

Correção da Criação de Grupos no Chat - OrganiZen

Data: 21 de Novembro de 2025

Versão: 3.2 - Bug Crítico na API Corrigido



Bug Crítico Reportado pelo Bruno

Situação:

- Usuário conseguia abrir o modal de criar grupo
- Conseguia selecionar membros (checkboxes funcionavam)
- **Mas ao clicar em “Criar” → NADA ACONTECIA ✗**
- Modal permanecia aberto
- Grupo não era criado
- Sem mensagem de erro visível

Gravidade: CRÍTICA 🛡

- Funcionalidade completamente quebrada
- Impede criação de qualquer grupo
- Nenhum feedback ao usuário



Diagnóstico do Problema

1. Investigação da API

Arquivo investigado: app/api/chat/conversations/route.ts

Código PROBLEMÁTICO (linha 159-166):

```
await prisma.chatGroupMember.createMany({
  data: uniqueMemberIds.map(userId => ({
    groupId: group.id,
    userId,
    role: userId === currentUserId ? 'admin' : 'member',
    companyId // ✗ CAMPO NÃO EXISTE NO SCHEMA!
  }))
});
```

Por que isso quebrava tudo:

- O código tentava inserir campo `companyId` na tabela `ChatGroupMember`
- Mas o **schema Prisma NÃO tem esse campo!** ✗
- Prisma lançava exceção no banco de dados
- API retornava erro 500
- Frontend não criava o grupo
- Mas sem feedback visual claro

2. Verificação do Schema Prisma

Schema correto (prisma/schema.prisma):

```
model ChatGroupMember {
    id      String  @id @default(cuid())
    groupId String // ✓ Existe
    userId  String // ✓ Existe
    role    String? // ✓ Existe
    joinedAt DateTime @default(now())
    isMuted Boolean @default(false)
    createdAt DateTime @default(now())
    updatedAt DateTime @updatedAt

    @@map("chat_group_members")
    @@unique([groupId, userId])
    @@index([groupId])
    @@index([userId])
}
```

Campos disponíveis:

- ✓ `id` - ID único
- ✓ `groupId` - ID do grupo
- ✓ `userId` - ID do utilizador
- ✓ `role` - Papel (admin, member)
- ✓ `joinedAt` - Data de entrada
- ✓ `isMuted` - Se está silenciado
- ✗ `companyId` NÃO EXISTE!

Erro gerado:

```
PrismaClientValidationError: Invalid `prisma.chatGroupMember.createMany()` invocation:
Unknown field `companyId` for model ChatGroupMember
```

3. Análise do Frontend

Arquivo: components/chat-group-content.tsx

Código da função (linha 507-547):

```

const handleCreateGroup = async () => {
  if (!groupName.trim() || selectedMembers.length === 0) {
    toast({
      title: language === 'pt' ? 'Erro' : 'Error',
      description: language === 'pt' ? 'Nome do grupo e membros são obrigatórios' : 'G
roup name and members are required',
      variant: 'destructive'
    });
    return;
  }

  try {
    const response = await fetch('/api/chat/conversations', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify({
        name: groupName,
        isGroup: true,
        memberIds: selectedMembers
      })
    });

    if (response.ok) {
      const newConv = await response.json();
      setConversations(prev => [newConv, ...prev]);
      setIsCreateGroupOpen(false);
      setGroupName('');
      setSelectedMembers([]);
      toast({
        title: language === 'pt' ? 'Sucesso' : 'Success',
        description: language === 'pt' ? 'Grupo criado com sucesso!' : 'Group created
successfully!'
      });
    }
    // ✗ SEM TRATAMENTO DE ERRO QUANDO response.ok = false
  } catch (error) {
    console.error('Failed to create group:', error);
    toast({
      title: language === 'pt' ? 'Erro' : 'Error',
      description: language === 'pt' ? 'Erro ao criar grupo' : 'Failed to create
group',
      variant: 'destructive'
    });
  }
};

```

Problemas identificados:

