

Fase 3B: Templates de Email Personalizados

Status: Implementado com Sucesso

Resumo da Implementação

A **Fase 3B** adiciona um sistema completo de templates de email personalizados ao OrganiZen, permitindo que cada empresa configure os emails enviados pelo sistema com seu próprio branding e mensagens.

Funcionalidades Implementadas

1. Editor de Templates

- Interface visual para edição de templates de email
- Preview em tempo real dos emails
- Suporte a variáveis dinâmicas

2. Tipos de Templates Disponíveis

Email de Boas-vindas

- Enviado quando um novo usuário se cadastra
- **Variáveis disponíveis:**
 - {{userName}} - Nome do usuário
 - {{userEmail}} - Email do usuário
 - {{companyName}} - Nome da empresa
 - {{loginUrl}} - Link para fazer login

Redefinição de Senha

- Enviado quando o usuário solicita redefinir a senha
- **Variáveis disponíveis:**
 - {{userName}} - Nome do usuário
 - {{resetLink}} - Link para redefinir senha
 - {{companyName}} - Nome da empresa
 - {{expiresIn}} - Tempo de expiração do link

Convite para Equipe

- Enviado ao convidar alguém para entrar na empresa
- **Variáveis disponíveis:**
 - {{inviterName}} - Nome de quem convidou
 - {{companyName}} - Nome da empresa
 - {{inviteLink}} - Link do convite
 - {{teamName}} - Nome da equipe

Notificação Geral

- Modelo base para notificações gerais do sistema
- **Variáveis disponíveis:**
 - {{userName}} - Nome do usuário
 - {{companyName}} - Nome da empresa
 - {{notificationTitle}} - Título da notificação

– {{notificationBody}} - Corpo da notificação

Estrutura do Banco de Dados

Campos Adicionados ao CompanyBranding

```
// Templates de Email Personalizados (Fase 3B)
emailSenderName      String? // Nome do remetente nos emails
emailFooter           String? @db.Text // Rodapé padrão para todos os emails

// Template: Email de Boas-vindas
emailWelcomeSubject   String? @default("Bem-vindo(a) ao {{companyName}}!")
emailWelcomeBody      String? @db.Text
emailWelcomeEnabled   Boolean @default(true)

// Template: Redefinição de Senha
emailResetSubject     String? @default("Redefinir sua senha")
emailResetBody        String? @db.Text
emailResetEnabled     Boolean @default(true)

// Template: Convite para Equipe
emailInviteSubject    String? @default("Você foi convidado(a) para {{companyName}}")
emailInviteBody       String? @db.Text
emailInviteEnabled    Boolean @default(true)

// Template: Notificação Geral
emailNotifySubject    String? @default("Notificação de {{companyName}}")
emailNotifyBody       String? @db.Text
emailNotifyEnabled    Boolean @default(true)
```

Arquivos Criados/Modificados

Novos Arquivos:

1. **components/branding/EmailTemplateEditor.tsx**
 - Componente principal do editor de templates
 - Interface com abas para cada tipo de template
 - Configurações gerais (nome do remetente, rodapé)
 - Preview em tempo real
2. **components/branding/EmailPreview.tsx**
 - Visualização do email renderizado
 - Substitui variáveis por dados de exemplo
 - Mostra logo e cores da empresa
3. **app/api/branding/email-templates/route.ts**
 - API GET: Busca templates configurados
 - API PUT: Salva/atualiza templates
 - Restrição: Apenas administradores
4. **app/settings/email-templates/page.tsx**
 - Página dedicada aos templates de email
 - Carrega dados do branding para preview
 - Controle de acesso (somente admins)
5. **lib/email-service.ts**

- Serviço para processar e enviar emails
- Substitui variáveis nos templates
- Funções auxiliares:
 - `sendWelcomeEmail()`
 - `sendPasswordResetEmail()`
 - `sendTeamInviteEmail()`
 - `sendNotificationEmail()`

Arquivos Modificados:

1. `prisma/schema.prisma`
 - Adicionados campos para templates de email
 - Suporte a habilitar/desabilitar cada template
 2. `app/settings/branding/page.tsx`
 - Adicionado card com link para templates de email
 - Destaque visual na seção de branding
-

Interface do Usuário

Página de Templates de Email

- Acessível em: `/settings/email-templates`
- Somente para administradores

Configurações Gerais:

- Nome do remetente
- Rodapé padrão (aplicado em todos os emails)

Para cada Template:

