**🚀 PRIMEM WHATSAPP - PROMPT TÉCNICO COMPLETO v16.0**

**📋 SITUAÇÃO ATUAL DETALHADA DO PROJETO**

**🎯 STATUS ATUAL: 95% MODULARIZADO + 10 FUNCIONALIDADES NOVAS**

* ✅ **Sistema FUNCIONANDO** - Login, chat, mensagens, arquivos operacional
* ✅ **58 arquivos modulares** criados com código funcional (926KB)
* ✅ **Estrutura modular completa** - Hooks, Contexts, Pages, Services prontos
* ❌ **App.js monolítico** (53KB) - Não refatorado para usar estrutura modular
* ❌ **31 arquivos vazios** (0 bytes) - Criados mas não implementados
* 🆕 **10 funcionalidades novas** definidas e especificadas tecnicamente

**🔍 REGRA CRÍTICA - VERSIONAMENTO DE ARQUIVOS:**

⚠️ SEMPRE analisar arquivo mais recente por data/hora:

📄 App.js (53KB - 12/09/2025 19:53) ← ATUAL EM USO

📄 App - Copia.js (97KB - 12/09/2025 18:19) ← BACKUP ANTIGO

🔧 PROCESSO:

1. Verificar data/hora de modificação

2. Usar o arquivo mais recente

3. Fazer backup antes de modificar

4. Nunca trabalhar com arquivos antigos

**🏗️ ESTRUTURA ATUAL COMPLETA MAPEADA**

**📁 CLIENT-SIDE - ESTRUTURA REAL DETALHADA**

C:\Users\david\Desktop\whatsapp\_sistem\client\src\

├── 📄 App.js (53KB - 12/09/2025 19:53) ❌ MONOLÍTICO

├── 📄 App - Copia.js (97KB - 12/09/2025 18:19) ❌ BACKUP

├── 📄 App.css (564 bytes) ✅

├── 📄 index.css (691 bytes) ✅

├── 📄 index.js (253 bytes) ✅

├── 📄 App.test.js (246 bytes) ✅

├── 📄 reportWebVitals.js (362 bytes) ✅

├── 📄 setupTests.js (241 bytes) ✅

│

├── 📁 components/ ✅ 31 ARQUIVOS COMPLETOS (459KB)

│ ├── 📁 auth/ ✅ 100% IMPLEMENTADO

│ │ ├── AuthGuard.js (15KB) ✅ - Proteção de rotas

│ │ ├── LoginForm.js (13KB) ✅ - Formulário de login

│ │ ├── UserProfile.js (21KB) ✅ - Perfil do usuário

│ │ └── index.js (3KB) ✅ - Exportações centralizadas

│ │

│ ├── 📁 chat/ ✅ 100% IMPLEMENTADO

│ │ ├── ChatHeader.js (16KB) ✅ - Cabeçalho do chat

│ │ ├── ChatList.js (22KB) ✅ - Lista de conversas

│ │ ├── ContactInfo.js (35KB) ✅ - Informações do contato

│ │ ├── MessageBubble.js (18KB) ✅ - Bolha de mensagem

│ │ ├── MessageInput.js (14KB) ✅ - Input de mensagem

│ │ └── MessageList.js (21KB) ✅ - Lista de mensagens

│ │

│ ├── 📁 layout/ ✅ 100% IMPLEMENTADO

│ │ ├── Footer.js (14KB) ✅ - Rodapé do sistema

│ │ ├── Header.js (17KB) ✅ - Cabeçalho principal

│ │ ├── Navigation.js (15KB) ✅ - Navegação principal

│ │ ├── Sidebar.js (25KB) ✅ - Barra lateral

│ │ └── index.js (10KB) ✅ - Exportações

│ │

│ ├── 📁 media/ ✅ 100% IMPLEMENTADO

│ │ ├── AudioPlayer.js (10KB) ✅ - Player de áudio

│ │ ├── AudioRecorder.js (10KB) ✅ - Gravador de áudio

│ │ └── ImageViewer.js (14KB) ✅ - Visualizador de imagem

│ │

│ ├── 📁 modals/ ✅ 100% IMPLEMENTADO

│ │ ├── AdvancedSettingsModal.js (21KB) ✅ - Config avançadas

│ │ ├── EditNameModal.js (11KB) ✅ - Editar nome

│ │ ├── KnowledgeModal.js (22KB) ✅ - Base de conhecimento

│ │ ├── ScheduleModal.js (18KB) ✅ - Agendamento

│ │ └── UserProfileModal.js (9KB) ✅ - Perfil modal

│ │

│ ├── 📁 ui/ ✅ 100% IMPLEMENTADO

│ │ ├── Avatar.js (11KB) ✅ - Avatar do usuário

│ │ ├── Button.js (7KB) ✅ - Botão personalizado

│ │ ├── EmojiPicker.js (12KB) ✅ - Seletor de emoji

│ │ ├── Input.js (13KB) ✅ - Input personalizado

│ │ ├── Loading.js (11KB) ✅ - Componente de loading

│ │ ├── Modal.js (12KB) ✅ - Modal base

│ │ └── SearchBar.js (16KB) ✅ - Barra de pesquisa

│ │

│ └── MessageStatus.js (1KB) ✅ - Status da mensagem

│

├── 📁 hooks/ ✅ 8 ARQUIVOS COMPLETOS (125KB)

│ ├── useAuth.js (10KB) ✅ - Autenticação

│ ├── useFileUpload.js (19KB) ✅ - Upload de arquivos

│ ├── useMessages.js (17KB) ✅ - Gerenciamento de mensagens

│ ├── useNotifications.js (18KB) ✅ - Notificações

│ ├── useSettings.js (13KB) ✅ - Configurações

│ ├── useSocket.js (14KB) ✅ - Socket.IO

│ ├── useWhatsApp.js (19KB) ✅ - WhatsApp principal

│ └── useWhatsAppFixed.js (15KB) ✅ - WhatsApp corrigido

│

├── 📁 contexts/ ✅ 3 ARQUIVOS COMPLETOS (50KB)

│ ├── AuthContext.js (18KB) ✅ - Contexto de autenticação

│ ├── SettingsContext.js (17KB) ✅ - Contexto de configurações

│ └── WhatsAppContext.js (15KB) ✅ - Contexto WhatsApp

│

├── 📁 pages/ ✅ 6 ARQUIVOS COMPLETOS (115KB)

│ ├── ChatPage.js (22KB) ✅ - Página principal do chat

│ ├── ConnectionPage.js (23KB) ✅ - Página de conexão

│ ├── DashboardPage.js (20KB) ✅ - Dashboard principal

│ ├── LoginPage.js (16KB) ✅ - Página de login

│ ├── SettingsPage.js (30KB) ✅ - Página de configurações

│ └── index.js (4KB) ✅ - Exportações centralizadas

│

├── 📁 layouts/ ✅ 4 ARQUIVOS COMPLETOS (56KB)

│ ├── AuthLayout.js (13KB) ✅ - Layout de autenticação

│ ├── ChatLayout.js (22KB) ✅ - Layout do chat

│ ├── DashboardLayout.js (13KB) ✅ - Layout principal

│ └── index.js (8KB) ✅ - Exportações

│

├── 📁 services/ ✅ 6 ARQUIVOS COMPLETOS (66KB)

│ ├── apiService.js (7KB) ✅ - Cliente HTTP

│ ├── authService.js (10KB) ✅ - Serviços de auth

│ ├── fileService.js (18KB) ✅ - Serviços de arquivo

│ ├── messageService.js (14KB) ✅ - Serviços de mensagem

│ ├── whatsappService.js (11KB) ✅ - Serviços WhatsApp

│ └── index.js (6KB) ✅ - Exportações

│

├── 📁 constants/ ❌ 3 ARQUIVOS VAZIOS (0KB)

│ ├── apiEndpoints.js ❌ VAZIO - URLs das APIs

│ ├── config.js ❌ VAZIO - Configurações gerais

│ └── themes.js ❌ VAZIO - Temas e cores

│

├── 📁 utils/ ❌ 4 DE 5 ARQUIVOS VAZIOS

│ ├── constants.js ❌ VAZIO - Constantes do sistema

│ ├── formatUtils.js ❌ VAZIO - Formatação

│ ├── messageUtils.js ❌ VAZIO - Utils de mensagem

│ ├── validationUtils.js ❌ VAZIO - Validações

│ └── NotificationManager.js (5KB) ✅ - Gerenciador

│

└── 📁 styles/ ❌ 4 ARQUIVOS VAZIOS (0KB)

├── components.css ❌ VAZIO - Estilos de componentes

├── global.css ❌ VAZIO - Estilos globais

├── themes.css ❌ VAZIO - Temas CSS

└── variables.css ❌ VAZIO - Variáveis CSS

**📁 SERVER-SIDE - ESTRUTURA REAL DETALHADA**

C:\Users\david\Desktop\whatsapp\_sistem\server\

├── 📄 server.js (18KB) ✅ MONOLÍTICO MAS FUNCIONAL

│

├── 📁 config/

│ └── database.js (3KB) ✅ - Configuração do banco

│

├── 📁 services/ ✅ 2 ARQUIVOS COMPLETOS (55KB)

│ ├── message.service.js (35KB) ✅ - Serviço de mensagens

│ └── whatsapp.service.js (20KB) ✅ - Serviço WhatsApp

│

├── 📁 media/ ✅ 2 ARQUIVOS COMPLETOS (27KB)

│ ├── AudioPlayer.js (13KB) ✅ - Player servidor

│ └── AudioRecorder.js (14KB) ✅ - Gravador servidor

│

├── 📁 utils/ ❌ 4 DE 5 ARQUIVOS VAZIOS

│ ├── encryption.js ❌ VAZIO - Criptografia

│ ├── helpers.js ❌ VAZIO - Funções auxiliares

│ ├── validators.js ❌ VAZIO - Validadores

│ └── logger.js (7KB) ✅ - Sistema de logs

│

├── 📁 controllers/ ❌ 5 ARQUIVOS VAZIOS (0KB)

│ ├── authController.js ❌ VAZIO - Controle de auth

│ ├── fileController.js ❌ VAZIO - Controle de arquivos

│ ├── messageController.js ❌ VAZIO - Controle de mensagens

│ ├── userController.js ❌ VAZIO - Controle de usuários

│ └── whatsappController.js ❌ VAZIO - Controle WhatsApp

│

├── 📁 middleware/ ❌ 5 ARQUIVOS VAZIOS (0KB)

│ ├── auth.js ❌ VAZIO - Middleware de auth

│ ├── errorHandler.js ❌ VAZIO - Tratamento de erros

│ ├── fileUpload.js ❌ VAZIO - Upload de arquivos

│ ├── rateLimit.js ❌ VAZIO - Rate limiting

│ └── validation.js ❌ VAZIO - Validação

│

└── 📁 routes/ ❌ 6 ARQUIVOS VAZIOS (0KB)

