
- ☒ components/whatsapp/proxies-manager.tsx - Gestão proxies
- ☒ app/admin/users/page.tsx - Gestão usuários
- ☒ app/admin/conversations/page.tsx - Tabulação conversas

Core (Bibliotecas)

- ☒ lib/redis.ts - Cliente Redis e cache
- ☒ lib/s3.ts - Integração AWS S3

- ✓ lib/whatsapp/proxy-pool.ts - Pool de proxies
- ✓ lib/whatsapp/conversation-cache.ts - Cache conversas
- ✓ lib/whatsapp/tim-api.ts - API TIM (simulada)
- ✓ lib/whatsapp/tim-sales-flow.ts - Fluxo vendas TIM
- ✓ lib/whatsapp/auto-reply-handler.ts - Respostas automáticas
- ✓ lib/whatsapp/baileys-service.ts - Serviço Baileys
- ✓ lib/whatsapp/instance-manager.ts - Gerenciador instâncias
- ✓ lib/whatsapp/campaign-manager.ts - Gerenciador campanhas

Database (Prisma)

- ✓ prisma/schema.prisma - Schema completo atualizado
- ✓ prisma/seed.ts - Seeds atualizados
- ✓ scripts/seed-chatbot.ts - Seed chatbot Centermed
- ✓ scripts/setup-oxylabs.ts - Setup automático proxies

Próximos Passos para Push no GitHub

Opção 1: Via SSH (Requer configuração de chave)

```
# Configurar chave SSH no GitHub primeiro
git push -u origin main
```

Opção 2: Via HTTPS (Mais simples)

```
# Alterar remote para HTTPS
git remote set-url origin https://github.com/daltonpinheiro1/DevSphere.git

# Fazer push (pedirá usuário e token)
git push -u origin main
```

Opção 3: Via GitHub CLI (Se disponível)

```
gh auth login
git push -u origin main
```

Comandos Úteis

Ver todos os commits

```
git log --oneline --all --graph
```

Ver estatísticas do último commit

```
git show --stat HEAD
```

Ver arquivos modificados

```
git diff HEAD~1 HEAD --name-status
```

Criar tag de versão

```
git tag -a v1.0.0 -m "DevSphere.ai - Sistema Completo"  
git push origin v1.0.0
```

Status Final

 **Total de arquivos no projeto:** 156 arquivos

 **Total de commits:** 45 commits

 **Status:** Working tree clean

 **Pronto para deploy:** Sim

 **Pronto para push GitHub:** Sim

Branch: main

Último commit: 69a6a4e - Sistema completo DevSphere.ai com proxies Oxylabs