**🏢 ÉPICO 1: GESTÃO DE CLIENTES**

**📋 Descrição**

Sistema completo para cadastro e gestão de clientes da agência, incluindo informações de marca e redes sociais.

**🎯 Objetivos**

* Centralizar informações dos clientes
* Gerenciar identidade visual (cores, logo, guidelines)
* Cadastrar redes sociais de cada cliente
* Facilitar a organização de campanhas por cliente

**👥 Responsáveis**

* **Product Owner:** [Nome]
* **Tech Lead:** [Nome]
* **Desenvolvedores:** [Nomes]

**📊 Métricas de Sucesso**

* [ ] 100% dos clientes migrados do sistema antigo
* [ ] Redução de 50% no tempo para encontrar informações do cliente
* [ ] Zero perda de arquivos de guidelines

**📝 USER STORIES**

**🔹 US1.1: Cadastro de Clientes**

**Como** gerente de contas  
**Quero** cadastrar novos clientes  
**Para** organizar as campanhas por cliente

**Critérios de Aceite:**

* [ ] Posso cadastrar um novo cliente com nome
* [ ] Posso fazer upload do logo do cliente
* [ ] Posso definir até 3 cores da marca
* [ ] Posso fazer upload do PDF de guidelines
* [ ] Sistema valida campos obrigatórios
* [ ] Cliente é salvo com status ativo por padrão

**Estimativa:** 8 pontos

**🔹 US1.2: Gestão de Redes Sociais do Cliente**

**Como** gerente de contas  
**Quero** cadastrar as redes sociais de cada cliente  
**Para** ter as informações centralizadas

**Critérios de Aceite:**

* [ ] Posso adicionar múltiplas redes sociais
* [ ] Sistema valida formato do username
* [ ] URL é gerada automaticamente
* [ ] Posso adicionar observações
* [ ] Suporta múltiplas contas da mesma plataforma

**Estimativa:** 5 pontos

**🔧 TAREFAS TÉCNICAS**

**📁 Backend - Models e Migrations**

**TASK-001: Criar Migration e Model Cliente**

**Descrição:** Criar estrutura de banco de dados para clientes  
**Checklist:**

* [ ] Migration com campos: nome, logo, cores\_marca, guidelines, ativo
* [ ] Model Cliente com fillable e casts
* [ ] Soft deletes implementado
* [ ] Relacionamento hasMany com RedeSocial
* [ ] Índices para performance

**Labels:** backend, database  
**Assignee:** [Developer]  
**Estimate:** 2h

**TASK-002: Criar Migration e Model RedeSocial**

**Descrição:** Criar estrutura para redes sociais dos clientes  
**Checklist:**

* [ ] Migration com campos definidos
* [ ] Model com relacionamento belongsTo
* [ ] Enum para plataformas
* [ ] Validações no model

**Labels:** backend, database  
**Assignee:** [Developer]  
**Estimate:** 2h

**🎨 Backend - Controllers e Requests**

**TASK-003: Criar ClienteController**

**Descrição:** Controller com todas as ações do CRUD  
**Checklist:**

* [ ] index() com paginação e filtros
* [ ] create() e store()
* [ ] show() com redes sociais
* [ ] edit() e update()
* [ ] destroy() com soft delete
* [ ] restore() para reativar

**Labels:** backend, api  
**Assignee:** [Developer]  
**Estimate:** 4h

**TASK-004: Criar FormRequests**

**Descrição:** Validações para cliente  
**Checklist:**

* [ ] StoreClienteRequest
* [ ] UpdateClienteRequest
* [ ] Validar cores hexadecimais
* [ ] Validar upload de imagem
* [ ] Validar upload de PDF

**Labels:** backend, validation  
**Assignee:** [Developer]  
**Estimate:** 2h

**🖼️ Frontend - Componentes Phoenix**

**TASK-005: Criar Página de Listagem**

**Descrição:** Tela para listar todos os clientes  
**Checklist:**

* [ ] Layout com Phoenix DataTable
* [ ] Busca por nome
* [ ] Filtro ativo/inativo
* [ ] Badge para status
* [ ] Ações (editar, deletar, ver)
* [ ] Paginação