1. ✗ Sem tratamento de erro quando `response.ok = false`
2. ✗ Sem estado de loading (usuário não sabe se está processando)
3. ✗ Sem proteção contra cliques duplicados
4. ✗ Botão não desabilitado durante criação
5. ✗ Sem feedback específico do erro do servidor

Resultado:

- API retorna erro 500
- `response.ok = false`
- Código cai no silêncio (sem `else`)
- Modal permanece aberto

- Sem mensagem de erro
 - Usuário não sabe o que aconteceu
-

✓ Solução Implementada

1. Correção da API - Remover Campo Inexistente

Arquivo: app/api/chat/conversations/route.ts

ANTES (ERRADO):

```
await prisma.chatGroupMember.createMany({
  data: uniqueMemberIds.map(userId => ({
    groupId: group.id,
    userId,
    role: userId === currentUserId ? 'admin' : 'member',
    companyId // ❌ Campo não existe
  }))
});
```

DEPOIS (CORRETO):

```
await prisma.chatGroupMember.createMany({
  data: uniqueMemberIds.map(userId => ({
    groupId: group.id,
    userId,
    role: userId === currentUserId ? 'admin' : 'member'
    // ✅ companyId removido
  }))
});
```

Mudança:

- ✅ Removido campo `companyId` que não existe no schema
 - ✅ Mantidos apenas campos válidos
 - ✅ API agora consegue criar membros sem erro
-

2. Melhorias no Frontend - Estados e Feedback

Arquivo: components/chat-group-content.tsx

A) Adicionado Estado de Loading

Código adicionado (linha 122):

```
const [isCreatingGroup, setIsCreatingGroup] = useState(false);
```

Objetivo:

- Controlar quando está criando grupo
 - Desabilitar botões durante processo
 - Mostrar feedback visual “Criando...”
-

B) Melhorado Tratamento de Erros

ANTES:

```
if (response.ok) {
    // ... sucesso
}
// ❌ NADA se response.ok = false
```

DEPOIS:

```
if (response.ok) {
    // ... sucesso
} else {
    // ✅ Tratar erro do servidor
    const errorData = await response.json().catch(() => ({}));
    toast({
        title: language === 'pt' ? 'Erro' : 'Error',
        description: errorData.error || (language === 'pt' ? 'Erro ao criar grupo' : 'Failed to create group'),
        variant: 'destructive'
    });
}
```

Melhorias:

- ✅ Detecta quando `response.ok = false`
- ✅ Tenta obter mensagem de erro do servidor
- ✅ Mostra toast com erro específico
- ✅ Fallback se não conseguir parsear JSON

C) Proteção Contra Cliques Duplicados

Código adicionado:

```
const handleCreateGroup = async () => {
    if (!groupName.trim() || selectedMembers.length === 0) {
        // ... validação
        return;
    }

    if (isCreatingGroup) return; // ✅ Evitar cliques duplicados

    setIsCreatingGroup(true); // ✅ Marcar como "criando"

    try {
        // ... lógica de criação
    } catch (error) {
        // ... tratamento de erro
    } finally {
        setIsCreatingGroup(false); // ✅ Sempre desmarcar no final
    }
};
```

Benefícios:

- ✅ Evita múltiplos cliques no botão “Criar”

- Previne criação de grupos duplicados
 - Usa `finally` para garantir reset do estado
-

D) Botão com Feedback Visual

ANTES:

```
<Button onClick={handleCreateGroup}>
  {language === 'pt' ? 'Criar' : 'Create'}
</Button>
```

DEPOIS:

```
<Button
  onClick={handleCreateGroup}
  disabled={isCreatingGroup || !groupName.trim() || selectedMembers.length === 0}
>
  {isCreatingGroup ? (
    <>
      <span className="mr-2">⏳</span>
      {language === 'pt' ? 'Criando...' : 'Creating...'}
    </>
  ) : (
    language === 'pt' ? 'Criar' : 'Create'
  )}
</Button>
```

Mudanças:

1. **Desabilitado quando:**

- Está criando (`isCreatingGroup`)
- Nome vazio (`!groupName.trim()`)
- Sem membros (`selectedMembers.length === 0`)

1. **Texto dinâmico:**

- Normal: “Criar” / “Create”
- Loading: “⏳ Criando...” / “⏳ Creating...”

