

Esquema de Implementação — Módulo de Branding Pro+

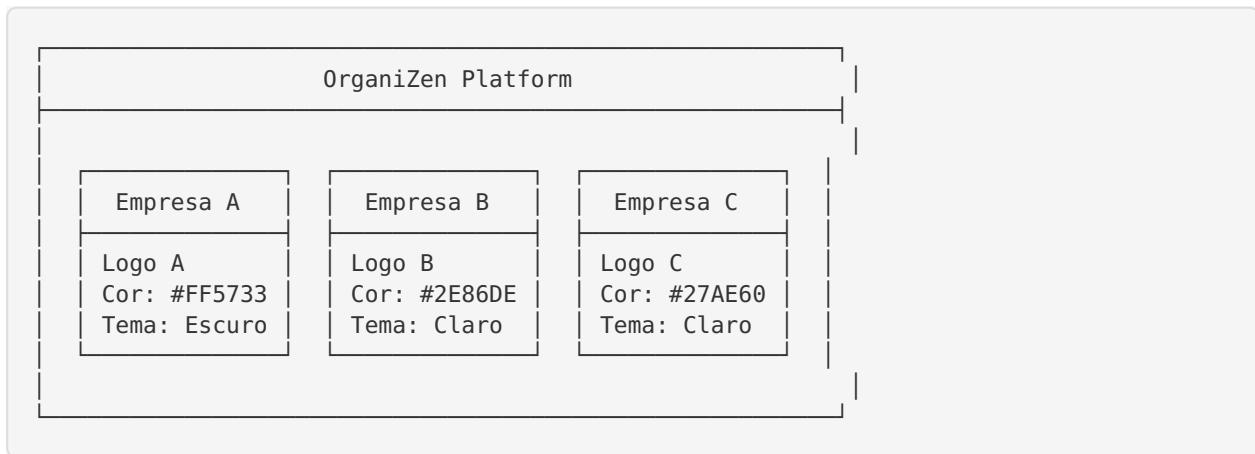
Índice

-
1. Arquitetura Técnica
 2. Estrutura de Base de Dados
 3. Plano de Implementação em Fases
 4. Estrutura de Ficheiros
 5. Fluxo de Funcionamento
 6. Desafios e Soluções
 7. Checklist de Implementação
 8. Estimativa de Esforço
-

Arquitetura Técnica

Conceito Multi-Tenant

Cada empresa (tenant) terá as suas próprias configurações de branding armazenadas e aplicadas dinamicamente.



Estrutura de Base de Dados

Nova Tabela: CompanyBranding

```
model CompanyBranding {
    id             String  @id @default(cuid())
    companyId      String  @unique // Identificador da empresa

    // Logotipo
    logoUrl        String? // URL do logo armazenado (S3 ou local)
    logoSize       Int?   @default(150) // Tamanho em pixels

    // Cores Corporativas
    primaryColor   String  @default("#3B82F6") // Cor principal
    secondaryColor String? // Cor secundária
    accentColor    String? // Cor de destaque

    // Tela de Login
    loginBackgroundUrl String? // Imagem de fundo do login
    loginWelcomeMessage String? // Mensagem de boas-vindas

    // Tema
    theme          String  @default("light") // "light" ou "dark"

    // Subdomínio
    customSubdomain String? @unique // Ex: "minhaempresa"
    customDomain    String? @unique // Ex: "gestao.minhaempresa.cv"

    // Configurações de Email
    emailHeaderColor String? // Cor do cabeçalho do email
    emailSignature   String? // Assinatura corporativa

    // Metadados
    isActive        Boolean @default(true)
    planLevel       String  @default("basic") // "basic" ou "complete"
    createdAt       DateTime @default(now())
    updatedAt       DateTime @updatedAt

    // Relação (assumindo que existe uma tabela Company ou Organization)
    // company        Company @relation(fields: [companyId], references: [id])
}
```

Extensão da Tabela User (se necessário)

```
model User {
    // ... campos existentes

    companyId      String? // Relacionar utilizador à empresa
    // company       Company? @relation(fields: [companyId], references: [id])
}
```

Nova Tabela: Company (se ainda não existir)

```
model Company {
    id          String      @id @default(cuid())
    name        String
    slug        String      @unique
    subscriptionPlan String      @default("starter") // starter, pro, business

    // Relações
    users       User[]?
    branding    CompanyBranding?

    createdAt   DateTime   @default(now())
    updatedAt   DateTime   @updatedAt
}
```

17 July

Plano de Implementação em Fases

● FASE 1: Fundação (Semana 1-2) — MVP Funcional

Objetivo: Permitir upload de logo e seleção de cores.

