

Sistema de Hierarquia de Temas - Validação Completa

Status da Implementação

Componentes Implementados

1. Schema do Prisma (Atualizado)

Status:  COMPLETO

Novos Campos no Model User :

```
themePreset      String?    // Tema pessoal (padrao, simples, moderado, moderno)
officeId         String?    // ID do escritório (futura implementação)
allowThemeOverride Boolean   @default(false) // Dono de escritório pode liberar temas
```

Novo Model GlobalSettings :

```
model GlobalSettings {
  id          Int      @id @default(1)
  superadminThemePreset String  @default("padrao")
  allowOfficeOverride Boolean @default(false) // Escritórios (futuro)
  allowUserOverride Boolean @default(true)    // Usuários podem personalizar
  createdAt    DateTime @default(now())
  updatedAt    DateTime @updatedAt
}
```

Migrações:

-  Schema atualizado com sucesso (prisma db push)
-  Banco de dados sincronizado

2. APIs Criadas

/api/admin/theme-settings (SuperAdmin)

Métodos: GET , PUT

Funcionalidades:

-  **GET:** Retorna configurações globais de tema
- Cria configurações padrão se não existir
- Retorna: superadminThemePreset , allowOfficeOverride , allowUserOverride
 -  **PUT:** Atualiza configurações globais
 - Valida tema (padrao, simples, moderado, moderno)
 - Upsert automático (cria se não existir)
 - Acesso restrito a SuperAdmin

Exemplo de Resposta (GET):

```
{
  "settings": {
    "id": 1,
    "superadminThemePreset": "padrao",
    "allowOfficeOverride": false,
    "allowUserOverride": true,
    "createdAt": "2025-11-25T00:00:00.000Z",
    "updatedAt": "2025-11-25T00:00:00.000Z"
  }
}
```

/api/user/theme (Usuários)

Métodos: GET , PUT

Funcionalidades:

- **GET**: Retorna hierarquia completa de temas
- Busca configurações globais
- Busca tema do usuário
- Placeholder para tema de escritório (futuro)
- Calcula tema efetivo com prioridade
 - **PUT**: Atualiza tema do usuário
 - Verifica permissão (allowUserOverride)
 - Valida tema
 - Permite null para resetar tema

Exemplo de Resposta (GET):

```
{
  "effectiveTheme": "moderno",
  "userTheme": "moderno",
  "officeTheme": null,
  "superadminTheme": "padrao",
  "canChangeTheme": true,
  "isOfficeOwner": false,
  "allowOfficeOverride": false
}
```

Lógica de Prioridade:

```
effectiveTheme = userTheme ?? officeTheme ?? superadminTheme ?? "padrao"
```

3. ThemeSelector (Atualizado)

Localização: components/theme-selector.tsx

Novas Funcionalidades:

- **Carregamento de Hierarquia**:
- Busca dados de /api/user/theme ao inicializar

- Fallback para localStorage se API falhar
- Aplica tema efetivo automaticamente

- **✓ Validação de Permissões:**

- Verifica `canChangeTheme` antes de permitir alteração
- Exibe mensagem de erro se bloqueado
- Mostra ícone de cadeado quando bloqueado

- **✓ Indicadores Visuais:**

- Badge “Tema definido pelo administrador” quando bloqueado
- Badge “Usando tema padrão do sistema” quando herdado
- Informação sobre hierarquia no preview modal

- **✓ Botão de Reset:**

- Aparece apenas se usuário tem tema personalizado
- Chama API para definir `themePreset = null`
- Recarrega hierarquia após reset

- **✓ Salvamento no Backend:**

- Salva no backend via API (não apenas localStorage)
- Mantém localStorage como backup
- Toast de feedback ao salvar

4. Página de Configuração de Tema (SuperAdmin)

Localização: `app/(protected)/admin/theme-config/page.tsx`

Funcionalidades:

- **✓ Seleção de Tema Padrão:**

- Dropdown com 4 opções (Padrão, Simples, Moderado, Moderno)
- Preview visual do tema selecionado
- Ícones únicos para cada tema

- **✓ Configuração de Permissões:**

- **Switch “Permitir Usuários Escolherem Tema”:**

- Habilita/desabilita personalização individual
- Explicação clara do comportamento

- **Switch “Permitir Donos de Escritório”:** (Desabilitado - Futuro)

- Preparado para futura implementação
- Badge “Em Breve” visível

- **✓ Explicação de Hierarquia:**

- Cards numerados (1º, 2º, 3º) mostrando ordem de prioridade
- Tema do Usuário → Tema do Escritório (futuro) → Tema Padrão
- Cores diferentes para cada nível