- Assunto personalizável
- Corpo do email personalizável
- Switch para ativar/desabilitar
- Lista de variáveis disponíveis
- Preview em tempo real

Preview Interativo:

- Mostra como o email ficará
 - Usa cores e logo da empresa
 - Exibe dados de exemplo
 - Atualização em tempo real
-

Como Usar

1. Acessar Configurações

Dashboard → Configurações → Branding Corporativo

2. Clicar em “Templates de Email Personalizados”

(Card destacado na página de branding)

3. Configurar Templates

1. Definir nome do remetente
 2. Adicionar rodapé padrão (opcional)
 3. Selecionar tipo de template (abas)
 4. Editar assunto e corpo
 5. Usar variáveis disponíveis
 6. Ativar/desativar template
 7. Visualizar preview
 8. Salvar alterações
-

Integração com Serviços de Email

Estado Atual (Demonstração)

O sistema está configurado para **log de emails no console**. Quando um email é enviado, as informações são exibidas no console do servidor.

Integração Futura

Para usar em produção, integre com um provedor de email:

Opções Recomendadas:

1. **SendGrid** (Fácil integração, ótimo para transacional)
2. **AWS SES** (Econômico, escalável)
3. **Mailgun** (Bom suporte técnico)
4. **Postmark** (Excelente deliverability)

Exemplo de Integração (SendGrid):

```
// Em lib/email-service.ts
import sgMail from '@sendgrid/mail';

sgMail.setApiKey(process.env.SENDGRID_API_KEY);

export async function sendTemplatedEmail(...) {
  // ... processar template ...

  await sgMail.send({
    to: options.to,
    from: process.env.EMAIL_FROM,
    subject,
    text: body,
    html: convertToHtml(body),
  });
}
```

Casos de Uso

1. Boas-vindas Personalizadas

Cenário: Novo funcionário se cadastra

Resultado: Recebe email com boas-vindas da empresa

2. Recuperação de Senha

Cenário: Usuário esqueceu a senha

Resultado: Recebe email com link personalizado

3. Convites para Equipes

Cenário: Admin convida novo membro

Resultado: Convidado recebe email com branding da empresa

4. Notificações Importantes

Cenário: Sistema envia alertas

Resultado: Emails com identidade visual da empresa

Destaques Técnicos

1. Variáveis Dinâmicas

- Sistema de substituição {{variavel}}
- Processamento seguro e eficiente
- Suporte a qualquer quantidade de variáveis

2. Preview em Tempo Real

- Renderização instantânea
- Dados de exemplo para cada tipo
- Integração com cores e logo

3. Controle Granular

- Cada template pode ser ativado/desativado
- Valores padrão inteligentes
- Rodapé compartilhado

4. Segurança

- Apenas administradores podem editar
 - Validação de dados no backend
 - Proteção contra injeção
-

Próximos Passos Sugeridos

Fase 4: Subdomínios Personalizados

- Permitir `suaempresa.organizen.com`
- Configuração de DNS
- SSL automático

Melhorias nos Templates:

- Editor visual WYSIWYG
- Mais variáveis disponíveis
- Anexos em emails
- Tradução automática

Analytics de Email:

- Taxa de abertura
 - Cliques em links
 - Emails entregues/rejeitados
-

Notas Importantes

Avisos de Build

Os avisos sobre “Dynamic server usage” são **normais** e **não afetam** o funcionamento: - APIs de email são dinâmicas por natureza - Next.js tenta renderizar estaticamente - Comportamento esperado para rotas de API

Testando Localmente

Para testar o envio de emails localmente: 1. Configure um provedor de email 2. Adicione credenciais no `.env` 3. Atualize `lib/email-service.ts` 4. Execute a aplicação

Acesso à Funcionalidade

URL: `/settings/email-templates`

Requisitos: - Estar autenticado - Ter role de ADMIN - Plano Pro+ (quando sistema de planos estiver ativo)

Conclusão

A **Fase 3B** foi implementada com sucesso! O OrganiZen agora oferece:

- Editor visual de templates de email
- 4 tipos de templates prontos para uso
- Preview em tempo real
- Variáveis dinâmicas
- Controle granular (ativar/desativar)
- Integração com branding corporativo
- Sistema pronto para produção (após configurar provedor)

O sistema está **100% funcional** e pronto para ser usado assim que um provedor de email for configurado!

Data de Implementação: 21 de Outubro de 2025

Status: Concluído e Testado

Build: Sem Erros