2. **Emoji visual:**

- Ampulheta (⏳) durante criação
- Indica processamento ativo

E) Botão Cancelar Também Desabilitado

Código:

```

<Button
  variant="outline"
  onClick={() => setIsCreateGroupOpen(false)}
  disabled={isCreatingGroup} // ✅ Desabilitar durante criação
>
  {language === 'pt' ? 'Cancelar' : 'Cancel'}
</Button>

```

Razão:

- Previne fechar modal durante criação
- Evita inconsistências de estado
- Força aguardar resposta do servidor



Comparação Antes/Depois

ANTES da Correção ✗

Fluxo do Usuário:

1. Clicar em "+ Grupo"
↓
2. Modal abre ✅
↓
3. Digitar nome: "Marketing"
↓
4. Selecionar membros: João, Maria ✅
↓
5. Clicar em "Criar"
↓
6. ✗ NADA ACONTECE
↓
7. Modal permanece aberto
↓
8. Sem feedback de erro
↓
9. Usuário confuso 😕

Experiência:

- ⚠️ Botão clicável mas não faz nada
- ⚠️ Sem indicação de processamento
- ⚠️ Sem mensagem de erro
- ⚠️ Modal não fecha
- ✗ Grupo não é criado

Erro no servidor:

```
PrismaClientValidationError: Unknown field `companyId` for model ChatGroupMember
```

No frontend:

- Sem mensagem de erro
- Console pode ter log, mas usuário não vê

- Botão continua clicável
 - Modal permanece aberto
-

DEPOIS da Correção

Fluxo do Usuário:

1. Clicar em "+ Grupo"

2. Modal abre 

3. Digitar nome: "Marketing"

4. Selecionar membros: João, Maria 

5. Clicar em "Criar"

6.  Botão mostra "⏳ Criando..."

7.  Botões desabilitados temporariamente

8.  Grupo criado no servidor

9.  Grupo aparece na lista

10.  Modal fecha automaticamente

11.  Toast: "Grupo criado com sucesso!"

Experiência:

-  Feedback imediato ("Criando...")
-  Botões desabilitados durante processo
-  Mensagem de sucesso clara
-  Modal fecha automaticamente
-  Grupo aparece na lista
-  Se houver erro, mensagem específica

No servidor:

-  Sem erros de Prisma
-  Membros criados corretamente
-  API retorna sucesso (201)

No frontend:

-  Estado visual claro
 -  Feedback em tempo real
 -  Proteção contra cliques duplicados
 -  Tratamento de erro robusto
-



Visualização da Correção

Modal ANTES (Bug) ❌

Criar Grupo

Nome **do** Grupo

Membros

João Silva
 Maria Santos
 Pedro Costa

[Cancelar]
[Criar]

← Clicável mas não funciona ❌

(Clicar em "Criar" → NADA acontece)

Modal DEPOIS (Corrigido) ✓

Estado Normal:

Criar Grupo

Nome **do** Grupo

Membros

João Silva
 Maria Santos
 Pedro Costa

[Cancelar]
[Criar]

← Clicável ✓

Estado Loading:

Criar Grupo

Nome do Grupo

Marketing

Membros

João Silva
 Maria Santos
 Pedro Costa

[Cancelar] [ Criando...]
 (desabilitado) (desabilitado)

← Processando ✓

Estado Sucesso:

 Sucesso
 Grupo criado com sucesso!

↓
 (Modal fecha automaticamente)
 ↓

Chat + Grupo

Todas 2 Diretas 1 Grupos 1

← Contador aumentou

 Marketing [2]
 Grupo criado

← Novo grupo!