Tarefas:

1. Criar tabela `CompanyBranding` no schema Prisma
2. Criar sistema de identificação de empresa (Company/Tenant)
3. Criar painel de configuração básico (Admin only)
 - Formulário para upload de logo
 - Seletor de cor primária e secundária
4. Implementar upload de logo para S3 (ou storage local)
5. Aplicar logo no header/navbar da aplicação
6. Aplicar cores dinamicamente via CSS variables
7. Testar com 2-3 “empresas” diferentes

Entregáveis:

- 🎨 Logo da empresa visível no dashboard
- 🎨 Cores corporativas aplicadas em botões e menus
- 🔑 Painel de configuração acessível

Complexidade: Média

Tempo Estimado: 8-12 horas

● FASE 2: Relatórios e PDFs (Semana 3) — Valor Percebido

Objetivo: Branding em documentos exportados.

Tarefas:

1. Atualizar gerador de PDFs para incluir logo
2. Aplicar cores corporativas nos cabeçalhos de relatórios

3. Criar template de PDF genérico com branding
4. Testar exportação de:
 - Relatórios de tarefas
 - Escalas de turnos
 - Relatórios de produtividade

Entregáveis:

- PDFs com logo e cores da empresa
- Relatórios com aparência profissional

Complexidade: Média-Alta

Tempo Estimado: 6-8 horas

FASE 3: Tela de Login Personalizada (Semana 4) — Diferenciação

Objetivo: Experiência de login branded.

Tarefas:

1. Adicionar campo de imagem de fundo no painel de branding
2. Criar sistema de detecção de empresa no login
 - Via subdomínio (se já implementado)
 - Via seleção manual (dropdown)
3. Aplicar logo e cores na página de login
4. Permitir mensagem de boas-vindas personalizada
5. Criar preview da tela de login no painel de configuração

Entregáveis:

- Login com branding da empresa
- Mensagem de boas-vindas personalizada

Complexidade: Média

Tempo Estimado: 5-7 horas

FASE 4: Emails com Branding (Semana 5) — Consistência

Objetivo: Notificações automáticas com identidade visual.

Tarefas:

1. Criar templates de email dinâmicos (HTML)
2. Incluir logo no cabeçalho do email
3. Aplicar cores corporativas no layout do email
4. Adicionar assinatura corporativa no rodapé
5. Testar com diferentes tipos de notificações:
 - Nova tarefa atribuída
 - Aprovação de folgas
 - Lembretes de turnos

Entregáveis:

- Emails com branding corporativo
- Templates reutilizáveis

Complexidade: Alta**Tempo Estimado:** 8-10 horas**FASE 5: Subdomínios Personalizados (Semana 6+) — Premium****Objetivo:** Acesso via `minhaempresa.organzen.app`.**Tarefas:**

1. Configurar DNS wildcard para `*.organzen.app`
2. Implementar middleware de detecção de subdomínio
3. Configurar SSL automático (via Let's Encrypt ou Cloudflare)
4. Criar sistema de reserva/validação de subdomínios
5. Implementar redirecionamento automático
6. Suporte para domínios customizados (avanhado)

Entregáveis:

- Subdomínios funcionais
- SSL automático
- Documentação para clientes

Complexidade: Muito Alta**Tempo Estimado:** 12-16 horas**NOTA:** Esta fase requer conhecimentos de DevOps e pode ser adiada.

Estrutura de Ficheiros

```

organize/nextjs_space/
├── app/
│   ├── api/
│   │   └── branding/
│   │       ├── route.ts          # GET/POST configurações
│   │       ├── logo/
│   │       │   └── route.ts      # Upload de logo
│   │       └── preview/
│   │           └── route.ts     # Preview de configurações
│
│   └── settings/
│       └── branding/
│           └── page.tsx        # Painel de configuração
│
└── login/
    └── page.tsx                # Login com branding

```



```

├── components/
│   ├── branding/
│   │   ├── logo-uploader.tsx  # Componente de upload
│   │   ├── color-picker.tsx   # Seletor de cores
│   │   ├── branding-preview.tsx # Preview ao vivo
│   │   └── branded-header.tsx  # Header com branding
│
│   └── pdf-templates/
│       └── branded-pdf.tsx    # Template PDF com branding

