📧 Sistema de Envio de Emails - OrganiZen

Resumo da Implementação

Foi implementado um **sistema completo de envio de emails** usando o serviço **Resend**, integrado com os templates personalizados de branding da empresa.

✨ Funcionalidades Implementadas

1. Email de Boas-Vindas

- ✅ Enviado automaticamente após cadastro (signup)
- ✅ Usa template personalizado do branding (se configurado)
- ✅ Inclui logo e cores da empresa

2. Email de Redefinição de Senha

- ✅ Endpoint para solicitar reset de senha (`/api/auth/forgot-password`)
- ✅ Endpoint para confirmar reset com token (`/api/auth/reset-password-confirm`)
- ✅ Geração de token seguro com expiração de 1 hora
- ✅ Email com link de reset personalizado

3. Email de Notificação de Reset por Admin

- ✅ Enviado quando admin reseta a senha de um usuário
- ✅ Alerta de segurança para o usuário

4. Email de Convite/Adição à Equipe

- ✅ Enviado quando usuário é adicionado a uma equipe
- ✅ Inclui nome da equipe e departamento

🔧 Arquitetura Técnica

Biblioteca de Email (`lib/email.ts`)

- **Função principal:** `sendEmail()` - centraliza todo o envio
- **Funções específicas:**
- `sendWelcomeEmail()`
- `sendPasswordResetEmail()`
- `sendTeamInviteEmail()`
- `sendNotificationEmail()`

Integração com Branding

- ✅ Busca automática de templates personalizados do `CompanyBranding`
- ✅ Aplicação de logo, cores e mensagens customizadas
- ✅ Fallback para templates padrão caso não haja personalização

Banco de Dados

- ✅ Novo modelo `PasswordResetToken` para gerenciar tokens de reset
- `email`: Email do usuário
- `token`: Token único gerado
- `expires`: Data de expiração (1 hora)
- `used`: Flag indicando se já foi usado

📍 Pontos de Integração

1. Signup (`/api/signup/route.ts`)

```
// Email de boas-vindas enviado após criação do usuário
await sendWelcomeEmail(email, fullName, companyId, companyName);
```

2. Reset de Senha por Admin (`/api/users/[id]/reset-password/route.ts`)

```
// Notificação enviada após admin alterar senha
await sendNotificationEmail(email, name, companyId, companyName, title, message);
```

3. Forgot Password (`/api/auth/forgot-password/route.ts`)

```
// Email com link de reset enviado
await sendPasswordResetEmail(email, name, companyId, companyName, resetLink);
```

4. Adicionar à Equipe (`/api/teams/[id]/members/route.ts`)

```
// Notificação quando usuário é adicionado à equipe
await sendNotificationEmail(email, name, companyId, companyName, title, message);
```

🎨 Templates de Email

Estrutura HTML

- **Header:** Logo e nome da empresa com gradiente de cores
- **Conteúdo:** Texto do email com variáveis substituídas
- **Footer:** Mensagem personalizada e copyright

Variáveis Dinâmicas Suportadas

- `{{companyName}}` - Nome da empresa
- `{{userName}}` - Nome do usuário
- `{{userEmail}}` - Email do usuário
- `{{resetLink}}` - Link para reset de senha
- `{{teamName}}` - Nome da equipe

- `{{inviterName}}` - Nome de quem convidou
- `{{inviteLink}}` - Link de convite
- `{{notificationTitle}}` - Título da notificação
- `{{notificationMessage}}` - Mensagem da notificação

🔐 Segurança

1. **Tokens de Reset**
   - Gerados com `crypto.randomBytes(32)`
   - Expiração de 1 hora
   - Uso único (flag `used`)
   - Validação rigorosa antes de aceitar

2. **Email Enumeration Protection**
   - Sempre retorna sucesso no forgot-password
   - Não revela se o email existe ou não

3. **Fallback Gracioso**
   - Erros de envio não bloqueiam operações principais
   - Logs detalhados para debugging

📊 Logs e Monitoramento

Console Logs

```
✅ Email enviado com sucesso: { id: '...', from: '...', to: '...' }
❌ Erro ao enviar email: { error details }
⏭️  Email não enviado - template desabilitado: WELCOME
⚠️  Erro ao enviar email de notificação: { error details }
```

🚀 Como Usar

Para Administradores

1. **Personalizar Templates**
   - Acesse `/settings/email-templates`
   - Edite assunto e corpo dos emails
   - Use variáveis dinâmicas (ex: `{{userName}}`)
   - Preview em tempo real

2. **Configurar Branding**
   - Acesse `/settings/branding`
   - Upload de logo
   - Escolha de cores corporativas
   - Mensagens personalizadas

Para Desenvolvedores

Enviar email customizado:

```
import { sendEmail } from '@/lib/email';

await sendEmail({
  to: 'usuario@exemplo.com',
  companyId: 'company-id',
  templateType: 'NOTIFICATION',
  variables: {
    userName: 'João Silva',
    companyName: 'Minha Empresa',
    notificationTitle: 'Novo Documento',
    notificationMessage: 'Um novo documento está disponível'
  }
});
```

📝 Endpoints de API

POST `/api/auth/forgot-password`

Body:

```
{
  "email": "usuario@exemplo.com"
}
```

POST `/api/auth/reset-password-confirm`

Body:

```
{
  "token": "abc123...",
  "newPassword": "novaSenha123"
}
```

🔄 Fluxo de Reset de Senha

1. **Usuário solicita reset**
   → POST `/api/auth/forgot-password`
2. **Sistema gera token**
   → Salva no banco com expiração de 1h
3. **Email enviado com link**
   → `https://app.com/reset-password?token=abc123`
4. **Usuário clica no link**
   → Formulário para nova senha

5. **Nova senha confirmada**
   → POST `/api/auth/reset-password-confirm`
   → Token marcado como usado

⚙️ Configuração do Resend

API Key

Armazenada em `.env`:

```
RESEND_API_KEY=re_CCRLEEP3_...
```

Domínio de Envio

- **Padrão:** `noreply@organizen.app`
- **Customizável** na função `sendEmail()`

Limites do Plano Gratuito

- **3.000 emails/mês**
- **Suficiente para testes e uso inicial**

🎯 Próximos Passos Sugeridos

1. **Configurar domínio customizado no Resend**
   - Adicionar DNS records
   - Verificar domínio
   - Usar `noreply@suaempresa.com`

2. **Adicionar mais tipos de emails**
   - Confirmação de tarefas
   - Lembretes de eventos
   - Relatórios periódicos

3. **Dashboard de emails enviados**
   - Histórico de envios
   - Taxa de entrega
   - Estatísticas

4. **Agendamento de emails**
   - Envios programados
   - Campanhas

✅ Testes Realizados

- ✅ Build sem erros TypeScript
- ✅ Integração com Prisma

- ✅ Validação de tipos
- ✅ Fallback para templates padrão
- ✅ Tratamento de erros
- ✅ Logs adequados

📞 Suporte

Para qualquer dúvida sobre o sistema de emails:
- Documentação Resend: https://resend.com/docs
- Logs do servidor: `console.log`
- Dashboard Resend: https://resend.com/emails

✨ **Sistema de emails totalmente funcional e pronto para uso!**