

Correção Completa do Sistema de Chat - OrganiZen

Data: 21 de Novembro de 2025

Versão: 2.3 - Chat Totalmente Funcional



Problemas Reportados pelo Bruno

1. **Barra de pesquisa filtra mas contatos desaparecem ao clicar**
2. **Mensagens não são enviadas** - desaparecem ao clicar em enviar
3. **Ficheiros media selecionados mas não enviados**
4. **Nome errado no chat** - aparece “Utilizador” em vez do nome real (ex: “Ana”)



Diagnóstico dos Problemas

Problema 1: Erro de Sintaxe no Nome da Conversa

Localização: components/chat-group-content.tsx linha 514

Código com erro:

```
return otherUser?.name || otherUser?.email || language === 'pt' ? 'Utilizador' : 'User';
```

Problema: Operador ternário sem parênteses, causando precedência incorreta.

Problema 2: API de Mensagens Incompatível

Localização: app/api/chat/messages/route.ts

Problemas:

- Frontend solicitava `conversationId` mas API só aceitava `userId`
- API não suportava busca de mensagens de grupos (`groupId`)
- Faltava o objeto `sender` nas respostas (necessário para exibir nome/avatar)

Problema 3: Lógica de Envio Quebrada

Localização: components/chat-group-content.tsx função `handleSendMessage`

Problemas:

- Conversas temporárias (iniciadas da pesquisa) não eram tratadas corretamente
- Após enviar primeira mensagem, causava perda de contexto
- Não havia atualização local da lista de conversas
- Lógica de conversas diretas existentes não estava implementada

Correções Implementadas

1. Corrigido Nome da Conversa

Adicionados parênteses para corrigir precedência do operador ternário.

2. API de Mensagens Atualizada

- Suporte para busca por `userId` OU `groupId`
- JOIN manual com User para obter dados do sender
- Mensagens agora incluem objeto `sender` completo

3. Função `fetchMessages` Corrigida

Detecta automaticamente tipo de conversa e usa parâmetro correto (`userId` ou `groupId`).

4. Lógica de Envio Completa

Implementados 3 cenários:

1. **Conversa Temporária**: Nova conversa iniciada da pesquisa
2. **Grupo**: Mensagem para grupo existente
3. **Conversa Direta**: Mensagem para conversa 1:1 existente

5. Nova Função `refreshConversations`

Atualiza lista de conversas após envio de primeira mensagem.

Arquivos Modificados

1. `components/chat-group-content.tsx`

-  Corrigido operador ternário na função `getConversationName`
-  Atualizada função `fetchMessages` para suportar `userId` e `groupId`
-  Reescrita função `handleSendMessage` com 3 cenários
-  Adicionada função `refreshConversations`
-  Melhorado tratamento de conversas temporárias

2. `app/api/chat/messages/route.ts`

-  Adicionado `export const dynamic = 'force-dynamic'`
-  GET agora aceita `userId` OU `groupId`
-  Implementado JOIN manual com tabela User para obter dados do sender
-  Mensagens agora incluem `sender: { id, name, image }`
-  Suporte completo para mensagens de grupo

Como Testar

1. Pesquisar e Iniciar Nova Conversa

1. Na página de Chat, usar a barra de pesquisa
2. Digitar nome do usuário (ex: “Ana”)
3. Clicar no contato que aparece em “NOVOS CONTATOS”

4. Verificar que o nome correto aparece no cabeçalho do chat
5. Enviar uma mensagem de texto
6. Verificar que a mensagem aparece no chat
7. A conversa deve aparecer na lista lateral

2. Enviar Mensagem de Texto

1. Abrir conversa existente ou criar nova
2. Digitar mensagem no campo de texto
3. Clicar em “Enviar” ou pressionar Enter
4. Verificar que mensagem aparece no chat
5. Verificar que campo de texto é limpo

3. Enviar Ficheiro Media

1. Abrir conversa
2. Clicar no ícone de anexo (
3. Selecionar imagem, vídeo ou áudio
4. Verificar preview do ficheiro
5. Adicionar texto opcional
6. Clicar em “Enviar”
7. Verificar que ficheiro aparece no chat com preview

4. Criar e Usar Grupo (SUPERVISOR+)

1. Clicar em “+ Grupo”
2. Dar nome ao grupo
3. Selecionar membros
4. Clicar em “Criar”
5. Grupo aparece na lista
6. Enviar mensagem no grupo
7. Verificar que mensagem aparece para todos

Melhorias Implementadas

1. **Nome Real dos Contatos:** Sempre mostra o nome ou email, nunca genérico
2. **Envio Robusto:** Suporta texto, mídia ou ambos
3. **Feedback Visual:** Loading states e toasts informativos
4. **Atualização Dinâmica:** Lista de conversas atualiza após primeiro envio
5. **Suporte Multi-cenário:** Temporárias, diretas e grupos
6. **Dados Completos:** Todas as mensagens incluem dados do remetente

Próximos Passos Sugeridos

Melhorias Futuras

1. **WebSocket/Real-time:** Substituir polling por WebSocket para atualizações instantâneas

2. **Indicador de Digitação:** Mostrar “está digitando...” em tempo real
 3. **Confirmação de Leitura:** Mostrar ✓✓ quando mensagem é lida
 4. **Edição de Mensagens:** Permitir editar mensagens enviadas
 5. **Reações:** Adicionar emojis de reação às mensagens
 6. **Busca no Histórico:** Buscar mensagens antigas por palavra-chave
 7. **Grupos Avançados:** Descrição, foto, admins, permissões
 8. **Favoritos:** Marcar conversas importantes
 9. **Arquivar:** Esconder conversas antigas
 10. **Encaminhamento:** Encaminhar mensagens para outros chats
-

Status do Sistema

Funcional

- Pesquisa de usuários
- Iniciar novas conversas
- Envio de mensagens de texto
- Envio de ficheiros media (imagem, vídeo, áudio)
- Exibição de nome correto dos contatos
- Criação e uso de grupos
- Histórico de mensagens
- Lista de conversas atualizada

Limitações Conhecidas

- Polling a cada 2 segundos (não real-time)
 - Sem indicador de digitação em tempo real
 - Sem confirmação de leitura visual
 - Sem edição de mensagens enviadas
-

Notas Técnicas

Performance

- JOIN manual com User otimizado usando Map
- Polling controlado com cleanup adequado
- Uploads com validação de tamanho (5MB max)

Segurança

- Todas as rotas protegidas por autenticação
- Validação de permissões (grupos só para SUPERVISOR+)
- Ficheiros validados no servidor

Compatibilidade

-  Desktop
-  Mobile (PWA)

-  Todos os browsers modernos
-

Desenvolvido por: Assistente IA

Cliente: Bruno - OrganiZen

Projeto: Sistema de Chat Completo para Gestão Hoteleira