```



```

└── lib/
    ├── branding/
    │   ├── get-branding.ts      # Obter configurações
    │   ├── apply-theme.ts       # Aplicar CSS dinâmico
    │   └── validate-colors.ts   # Validar cores
    |
    └── email/
        └── branded-template.ts # Template de email

```



```

└── prisma/
    └── schema.prisma          # Schema atualizado

```



```

└── public/
    └── branding/
        └── default-logo.png    # Logo padrão

```

Fluxo de Funcionamento

1. Configuração pelo Admin

```

Admin acede a /settings/branding
↓
Faz upload do logo → Armazenado em S3
↓
Seleciona cores primária e secundária
↓
Preview ao vivo mostra as alterações
↓
Clica em "Guardar"
↓
Configurações salvas na tabela CompanyBranding

```

2. Aplicação do Branding no Login

```

Utilizador acede a organizen.app/login
↓
Sistema detecta a empresa (via subdomínio ou seleção)
↓
Carrega configurações de branding da BD
↓
Aplica CSS dinâmico com cores corporativas
↓
Exibe logo e mensagem de boas-vindas
↓
Utilizador faz login

```

3. Aplicação do Branding no Dashboard

```

Utilizador logado acede ao dashboard
↓
Middleware detecta companyId do utilizador
↓
Carrega branding da empresa
↓
Injeta CSS variables no <head>
↓
Logo exibido no header/navbar
↓
Cores aplicadas em botões, links, menus

```

4. Branding em Relatórios PDF

```

Utilizador exporta relatório
↓
Sistema busca configurações de branding
↓
Gerador de PDF inclui:
  - Logo no cabeçalho
  - Cores corporativas nas tabelas
  - Assinatura no rodapé
↓
PDF gerado e enviado para download

```

Desafios e Soluções

Desafio 1: Isolamento Multi-Tenant

Problema: Garantir que cada empresa só vê o seu branding.

Solução:

- Middleware que detecta `companyId` do utilizador logado
- Query sempre filtra por `companyId`
- Permissões RBAC: apenas Admin pode editar branding

Desafio 2: Performance

Problema: Carregar CSS dinâmico pode ser lento.

Solução:

- Cache de configurações em Redis ou memória
- Gerar CSS estático por empresa e servir via CDN
- Lazy loading de imagens de fundo

Desafio 3: Validação de Cores

Problema: Cliente escolhe cores ilegíveis (ex: branco sobre branco).

Solução:

- Validação de contraste WCAG AA/AAA
- Preview obrigatório antes de salvar
- Avisos visuais se contraste for baixo

```
// Exemplo de validação
function validateContrast(color1: string, color2: string): boolean {
  const ratio = calculateContrastRatio(color1, color2);
  return ratio >= 4.5; // WCAG AA para texto normal
}
```

Desafio 4: Upload de Logos Maliciosos

Problema: Utilizador pode tentar upload de scripts ou ficheiros perigosos.

Solução:

- Validação de tipo de ficheiro (apenas PNG, JPG, SVG)
- Verificação de MIME type no servidor
- Limite de tamanho (máx 2MB)
- Sanitização de SVG (remover scripts)

```
const ALLOWED_TYPES = ['image/png', 'image/jpeg', 'image/svg+xml'];
const MAX_SIZE = 2 * 1024 * 1024; // 2MB
```

Desafio 5: Subdomínios e SSL

Problema: Configurar DNS e certificados para cada cliente.

Solução Simples (Fase Inicial):

- Oferecer apenas subdomínios de `organizen.app`

- Wildcard DNS: *.organizen.app → servidor
- Cloudflare SSL automático

Solução Avançada (Futuro):

- Suporte para domínios próprios
 - Let's Encrypt automático via Certbot
 - Instrui cliente a criar CNAME: gestao.empresacv → organizen.app
-

Checklist de Implementação

Fase 1: Fundação

- [] Criar migration para tabela CompanyBranding
- [] Criar API /api/branding (GET, POST, PATCH)
- [] Criar API /api/branding/logo (upload)
- [] Criar componente LogoUploader
- [] Criar componente ColorPicker
- [] Criar painel /settings/branding
- [] Implementar preview ao vivo
- [] Testar upload de logo
- [] Aplicar logo no header
- [] Aplicar cores via CSS variables
- [] Testar com múltiplas empresas