**Labels:** frontend, phoenix  
**Assignee:** [Developer]  
**Estimate:** 4h

**TASK-006: Criar Formulário de Cliente**

**Descrição:** Form para criar/editar cliente  
**Checklist:**

* [ ] Phoenix Form components
* [ ] Upload de logo com preview
* [ ] Color picker para 3 cores
* [ ] Upload de PDF
* [ ] Validações client-side
* [ ] Loading states

**Labels:** frontend, phoenix  
**Assignee:** [Developer]  
**Estimate:** 6h

**TASK-007: Criar Componente de Redes Sociais**

**Descrição:** Gerenciar redes sociais do cliente  
**Checklist:**

* [ ] Adicionar nova rede social
* [ ] Ícones das plataformas
* [ ] Validação de username
* [ ] Auto-gerar URL
* [ ] Remover rede social
* [ ] Ordenação drag-and-drop

**Labels:** frontend, phoenix  
**Assignee:** [Developer]  
**Estimate:** 4h

**🧪 Testes**

**TASK-008: Testes Unitários Models**

**Descrição:** Testar models e relacionamentos  
**Checklist:**

* [ ] Testes do Model Cliente
* [ ] Testes do Model RedeSocial
* [ ] Testes de relacionamentos
* [ ] Testes de soft delete

**Labels:** testing, backend  
**Assignee:** [Developer]  
**Estimate:** 2h

**TASK-009: Testes Feature Cliente**

**Descrição:** Testes de integração do fluxo completo  
**Checklist:**

* [ ] Teste criar cliente
* [ ] Teste editar cliente
* [ ] Teste deletar/restaurar
* [ ] Teste upload de arquivos
* [ ] Teste validações

**Labels:** testing, feature  
**Assignee:** [Developer]  
**Estimate:** 3h

**📚 Documentação**

**TASK-010: Documentar API**

**Descrição:** Documentação dos endpoints  
**Checklist:**

* [ ] Documentar endpoints no README
* [ ] Exemplos de request/response
* [ ] Códigos de erro
* [ ] Postman collection

**Labels:** documentation  
**Assignee:** [Developer]  
**Estimate:** 1h

**📅 CRONOGRAMA**

**Sprint 1 (1 semana)**

* [ ] TASK-001: Migration Cliente
* [ ] TASK-002: Migration RedeSocial
* [ ] TASK-003: ClienteController
* [ ] TASK-004: FormRequests
* [ ] TASK-008: Testes Models

**Sprint 2 (1 semana)**

* [ ] TASK-005: Página Listagem
* [ ] TASK-006: Formulário Cliente
* [ ] TASK-007: Componente Redes
* [ ] TASK-009: Testes Feature
* [ ] TASK-010: Documentação

**✅ DEFINITION OF DONE**

* [ ] Código revisado por pelo menos 1 dev
* [ ] Testes passando (min 80% coverage)
* [ ] Sem erros no Laravel Debugbar
* [ ] Documentação atualizada
* [ ] Funciona em todos os browsers
* [ ] Responsivo (mobile/tablet/desktop)
* [ ] Acessível (Phoenix a11y)
* [ ] Deploy em staging aprovado

**🚀 ENTREGÁVEIS**

1. **CRUD completo de clientes**
   * Listar, criar, editar, deletar, restaurar
   * Upload de logo e guidelines
   * Gerenciamento de cores da marca
2. **Gestão de redes sociais**
   * Múltiplas redes por cliente
   * Validações específicas por plataforma
   * Interface intuitiva
3. **Documentação**
   * API documentada
   * Manual do usuário
   * Vídeo de demonstração

**📝 NOTAS**

* Usar componentes Phoenix para manter consistência
* Seguir padrão Repository Pattern do Laravel
* Implementar cache para listagem
* Considerar lazy loading para imagens