├── auth.js ❌ VAZIO - Rotas de auth

├── files.js ❌ VAZIO - Rotas de arquivos

├── index.js ❌ VAZIO - Rota principal

├── messages.js ❌ VAZIO - Rotas de mensagens

├── users.js ❌ VAZIO - Rotas de usuários

└── whatsapp.js ❌ VAZIO - Rotas WhatsApp

**📁 ARQUIVOS DE CONFIGURAÇÃO**

C:\Users\david\Desktop\whatsapp\_sistem\

├── 📄 package.json (793 bytes) ✅ - Dependências servidor

├── 📄 .env (45 bytes) ✅ - Variáveis ambiente

├── 📄 database.sqlite (57KB) ✅ - Banco de dados

├── 📄 LOGO.png (23KB) ✅ - Logo PRIMEM

├── 📁 uploads/ ✅ - Arquivos enviados

├── 📁 sessions/ ✅ - Sessões WhatsApp

├── 📁 logs/ ✅ - Logs do sistema

└── 📁 client/package.json (627 bytes) ✅ - Deps React

**🎯 ESPECIFICAÇÃO TÉCNICA COMPLETA - 30 FUNCIONALIDADES**

**✅ FUNCIONALIDADES BÁSICAS (JÁ IMPLEMENTADAS - 6)**

**1. AUTENTICAÇÃO COMPLETA**

javascript

*// ATUAL: useAuth.js (10KB) ✅ IMPLEMENTADO*

const useAuth = () => {

*// Estados: user, isLoggedIn, loading, error*

*// Funções: login, logout, checkAuth, refreshToken*

*// Persistência: localStorage, session management*

*// Usuários teste: teste@teste.com/123, admin@primem.com/admin123*

}

*// LOCALIZAÇÃO:*

*// - Hook: client/src/hooks/useAuth.js*

*// - Context: client/src/contexts/AuthContext.js*

*// - Page: client/src/pages/LoginPage.js*

*// - Components: client/src/components/auth/*

**2. CONEXÃO WHATSAPP**

javascript

*// ATUAL: useWhatsApp.js (19KB) ✅ IMPLEMENTADO*

const useWhatsApp = () => {

*// Estados: isConnected, qrCode, clientInfo, status*

*// Funções: connect, disconnect, sendMessage, getChats*

*// Eventos: onQR, onReady, onMessage, onDisconnected*

*// QR Code: Base64 generation, display, refresh*

}

*// LOCALIZAÇÃO:*

*// - Hook: client/src/hooks/useWhatsApp.js*

*// - Context: client/src/contexts/WhatsAppContext.js*

*// - Service: server/services/whatsapp.service.js*

*// - Page: client/src/pages/ConnectionPage.js*

**3. SISTEMA DE MENSAGENS**

javascript

*// ATUAL: useMessages.js (17KB) ✅ IMPLEMENTADO*

const useMessages = () => {

*// Estados: messages, conversations, selectedChat*

*// Funções: sendText, sendMedia, markRead, loadHistory*

*// Socket: real-time sync, typing indicators*

*// Media: images, videos, audio, documents*

}

*// LOCALIZAÇÃO:*

*// - Hook: client/src/hooks/useMessages.js*

*// - Service: server/services/message.service.js*

*// - Components: client/src/components/chat/*

*// - Page: client/src/pages/ChatPage.js*

**4. UPLOAD/DOWNLOAD DE ARQUIVOS**

javascript

*// ATUAL: useFileUpload.js (19KB) ✅ IMPLEMENTADO*

const useFileUpload = () => {

*// Funções: uploadFile, validateFile, compressImage*

*// Suporte: images, videos, audio, documents (16MB max)*

*// Preview: image/video preview before send*

*// Progress: upload progress tracking*

}

*// LOCALIZAÇÃO:*

*// - Hook: client/src/hooks/useFileUpload.js*

*// - Service: client/src/services/fileService.js*

*// - Server: multer configuration in server.js*

**5. INTERFACE DE CHAT COMPLETA**

javascript

*// ATUAL: components/chat/ (126KB) ✅ IMPLEMENTADO*

*// - ChatHeader.js: Nome, status, ações do contato*

*// - ChatList.js: Lista de conversas com busca*

*// - MessageBubble.js: Bolhas de mensagem (sent/received)*

*// - MessageInput.js: Input com emoji, anexos, gravação*

*// - MessageList.js: Lista infinita de mensagens*

*// - ContactInfo.js: Detalhes do contato/grupo*

**6. COMPONENTES UI BÁSICOS**

javascript

*// ATUAL: components/ui/ (83KB) ✅ IMPLEMENTADO*

*// - Button.js: Botões personalizados*

*// - Input.js: Inputs com validação*

*// - Modal.js: Modais reutilizáveis*

*// - Loading.js: Componentes de carregamento*

*// - Avatar.js: Avatares de usuário*

*// - EmojiPicker.js: Seletor de emojis*

*// - SearchBar.js: Barra de pesquisa*

**🚧 FUNCIONALIDADES INTERMEDIÁRIAS (PARCIAIS - 5)**

**7. SISTEMA DE ÁUDIO**

javascript

*// ATUAL: components/media/ (33KB) ✅ BÁSICO IMPLEMENTADO*

*// FALTA IMPLEMENTAR:*

*// - Visualização de ondas sonoras*

*// - Controles avançados (velocidade, volume)*

*// - Compressão automática de áudio*

*// - Transcrição de áudio (opcional)*

*// - Player com timeline interativa*

*// LOCALIZAÇÃO:*

*// - Client: client/src/components/media/AudioPlayer.js*

*// - Client: client/src/components/media/AudioRecorder.js*

*// - Server: server/media/AudioPlayer.js, AudioRecorder.js*

**8. SISTEMA DE NOTIFICAÇÕES**

javascript

*// ATUAL: utils/NotificationManager.js (5KB) ✅ BÁSICO*

*// FALTA IMPLEMENTAR:*

*// - Notificações desktop nativas*

*// - Sons personalizados por tipo*

*// - Centro de notificações*

*// - Notificações por email (opcional)*

*// - Configurações de notificação por usuário*

*// Hook: client/src/hooks/useNotifications.js (18KB) ✅*

**9. CONFIGURAÇÕES DE SISTEMA**

javascript

*// ATUAL: SettingsContext.js (17KB) ✅ ESTRUTURA PRONTA*

*// FALTA IMPLEMENTAR:*

*// - Interface completa de configurações*

*// - Configurações por usuário/grupo*

*// - Import/export de configurações*

*// - Reset para padrão*

*// LOCALIZAÇÃO:*

*// - Context: client/src/contexts/SettingsContext.js*

*// - Hook: client/src/hooks/useSettings.js*

*// - Page: client/src/pages/SettingsPage.js (30KB) ✅*

**10. SISTEMA DE TEMAS**

javascript

*// ATUAL: Cores PRIMEM definidas no código*

*// FALTA IMPLEMENTAR:*

*// - Aplicação completa do tema em todos componentes*

*// - Modo escuro/claro funcional*

*// - CSS variables para cores*

*// - Transições suaves entre temas*

*// CORES PRIMEM OFICIAIS:*

*// --primem-primary: #2B4C8C (azul principal)*

*// --primem-secondary: #C97A4A (laranja secundário)*

*// --primem-accent: #8B9DC3 (azul claro)*

**11. LAYOUTS MODULARES**

javascript

*// ATUAL: layouts/ (56KB) ✅ CRIADOS MAS NÃO INTEGRADOS*

*// - AuthLayout.js: Layout para login/registro*

*// - DashboardLayout.js: Layout principal do sistema*

*// - ChatLayout.js: Layout específico do chat*

*// FALTA: Integração com React Router e páginas*

**❌ FUNCIONALIDADES AVANÇADAS (NÃO IMPLEMENTADAS - 17)**

**12. DASHBOARD COM ESTATÍSTICAS**

javascript

*// ESPECIFICAÇÃO TÉCNICA:*

const dashboardFeatures = {

charts: {

messagesPerDay: 'Gráfico de mensagens por dia',

responseTime: 'Tempo médio de resposta',

contactsActivity: 'Atividade de contatos',

fileTypes: 'Tipos de arquivo mais enviados'

},

widgets: {

totalMessages: 'Total de mensagens hoje/semana/mês',

activeChats: 'Conversas ativas no momento',

pendingMessages: 'Mensagens não respondidas',

onlineUsers: 'Usuários online no sistema'

},

reports: {

daily: 'Relatório diário automático',

weekly: 'Relatório semanal',

monthly: 'Relatório mensal',

custom: 'Relatórios personalizados por período'

}

}

*// DEPENDÊNCIAS:*

*// - Chart.js ou Recharts para gráficos*

*// - Cálculos de estatísticas no backend*

*// - Cache de dados para performance*

*// - Filtros por usuário/departamento/período*

**13. GERENCIAMENTO DE MÚLTIPLOS USUÁRIOS**

javascript

*// ESPECIFICAÇÃO TÉCNICA:*

const userManagement = {

roles: {

admin: {

permissions: ['all'],

description: 'Acesso total ao sistema'

},

supervisor: {

permissions: ['view\_all\_chats', 'manage\_users', 'view\_reports'],

description: 'Supervisiona atendimento'

},

agent: {

permissions: ['view\_assigned\_chats', 'send\_messages', 'upload\_files'],

description: 'Agente de atendimento'

},

viewer: {

permissions: ['view\_assigned\_chats'],

description: 'Apenas visualização'

}

},

features: {

userCrud: 'Criar, editar, deletar usuários',

roleAssignment: 'Atribuir roles e permissões',

activityLog: 'Log de atividades por usuário',

sessionManagement: 'Controle de sessões ativas',

departmentAssignment: 'Usuários por departamento'

}

}

*// ESTRUTURA DE DADOS:*

const user = {

id: 'uuid',

name: 'Nome Completo',

email: 'email@empresa.com',

password: 'hashed\_password',

role: 'admin|supervisor|agent|viewer',

department: 'comercial|operacional|financeiro|suporte',

permissions: ['permission1', 'permission2'],

isActive: true,

lastLogin: Date,

createdAt: Date,

settings: {

notifications: true,

theme: 'light|dark',

language: 'pt-BR'

}

}

**14. SISTEMA DE ASSINATURA AUTOMÁTICA**

javascript

*// ESPECIFICAÇÃO TÉCNICA:*

const signatureSystem = {

globalSignature: {

enabled: true,

template: 'Atenciosamente,\n{userName}\n{userDepartment}\n{companyName}\n{companyPhone}',

applyTo: 'all|department|user'

},

userSignatures: {

[userId]: {

enabled: true,

customTemplate: 'Minha assinatura personalizada\n{userName}',

overrideGlobal: true

}

},

variables: {

userName: 'Nome do usuário atual',

userEmail: 'Email do usuário',

userDepartment: 'Departamento do usuário',

companyName: 'PRIMEM COMEX',

companyPhone: '+55 11 99999-9999',

currentDate: 'Data atual',

currentTime: 'Hora atual'

},

rules: {

autoAppend: true, *// Adicionar automaticamente*

manualOverride: true, *// Permitir remover manualmente*

onlyFirstMessage: false, *// Só na primeira mensagem*

exceptReplies: true *// Não aplicar em respostas*

}

}

**15. BACKUP E RESTORE**

javascript

*// ESPECIFICAÇÃO TÉCNICA:*

const backupSystem = {

automatic: {

enabled: true,

frequency: 'daily|weekly|monthly',

time: '02:00', *// 2h da manhã*

retention: 30, *// Manter 30 backups*

location: './backups/'

},

manual: {

fullBackup: 'Backup completo (BD + arquivos + configurações)',

configOnly: 'Apenas configurações',

messagesOnly: 'Apenas mensagens',

filesOnly: 'Apenas arquivos'

},

restore: {

selective: 'Restaurar partes específicas',

preview: 'Preview antes de restaurar',

migration: 'Migração entre versões'

},

content: {

database: 'SQLite database completo',

uploads: 'Pasta uploads/',

sessions: 'Sessões WhatsApp',

configurations: 'Configurações do sistema',

userSettings: 'Configurações por usuário'

}

}

**16. AGENDAMENTO DE MENSAGENS**

javascript

*// ESPECIFICAÇÃO TÉCNICA:*

const schedulingSystem = {

types: {

single: 'Mensagem única em data/hora específica',

recurring: 'Mensagem recorrente (diária, semanal, mensal)',

conditional: 'Baseada em eventos (novo cliente, aniversário)',

bulk: 'Envio em massa para múltiplos contatos'

},

interface: {

calendar: 'Visualização em calendário',

list: 'Lista de mensagens agendadas',

templates: 'Templates de mensagem',

preview: 'Preview da mensagem antes de agendar'

},

features: {

timezone: 'Suporte a fusos horários',

retry: 'Retry automático se falhar',

confirmation: 'Confirmação de envio',

analytics: 'Estatísticas de mensagens agendadas'

}

}

*// ESTRUTURA DE DADOS:*

const scheduledMessage = {

id: 'uuid',

contactId: 'whatsapp\_contact\_id',

message: 'Texto da mensagem',

scheduledFor: Date,

type: 'single|recurring|conditional|bulk',

status: 'pending|sent|failed|cancelled',

createdBy: 'user\_id',

recurrence: {

frequency: 'daily|weekly|monthly',

interval: 1, *// A cada 1 dia/semana/mês*

endDate: Date

}

}

**17. AUTO-RESPOSTA CONFIGURÁVEL**

javascript

*// ESPECIFICAÇÃO TÉCNICA:*

const autoResponseSystem = {

triggers: {

firstMessage: 'Primeira mensagem do contato',

keywords: ['ola', 'oi', 'bom dia', 'ajuda'],

businessHours: 'Fora do horário comercial',

absence: 'Quando usuário está ausente'

},

responses: {

welcome: 'Mensagem de boas-vindas',

businessHours: 'Mensagem de horário comercial',

absence: 'Mensagem de ausência',

keywords: 'Respostas específicas por palavra-chave'

},

config: {

enabled: true,

delay: 2000, *// 2 segundos de delay para parecer humano*

onlyFirstTime: true, *// Só responder uma vez por contato*

workingHours: {

start: '08:00',

end: '18:00',

timezone: 'America/Sao\_Paulo',

workDays: [1, 2, 3, 4, 5] *// Segunda a sexta*

}

}

}

**🆕 FUNCIONALIDADES NOVAS (NÃO IMPLEMENTADAS - 10)**

**21. CONFIGURAÇÕES DE APARÊNCIA**

javascript

*// ESPECIFICAÇÃO TÉCNICA COMPLETA:*

const appearanceSystem = {

logo: {

upload: {

acceptedFormats: ['png', 'jpg', 'jpeg', 'svg'],

maxSize: '2MB',

dimensions: {

min: { width: 100, height: 40 },

max: { width: 500, height: 200 }

},

validation: 'Aspect ratio, file size, format'

},

display: {

loginPage: {

maxWidth: 200,

maxHeight: 80,

position: 'center-top'

},

systemHeader: {

maxWidth: 120,

maxHeight: 40,

position: 'left'

},

responsive: {

mobile: 'Scale appropriately',

tablet: 'Maintain proportions'

}

},

fallback: {

textMode: 'Use company name as text',

fontSize: 'Dynamic based on space',

fontFamily: 'PRIMEM brand font'

}

},

colors: {

customizable: {

primary: '#2B4C8C', *// Cor principal*

secondary: '#C97A4A', *// Cor secundária*

accent: '#8B9DC3', *// Cor de destaque*

background: '#f0f2f5', *// Fundo*

text: '#333333', *// Texto principal*

border: '#dee2e6' *// Bordas*

},

presets: [

{ name: 'PRIMEM Original', colors: {...} },

{ name: 'Dark Mode', colors: {...} },

{ name: 'High Contrast', colors: {...} }

],

preview: 'Real-time preview in interface',

cssVariables: 'Generate CSS custom properties',

export: 'Export theme configuration'

}

}

*// IMPLEMENTAÇÃO:*

*// 1. Component: ThemeCustomizer*

*// 2. Service: themeService.js*

*// 3. Utils: logoUtils.js (resize, validate)*

*// 4. Storage: theme config in localStorage/database*

*// 5. CSS: CSS custom properties system*

**22. MARCAR COMO NÃO LIDA**

javascript

*// ESPECIFICAÇÃO TÉCNICA:*

const unreadSystem = {

functionality: {

markUnread: 'Marcar conversa como não lida manualmente',

visualIndicator: 'Indicador visual diferenciado',

counter: 'Contador de não lidas incluindo manuais',

persistence: 'Manter estado entre sessões'

},

implementation: {

database: {

conversation: {

id: 'conversation\_id',

unreadCount: 'number', *// Mensagens não lidas reais*

manuallyMarked: 'boolean', *// Marcada manualmente*

lastReadAt: 'timestamp',

manuallyMarkedAt: 'timestamp'

}

},

interface: {

contextMenu: 'Botão direito -> Marcar como não lida',

button: 'Botão na conversa aberta',

keyboard: 'Atalho Ctrl+U',

indicator: 'Ícone diferenciado para manual vs real'

}

},

rules: {

priority: 'Manual override tem prioridade',

clearing: 'Como limpar marca manual (ao ler ou manual)',

notifications: 'Notificar sobre conversas marcadas'

}

}

**23. ABA LATERAL DE ARQUIVOS**

javascript

*// ESPECIFICAÇÃO TÉCNICA COMPLETA:*

const fileSidebarSystem = {

layout: {

position: 'right',

defaultWidth: 300,

minWidth: 250,

maxWidth: 500,

resizable: true,

collapsible: true,

persistState: 'Remember expanded/collapsed'

},

content: {

fileList: {

groupBy: 'date|type|sender',

sortBy: 'date|name|size|type',

filters: {

type: ['image', 'video', 'audio', 'document'],

dateRange: 'Custom date range picker',

size: 'File size range',

sender: 'Filter by who sent'

}

},

preview: {

images: 'Thumbnail preview',

videos: 'Video thumbnail + duration',

documents: 'File icon + name + size',

audio: 'Waveform preview'

