

ESPECIFICAÇÃO TÉCNICA – SISTEMA REISTECH

****Versão:** 2.0.0 | ****Data:** 2024-01-15 | ****Status:** IMPLEMENTADO******

1. VISÃO GERAL DO SISTEMA

1.1. Objetivo Principal

Plataforma de automação de atendimento e vendas via WhatsApp com motor de inteligência conversacional para múltiplos nichos de mercado.

- ### 1.2. Público-Alvo
- Pequenas e médias empresas que atendem via WhatsApp
 - Empreendedores digitais em nichos específicos
 - Equipes de vendas e suporte ao cliente
 - Profissionais de serviços (advogados, mecânicos, designers)

2. ARQUITETURA DO SISTEMA

2.1. Stack Tecnológico

BACKEND:

- Node.js 18+ (Runtime)
- Express.js 4.x (Framework)
- PostgreSQL 15 (Banco de dados)
- Redis 7 (Cache e sessões)
- WebSocket (Comunicação em tempo real)
- JWT + RBAC (Autenticação)

FRONTEND:

- React 18 + Vite
- Material-UI 5 (UI Components)
- Redux Toolkit (Gerenciamento de estado)
- WebSocket Client
- Notistack (Notificações)

INFRAESTRUTURA:

- Docker + Docker Compose
- Nginx (Reverse proxy)
- GitHub Actions (CI/CD)
- PM2 (Process manager – produção)

2.2. Diagrama de Componentes





3. MÓDULOS PRINCIPAIS

3.1. Módulo de Autenticação

- Login com email/senha
- JWT tokens (access + refresh)
- RBAC (Admin, Operador)
- Workspace isolation

3.2. Motor ReisTech (FSM)

ESTADOS POR NICHOS:

iPhone Store:

[início] → [consulta] → [orçamento] → [venda] → [pós-venda]

Advocacia:

[início] → [triagem] → [coleta_dados] → [agendamento] → [followup]

Mecânica:

[início] → [diagnóstico] → [orçamento] → [agendamento] → [conclusão]

Nail Designer:

[início] → [catálogo] → [agendamento] → [confirmação] → [lembrete]

Detetive:

[início] → [confidencialidade] → [coleta] → [proposta] → [contrato]

3.3. Sistema de Fila Humana

- Distribuição automática por disponibilidade
- Locks exclusivos por operador
- Priorização por tempo de espera
- Transferência entre operadores

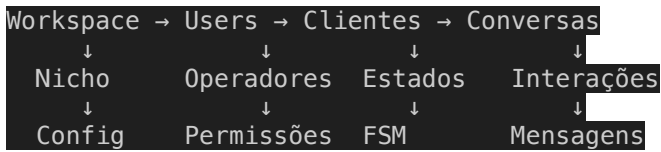
3.4. Integração WhatsApp

- Conexão via QR Code
- Multi-números simultâneos
- Envio de mídia (imagens, PDFs)

- Respostas automáticas configuráveis

4. MODELO DE DADOS

4.1. Entidades Principais



4.2. Relacionamentos

- 1 Workspace : N Users
- 1 User : N Clientes (atribuídos)
- 1 Cliente : N Conversas
- 1 Conversa : N Interações

5. API ENDPOINTS

5.1. Autenticação

POST /api/auth/login
POST /api/auth/refresh
POST /api/auth/logout

5.2. WhatsApp

GET /api/whatsapp/status
POST /api/whatsapp/send
GET /api/whatsapp/conversas
GET /api/whatsapp/conversas/:id/mensagens

5.3. Fila Humana

GET /api/fila
POST /api/fila/:id/assumir
POST /api/fila/:id/liberar

5.4. CMS (Textos)

GET /api/cms/textos
POST /api/cms/textos
PUT /api/cms/textos/:id

5.5. Catálogo

GET /api/catalogo
POST /api/catalogo
PUT /api/catalogo/:id
DELETE /api/catalogo/:id

6. WEBSOCKET EVENTS

6.1. Eventos do Cliente → Servidor

```
javascript
{
  "type": "subscribe",
```

```
"channel": "fila_updates",
"filter": { "workspace_id": 1 }
}
```

6.2. Eventos do Servidor → Cliente

```
javascript
// Nova mensagem WhatsApp
{
  "event": "new_message",
  "data": {
    "clienteId": 123,
    "message": { ... }
  }
}
```

```
// Atualização de fila
{
  "event": "fila_update",
  "data": {
    "action": "added|removed|updated|assumed|released",
    "filaItem": { ... }
  }
}
```

```
// Notificação push
{
  "event": "new_notification",
  "data": {
    "notification": { ... }
  }
}
```

7. CONFIGURAÇÃO DE NICHOS

7.1. iPhone Store

```
json
{
  "nicho": "iphone_store",
  "etapas": ["consulta", "orçamento", "venda", "entrega"],
  "gatilhos": {
    "consulta": ["preço", "modelo", "estoque"],
    "orçamento": ["acessórios", "garantia"],
    "venda": ["pagamento", "entrega"]
  }
}
```

7.2. Advocacia

```
json
{
  "nicho": "law_firm",
  "etapas": ["triagem", "documentação", "agendamento", "followup"],
  "gatilhos": {
    "triagem": ["tipo_caso", "urgência"],
    "documentação": ["documentos", "provas"]
  }
}
```

8. REQUISITOS DE SISTEMA

8.1. Desenvolvimento

- Node.js 18+
- PostgreSQL 15+
- Redis 7+
- Docker + Docker Compose
- 8GB RAM mínimo
- 20GB de espaço livre

8.2. Produção

- Linux Ubuntu 20.04+
- SSL Certificate (Let's Encrypt)
- Backup automático diário
- Monitoramento (PM2 + Logging)
- CDN para arquivos estáticos

9. SEGURANÇA

9.1. Medidas Implementadas

- JWT com expiration curta (15min)
- Refresh tokens com revogação
- Rate limiting por IP
- SQL injection prevention
- XSS protection
- CORS configurado estritamente

9.2. Auditoria

- Log de todas as ações administrativas
- Histórico de login/logout
- Registro de alterações em dados sensíveis
- Backup de mensagens (retenção: 180 dias)

10. DEPLOY

10.1. Desenvolvimento Local

```
bash
git clone <repo>
docker-compose up -d
# Acesse: http://localhost:5173
```

10.2. Produção

```
bash
./deploy-production.sh
# Script inclui:
# 1. Build das imagens
# 2. Migrations do banco
# 3. Configuração SSL
# 4. Inicialização dos serviços
```

11. MANUTENÇÃO

11.1. Backup

```
bash
./scripts/backup.sh
# Gera backup de:
# - Banco de dados (dump SQL)
# - Arquivos de sessão WhatsApp
# - Logs do sistema
```

11.2. Monitoramento

- Health checks automáticos
- Alertas por email em caso de falha
- Dashboard de métricas em tempo real

- Logs centralizados

****Documentação Atualizada em:**** 15/01/2024
****Próxima Revisão:**** 15/04/2024
****Contato Técnico:**** tech@reistech.com
****Repositório:**** github.com/reistech/platform