- **Integração no Menu:**

- Item “Temas” adicionado no menu do SuperAdmin
 - Ícone: Palette (
 - Posição: Entre “Anúncios” e “Configurações”
-

Fluxo de Funcionamento

Cenário 1: SuperAdmin Define Tema “Moderno”

1. SuperAdmin acessa `/admin/theme-config`
2. Seleciona “Moderno” no dropdown
3. Habilita “Permitir Usuários Escolherem Tema”
4. Clica em “Salvar Configurações”

5. Resultado:

- Todos os usuários que não têm tema personalizado veem o tema “Moderno”
 - Usuários podem escolher outro tema se quiserem
-

Cenário 2: Usuário Personaliza Tema

1. Usuário acessa a sidebar
2. Clica no ThemeSelector (seção “Aparência”)
3. Seleciona “Simples” no dropdown

4. Resultado:

- Tema “Simples” é aplicado instantaneamente
 - Tema é salvo no banco de dados
 - Badge “Usando tema padrão do sistema” desaparece
 - Botão “Resetar” aparece
-

Cenário 3: SuperAdmin Bloqueia Personalização

1. SuperAdmin acessa `/admin/theme-config`
2. Define tema “Padrão”
3. **Desabilita** “Permitir Usuários Escolherem Tema”
4. Clica em “Salvar Configurações”

5. Resultado:

- Todos os usuários veem o tema “Padrão”
 - ThemeSelector fica desabilitado para usuários
 - Badge “Tema definido pelo administrador” aparece
 - Tentativa de alterar tema mostra erro
-

Cenário 4: Usuário Reseta Tema

1. Usuário tem tema “Moderado” personalizado

2. Clica no botão “Resetar” no ThemeSelector

3. Resultado:

- `themePreset` do usuário vira `null`
 - Sistema aplica tema padrão do SuperAdmin
 - Badge “Usando tema padrão do sistema” aparece
 - Botão “Resetar” desaparece
-

Como Testar

Teste 1: Configuração Global (SuperAdmin)

1. Faça login como SuperAdmin (`admin@clivus.com` / `admin123`)

2. Acesse `/admin/theme-config` (menu “Temas”)

3. Selecione “Moderno” no dropdown

4. Clique em “Salvar Configurações”

5.  Resultado Esperado:

- Toast “✓ Configurações salvas com sucesso!”
 - Preview do tema “Moderno” aparece
-

Teste 2: Usuário Personaliza Tema

1. Faça login como cliente (`cliente@teste.com` / `teste123`)

2. Na sidebar, localize “Aparência”

3. Abra o ThemeSelector

4. Selecione “Simples”

5.  Resultado Esperado:

- Tema muda imediatamente
 - Toast “Tema salvo com sucesso!”
 - Botão “Resetar” aparece
 - Badge “Usando tema padrão do sistema” desaparece
-

Teste 3: Bloqueio de Personalização

1. Como SuperAdmin, acesse `/admin/theme-config`

2. Desabilite “Permitir Usuários Escolherem Tema”

3. Salve as configurações

4. Faça logout e login como cliente

5. Na sidebar, tente mudar o tema no ThemeSelector

6.  Resultado Esperado:

- Dropdown do ThemeSelector está desabilitado (grayed out)
 - Badge “Tema definido pelo administrador” aparece
 - Tentativa de mudar tema mostra toast de erro
-

Teste 4: Reset de Tema

1. Como cliente, personalize o tema para “Moderado”
 2. Clique no botão “Resetar”
 3. **Resultado Esperado:**
 - Toast “Tema resetado para padrão!”
 - Tema volta para o padrão do sistema
 - Badge “Usando tema padrão do sistema” aparece
 - Botão “Resetar” desaparece
-

Teste 5: Preview Modal

1. Como cliente, clique no ícone de paleta (👁️) ao lado do dropdown
 2. **Resultado Esperado:**
 - Modal abre com 4 cards de preview
 - Card do tema ativo tem borda azul e checkmark
 - Seção “Hierarquia de Temas” mostra:
 - Seu tema (se personalizado)
 - Tema do escritório (null)
 - Tema padrão do sistema
-

Teste 6: Persistência ao Recarregar

1. Como cliente, selecione tema “Moderno”
 2. Atualize a página (F5)
 3. **Resultado Esperado:**
 - Tema “Moderno” permanece ativo
 - ThemeSelector mostra “Moderno” selecionado
-



Checklist de Validação Final

Database Schema

- **✓** Campo `themePreset` adicionado ao `User`
- **✓** Campos `officeId`, `allowThemeOverride` adicionados (preparação futura)
- **✓** Model `GlobalSettings` criado
- **✓** Migração aplicada com sucesso (`prisma db push`)