},

actions: {

download: 'Download file',

navigateToMessage: 'Scroll to original message',

forward: 'Forward to another chat',

delete: 'Delete file (with confirmation)'

}

},

navigation: {

toMessage: {

scrollTo: 'Scroll chat to specific message',

highlight: 'Highlight message temporarily',

context: 'Show messages before/after',

animation: 'Smooth scroll animation'

}

}

}

*// ESTRUTURA DE DADOS:*

const fileEntry = {

id: 'file\_id',

messageId: 'message\_id\_reference',

fileName: 'document.pdf',

originalName: 'Original file name',

fileType: 'pdf|jpg|mp4|mp3',

mimeType: 'application/pdf',

fileSize: 1024000, *// bytes*

thumbnailUrl: '/thumbnails/thumb\_123.jpg',

downloadUrl: '/api/files/download/123',

sentAt: Date,

sentBy: 'contact\_id|user\_id',

isReceived: false, *// false = sent by us*

conversationId: 'conversation\_id'

}

**24. SISTEMA DE IDs BITRIX**

javascript

*// ESPECIFICAÇÃO TÉCNICA COMPLETA:*

const bitrixIdSystem = {

dataStructure: {

bitrixId: {

id: '11679', *// ID do deal/lead no Bitrix*

title: 'Nome do Deal (opcional)',

status: 'active|finished',

type: 'deal|lead|contact',

link: 'https://primem.bitrix24.com.br/crm/deal/details/11679/',

createdAt: Date,

updatedAt: Date,

createdBy: 'user\_id',

description: 'Descrição opcional'

},

association: {

conversationId: 'whatsapp\_conversation\_id',

bitrixId: '11679',

associatedAt: Date,

associatedBy: 'user\_id',

isAutomatic: true, *// Auto-associado ou manual*

confidence: 0.95 *// Confiança da associação automática*

}

},

features: {

crud: {

create: 'Criar novo ID manualmente',

read: 'Listar IDs (ativos/finalizados)',

update: 'Editar ID existente',

delete: 'Remover ID (com confirmação)'

},

association: {

manual: 'Associar ID a conversa manualmente',

automatic: 'Auto-associar baseado em histórico',

validation: 'Validar se ID existe no Bitrix',

conflict: 'Resolver conflitos de associação'

},

interface: {

sidebar: 'Lista de IDs na sidebar',

conversationHeader: 'ID associado no header',

quickSelect: 'Seletor rápido de ID',

search: 'Buscar por ID ou título'

}

},

automation: {

autoAssociation: {

rules: {

sameContact: 'Mesmo contato = usar última ID',

singleId: 'Contato com única ID = auto-associar',

multipleIds: 'Múltiplas IDs = perguntar ao usuário',

newContact: 'Novo contato = criar nova ID ou associar existente'

},

algorithm: {

contactHistory: 'Verificar histórico do contato',

timeWindow: 'Considerar IDs dos últimos X dias',

confidence: 'Calcular confiança da associação',

fallback: 'Ação quando não conseguir decidir'

}

}

}

}

*// IMPLEMENTAÇÃO:*

*// 1. Database: tabelas bitrix\_ids, id\_associations*

*// 2. API: endpoints para CRUD de IDs*

*// 3. Components: IdSelector, IdManager, IdDisplay*

*// 4. Services: bitrixService.js para validação*

*// 5. Hooks: useBitrixIds.js para gerenciamento*

**25. PESQUISA AVANÇADA COM FILTROS**

javascript

*// ESPECIFICAÇÃO TÉCNICA COMPLETA:*

const advancedSearchSystem = {

interface: {

searchBar: {

expandable: 'Clique na seta para expandir filtros',

placeholder: 'Buscar em conversas, arquivos, IDs...',

autocomplete: 'Sugestões baseadas em histórico',

shortcuts: 'Ctrl+F para focar'

},

filters: {

location: {

conversations: 'Buscar em mensagens de texto',

activeBitrixIds: 'Buscar em IDs ativos',

finishedBitrixIds: 'Buscar em IDs finalizados',

files: 'Buscar em nomes de arquivos',

all: 'Buscar em todos os locais'

},

content: {

exactPhrase: 'Frase exata (entre aspas)',

anyWord: 'Qualquer palavra',

allWords: 'Todas as palavras',

regex: 'Expressão regular (usuários avançados)'

},

time: {

dateRange: 'Período específico',

lastHour: 'Última hora',

today: 'Hoje',

yesterday: 'Ontem',

thisWeek: 'Esta semana',

thisMonth: 'Este mês',

custom: 'Período personalizado'

},

media: {

fileType: ['image', 'video', 'audio', 'document'],

fileSize: 'Tamanho do arquivo',

hasMedia: 'Mensagens com/sem mídia'

},

contact: {

specific: 'Contato específico',

group: 'Grupos apenas',

individual: 'Conversas individuais'

}

},

results: {

grouping: 'Agrupar por tipo (conversa, arquivo, ID)',

sorting: 'Ordenar por relevância/data/nome',

highlighting: 'Destacar termo buscado',

pagination: 'Paginar resultados longos',

export: 'Exportar resultados da busca'

}

},

implementation: {

backend: {

indexing: 'Índices otimizados para busca',

fullText: 'Full-text search no SQLite/PostgreSQL',

caching: 'Cache de buscas frequentes',

performance: 'Otimização para grandes volumes'

},

frontend: {

debouncing: 'Delay de 300ms para evitar spam',

loading: 'Indicadores de carregamento',

emptyState: 'Estado quando não há resultados',

errorHandling: 'Tratamento de erros de busca'

}

}

}

*// ESTRUTURA DE DADOS:*

const searchQuery = {

query: 'texto a buscar',

filters: {

locations: ['conversations', 'files', 'activeBitrixIds'],

dateRange: { from: Date, to: Date },

fileTypes: ['image', 'video'],

contactIds: ['contact1', 'contact2'],

exactMatch: false,

caseSensitive: false,

includeArchived: false

},

pagination: {

page: 1,

limit: 20

},

sorting: {

field: 'relevance|date|name',

order: 'desc|asc'

}

}

const searchResult = {

type: 'message|file|bitrixId',

id: 'result\_id',

title: 'Result title',

excerpt: 'Text excerpt with highlighted terms',

conversationId: 'related\_conversation',

timestamp: Date,

relevanceScore: 0.95,

metadata: {

*// Dados específicos por tipo*

messageId: 'for messages',

fileName: 'for files',

bitrixId: 'for bitrix IDs'

}

}

**26. HISTÓRICO POR ID BITRIX**

javascript

*// ESPECIFICAÇÃO TÉCNICA:*

const bitrixHistorySystem = {

functionality: {

grouping: 'Agrupar todas as mensagens de uma ID',

timeline: 'Linha do tempo de interações',

statistics: 'Estatísticas por ID (mensagens, arquivos, duração)',

export: 'Exportar histórico da ID para PDF/Excel'

},

interface: {

idView: {

header: 'Informações da ID (título, status, link)',

timeline: 'Timeline de todas as mensagens',

participants: 'Todos os contatos relacionados à ID',

files: 'Todos os arquivos relacionados',

statistics: 'Stats da ID (duração, mensagens, etc.)'

},

navigation: {

fromChat: 'Ver histórico da ID atual',

fromIdList: 'Clicar em ID para ver histórico',

breadcrumbs: 'Navegação clara'

}

},

reports: {

summary: 'Resumo executivo da ID',

detailed: 'Relatório detalhado',

timeline: 'Timeline visual',

export: 'Múltiplos formatos (PDF, Excel, CSV)'

}

}

**27. SISTEMA DE CHATBOT/ROBÔ**

javascript

*// ESPECIFICAÇÃO TÉCNICA COMPLETA:*

const chatbotSystem = {

engine: {

states: {

INACTIVE: 'Robô desativado para este contato',

WAITING\_MENU: 'Aguardando seleção do menu',

PROCESSING\_OPTION: 'Processando opção escolhida',

ROUTING: 'Redirecionando para departamento',

TRANSFERRED: 'Transferido para atendente humano',

FALLBACK: 'Opção inválida, reapresentando menu'

},

transitions: {

newContact: 'INACTIVE -> WAITING\_MENU',

validOption: 'WAITING\_MENU -> PROCESSING\_OPTION',

invalidOption: 'WAITING\_MENU -> FALLBACK',

routingComplete: 'PROCESSING\_OPTION -> TRANSFERRED',

fallbackComplete: 'FALLBACK -> WAITING\_MENU'

}

},

configuration: {

menu: {

welcomeMessage: {

default: '🤖 Olá! Sou o assistente virtual da PRIMEM COMEX.\n\nComo posso ajudá-lo?\n\n1️⃣ Comercial\n2️⃣ Operacional\n3️⃣ Financeiro\n4️⃣ Suporte Técnico\n0️⃣ Falar com Atendente\n\nDigite o número da opção desejada:',

customizable: true,

variables: ['{companyName}', '{currentTime}', '{contactName}']

},

options: {

'1': {

label: 'Comercial',

department: 'comercial',

users: ['user1@primem.com', 'user2@primem.com'],

message: 'Redirecionando você para nossa equipe comercial...'

},

'2': {

label: 'Operacional',

department: 'operacional',

users: ['user3@primem.com'],

message: 'Conectando com operacional...'

},

'0': {

label: 'Atendente Geral',

department: 'geral',

users: ['supervisor@primem.com'],

message: 'Transferindo para atendente...'

}

},

fallbackMessage: '❌ Opção inválida.\n\nPor favor, digite apenas o número da opção desejada (1, 2, 3, 4 ou 0).',

maxRetries: 3,

afterMaxRetries: 'transfer\_to\_human'

},

scheduling: {

workingHours: {

timezone: 'America/Sao\_Paulo',

monday: { start: '08:00', end: '18:00', enabled: true },

tuesday: { start: '08:00', end: '18:00', enabled: true },

wednesday: { start: '08:00', end: '18:00', enabled: true },

thursday: { start: '08:00', end: '18:00', enabled: true },

friday: { start: '08:00', end: '18:00', enabled: true },

saturday: { start: '08:00', end: '12:00', enabled: true },

sunday: { enabled: false }

},

offHoursMessage: '🕒 Nosso atendimento funciona de segunda a sexta das 8h às 18h, e sábados das 8h às 12h.\n\nDeixe sua mensagem que retornaremos assim que possível.',

holidayMessage: '🎉 Hoje é feriado. Retornaremos no próximo dia útil.'

},

personalization: {

contactNameDetection: 'Detectar nome do contato automaticamente',

repeatCustomerHandling: 'Tratamento diferenciado para clientes recorrentes',

contextAwareness: 'Considerar histórico de conversas anteriores'

}

},

specialGroup: {

name: 'Robô',

description: 'Grupo para monitorar interações do chatbot',

members: ['admin@primem.com', 'supervisor@primem.com'],

permissions: {

viewAllBotInteractions: true,

modifyBotSettings: true,

overrideBotDecisions: true,

accessBotAnalytics: true

},

notifications: {

newBotInteraction: false,

failedRouting: true,

maxRetriesReached: true,

botErrors: true

}

},

analytics: {

metrics: {

totalInteractions: 'Total de interações com o bot',

optionDistribution: 'Distribuição de opções escolhidas',

conversionRate: 'Taxa de conversão para atendente',

averageResolutionTime: 'Tempo médio até transferência',

failureRate: 'Taxa de falhas (max retries atingido)'

},

reports: {

daily: 'Relatório diário de atividade do bot',

weekly: 'Relatório semanal com insights',

monthly: 'Relatório mensal completo'

}

}

}

*// ESTRUTURA DE DADOS:*

const botConversation = {

id: 'bot\_conversation\_id',

contactId: 'whatsapp\_contact\_id',

currentState: 'WAITING\_MENU',

startedAt: Date,

lastInteraction: Date,

selectedOption: null,

retryCount: 0,

assignedDepartment: null,

assignedUsers: [],

transferredAt: null,

transferredTo: 'user\_id',

isActive: true,

history: [

{

timestamp: Date,

state: 'WAITING\_MENU',

message: 'Menu sent to user',

response: 'User typed "1"'

}

]

}

**28. AUTO-RESPOSTA PARA LIGAÇÕES**

javascript

*// ESPECIFICAÇÃO TÉCNICA COMPLETA:*

const callAutoResponseSystem = {

detection: {

whatsappEvents: {

call\_received: 'Chamada recebida',

call\_missed: 'Chamada perdida',

call\_rejected: 'Chamada rejeitada pelo destinatário'

},

triggers: {

onlyMissedCalls: true, *// Só responder chamadas perdidas*

excludeRejected: true, *// Não responder chamadas rejeitadas*

workingHoursOnly: true, *// Só durante horário comercial*

cooldownRespect: true *// Respeitar cooldown entre respostas*

}

},

response: {

template: {

default: '📞 Perdemos sua ligação!\n\nEstamos disponíveis para atendê-lo por aqui no WhatsApp, ou você pode ligar diretamente para nosso atendimento:\n\n{callButton}\n\nHorário de atendimento: Segunda a Sexta, 8h às 18h.',

customizable: true,

variables: {

contactName: '{contactName}',

companyName: '{companyName}',

currentTime: '{currentTime}',

workingHours: '{workingHours}',

departmentPhone: '{departmentPhone}'

}

},

button: {

text: '📞 LIGAR AGORA',

type: 'url', *// WhatsApp button type*

url: 'tel:+5511999999999', *// ou sip:atendimento@primem.bitrix24.com.br*

departmentMapping: {

comercial: 'sip:comercial@primem.bitrix24.com.br',

operacional: 'sip:operacional@primem.bitrix24.com.br',

financeiro: 'sip:financeiro@primem.bitrix24.com.br',

suporte: 'sip:suporte@primem.bitrix24.com.br',

geral: 'tel:+551140028900' *// Número principal*

}

},

personalization: {

departmentDetection: 'Detectar departamento baseado no contato',

contextualMessage: 'Personalizar mensagem baseada no histórico',

urgencyLevel: 'Diferentes respostas para diferentes urgências'