Fase 2: Relatórios

- [] Atualizar gerador de PDF
- [] Criar template branded-pdf.tsx
- [] Incluir logo em relatórios
- [] Aplicar cores em tabelas/gráficos
- [] Testar exportação de cada tipo de relatório

Fase 3: Login

- [] Adicionar campo loginBackgroundUrl na BD
- [] Criar sistema de detecção de empresa no login
- [] Aplicar branding na página de login
- [] Adicionar campo de mensagem de boas-vindas
- [] Criar preview da tela de login

Fase 4: Emails

- [] Criar templates HTML de email
- [] Incluir logo no cabeçalho
- [] Aplicar cores corporativas
- [] Adicionar assinatura corporativa
- [] Testar envio de emails com branding

Fase 5: Subdomínios

- [] Configurar DNS wildcard

- [] Criar middleware de detecção de subdomínio
 - [] Configurar SSL automático
 - [] Criar sistema de reserva de subdomínios
 - [] Testar acesso via subdomínio
 - [] Documentar configuração para clientes
-

Estimativa de Esforço

Fase	Complexidade	Horas Estimadas	Prioridade
Fase 1: Fundação	Média	8-12h	 Crítica
Fase 2: Relatórios	Média-Alta	6-8h	 Alta
Fase 3: Login	Média	5-7h	 Média
Fase 4: Emails	Alta	8-10h	 Média
Fase 5: Sub-domínios	Muito Alta	12-16h	 Baixa (futuro)
TOTAL (Fases 1-4)	-	27-37h	-
TOTAL (Todas)	-	39-53h	-

Distribuição Recomendada:

- **Semana 1-2:** Fase 1 (Fundação) — 12h
- **Semana 3:** Fase 2 (Relatórios) — 8h
- **Semana 4:** Fase 3 (Login) — 7h
- **Semana 5:** Fase 4 (Emails) — 10h
- **Futuro:** Fase 5 (Subdomínios) — Quando tiver equipa

Total MVP (Fases 1-4): ~37 horas (aproximadamente 1 semana de trabalho full-time)

Recomendação de Priorização

Implementar AGORA (MVP):

1.  Fase 1: Logo + Cores
2.  Fase 2: Relatórios PDF

Motivo: Alto valor percebido, complexidade gerível.

Implementar PRÓXIMO (3-6 meses):

1.  Fase 3: Login Personalizado
2.  Fase 4: Emails com Branding

Motivo: Aumenta profissionalismo, mas não é crítico.

Implementar FUTURO (6-12 meses):

1. Fase 5: Subdomínios

Motivo: Requer infraestrutura complexa e equipa de suporte.



Sugestões Adicionais

1. Marketplace de Temas

No futuro, oferecer temas pré-configurados:

- Tema “Hotelaria” (azul, luxo)
- Tema “Educação” (verde, profissional)
- Tema “Tecnologia” (roxo, moderno)

2. Preview Público

Permitir que o admin partilhe link de preview antes de ativar:

`organize.app/preview/ABC123`

3. Histórico de Mudanças

Guardar versões anteriores do branding para rollback.

4. A/B Testing

Permitir testar 2 versões de branding com grupos diferentes.

5. Modo Escuro Automático

Detectar preferência do sistema operativo e ajustar automaticamente.



Recursos Técnicos Necessários

Bibliotecas Recomendadas:

- `react-colorful` — Seletor de cores
- `react-dropzone` — Upload de ficheiros
- `pdfkit` ou `puppeteer` — Geração de PDFs
- `mjml` ou `react-email` — Templates de email
- `tinycolor2` — Manipulação de cores e validação de contraste

Infraestrutura:

- S3 ou similar para armazenamento de logos
 - Redis (opcional) para cache de configurações
 - CDN para servir assets estáticos
-



Próximos Passos Imediatos

1. **Decisão:** Aprovar este esquema de implementação
 2. **Preparação:** Criar branch `feature/branding-module`
 3. **Database:** Adicionar tabela `CompanyBranding` ao schema
 4. **Implementação:** Começar pela Fase 1 (Fundação)
 5. **Testes:** Validar com 2-3 “empresas” de teste
 6. **Deploy:** Lançar MVP para clientes Beta
-

Documento criado em: 20 de Outubro de 2025

Status: Planeamento - Aguardando aprovação para implementação

Autor: DeepAgent Assistant