APIs

- **✓** `/api/admin/theme-settings` (GET/PUT) funcionando
- **✓** `/api/user/theme` (GET/PUT) funcionando
- **✓** Validação de permissões implementada
- **✓** Hierarquia de temas calculada corretamente

Frontend

- **✓** ThemeSelector atualizado com hierarquia

- Indicadores visuais de permissões
- Botão de reset funcionando
- Preview modal atualizado
- Toast de feedback implementado

SuperAdmin

- Página `/admin/theme-config` criada
- Item “Temas” adicionado ao menu
- Configuração de tema padrão funcionando
- Configuração de permissões funcionando
- Explicação de hierarquia clara

Build e Deploy

- TypeScript compila sem erros (`exit_code=0`)
- Build de produção bem-sucedido
- 33 páginas geradas com sucesso (incluindo `/admin/theme-config`)
- APIs `/api/admin/theme-settings` e `/api/user/theme` listadas
- Checkpoint salvo com sucesso



Resultado da Validação

Status Geral: SISTEMA TOTALMENTE FUNCIONAL

Resumo:

- Hierarquia de temas implementada (User → Office → SuperAdmin → Padrão)
- SuperAdmin pode definir tema padrão do sistema
- SuperAdmin pode habilitar/desabilitar personalização de usuários
- Usuários podem personalizar tema (se permitido)
- Usuários podem resetar tema para padrão
- Preparação para futura implementação de escritórios
- Build de produção sem erros
- Interface intuitiva e responsiva

Nenhuma correção adicional necessária.



Detalhes Técnicos

Prioridade de Temas

A lógica de prioridade é implementada na API `/api/user/theme` :

```
const effectiveTheme =
  user?.themePreset ||           // 1º: Tema do usuário
  officeTheme ||                 // 2º: Tema do escritório (futuro)
  globalSettings.superadminThemePreset || // 3º: Tema padrão
  "padrao";                      // 4º: Fallback
```

Permissões

O sistema verifica permissões em 2 níveis:

1. Frontend (ThemeSelector):

- Desabilita UI se `canChangeTheme === false`
- Mostra mensagens informativas

2. Backend (API /api/user/theme):

- Retorna 403 se `allowUserOverride === false`
- Validação adicional de segurança

Estrutura de Arquivos

<code>clivus_landing_page/nextjs_space/</code>	
└── prisma/schema.prisma	# Schema atualizado
└── app/	
└── api/	
└── admin/theme-settings/route.ts	# API SuperAdmin
└── user/theme/route.ts	# API Usuário
└── (protected)/admin/theme-config/page.tsx	# Página Config
└── components/	
└── theme-selector.tsx	# Atualizado
└── sidebar.tsx	# Menu atualizado
└── app/globals.css	# (CSS vars existentes)



Próximos Passos (Opcional)

Para expansão futura:

- [] Implementar escritórios (model `Office`)
- [] Criar API de gestão de escritórios
- [] Adicionar campo `officeThemePreset` em `Office`
- [] Atualizar ThemeSelector para mostrar tema do escritório
- [] Permitir donos de escritório configurarem tema para membros



Casos de Uso Práticos

Caso 1: Empresa com Visual Corporativo

- SuperAdmin define tema “Moderado” (profissional)
- Desabilita personalização de usuários
- **Resultado:** Todos usam o tema corporativo padronizado

Caso 2: Flexibilidade Total

- SuperAdmin define tema “Padrão”
- Habilita personalização de usuários

- **Resultado:** Cada usuário pode escolher seu tema favorito

Caso 3: Escritórios Multi-Tenant (Futuro)

- Empresa A: Tema “Moderno”
 - Empresa B: Tema “Simples”
 - Usuários de cada escritório herdam o tema do escritório
 - Usuários podem personalizar se o dono permitir
-

Temas Disponíveis

Padrão

- Visual: Clássico e equilibrado
- Cores: Blue (#3b82f6) + Purple (#8b5cf6)
- Uso: Geral, padrão do sistema

Simples

- Visual: Minimalista e limpo
- Cores: Grayscale monocromático
- Uso: Usuários que preferem minimalismo

Moderado

- Visual: Balanceado e profissional
- Cores: Blue (#2563eb) + Purple (#7c3aed)
- Uso: Ambientes corporativos

Moderno

- Visual: Ousado e contemporâneo
 - Cores: Purple (#8b5cf6) + Pink (#ec4899)
 - Tema escuro por padrão
 - Uso: Usuários avançados, startups
-

Sistema de hierarquia de temas totalmente funcional e pronto para produção! 

Observação: O warning sobre “logo-clivus.png duplicado” é esperado e correto (branding consistente na landing page). Não afeta o funcionamento do sistema de hierarquia de temas.