M Marisia
 hello

Testes de Validação

Teste 1: Criar Grupo com Sucesso

1. Abrir modal "+ Grupo"
2. VERIFICAR:
 - Modal abre
 - Lista** de usuários visível
3. Digitar nome: "Equipa Marketing"
4. Selecionar membros: João, Maria
5. Clicar em "Criar"
6. VERIFICAR:
 - Botão muda para "🕒 Criando..."
 - Botões ficam desabilitados
 - Após 1-2s: modal fecha
 - Toast:** "Grupo criado com sucesso!"
 - Grupo aparece na **lista**
 - Contador** "Grupos" aumenta (0 → 1)
7. Clicar no grupo
8. VERIFICAR:
 - Grupo abre normalmente
 - Membros corretos (você + João + Maria)
 - Pode enviar mensagens

Teste 2: Validação de Campos Vazios

1. Abrir modal "+ Grupo"
2. NÃO digitar nome
3. NÃO selecionar membros
4. VERIFICAR:
 - Botão "Criar" está DESABILITADO
5. Digitar nome: "Teste"
6. VERIFICAR:
 - Botão ainda DESABILITADO (sem membros)
7. Selecionar 1 membro
8. VERIFICAR:
 - Botão agora HABILITADO 

Teste 3: Nome Vazio mas Membros Selecionados

1. Abrir modal "+ Grupo"
2. Selecionar membros: João, Maria
3. Clicar em campo nome mas NÃO digitar
4. Clicar em "Criar"
5. VERIFICAR:
 - Toast** de erro aparece
 - Mensagem: "Nome do grupo e membros são obrigatórios"
 - Modal permanece aberto
 - Membros ainda selecionados

Teste 4: Cliques Duplos

1. Abrir modal "+ Grupo"
2. Preencher nome e membros
3. Clicar em "Criar" DUAS VEZES rapidamente
4. VERIFICAR:
 - Botão desabilita após primeiro clique
 - Segundo clique não faz nada
 - Só 1 grupo é criado (não duplica)
 - Toast aparece 1 vez só

Teste 5: Cancelar Durante Criação

1. Abrir modal "+ Grupo"
2. Preencher nome e membros
3. Clicar em "Criar"
4. IMEDIATAMENTE tentar clicar "Cancelar"
5. VERIFICAR:
 - Botão "Cancelar" está desabilitado
 - Não consegue fechar modal
 - Força aguardar resposta do servidor
 - Modal só fecha após sucesso/erro

Teste 6: Criar Múltiplos Grupos

1. Criar grupo "Marketing"
2. Criar grupo "Vendas"
3. Criar grupo "Suporte"
4. VERIFICAR:
 - Todos 3 grupos aparecem na lista
 - Contador "Grupos" = 3
 - Cada grupo tem membros corretos
 - Pode abrir qualquer um

Teste 7: Persistência (Recarregar)

1. Criar grupo "Teste"
2. Recarregar página (F5)
3. VERIFICAR:
 - Grupo "Teste" ainda aparece ✓
 - Membros corretos
 - Pode enviar mensagens
 - Tudo funciona normalmente



Detalhes Técnicos

Por que companyId estava no código?

Hipótese:

- Provavelmente foi copiado de outro modelo
- Ou o schema mudou e código não foi atualizado
- Comum em refatorações rápidas

Modelos que TÊM companyID :

- ✓ ChatGroup - tem companyID
- ✓ ChatMessage - tem companyID
- ✗ ChatGroupMember - NÃO tem

Confusão comum:

```
// ChatGroup - TEM companyID ✓
await prisma.chatGroup.create({
  data: {
    name,
    companyId, // ✓ OK aqui
    ...
  }
});

// ChatGroupMember - NÃO TEM companyID ✗
await prisma.chatGroupMember.create({
  data: {
    groupId,
    userId,
    companyId, // ✗ ERRO aqui!
  }
});
```

Por que Não Adicionar companyID ao Schema?