}

},

configuration: {

cooldown: {

enabled: true,

duration: 60, *// minutos*

perContact: true, *// Cooldown por contato individual*

globalCooldown: false *// Cooldown global (todos os contatos)*

},

workingHours: {

timezone: 'America/Sao\_Paulo',

schedule: {

monday: { start: '08:00', end: '18:00' },

tuesday: { start: '08:00', end: '18:00' },

wednesday: { start: '08:00', end: '18:00' },

thursday: { start: '08:00', end: '18:00' },

friday: { start: '08:00', end: '18:00' },

saturday: { enabled: false },

sunday: { enabled: false }

}

},

exceptions: {

blacklist: ['contact\_id\_1', 'contact\_id\_2'], *// Nunca responder*

whitelist: [], *// Se definido, só responder para estes*

vipList: ['important\_client\_id'], *// Resposta imediata, sem cooldown*

internalNumbers: ['+551199999999'] *// Números internos, não responder*

}

},

analytics: {

tracking: {

callsReceived: 'Total de chamadas recebidas',

callsMissed: 'Total de chamadas perdidas',

autoResponsesSent: 'Auto-respostas enviadas',

clickThroughRate: 'Taxa de cliques no botão de ligação',

conversionRate: 'Conversões (ligações retornadas)'

},

reports: {

daily: 'Relatório diário de chamadas',

effectiveness: 'Efetividade das auto-respostas',

departmentBreakdown: 'Breakdown por departamento'

}

},

integration: {

bitrixPBX: {

sipProtocol: 'sip:user@domain',

authentication: 'SIP authentication if required',

callRouting: 'Route to correct department/agent',

callLogging: 'Log calls in Bitrix CRM'

},

whatsappBusiness: {

buttonSupport: 'WhatsApp Business API buttons',

interactiveMessages: 'Rich interactive messages',

templateMessages: 'Pre-approved message templates'

}

}

}

*// ESTRUTURA DE DADOS:*

const callEvent = {

id: 'call\_event\_id',

contactId: 'whatsapp\_contact\_id',

callType: 'received|missed|rejected',

timestamp: Date,

duration: 0, *// segundos (0 para missed/rejected)*

autoResponseSent: false,

autoResponseAt: null,

department: 'comercial|operacional|etc',

lastAutoResponseAt: Date, *// Para cooldown*

clickedCallButton: false,

clickedAt: null,

returnCallMade: false, *// Se o cliente ligou de volta*

returnCallAt: null

}

**🔥 FUNCIONALIDADE NOVA CRÍTICA - 29**

**29. SISTEMA DE CONFIGURAÇÃO CENTRALIZADA DE TEXTOS E MENSAGENS**

javascript

*// ESPECIFICAÇÃO TÉCNICA COMPLETA:*

const centralizedTextConfigSystem = {

userRoles: {

admin: {

permissions: ['all'],

canAccess: ['all\_configurations', 'user\_management', 'system\_settings'],

description: 'Acesso total ao sistema de configurações'

},

manager: {

permissions: ['view\_configurations', 'edit\_department\_messages'],

canAccess: ['department\_specific\_configs'],

description: 'Acesso limitado às configurações do departamento'

},

agent: {

permissions: ['view\_only'],

canAccess: ['read\_only\_view'],

description: 'Apenas visualização das configurações'

},

viewer: {

permissions: ['none'],

canAccess: [],

description: 'Sem acesso às configurações'

}

},

configurableTexts: {

chatbot: {

welcomeMessage: {

key: 'chatbot.welcome',

category: 'Chatbot',

label: 'Mensagem de Boas-vindas',

description: 'Primeira mensagem enviada pelo chatbot',

defaultValue: '🤖 Olá! Sou o assistente virtual da {companyName}.\n\nComo posso ajudá-lo?\n\n1️⃣ Comercial\n2️⃣ Operacional\n3️⃣ Financeiro\n4️⃣ Suporte Técnico\n0️⃣ Falar com Atendente\n\nDigite o número da opção desejada:',

variables: ['{companyName}', '{currentTime}', '{contactName}', '{userDepartment}'],

maxLength: 4096,

validation: 'required',

adminOnly: true

},

fallbackMessage: {

key: 'chatbot.fallback',

category: 'Chatbot',

label: 'Mensagem de Opção Inválida',

description: 'Mensagem quando usuário digita opção inválida',

defaultValue: '❌ Opção inválida.\n\nPor favor, digite apenas o número da opção desejada (1, 2, 3, 4 ou 0).',

variables: [],

maxLength: 1024,

validation: 'required',

adminOnly: true

},

departmentMessages: {

comercial: {

key: 'chatbot.department.comercial',

category: 'Chatbot - Departamentos',

label: 'Mensagem Comercial',

description: 'Mensagem ao direcionar para comercial',

defaultValue: 'Redirecionando você para nossa equipe comercial...\n\nAguarde que um de nossos consultores irá atendê-lo em breve.',

variables: ['{estimatedWaitTime}'],

maxLength: 512,

validation: 'required',

adminOnly: false

},

operacional: {

key: 'chatbot.department.operacional',

category: 'Chatbot - Departamentos',

label: 'Mensagem Operacional',

description: 'Mensagem ao direcionar para operacional',

defaultValue: 'Conectando com nossa equipe operacional...\n\nEm instantes você será atendido por um especialista.',

variables: ['{estimatedWaitTime}'],

maxLength: 512,

validation: 'required',

adminOnly: false

},

financeiro: {

key: 'chatbot.department.financeiro',

category: 'Chatbot - Departamentos',

label: 'Mensagem Financeiro',

description: 'Mensagem ao direcionar para financeiro',

defaultValue: 'Direcionando para nosso setor financeiro...\n\nNosso time financeiro irá auxiliá-lo shortly.',

variables: ['{estimatedWaitTime}'],

maxLength: 512,

validation: 'required',

adminOnly: false

}

},

offHoursMessage: {

key: 'chatbot.offHours',

category: 'Chatbot',

label: 'Mensagem Fora de Horário',

description: 'Mensagem enviada fora do horário comercial',

defaultValue: '🕒 Nosso atendimento funciona de segunda a sexta das {workingHours}.\n\nDeixe sua mensagem que retornaremos assim que possível.',

variables: ['{workingHours}', '{nextBusinessDay}', '{companyName}'],

maxLength: 1024,

validation: 'required',

adminOnly: true

}

},

autoResponse: {

firstContactMessage: {

key: 'autoResponse.firstContact',

category: 'Auto-Resposta',

label: 'Primeira Mensagem de Contato',

description: 'Resposta automática para novos contatos',

defaultValue: '👋 Olá! Obrigado por entrar em contato com a {companyName}.\n\nRecebemos sua mensagem e retornaremos em breve.',

variables: ['{companyName}', '{contactName}', '{estimatedResponseTime}'],

maxLength: 1024,

validation: 'required',

adminOnly: true

},

keywordResponses: {

ola: {

key: 'autoResponse.keyword.ola',

category: 'Auto-Resposta - Palavras-chave',

label: 'Resposta para "olá"',

description: 'Resposta quando detectar cumprimento',

defaultValue: '👋 Olá! Como posso ajudá-lo hoje?',

variables: ['{contactName}'],

maxLength: 512,

validation: 'required',

adminOnly: false

},

ajuda: {

key: 'autoResponse.keyword.ajuda',

category: 'Auto-Resposta - Palavras-chave',

label: 'Resposta para "ajuda"',

description: 'Resposta quando solicitar ajuda',

defaultValue: '🆘 Estou aqui para ajudar! Descreva sua necessidade que direcionarei para o setor adequado.',

variables: ['{contactName}'],

maxLength: 512,

validation: 'required',

adminOnly: false

}

}

},

bitrixIntegration: {

newDealMessage: {

key: 'bitrix.newDeal',

category: 'Integração Bitrix',

label: 'Mensagem Novo Deal',

description: 'Mensagem enviada quando deal é criado no Bitrix',

defaultValue: 'Olá {contactName}! 👋\n\nRecebemos sua solicitação "{dealTitle}" no valor de {dealAmount}.\n\nEm breve nossa equipe entrará em contato.\n\nAtenciosamente,\n{companyName}',

variables: ['{contactName}', '{dealTitle}', '{dealAmount}', '{companyName}', '{responsibleUser}'],

maxLength: 1024,

validation: 'required',

adminOnly: true

},

followUpMessage: {

key: 'bitrix.followUp',

category: 'Integração Bitrix',

label: 'Mensagem Follow-up',

description: 'Mensagem de follow-up automático',

defaultValue: '📞 Follow-up\n\nOlá {contactName}!\n\nO deal "{dealTitle}" está sem atividade há {daysInactive} dias.\n\nPodemos ajudar com alguma coisa?\n\nResponda esta mensagem ou ligue para {companyPhone}.',

variables: ['{contactName}', '{dealTitle}', '{daysInactive}', '{companyPhone}'],

maxLength: 1024,

validation: 'required',

adminOnly: true

},

paymentReminderMessage: {

key: 'bitrix.paymentReminder',

category: 'Integração Bitrix',

label: 'Lembrete de Pagamento',

description: 'Mensagem de lembrete de pagamento',

defaultValue: '💰 Lembrete de Pagamento\n\nOlá {contactName},\n\nO pagamento do deal "{dealTitle}" vence em {daysUntilDue} dias.\n\nValor: {dealAmount}\nVencimento: {dueDate}\n\nPor favor, providencie o pagamento.',

variables: ['{contactName}', '{dealTitle}', '{daysUntilDue}', '{dealAmount}', '{dueDate}'],

maxLength: 1024,

validation: 'required',

adminOnly: true

}

},

callAutoResponse: {

missedCallMessage: {

key: 'callAutoResponse.missedCall',

category: 'Auto-Resposta Ligações',

label: 'Mensagem Chamada Perdida',

description: 'Mensagem enviada após chamada perdida',

defaultValue: '📞 Perdemos sua ligação!\n\nEstamos disponíveis para atendê-lo por aqui no WhatsApp, ou você pode ligar diretamente:\n\n{callButton}\n\nHorário: {workingHours}',

variables: ['{contactName}', '{callButton}', '{workingHours}', '{departmentPhone}'],

maxLength: 1024,

validation: 'required',

adminOnly: true

}

},

signatures: {

globalSignature: {

key: 'signatures.global',

category: 'Assinaturas',

label: 'Assinatura Global',

description: 'Assinatura padrão para todas as mensagens',

defaultValue: '\n---\nAtenciosamente,\n{userName}\n{userDepartment}\n{companyName}\n{companyPhone}',

variables: ['{userName}', '{userEmail}', '{userDepartment}', '{companyName}', '{companyPhone}', '{currentDate}'],

maxLength: 512,

validation: 'optional',

adminOnly: true

},

departmentSignatures: {

comercial: {

key: 'signatures.department.comercial',

category: 'Assinaturas - Departamentos',

label: 'Assinatura Comercial',

description: 'Assinatura específica do comercial',

defaultValue: '\n---\n{userName} - Consultor Comercial\n{companyName}\n📞 {departmentPhone}\n📧 {userEmail}',

variables: ['{userName}', '{userEmail}', '{departmentPhone}', '{companyName}'],

maxLength: 512,

validation: 'optional',

adminOnly: false

}

}

},

notifications: {

systemMessages: {

connectionEstablished: {

key: 'notifications.system.connected',

category: 'Notificações Sistema',

label: 'WhatsApp Conectado',

description: 'Mensagem quando WhatsApp conecta',

defaultValue: '✅ WhatsApp conectado com sucesso!',

variables: ['{timestamp}', '{deviceName}'],

maxLength: 256,

validation: 'required',

adminOnly: true

},

connectionLost: {

key: 'notifications.system.disconnected',

category: 'Notificações Sistema',

label: 'WhatsApp Desconectado',

description: 'Mensagem quando WhatsApp desconecta',

defaultValue: '⚠️ Conexão com WhatsApp perdida. Tentando reconectar...',

variables: ['{timestamp}', '{retryAttempt}'],

maxLength: 256,

validation: 'required',

adminOnly: true

}

}

},

errors: {

failedToSend: {

key: 'errors.failedToSend',

category: 'Mensagens de Erro',

label: 'Falha no Envio',

description: 'Mensagem quando falha ao enviar',

defaultValue: '❌ Falha ao enviar mensagem. Tentando novamente...',

variables: ['{errorCode}', '{retryIn}'],

maxLength: 256,

validation: 'required',

adminOnly: true

},

fileTooBig: {

key: 'errors.fileTooBig',

category: 'Mensagens de Erro',

label: 'Arquivo Muito Grande',

description: 'Mensagem quando arquivo excede limite',

defaultValue: '📎 Arquivo muito grande. Limite máximo: {maxFileSize}MB',

variables: ['{maxFileSize}', '{actualSize}'],

maxLength: 256,

validation: 'required',

adminOnly: true

}

}

},

interface: {

settingsPage: {

layout: 'Tabs organizadas por categoria',

categories: [

'Chatbot',

'Auto-Resposta',

'Integração Bitrix',

'Assinaturas',

'Notificações Sistema',

'Mensagens de Erro'

],

features: {

search: 'Buscar configurações por nome/categoria',

bulkEdit: 'Edição em lote de configurações similares',

preview: 'Preview em tempo real das mensagens',

variables: 'Lista de variáveis disponíveis',

validation: 'Validação de campos obrigatórios',

reset: 'Resetar para valor padrão',

export: 'Exportar configurações como JSON',

import: 'Importar configurações de arquivo'

}

},

textEditor: {

type: 'Rich text editor with WhatsApp preview',

features: [

'Emoji picker integrado',

'Contagem de caracteres',

'Preview em formato WhatsApp',

'Inserção de variáveis via dropdown',

'Validação em tempo real',

'Auto-save a cada 30 segundos'

],

variableInsertion: {

dropdown: 'Dropdown com variáveis disponíveis',

validation: 'Validar se variável existe',

preview: 'Mostrar valor exemplo da variável'

}

},

permissionControl: {

adminOnly: 'Apenas admin pode editar campos críticos',

departmentAccess: 'Managers podem editar apenas seu departamento',

auditLog: 'Log de todas as alterações com timestamp e usuário',

approval: 'Mudanças críticas precisam de aprovação'

}

},

storage: {

database: {

table: 'text\_configurations',

structure: {

id: 'Primary key',

key: 'Chave única (ex: chatbot.welcome)',

category: 'Categoria da configuração',

label: 'Nome amigável',

description: 'Descrição da configuração',

value: 'Valor atual da configuração',

defaultValue: 'Valor padrão original',

variables: 'Array de variáveis disponíveis',

validation: 'Regras de validação',

adminOnly: 'Boolean se apenas admin pode editar',

maxLength: 'Limite de caracteres',

createdAt: 'Data de criação',

updatedAt: 'Data da última atualização',

updatedBy: 'ID do usuário que alterou'

}

},

caching: {

memory: 'Cache em memória para performance',

redis: 'Cache distribuído se necessário',

invalidation: 'Invalidar cache quando configuração muda'

}

},

variableSystem: {

globalVariables: {

companyName: 'PRIMEM COMEX',

companyPhone: '+55 11 4002-8900',

companyEmail: 'contato@primem.com.br',

companyWebsite: 'https://primem.com.br',

currentDate: 'Data atual formatada',

currentTime: 'Hora atual formatada',

currentDateTime: 'Data e hora atual'

},

userVariables: {

userName: 'Nome do usuário logado',

userEmail: 'Email do usuário logado',

userDepartment: 'Departamento do usuário',

userRole: 'Role do usuário (admin, manager, etc.)'