Opção 1: Adicionar campo ao schema ✗

```
model ChatGroupMember {
  // ...
  companyId String // Adicionar isso?
}
```

Problemas:

- Redundante (já está em ChatGroup)
- Aumenta tamanho do banco
- Pode ficar dessincronizado
- Não é necessário

Opção 2: Remover do código ✓ (escolhida)

```
// Não usar companyId
await prisma.chatGroupMember.createMany({
  data: uniqueMemberIds.map(userId => ({
    groupId: group.id,
    userId,
    role: userId === currentUserId ? 'admin' : 'member'
    // Sem companyId ✓
  }))
});
```

Vantagens:

- Código correto com schema existente
- Sem mudança no banco de dados

- Fix rápido e seguro
- Sem risco de migration

Como obter companyId se precisar:

```
// Via JOIN com ChatGroup
const member = await prisma.chatGroupMember.findUnique({
  where: { id: memberId },
  include: {
    group: {
      select: { companyId: true } // ✓ Vem do grupo
    }
  }
});

const companyId = member.group.companyId;
```

Por que finally no Try-Catch?

Código:

```
try {
  // ... criar grupo
} catch (error) {
  // ... tratar erro
} finally {
  setIsCreatingGroup(false); // ✓ SEMPRE executa
}
```

Benefícios:

- finally SEMPRE executa
- Mesmo com sucesso
- Mesmo com erro
- Mesmo com return dentro do try
- Garante reset do estado

Sem finally (ERRADO):

```
try {
  // ... criar grupo
  setIsCreatingGroup(false); // ✗ Só executa se sucesso
} catch (error) {
  // ... tratar erro
  setIsCreatingGroup(false); // ✗ Precisa repetir código
}
```

Problema:

- Código duplicado
- Fácil esquecer em um dos blocos
- Se houver return, não executa



Resumo das Mudanças

Arquivos Modificados

1. app/api/chat/conversations/route.ts

- ✓ Removido campo `companyId` do `prisma.chatGroupMember.createMany()`
- ✓ API agora cria membros sem erro
- ✓ Retorna sucesso (201) corretamente

2. components/chat-group-content.tsx

- ✓ Adicionado estado `isCreatingGroup`
- ✓ Proteção contra cliques duplicados
- ✓ Melhor tratamento de erros (`else` para `response.ok = false`)
- ✓ Feedback visual no botão (“⏳ Criando...”)
- ✓ Botões desabilitados durante criação
- ✓ Toast com mensagem específica do servidor
- ✓ Uso de `finally` para garantir reset de estado



Status Final

Funcionalidade	Antes	Agora
API criar grupo	✗ Erro Prisma	✓ Funciona
Botão “Criar”	✗ Não funciona	✓ Funciona
Feedback loading	✗ Não tinha	✓ “Criando...”
Tratamento de erro	✗ Silencioso	✓ Toast com erro
Clique duplicados	⚠ Permitido	✓ Bloqueado
Modal fecha	✗ Permanece aberto	✓ Fecha após sucesso
Grupo aparece na lista	✗ Não	✓ Sim
Mensagem de sucesso	✗ Não	✓ Sim
Persistência	✗ Não criava	✓ Grupo salvo no BD



Conclusão

Bug crítico corrigido com sucesso! ✓

O que estava errado:

- API tentava criar membros com campo inexistente (`companyId`)

- Prisma lançava erro de validação
- Frontend não tratava erro de resposta
- Sem feedback visual para o usuário

O que foi feito:

- Removido campo `companyId` da criação de membros
- Adicionado estado de loading
- Melhorado tratamento de erros
- Feedback visual claro para o usuário
- Proteção contra cliques duplicados

Resultado:

- Criação de grupos funciona perfeitamente
- Feedback visual em todas as etapas
- Mensagens de erro claras
- UX profissional e polida
- Sem bugs ou comportamentos inesperados

Bruno, agora pode criar grupos sem problemas!

Desenvolvido por: Assistente IA

Cliente: Bruno - OrganiZen

Projeto: Sistema de Chat - Correção Criação de Grupos

Data: 21 de Novembro de 2025