},

contactVariables: {

contactName: 'Nome do contato WhatsApp',

contactNumber: 'Número do contato',

contactFirstMessage: 'Primeira mensagem do contato'

},

bitrixVariables: {

dealTitle: 'Título do deal no Bitrix',

dealAmount: 'Valor do deal formatado',

dealStage: 'Estágio atual do deal',

responsibleUser: 'Usuário responsável pelo deal',

daysInactive: 'Dias sem atividade',

daysUntilDue: 'Dias até vencimento',

dueDate: 'Data de vencimento'

},

contextVariables: {

currentConversation: 'ID da conversa atual',

messageCount: 'Número de mensagens na conversa',

lastMessageTime: 'Timestamp da última mensagem',

estimatedWaitTime: 'Tempo estimado de espera',

queuePosition: 'Posição na fila de atendimento'

},

processing: {

syntax: 'Usar {variableName} para inserir variáveis',

escaping: 'Escapar caracteres especiais',

fallback: 'Valor padrão se variável não existir',

validation: 'Validar se todas as variáveis existem'

}

},

api: {

endpoints: [

'GET /api/admin/text-configs - Listar todas as configurações',

'GET /api/admin/text-configs/:key - Buscar configuração específica',

'PUT /api/admin/text-configs/:key - Atualizar configuração',

'POST /api/admin/text-configs/bulk - Atualização em lote',

'POST /api/admin/text-configs/export - Exportar configurações',

'POST /api/admin/text-configs/import - Importar configurações',

'GET /api/admin/text-configs/variables - Listar variáveis disponíveis',

'POST /api/admin/text-configs/preview - Preview de mensagem'

],

permissions: {

middleware: 'Verificar se usuário tem permissão para editar',

roleBasedAccess: 'Controle baseado em roles',

auditLog: 'Log de todas as operações de admin'

}

},

implementation: {

components: [

'TextConfigManager.js - Componente principal',

'TextEditor.js - Editor de texto com preview',

'VariableInserter.js - Inserção de variáveis',

'CategoryTabs.js - Tabs por categoria',

'PermissionGate.js - Controle de acesso',

'ConfigPreview.js - Preview em tempo real'

],

hooks: [

'useTextConfigs.js - Gerenciar configurações',

'useVariables.js - Sistema de variáveis',

'usePermissions.js - Controle de permissões'

],

services: [

'textConfigService.js - API calls',

'variableService.js - Processamento de variáveis',

'permissionService.js - Verificação de permissões'

]

}

}

*// ESTRUTURA DE DADOS:*

const textConfiguration = {

id: 'uuid',

key: 'chatbot.welcome',

category: 'Chatbot',

label: 'Mensagem de Boas-vindas',

description: 'Primeira mensagem enviada pelo chatbot',

value: '🤖 Olá! Sou o assistente virtual da PRIMEM COMEX...',

defaultValue: '🤖 Olá! Sou o assistente virtual da {companyName}...',

variables: ['{companyName}', '{currentTime}', '{contactName}'],

validation: 'required',

adminOnly: true,

maxLength: 4096,

createdAt: Date,

updatedAt: Date,

updatedBy: 'user\_id'

}

const auditLog = {

id: 'uuid',

configKey: 'chatbot.welcome',

action: 'update|create|delete',

oldValue: 'Valor anterior',

newValue: 'Novo valor',

userId: 'user\_id',

userRole: 'admin',

timestamp: Date,

ipAddress: '192.168.1.1',

userAgent: 'Browser info'

}

*// CRITÉRIOS DE ACEITAÇÃO:*

*// ✅ Interface completa para configurar todos os textos*

*// ✅ Controle de acesso baseado em roles (Admin, Manager, Agent, Viewer)*

*// ✅ Sistema de variáveis dinâmicas funcionando*

*// ✅ Preview em tempo real das mensagens*

*// ✅ Auditoria completa de mudanças*

*// ✅ Export/import de configurações*

*// ✅ Aplicação automática das configurações no sistema*

*// ✅ Validação de campos e variáveis*

*// ✅ Cache para performance*

*// ✅ Backup automático antes de mudanças críticas*

**🔥 FUNCIONALIDADE CRÍTICA - 30**

**30. INTEGRAÇÃO COMPLETA COM BITRIX24 PARA AUTOMAÇÕES**

javascript

*// ESPECIFICAÇÃO TÉCNICA COMPLETA:*

const bitrixIntegrationSystem = {

architecture: {

connectionType: 'HTTP REST API + Webhooks bidirecionais',

protocol: 'HTTPS',

authentication: 'Bearer Token + Webhook Secret',

dataFormat: 'JSON',

realTime: 'Webhooks para eventos em tempo real',

rateLimit: '2-10 requests/second (baseado no plano Bitrix)'

},

webhookConfiguration: {

outgoing: {

*// Bitrix envia para nosso sistema*

endpoint: 'https://seu-sistema.com/api/bitrix/webhook',

events: [

'ONCRMDEALADD', *// Deal criado*

'ONCRMDEALUPDATE', *// Deal atualizado*

'ONCRMLEADADD', *// Lead criado*

'ONCRMCONTACTADD', *// Contato criado*

'ONCRMCONTACTUPDATE', *// Contato atualizado*

'ONCRM\_TIMELINE\_COMMENT\_ADD' *// Comentário adicionado*

],

authentication: {

type: 'bearer',

token: 'api\_key\_gerada\_automaticamente'

},

security: {

signature: 'HMAC-SHA256 validation',

ipWhitelist: ['185.54.36.0/24', '185.54.37.0/24'], *// IPs Bitrix24*

maxPayloadSize: '1MB'

}

},

incoming: {

*// Nosso sistema consulta Bitrix*

baseUrl: 'https://primem.bitrix24.com.br/rest/',

authentication: {

webhookUrl: '/123/abc123def456/', *// URL do webhook de entrada*

*// ou OAuth para maior segurança*

oauthToken: 'access\_token\_via\_oauth'

},

methods: [

'crm.deal.get',

'crm.deal.list',

'crm.contact.get',

'crm.contact.list',

'crm.activity.add' *// Para registrar atividades*

]

}

},

messageTemplates: {

structure: {

id: 'template\_uuid',

name: 'Nome amigável do template',

description: 'Descrição do template',

bitrixEvent: 'ONCRMDEALADD', *// Evento que dispara*

conditions: {

dealStage: ['NEW', 'PROPOSAL'], *// Estágios que ativam*

dealAmount: { min: 1000, max: null }, *// Valor mínimo*

contactType: 'LEAD', *// Tipo de contato*

customFields: {

'UF\_SEND\_WHATSAPP': 'Y' *// Campos personalizados*

}

},

content: {

text: 'Olá {contactName}! Recebemos sua solicitação...',

variables: ['contactName', 'dealTitle', 'dealAmount'],

hasSignature: true,

mediaType: null, *// text|image|document|video*

mediaUrl: null

},

scheduling: {

immediate: true,

delay: 0, *// minutos*

businessHoursOnly: true,

respectTimezone: true

},

priority: 'normal', *// high|normal|low*

isActive: true,

createdBy: 'user\_id',

createdAt: Date,

updatedAt: Date

},

predefinedTemplates: {

newDeal: {

name: 'Novo Deal Criado',

bitrixEvent: 'ONCRMDEALADD',

content: 'Olá {contactName}! 👋\n\nRecebemos sua solicitação "{dealTitle}" no valor de {dealAmount}.\n\nEm breve nossa equipe entrará em contato.\n\nAtenciosamente,\n{companyName}',

variables: ['contactName', 'dealTitle', 'dealAmount', 'companyName']

},

dealWon: {

name: 'Deal Ganho - Agradecimento',

bitrixEvent: 'ONCRMDEALUPDATE',

conditions: { dealStage: ['WON'] },

content: '🎉 Parabéns {contactName}!\n\nSua solicitação "{dealTitle}" foi aprovada!\n\nValor: {dealAmount}\n\nNossa equipe entrará em contato para os próximos passos.\n\n{companyName}',

variables: ['contactName', 'dealTitle', 'dealAmount', 'companyName']

},

followUp: {

name: 'Follow-up Automático',

bitrixEvent: 'CUSTOM\_AUTOMATION',

content: '📞 Follow-up\n\nOlá {contactName}!\n\nO deal "{dealTitle}" está sem atividade há {daysInactive} dias.\n\nPodemos ajudar?\n\nResponda ou ligue: {companyPhone}',

variables: ['contactName', 'dealTitle', 'daysInactive', 'companyPhone']

},

paymentReminder: {

name: 'Lembrete de Pagamento',

bitrixEvent: 'CUSTOM\_AUTOMATION',

content: '💰 Lembrete\n\nOlá {contactName},\n\nPagamento do deal "{dealTitle}" vence em {daysUntilDue} dias.\n\nValor: {dealAmount}\nVencimento: {dueDate}',

variables: ['contactName', 'dealTitle', 'daysUntilDue', 'dealAmount', 'dueDate']

}

},

variableMapping: {

*// Mapeamento de campos Bitrix para variáveis do template*

bitrixToTemplate: {

'TITLE': 'dealTitle',

'OPPORTUNITY': 'dealAmount',

'STAGE\_ID': 'dealStage',

'CONTACT\_ID.NAME': 'contactName',

'CONTACT\_ID.PHONE[0].VALUE': 'contactPhone',

'ASSIGNED\_BY\_ID.NAME': 'responsibleUser',

'DATE\_CREATE': 'dateCreated',

'COMPANY\_ID.TITLE': 'companyTitle'

},

formatting: {

dealAmount: 'formatCurrency', *// R$ 50.000,00*

dateCreated: 'formatDate', *// 13/09/2025*

contactPhone: 'formatPhone' *// +55 11 99999-9999*

}

}

},

webhookProcessor: {

validation: {

signature: 'Validar HMAC-SHA256',

ipWhitelist: 'Verificar IP de origem',

timestamp: 'Verificar timestamp (max 5 min)',

structure: 'Validar estrutura do payload'

},

processing: {

async: true, *// Processamento assíncrono*

queue: 'Redis Bull Queue',

retry: {

maxAttempts: 3,

backoff: 'exponential',

delay: [1000, 5000, 15000] *// ms*

},

timeout: 30000 *// 30 segundos*

},

workflow: [

'1. Receber webhook do Bitrix',

'2. Validar autenticidade',

'3. Responder 200 OK imediatamente',

'4. Processar assincronamente',

'5. Buscar templates matching',

'6. Buscar dados do contato',

'7. Renderizar mensagem',

'8. Adicionar à fila WhatsApp',

'9. Registrar atividade no Bitrix'

]

},

messageQueue: {

architecture: {

primary: 'Redis + Bull Queue',

fallback: 'SQLite queue table',

concurrency: 5, *// Mensagens simultâneas*

priorities: {

high: 1, *// Urgent (payment due)*

normal: 5, *// Standard notifications*

low: 10 *// Informational*

}

},

scheduling: {

immediate: 'Envio imediato',

delayed: 'Envio com delay configurável',

businessHours: {

enabled: true,

timezone: 'America/Sao\_Paulo',

schedule: {

monday: { start: '08:00', end: '18:00' },

tuesday: { start: '08:00', end: '18:00' },

wednesday: { start: '08:00', end: '18:00' },

thursday: { start: '08:00', end: '18:00' },

friday: { start: '08:00', end: '18:00' },

saturday: { enabled: false },

sunday: { enabled: false }

}

}

},

monitoring: {

metrics: ['success\_rate', 'avg\_processing\_time', 'queue\_size'],

alerts: ['queue\_backup', 'high\_failure\_rate', 'processing\_timeout']

}

},

security: {

authentication: {

apiKey: {

generation: 'UUID v4 + timestamp',

rotation: 'Automática a cada 30 dias',

storage: 'Criptografado com AES-256',

validation: 'Header: Authorization: Bearer {apiKey}'

},

webhookSecret: {

purpose: 'Validar origem dos webhooks',

algorithm: 'HMAC-SHA256',

header: 'X-Bitrix-Signature',

validation: 'hash\_hmac("sha256", payload, secret)'

}

},

dataProtection: {

encryption: 'AES-256 para dados sensíveis',

masking: 'Telefones/emails em logs',

retention: '90 dias para logs',

anonymization: 'Dados antigos anonimizados'

},

rateLimiting: {

incoming: '100 requests/minute per IP',

outgoing: 'Respeitará limits do Bitrix24',

burst: 'Max 10 requests em 1 segundo',

backoff: 'Exponential backoff on rate limit'

}

},

implementation: {

backend: {

controllers: [

'bitrixController.js - Controle principal da integração',

'webhookController.js - Processar webhooks recebidos',

'templateController.js - Gerenciar templates',

'queueController.js - Monitorar filas'

],

services: [

'bitrixConnectionService.js - Gerenciar conexões',

'bitrixApiService.js - Chamadas para API',

'webhookProcessor.js - Processar eventos',

'templateEngine.js - Renderizar templates',

'messageQueue.js - Fila de mensagens',

'activityLogger.js - Log no Bitrix'

],

middleware: [

'bitrixAuth.js - Autenticar requests',

'webhookValidator.js - Validar webhooks',

'rateLimiter.js - Rate limiting',

'errorHandler.js - Tratamento de erros'

]

},

frontend: {

pages: [

'BitrixIntegrationPage.js - Dashboard principal',

'TemplateManagerPage.js - Gerenciar templates',

'WebhookLogsPage.js - Logs de webhooks'

],

components: [

'BitrixConnectionStatus.js - Status da conexão',

'TemplateEditor.js - Editor de templates',

'WebhookTester.js - Testar webhooks',

'QueueMonitor.js - Monitor da fila'

],

hooks: [

'useBitrixIntegration.js - Gerenciar integração',

'useTemplates.js - Gerenciar templates',

'useWebhookLogs.js - Logs de webhooks'

]

},

database: {

tables: [

'bitrix\_integrations - Configurações da integração',

'bitrix\_templates - Templates de mensagem',

'bitrix\_webhook\_logs - Log de webhooks',

'bitrix\_message\_queue - Fila de mensagens',

'bitrix\_activities - Atividades registradas'

]

}

},

monitoring: {

metrics: {

webhooksReceived: 'Total de webhooks recebidos',

webhooksProcessed: 'Webhooks processados com sucesso',

messagesQueued: 'Mensagens adicionadas à fila',

messagesSent: 'Mensagens enviadas com sucesso',

processingTime: 'Tempo médio de processamento',

errorRate: 'Taxa de erro',

queueSize: 'Tamanho atual da fila'

},

alerts: {

webhookFailure: 'Falha no processamento de webhook',

queueBacklog: 'Fila com muitas mensagens pendentes',

highErrorRate: 'Taxa de erro acima de 5%',

connectionLost: 'Perda de conexão com Bitrix24'

},

dashboard: {

realTime: 'Métricas em tempo real',

charts: 'Gráficos de atividade',

logs: 'Logs detalhados',

health: 'Status de saúde do sistema'

}

}

}

*// ESTRUTURAS DE DADOS:*

*// Configuração da Integração*

const bitrixIntegration = {

id: 'uuid',

companyId: 'company\_id',

bitrixDomain: 'primem.bitrix24.com.br',

webhookUrl: 'https://primem.bitrix24.com.br/rest/123/abc123/',

apiKey: 'encrypted\_api\_key',

webhookSecret: 'encrypted\_webhook\_secret',

oauthToken: 'encrypted\_oauth\_token', *// Se usar OAuth*

isActive: true,

settings: {

businessHoursOnly: true,

workingHours: {

start: '08:00',

end: '18:00',

timezone: 'America/Sao\_Paulo'

},

rateLimits: {

maxPerSecond: 2,

maxPerMinute: 100

},

retryPolicy: {

maxRetries: 3,

backoffMultiplier: 2

}

},

createdAt: Date,

updatedAt: Date

}

*// Template de Mensagem*

const messageTemplate = {

id: 'uuid',

integrationId: 'bitrix\_integration\_id',

name: 'Novo Deal - Boas Vindas',

description: 'Mensagem enviada quando deal é criado',

bitrixEvent: 'ONCRMDEALADD',

conditions: {

dealStage: ['NEW'],

dealAmount: { min: 1000 },

customFields: {

'UF\_SEND\_WHATSAPP': 'Y'

}

},

content: {

text: 'Olá {contactName}! Recebemos sua solicitação "{dealTitle}"...',

variables: ['contactName', 'dealTitle', 'dealAmount'],

hasSignature: true

},

scheduling: {

immediate: false,

delay: 5, *// 5 minutos*

businessHoursOnly: true

},

priority: 'normal',

isActive: true,

createdBy: 'user\_id',

createdAt: Date

}

*// Log de Webhook*

const webhookLog = {

id: 'uuid',

integrationId: 'bitrix\_integration\_id',

event: 'ONCRMDEALADD',

payload: 'JSON do Bitrix',

headers: 'Headers do request',

ipAddress: '185.54.36.100',

signature: 'HMAC signature',

status: 'success|failed|processing',

processingTime: 250, *// ms*

messagesCreated: 1,

templatesMatched: ['template1'],

errors: [],

createdAt: Date

}

*// Item da Fila*

const messageQueueItem = {

id: 'uuid',

integrationId: 'bitrix\_integration\_id',

templateId: 'template\_id',

dealId: '12345',

contactPhone: '+5511999999999',

priority: 'normal',

scheduledFor: Date,

attempts: 0,

maxAttempts: 3,

status: 'pending|processing|sent|failed',

payload: {

contactName: 'João Silva',

dealTitle: 'Importação Equipamentos',

dealAmount: 'R$ 50.000,00'

},

result: {

messageId: 'whatsapp\_message\_id',

sentAt: Date,

error: null

},

createdAt: Date

}

*// CRITÉRIOS DE ACEITAÇÃO:*

*// ✅ Conexão bidirecional com Bitrix24 funcionando*

*// ✅ Webhooks recebidos e processados corretamente*

*// ✅ Templates de mensagem configuráveis*

*// ✅ Fila de mensagens com retry automático*

*// ✅ Processamento assíncrono de alta performance*

*// ✅ Segurança robusta com validação de origem*

*// ✅ Logs detalhados de todas operações*

*// ✅ Interface para configurar e monitorar*

*// ✅ Rate limiting respeitando limites do Bitrix*

*// ✅ Registro de atividades de volta no Bitrix*

**📋 PLANO DE EXECUÇÃO LINEAR DETALHADO**

**🔥 FASE 1: FINALIZAR MODULARIZAÇÃO (CRÍTICA) - 2-3 DIAS**

**FASE 1.1: ANÁLISE COMPLETA DO App.js ATUAL**

javascript

*// OBJETIVOS:*

*// 1. Mapear todas as funcionalidades do App.js (53KB)*

*// 2. Identificar dependências críticas*

*// 3. Documentar fluxos de navegação*

*// 4. Planejar migração sem quebrar funcionalidades*

*// PROCESSO:*

*// 1. Fazer backup: cp App.js App\_backup\_$(date).js*

*// 2. Analisar imports e dependências*

*// 3. Mapear estados (useState, useEffect)*

*// 4. Identificar handlers de eventos*

*// 5. Documentar componentes inline*

*// 6. Verificar integração com hooks existentes*

*// ENTREGÁVEIS:*

*// - Documento de análise técnica*

*// - Lista de funcionalidades mapeadas*

*// - Plano de migração passo a passo*

*// - Identificação de riscos e dependências*

**FASE 1.2: CRIAR SISTEMA DE ROTEAMENTO**

javascript

*// ARQUIVO: client/src/routes/index.js (NOVO)*

import { Routes, Route } from 'react-router-dom';

import { LoginPage, DashboardPage, ChatPage, ConnectionPage, SettingsPage } from '../pages';

import { AuthGuard, AdminOnly } from '../components/auth';

export function AppRoutes() {

return (

<Routes>

<Route path="/" element={<LoginPage />} />

<Route path="/login" element={<LoginPage />} />

<Route path="/dashboard" element={

<AuthGuard>

<DashboardPage />

</AuthGuard>

} />

<Route path="/chat" element={

<AuthGuard>

<ChatPage />

</AuthGuard>

} />

<Route path="/chat/:contactId" element={

<AuthGuard>

<ChatPage />

</AuthGuard>

} />

<Route path="/connection" element={

<AuthGuard>

<ConnectionPage />

</AuthGuard>

} />

<Route path="/settings" element={

<AuthGuard>

<SettingsPage />

</AuthGuard>

} />

<Route path="/admin/text-configs" element={

<AdminOnly>

<TextConfigPage />

</AdminOnly>

} />

<Route path="/admin/bitrix-integration" element={

<AdminOnly>

<BitrixIntegrationPage />

</AdminOnly>

} />

</Routes>

);

}

*// INSTALAÇÕES NECESSÁRIAS:*

*// npm install react-router-dom*

**FASE 1.3: REFATORAR App.js PARA ESTRUTURA MODULAR**

javascript

*// ARQUIVO: client/src/App.js (NOVO - SUBSTITUIR O ATUAL)*

import React from 'react';

import { BrowserRouter } from 'react-router-dom';

import { AuthProvider } from './contexts/AuthContext';

import { WhatsAppProvider } from './contexts/WhatsAppContext';

import { SettingsProvider } from './contexts/SettingsContext';

import { TextConfigProvider } from './contexts/TextConfigContext';

import { BitrixIntegrationProvider } from './contexts/BitrixIntegrationContext';

import { AppRoutes } from './routes';

import './styles/global.css';

export default function App() {

return (

<BrowserRouter>

<AuthProvider>

<WhatsAppProvider>

<SettingsProvider>

<TextConfigProvider>

<BitrixIntegrationProvider>

<div className="app">

<AppRoutes />

</div>

</BitrixIntegrationProvider>

</TextConfigProvider>

</SettingsProvider>

</WhatsAppProvider>

</AuthProvider>

</BrowserRouter>

);

}

*// CRITÉRIOS DE ACEITAÇÃO:*

*// ✅ App.js reduzido de 53KB para <5KB*

*// ✅ Navegação funcionando entre páginas*

*// ✅ Contextos carregando corretamente*

*// ✅ Autenticação funcionando*

*// ✅ WhatsApp conectando normalmente*

*// ✅ Chat enviando/recebendo mensagens*

*// ✅ Upload de arquivos funcionando*

*// ✅ Sistema de configuração de textos operacional*

*// ✅ Integração Bitrix configurável*

**FASE 1.4: TESTE E VALIDAÇÃO COMPLETA**

javascript

*// CHECKLIST DE TESTES:*

*// ✅ Login com usuários teste (teste@teste.com/123, admin@primem.com/admin123)*

*// ✅ Navegação entre páginas sem erros*

*// ✅ Conexão WhatsApp + QR Code*

*// ✅ Envio/recebimento de mensagens*

*// ✅ Upload de imagens, vídeos, documentos*

*// ✅ Gravação e reprodução de áudio*

*// ✅ Lista de conversas carregando*

*// ✅ Estados de loading e erro funcionando*

*// ✅ Responsividade em diferentes telas*

*// ✅ Performance não degradada*

*// ✅ Controle de acesso por perfil funcionando*

*// ✅ Admin acessa configurações, outros perfis não*

*// ✅ Integração Bitrix testável*

*// FERRAMENTAS DE TESTE:*

*// - Chrome DevTools para performance*

*// - React DevTools para componentes*

*// - Network tab para requests*

*// - Console para erros JavaScript*

**📦 FASE 2: COMPLETAR ARQUIVOS VAZIOS + INTEGRAÇÃO BITRIX - 4-5 DIAS**

**FASE 2.1: CLIENT-SIDE CONSTANTS & UTILS**

javascript

*// ARQUIVO: client/src/constants/apiEndpoints.js*

const API\_BASE\_URL = process.env.REACT\_APP\_API\_URL || 'http://localhost:3001';

export const API\_ENDPOINTS = {

auth: {

login: `${API\_BASE\_URL}/api/auth/login`,

logout: `${API\_BASE\_URL}/api/auth/logout`,

refresh: `${API\_BASE\_URL}/api/auth/refresh`

},

whatsapp: {

connect: `${API\_BASE\_URL}/api/whatsapp/connect`,

disconnect: `${API\_BASE\_URL}/api/whatsapp/disconnect`,

status: `${API\_BASE\_URL}/api/whatsapp/status`,

qr: `${API\_BASE\_URL}/api/whatsapp/qr`,

chats: `${API\_BASE\_URL}/api/whatsapp/chats`,

send: `${API\_BASE\_URL}/api/whatsapp/send`,

sendMedia: `${API\_BASE\_URL}/api/whatsapp/send-media`

},

messages: {

list: `${API\_BASE\_URL}/api/messages`,

history: `${API\_BASE\_URL}/api/messages/history`,

markRead: `${API\_BASE\_URL}/api/messages/mark-read`

},

files: {

upload: `${API\_BASE\_URL}/api/files/upload`,

download: `${API\_BASE\_URL}/api/files/download`,

delete: `${API\_BASE\_URL}/api/files/delete`

},

users: {

list: `${API\_BASE\_URL}/api/users`,

profile: `${API\_BASE\_URL}/api/users/profile`,

update: `${API\_BASE\_URL}/api/users/update`

},

textConfigs: {

list: `${API\_BASE\_URL}/api/admin/text-configs`,

get: `${API\_BASE\_URL}/api/admin/text-configs`,

update: `${API\_BASE\_URL}/api/admin/text-configs`,

bulk: `${API\_BASE\_URL}/api/admin/text-configs/bulk`,

export: `${API\_BASE\_URL}/api/admin/text-configs/export`,

import: `${API\_BASE\_URL}/api/admin/text-configs/import`,

variables: `${API\_BASE\_URL}/api/admin/text-configs/variables`,

preview: `${API\_BASE\_URL}/api/admin/text-configs/preview`

},

bitrix: {

webhook: `${API\_BASE\_URL}/api/bitrix/webhook`,

sendMessage: `${API\_BASE\_URL}/api/bitrix/send-message`,

integration: `${API\_BASE\_URL}/api/bitrix/integration`,

templates: `${API\_BASE\_URL}/api/bitrix/templates`,

status: `${API\_BASE\_URL}/api/bitrix/status`,

logs: `${API\_BASE\_URL}/api/bitrix/logs`,

testConnection: `${API\_BASE\_URL}/api/bitrix/test-connection`

}

};

*// ARQUIVO: client/src/constants/config.js*

export const APP\_CONFIG = {

name: 'PRIMEM WhatsApp Business',

version: '2.0.0',

company: {

name: 'PRIMEM COMEX',

phone: '+55 11 4002-8900',

email: 'contato@primem.com.br',

website: 'https://primem.com.br'

},

features: {

audio: true,

files: true,

notifications: true,

darkMode: true,

multiUser: true,

textConfigs: true,

chatbot: true,

autoResponse: true,

bitrixIntegration: true

},

limits: {

fileSize: 16 \* 1024 \* 1024, *// 16MB*

messageLength: 4096,

conversationHistory: 1000,

textConfigLength: 4096,

webhookTimeout: 30000

},

socket: {

url: process.env.REACT\_APP\_SOCKET\_URL || 'http://localhost:3001',

options: {

transports: ['websocket', 'polling'],

reconnectionAttempts: 10,

reconnectionDelay: 1000

}

},

roles: {

admin: {

permissions: ['all'],

canAccessTextConfigs: true,

canAccessUserManagement: true,

canAccessSystemSettings: true,

canAccessBitrixIntegration: true

},

manager: {

permissions: ['view\_all\_chats', 'manage\_department', 'view\_reports'],

canAccessTextConfigs: false,

canAccessUserManagement: false,

canAccessSystemSettings: false,

canAccessBitrixIntegration: false

},

agent: {

permissions: ['view\_assigned\_chats', 'send\_messages', 'upload\_files'],

canAccessTextConfigs: false,

canAccessUserManagement: false,

canAccessSystemSettings: false,

canAccessBitrixIntegration: false

},

viewer: {

permissions: ['view\_assigned\_chats'],

canAccessTextConfigs: false,

canAccessUserManagement: false,

canAccessSystemSettings: false,

canAccessBitrixIntegration: false

}

},

bitrix: {

allowedDomains: ['bitrix24.com.br', 'bitrix24.com', 'bitrix24.ru'],

webhookTimeout: 30000,

retryAttempts: 3,

queueConcurrency: 5,

rateLimits: {

basic: 2, *// requests/second*

standard: 10,

premium: 'unlimited'

}

}

};

**FASE 2.6: IMPLEMENTAÇÃO DA INTEGRAÇÃO BITRIX**

javascript

*// ARQUIVO: server/controllers/bitrixController.js*

const bitrixConnectionService = require('../services/bitrixConnectionService');

const webhookProcessor = require('../services/webhookProcessor');

const templateEngine = require('../services/templateEngine');

const messageQueue = require('../services/messageQueue');

class BitrixController {

*// Receber webhook do Bitrix*

async handleWebhook(req, res) {

try {

const { headers, body } = req;

const integration = req.bitrixIntegration; *// From middleware*

*// Validar assinatura do webhook*

const isValid = await webhookProcessor.validateSignature(

headers,

body,

integration.webhookSecret

);

if (!isValid) {

return res.status(401).json({

success: false,

error: 'Invalid webhook signature'

});

}

*// Log do webhook*

const logEntry = await webhookProcessor.logWebhook({

integrationId: integration.id,

event: body.event,

payload: body,

headers,

ipAddress: req.ip

});

*// Processar webhook de forma assíncrona*

setImmediate(() => {

this.processWebhookAsync(integration, body, logEntry.id);

});

*// Resposta imediata para o Bitrix*

res.status(200).json({

success: true,

message: 'Webhook received and queued for processing'

});

} catch (error) {

console.error('Bitrix webhook error:', error);

res.status(500).json({

success: false,

error: 'Internal server error'

});

}

}

*// Processar webhook de forma assíncrona*

async processWebhookAsync(integration, payload, logId) {

try {

*// Encontrar templates que correspondem ao evento*

const matchingTemplates = await templateEngine.findMatchingTemplates(

integration.id,

payload.event,

payload.data

);

if (matchingTemplates.length === 0) {

await webhookProcessor.updateLog(logId, {

status: 'completed',

messagesSent: 0,

note: 'No matching templates found'

});

return;

}

*// Processar cada template*

for (const template of matchingTemplates) {

await this.processTemplate(integration, template, payload.data, logId);

}

await webhookProcessor.updateLog(logId, {

status: 'completed',

messagesSent: matchingTemplates.length

});

} catch (error) {

console.error('Webhook processing error:', error);

await webhookProcessor.updateLog(logId, {

status: 'failed',

errors: [error.message]

});

}

}

*// Processar template específico*

async processTemplate(integration, template, bitrixData, logId) {

try {

*// Obter dados do contato no Bitrix*

const contactData = await bitrixConnectionService.getContactData(

integration,

bitrixData.CONTACT\_ID

);

if (!contactData.phone) {

throw new Error('Contact has no phone number');

}

*// Renderizar template com dados*

const messageContent = await templateEngine.renderTemplate(template, {

...bitrixData,

...contactData

});

*// Adicionar à fila de mensagens*

await messageQueue.addMessage({

integrationId: integration.id,

templateId: template.id,

contactPhone: contactData.phone,

dealId: bitrixData.ID,

priority: template.priority,

scheduledFor: this.calculateScheduledTime(template),

payload: messageContent

});

} catch (error) {

console.error('Template processing error:', error);

throw error;

}

}

*// Calcular horário de envio*

calculateScheduledTime(template) {

const now = new Date();

if (template.scheduling.immediate) {

return now;

}

*// Adicionar delay configurado*

const scheduledTime = new Date(now.getTime() + template.scheduling.delay \* 60000);

*// Verificar horário comercial*

if (template.scheduling.businessHoursOnly) {

return this.adjustToBusinessHours(scheduledTime);

}

return scheduledTime;

}

*// Ajustar para horário comercial*

adjustToBusinessHours(date) {

*// Implementar lógica de horário comercial*

*// Considerar fuso horário, fins de semana, feriados*

const businessStart = 8; *// 8h*

const businessEnd = 18; *// 18h*

const hour = date.getHours();

const day = date.getDay();

*// Se é fim de semana, mover para segunda-feira*

if (day === 0 || day === 6) {

const daysToAdd = day === 0 ? 1 : 2;

date.setDate(date.getDate() + daysToAdd);

date.setHours(businessStart, 0, 0, 0);

return date;

}

*// Se é antes do horário comercial, mover para início*

if (hour < businessStart) {

date.setHours(businessStart, 0, 0, 0);

return date;

}

*// Se é depois do horário comercial, mover para próximo dia*

if (hour >= businessEnd) {

date.setDate(date.getDate() + 1);

date.setHours(businessStart, 0, 0, 0);

return date;

}

return date;

}

}

module.exports = new BitrixController();

*// ARQUIVO: client/src/pages/BitrixIntegrationPage.js*

import React, { useState, useEffect } from 'react';

import { useAuth } from '../contexts/AuthContext';

import { useBitrixIntegration } from '../hooks/useBitrixIntegration';

import { Button } from '../components/ui/Button';

import { Modal } from '../components/ui/Modal';

import { BitrixSetupForm } from '../components/bitrix/BitrixSetupForm';

import { BitrixStats } from '../components/bitrix/BitrixStats';

import { TemplateManager } from '../components/bitrix/TemplateManager';

const BitrixIntegrationPage = () => {

const { user } = useAuth();

const {

integration,

loading,

testConnection,

createIntegration

} = useBitrixIntegration();

const [showSetupModal, setShowSetupModal] = useState(false);

const [activeTab, setActiveTab] = useState('overview');

*// Verificar permissão*

if (user?.role !== 'admin') {

return (

<div className="no-access">

<h2>🔒 Acesso Restrito</h2>

<p>Apenas administradores podem configurar a integração com Bitrix24.</p>

</div>

);

}

const handleTestConnection = async () => {

try {

const result = await testConnection();

if (result.success) {

alert('✅ Conexão com Bitrix24 estabelecida com sucesso!');

} else {

alert('❌ Falha na conexão: ' + result.error);

}

} catch (error) {

alert('❌ Erro ao testar conexão: ' + error.message);

}

};

if (loading) {

return <div className="loading">Carregando configurações da integração...</div>;

}

return (

<div className="bitrix-integration-page">

<header className="page-header">

<h1>🔗 Integração Bitrix24</h1>

<p>Configure a integração para receber automações do Bitrix24</p>

</header>

{!integration ? (

<div className="setup-section">

<div className="setup-card">

<h2>🚀 Configurar Integração</h2>

<p>Configure a integração com Bitrix24 para enviar mensagens automáticas via WhatsApp.</p>

<div className="features-list">

<div className="feature">✅ Mensagens automáticas quando deal é criado</div>

<div className="feature">✅ Follow-up automático após período de inatividade</div>

<div className="feature">✅ Lembretes de pagamento</div>

<div className="feature">✅ Notificações de mudança de status</div>

<div className="feature">✅ Templates personalizáveis</div>

</div>

<Button

onClick={() => setShowSetupModal(true)}

variant="primary"

size="large"

>

Configurar Integração

</Button>

</div>

</div>

) : (

<div className="integration-content">

<nav className="tab-navigation">

<button

className={activeTab === 'overview' ? 'active' : ''}

onClick={() => setActiveTab('overview')}

>

📊 Visão Geral

</button>

<button

className={activeTab === 'templates' ? 'active' : ''}

onClick={() => setActiveTab('templates')}

>

📝 Templates

</button>

<button

className={activeTab === 'logs' ? 'active' : ''}

onClick={() => setActiveTab('logs')}

>

📋 Logs

</button>

</nav>

<div className="tab-content">

{activeTab === 'overview' && (

<div className="overview-tab">

<div className="integration-status">

<div className="status-header">

<h3>Status da Integração</h3>

<div className={`status-badge ${integration.isActive ? 'active' : 'inactive'}`}>

{integration.isActive ? 'Ativa' : 'Inativa'}

</div>

</div>

<div className="status-details">

<div className="status-item">

<span>Domínio Bitrix:</span>

<span>{integration.bitrixDomain}</span>

</div>

<div className="status-item">

<span>URL do Webhook:</span>

<code className="webhook-url">

{window.location.origin}/api/bitrix/webhook

</code>

</div>

</div>

<div className="actions">

<Button onClick={handleTestConnection} variant="outline">

🔍 Testar Conexão

</Button>

</div>

</div>

<BitrixStats integration={integration} />

</div>

)}

{activeTab === 'templates' && (

<TemplateManager integrationId={integration.id} />

)}

{activeTab === 'logs' && (

<div className="logs-tab">

{*/\* Implementar componente de logs \*/*}

<p>Logs de webhooks e atividades serão exibidos aqui</p>

</div>

)}

</div>

</div>

)}

<Modal

isOpen={showSetupModal}

onClose={() => setShowSetupModal(false)}

title="Configurar Integração Bitrix24"

>

<BitrixSetupForm

onSuccess={() => {

setShowSetupModal(false);

window.location.reload();

}}

onCancel={() => setShowSetupModal(false)}

/>

</Modal>

</div>

);

};

export default BitrixIntegrationPage;

**⏰ CRONOGRAMA DETALHADO COM CHECKPOINTS**

**📅 SEMANA 1: FINALIZAÇÃO DA MODULARIZAÇÃO + INTEGRAÇÃO BITRIX**

* **Dia 1:** FASE 1.1 - Análise completa do App.js atual
* **Dia 2:** FASE 1.2 - Criar sistema de roteamento
* **Dia 3:** FASE 1.3 - Refatorar App.js para estrutura modular
* **Dia 4:** FASE 2.6 - Implementar integração Bitrix (backend)
* **Dia 5:** FASE 1.4 - Testes e validação completa

**📅 SEMANA 2: PREENCHIMENTO DE ARQUIVOS VAZIOS + INTERFACE BITRIX**

* **Dia 1-2:** FASE 2.1-2.2 - Constants, Utils, Styles
* **Dia 3:** FASE 2.3 - Controllers do servidor
* **Dia 4:** Interface da integração Bitrix (frontend)
* **Dia 5:** Testes de integração e validação

**📅 SEMANA 3: SISTEMA DE CONFIGURAÇÃO DE TEXTOS**

* **Dia 1-2:** Sistema de configuração centralizada
* **Dia 3-4:** Interface de configuração de textos
* **Dia 5:** Integração textos ↔ Bitrix

**📅 SEMANA 4: FUNCIONALIDADES AVANÇADAS - PARTE 1**

* **Dia 1-2:** FASE 3.1 - Sistema de áudio avançado
* **Dia 3-4:** FASE 3.2 - Sistema de notificações
* **Dia 5:** Testes e ajustes

**📅 SEMANA 5: FUNCIONALIDADES AVANÇADAS - PARTE 2**

* **Dia 1-2:** FASE 3.3 - Sistema de IDs Bitrix
* **Dia 3-4:** FASE 3.4 - Aba lateral de arquivos
* **Dia 5:** Integração e testes

**📅 SEMANA 6: CHATBOT E AUTOMAÇÕES**

* **Dia 1-2:** FASE 3.8 - Sistema de chatbot/robô
* **Dia 3-4:** Testes e refinamento do chatbot
* **Dia 5:** Integração chatbot ↔ Bitrix

**📅 SEMANA 7: INTERFACE E CONFIGURAÇÕES**

* **Dia 1-2:** FASE 4.1 - Configurações de aparência
* **Dia 3-4:** FASE 4.2-4.3 - Melhorias no chat e pesquisa
* **Dia 5:** FASE 4.8 - Auto-resposta para ligações

**📅 SEMANA 8: QUALIDADE E DEPLOY**

* **Dia 1-3:** Testes completos, correções, otimizações
* **Dia 4-5:** Documentação final e deploy

**✅ CRITÉRIOS DE ACEITAÇÃO GLOBAIS**

**📊 MÉTRICAS TÉCNICAS**

* **App.js reduzido** de 53KB para <5KB
* **Zero arquivos vazios** (31 → 0)
* **Cobertura de testes** >80%
* **Performance score** >90 (Lighthouse)
* **Bundle size** <500KB (otimizado)
* **Tempo de carregamento** <3s
* **Zero erros** no console

**🎨 FUNCIONALIDADES OPERACIONAIS**

* **30 funcionalidades** implementadas e testadas
* **Login funcionando** com todos usuários teste
* **WhatsApp conectando** e mantendo conexão
* **Mensagens enviadas/recebidas** em tempo real
* **Upload/download** de todos tipos de arquivo
* **Áudio gravado/reproduzido** com qualidade
* **Notificações desktop** funcionando
* **Chatbot respondendo** adequadamente
* **Pesquisa retornando** resultados corretos
* **Temas aplicados** corretamente
* **Sistema de configuração de textos** operacional
* **Integração Bitrix24** enviando mensagens automaticamente

**🔐 CONTROLE DE ACESSO**

* **Admin:** Acesso total às configurações de textos e Bitrix
* **Manager:** Acesso limitado ao seu departamento
* **Agent:** Sem acesso às configurações
* **Viewer:** Sem acesso às configurações
* **Auditoria:** Log de todas as mudanças
* **Permissões:** Validação em frontend e backend

**🔗 INTEGRAÇÕES FUNCIONAIS**

* **IDs Bitrix** associadas corretamente
* **Links SIP** direcionando para PABX
* **Socket.IO** mantendo conexão estável
* **Auto-respostas** sendo enviadas
* **Aba lateral** navegando para mensagens
* **Histórico por ID** completo e preciso
* **Textos configuráveis** aplicados automaticamente
* **Webhooks Bitrix** processados corretamente
* \*\*

Tentar novamente

DF

Continuar

Editar

Planned to continue technical prompt from integration section.

O usuário quer que eu continue o prompt técnico que foi cortado. Vou continuar a partir de onde parou, que foi nos critérios de aceitação na seção de integrações funcionais. Preciso completar essa seção e finalizar o prompt técnico.

**🔗 INTEGRAÇÕES FUNCIONAIS (CONTINUAÇÃO)**

* **Webhooks Bitrix** processados corretamente
* **Templates de mensagem** renderizados com variáveis
* **Fila de mensagens** funcionando com retry
* **Rate limiting** respeitado com Bitrix24
* **Atividades registradas** de volta no Bitrix
* **Validação de origem** dos webhooks
* **Processamento assíncrono** de alta performance

**👥 EXPERIÊNCIA DO USUÁRIO**

* **Interface intuitiva** sem necessidade de treinamento
* **Responsiva** em desktop, tablet, mobile
* **Performance fluida** em todas as operações
* **Erros tratados** com mensagens claras
* **Loading states** em todas as operações assíncronas
* **Feedback visual** para todas as ações
* **Acessibilidade** básica (WCAG AA)
* **Configurações salvas** automaticamente
* **Preview em tempo real** das configurações

**📊 MONITORING E ANALYTICS**

* **Logs detalhados** de todas operações
* **Métricas de performance** coletadas
* **Dashboard de monitoring** operacional
* **Alertas automáticos** para falhas
* **Estatísticas de uso** disponíveis
* **Health checks** funcionando

**🚀 PRÓXIMO PASSO IMEDIATO - CHECKPOINT INICIAL**

**🔍 FASE 1.1 - ANÁLISE DETALHADA DO App.js ATUAL (53KB)**

**PROCESSO DETALHADO:**

1. **FAZER BACKUP COMPLETO**

bash

cd C:\Users\david\Desktop\whatsapp\_sistem\client\src

cp App.js App\_backup\_$(date +%Y%m%d\_%H%M%S).js

cp -r . ../app\_backup\_complete\_$(date +%Y%m%d)

1. **ANALISAR ESTRUTURA DO ARQUIVO**
   * Mapear todos os imports e dependências
   * Listar todos os useState e useEffect
   * Identificar todos os handlers de eventos
   * Documentar componentes definidos inline
   * Verificar integração com hooks existentes
   * Mapear rotas e navegação atual
2. **IDENTIFICAR FUNCIONALIDADES CRÍTICAS**
   * Sistema de autenticação (login/logout)
   * Conexão WhatsApp (QR code, status)
   * Envio/recebimento de mensagens
   * Upload/download de arquivos
   * Gravação/reprodução de áudio
   * Gerenciamento de conversas
   * Estados de UI (modais, loading, etc.)
3. **DOCUMENTAR DEPENDÊNCIAS**
   * Quais hooks são utilizados
   * Quais contextos são acessados
   * Quais services são chamados
   * Quais componentes são renderizados
4. **PLANEJAR MIGRAÇÃO**
   * Definir ordem de migração das funcionalidades
   * Identificar riscos e pontos críticos
   * Criar checklist de validação
   * Definir estratégia de rollback se necessário

**ENTREGÁVEIS:**

* 📋 Documento de análise técnica detalhada
* 📊 Mapeamento completo de funcionalidades
* 🗺️ Plano de migração passo a passo
* ⚠️ Lista de riscos e mitigações
* ✅ Checklist de testes para validação

**TEMPO ESTIMADO:** 4-6 horas

**CRITÉRIO DE CONCLUSÃO:** Documento completo aprovado e plano de migração validado

**🔧 GUIA DE CONFIGURAÇÃO BITRIX24**

**📝 PASSO A PASSO PARA CONFIGURAR NO BITRIX24**

**1. CRIAR WEBHOOK DE ENTRADA**

javascript

*// 1. Acessar: Configurações > Desenvolvedor > Webhooks de entrada*

*// 2. Clicar em "Criar webhook"*

*// 3. Configurar permissões:*

const webhookPermissions = {

required: [

'crm', *// Acesso ao CRM*

'user', *// Informações de usuários*

'contact', *// Gerenciar contatos*

'company', *// Gerenciar empresas*

'deal', *// Gerenciar negócios*

'lead' *// Gerenciar leads*

],

optional: [

'entity', *// Entidades personalizadas*

'telephony', *// Telefonia (para registrar chamadas)*

'activity' *// Atividades*

]

}

*// 4. O Bitrix gerará uma URL como:*

*// https://primem.bitrix24.com.br/rest/123/abc123def456/*

*// 5. Copiar esta URL para usar no sistema*

**2. CONFIGURAR WEBHOOK DE SAÍDA**

javascript

*// 1. Ir em: Configurações > Desenvolvedor > Webhooks de saída*

*// 2. Clicar em "Adicionar webhook"*

*// 3. Configurar:*

const outgoingWebhook = {

name: 'PRIMEM WhatsApp Integration',

url: 'https://seu-sistema.com/api/bitrix/webhook',

events: [

'ONCRMDEALADD', *// Deal criado*

'ONCRMDEALUPDATE', *// Deal atualizado*

'ONCRMLEADADD', *// Lead criado*

'ONCRMCONTACTADD', *// Contato criado*

'ONCRMCONTACTUPDATE' *// Contato atualizado*

],

authentication: 'Bearer seu\_api\_key\_aqui',

isActive: true

}

*// 4. Configurar headers personalizados:*

*// Authorization: Bearer sua\_api\_key\_aqui*

*// X-Bitrix-Secret: webhook\_secret\_key*

*// Content-Type: application/json*

**3. CONFIGURAR AUTOMAÇÕES**

javascript

*// 1. Ir em: CRM > Automação > Criar regra*

*// 2. Configurar gatilhos:*

const automationRules = [

{

name: 'Enviar Boas-vindas - Novo Deal',

entity: 'Deal',

trigger: {

event: 'Deal criado',

conditions: [

'Estágio = "Novo"',

'Valor > R$ 1.000',

'Campo "Enviar WhatsApp" = Sim'

]

},

action: {

type: 'Webhook',

url: 'https://seu-sistema.com/api/bitrix/send-message',

method: 'POST',

data: {

templateName: 'new\_deal\_welcome',

dealId: '{ID}',

contactId: '{CONTACT\_ID}',

variables: {

dealTitle: '{TITLE}',

dealAmount: '{OPPORTUNITY}',

contactName: '{CONTACT\_ID.NAME}'

}

}

}

},

{

name: 'Follow-up Inatividade',

entity: 'Deal',

trigger: {

event: 'Deal sem atividade há X dias',

conditions: [

'Dias sem atividade >= 7',

'Estágio != "Ganho"',

'Estágio != "Perdido"'

]

},

action: {

type: 'Webhook',

url: 'https://seu-sistema.com/api/bitrix/send-message',

method: 'POST',

data: {

templateName: 'follow\_up\_inactive',

dealId: '{ID}',

contactId: '{CONTACT\_ID}',

variables: {

dealTitle: '{TITLE}',

daysInactive: '{DAYS\_INACTIVE}',

contactName: '{CONTACT\_ID.NAME}'

}

}

}

}

];

**4. CRIAR CAMPOS PERSONALIZADOS**

javascript

*// 1. Ir em: CRM > Configurações > Campos personalizados*

*// 2. Criar campos:*

const customFields = [

{

entity: 'Deal',

field: {

name: 'UF\_SEND\_WHATSAPP',

type: 'boolean',

title: 'Enviar WhatsApp',

description: 'Permitir envio de mensagens automáticas',

defaultValue: 'Y',

isRequired: false,

showInList: true

}

},

{

entity: 'Contact',

field: {

name: 'UF\_WHATSAPP\_PHONE',

type: 'string',

title: 'WhatsApp',

description: 'Número do WhatsApp se diferente do telefone principal',

defaultValue: '',

isRequired: false,

showInList: true

}

},

{

entity: 'Contact',

field: {

name: 'UF\_PREFERRED\_TIME',

type: 'list',

title: 'Horário Preferido',

description: 'Melhor horário para contato',

options: [

'Manhã (8h-12h)',

'Tarde (12h-18h)',

'Noite (18h-22h)',

'Qualquer horário'

],

defaultValue: 'Qualquer horário',

isRequired: false

}

},

{

entity: 'Deal',

field: {

name: 'UF\_WHATSAPP\_SENT',

type: 'datetime',

title: 'Último WhatsApp Enviado',

description: 'Data/hora do último WhatsApp enviado',

defaultValue: null,

isRequired: false,

showInList: false

}

}

];

**📋 FLUXO COMPLETO DE CONFIGURAÇÃO**

mermaid

graph TD

A[Acessar Bitrix24] --> B[Configurações]

B --> C[Desenvolvedor]

C --> D[Webhooks de Entrada]

D --> E[Criar Webhook]

E --> F[Definir Permissões]

F --> G[Copiar URL Gerada]

C --> H[Webhooks de Saída]

H --> I[Adicionar Webhook]

I --> J[Configurar URL Sistema]

J --> K[Definir Eventos]

K --> L[Configurar Autenticação]

B --> M[CRM]

M --> N[Automação]

N --> O[Criar Regras]

O --> P[Definir Gatilhos]

P --> Q[Configurar Ações]

M --> R[Configurações]

R --> S[Campos Personalizados]

S --> T[Criar Campos]

T --> U[Testar Integração]

**🎯 CONCLUSÃO E PRÓXIMOS PASSOS**

**📊 RESUMO DO PROMPT TÉCNICO COMPLETO**

Este prompt técnico fornece:

✅ **Estrutura atual mapeada** - Todos os 89 arquivos catalogados  
✅ **30 funcionalidades especificadas** - Detalhes técnicos completos  
✅ **Integração Bitrix24 completa** - Webhooks bidirecionais e automações  
✅ **Sistema de configuração de textos** - Controle centralizado de mensagens  
✅ **Controle de acesso por perfis** - Admin, Manager, Agent, Viewer  
✅ **Plano linear de 8 semanas** - Passo a passo detalhado  
✅ **Códigos de exemplo** - Implementações reais  
✅ **Critérios de aceitação** - Métricas claras de sucesso  
✅ **Checkpoints de validação** - Pontos de controle em cada fase  
✅ **Cronograma com datas** - Timeline executável  
✅ **Configuração Bitrix24** - Guia passo a passo

**🔥 PRINCIPAIS NOVIDADES v16.0:**

1. **Sistema de Configuração Centralizada de Textos** - Controle total sobre todas as mensagens
2. **Integração Completa Bitrix24** - Webhooks bidirecionais, templates, fila inteligente
3. **Controle de Acesso Granular** - 4 perfis de usuário com permissões específicas
4. **Processamento Assíncrono** - Alta performance para grandes volumes
5. **Auditoria Completa** - Log de todas as alterações e atividades
6. **Interface de Administração** - Painel completo para configurações

**❓ QUAL FASE VOCÊ QUER INICIAR?**

**A.** 🔍 **FASE 1.1** - Análise detalhada do App.js atual (53KB)  
**B.** 🔗 **INTEGRAÇÃO BITRIX** - Implementar webhooks e automações  
**C.** ⚙️ **SISTEMA DE TEXTOS** - Configuração centralizada de mensagens  
**D.** 🧹 **LIMPEZA** - Preencher arquivos vazios primeiro  
**E.** 🤖 **TESTE TÉCNICO** - Validar viabilidade específica

**🎯 RECOMENDAÇÃO ESTRATÉGICA:**

Iniciar com **A (FASE 1.1)** seguindo o plano linear detalhado, pois:

1. **Modularização** resolve a base técnica crítica
2. **Integração Bitrix** pode ser testada imediatamente após
3. **Sistema de textos** depende da estrutura modular
4. **Todas outras funcionalidades** dependem da base sólida

**📝 CONFIGURAÇÃO PRÉ-REQUISITOS:**

Antes de iniciar qualquer fase, garantir:

* **Backup completo** do projeto atual
* **Bitrix24 configurado** com webhooks de entrada/saída
* **Ambiente de desenvolvimento** preparado
* **Banco de dados** com estrutura para novas tabelas
* **Variáveis de ambiente** configuradas (.env)

**🔐 CONFIGURAÇÃO DE SEGURANÇA BITRIX**

**🛡️ CHECKLIST DE SEGURANÇA**

javascript

const securityChecklist = {

authentication: {

apiKeys: {

generation: 'UUID v4 + timestamp + random salt',

storage: 'Encrypted with AES-256-GCM',

rotation: 'Automatic every 30 days',

validation: 'HMAC signature verification'

},

webhookSecrets: {

algorithm: 'HMAC-SHA256',

validation: 'Compare computed vs received signature',

tolerance: '5 minutes timestamp drift',

ipWhitelist: 'Strict IP validation'

}

},

networking: {

https: 'Required for all communications',

tls: 'Minimum TLS 1.2',

certificates: 'Valid SSL certificates only',

cors: 'Configured for Bitrix24 domains only'

},

dataProtection: {

encryption: 'AES-256 for sensitive data at rest',

masking: 'PII masked in logs and responses',

retention: '90 days for operational logs',

anonymization: 'Automatic after retention period'

},

monitoring: {

logging: 'All API calls logged with sanitized data',

alerting: 'Real-time alerts for security events',

anomaly: 'Rate limiting and anomaly detection',

audit: 'Complete audit trail for compliance'

}

}

*// IMPLEMENTAÇÃO DE VALIDAÇÃO DE WEBHOOK:*

function validateBitrixWebhook(req) {

*// 1. Verificar IP de origem*

const allowedIPs = [

'185.54.36.0/24', *// Bitrix24 EU servers*

'185.54.37.0/24' *// Bitrix24 EU servers*

];

if (!isIpInRange(req.ip, allowedIPs)) {

throw new Error('IP not allowed');

}

*// 2. Verificar assinatura HMAC*

const signature = req.headers['x-bitrix-signature'];

const payload = JSON.stringify(req.body);

const secret = process.env.BITRIX\_WEBHOOK\_SECRET;

const expectedSignature = crypto

.createHmac('sha256', secret)

.update(payload)

.digest('hex');

if (!crypto.timingSafeEqual(

Buffer.from(signature, 'hex'),

Buffer.from(expectedSignature, 'hex')

)) {

throw new Error('Invalid signature');

}

*// 3. Verificar timestamp*

const timestamp = req.body.ts || req.headers['x-timestamp'];

const now = Math.floor(Date.now() / 1000);

if (Math.abs(now - timestamp) > 300) { *// 5 minutos*

throw new Error('Request too old');

}

return true;

}

*📅 Prompt técnico criado: Setembro 2025*  
*👨‍💻 Projeto: PRIMEM WhatsApp Business System*  
*📊 Escopo: 89 arquivos, 30 funcionalidades, 8 semanas*  
*🔗 Novidade: Integração completa Bitrix24 + Sistema de configuração de textos*  
*🎯 Status: Pronto para execução linear com prioridade na modularização*

**O projeto está completamente especificado e pode ser retomado de qualquer ponto seguindo este roadmap detalhado. A integração com Bitrix24 está tecnicamente viável e documentada passo a passo.**