# **Documentação Técnica: Sistema de Gerenciamento e Venda de Couro Bovino (Sunshine Cowhides)**

## **1. Visão Geral do Sistema**

O Sistema de Gerenciamento e Venda de Couro Bovino (Sunshine Cowhides) é uma aplicação web completa projetada para facilitar a comercialização de produtos de couro premium através de um catálogo digital. O sistema permite que clientes visualizem uma seleção personalizada de produtos, façam pedidos digitais, e que administradores gerenciem o inventário, os clientes e processem os pedidos.

Este sistema atende a um modelo de negócio B2B (Business-to-Business) baseado em acesso controlado, onde cada cliente recebe um código único para acessar o catálogo, frequentemente com preços e produtos personalizados para suas necessidades específicas.

## **2. Arquitetura do Sistema**

### **2.1 Visão Geral da Arquitetura**

O aplicativo segue uma arquitetura cliente-servidor moderna:

* **Front-end**: Interface de usuário responsiva com HTML5, CSS3 e JavaScript
* **Back-end**: Servidor Node.js com Express para API REST
* **Armazenamento de dados**: Combinação de MongoDB e Google Drive
* **Integração**: Google Drive API para gerenciamento de arquivos
* **Autenticação**: Sistema baseado em códigos de acesso para clientes e login tradicional para administradores

### **2.2 Fluxo de Dados**

Cliente Web → API (Express.js) → Controladores → Serviços

↓

Notificação por Email ← ← ← ← ← MongoDB

↓

Google Drive

### **2.3 Componentes do Sistema**

* **Aplicação Web**: Interface para clientes e administradores
* **API REST**: Endpoints para comunicação com o front-end
* **Serviços de Armazenamento**: MongoDB para dados estruturados, Google Drive para imagens
* **Sistema de Cache**: Armazenamento local e em memória para otimização de desempenho
* **Serviço de Notificações**: Envio de e-mails automatizados

## **3. Tecnologias Utilizadas**

### **3.1 Front-end**

* **HTML5**: Estrutura semântica da página
* **CSS3**: Estilização com design de luxo, responsivo com Grid e Flexbox
* **JavaScript**: Manipulação DOM, carregamento assíncrono, gerenciamento de estado
* **Lazy Loading**: IntersectionObserver para carregamento otimizado de imagens
* **Responsive Design**: Adaptação para múltiplos dispositivos

### **3.2 Back-end**

* **Node.js**: Ambiente de execução JavaScript
* **Express.js**: Framework para API RESTful
* **MongoDB**: Banco de dados NoSQL para armazenamento de dados estruturados
* **Google Drive API**: Armazenamento e gerenciamento de imagens
* **Sharp**: Processamento e otimização de imagens
* **Nodemailer**: Serviço de envio de e-mails
* **Bcrypt**: Criptografia para senhas de administrador

### **3.3 DevOps e Infraestrutura**

* **Controle de Versão**: Git
* **Hospedagem**: Planejado para Oracle Cloud Free Tier
* **Gerenciamento de Dependências**: NPM
* **Variáveis de Ambiente**: dotenv para configuração

## **4. Estrutura do Banco de Dados**

### **4.1 Coleções MongoDB**

#### **CustomerCode**

javascript

{

code: String, *// Código de acesso único de 4 dígitos*

customerName: String, *// Nome do cliente*

createdAt: Date, *// Data de criação do código*

lastAccess: Date, *// Último acesso do cliente*

lastUpdated: Date, *// Última atualização dos dados*

items: [String], *// IDs das fotos no carrinho*

orderInProgress: Boolean, *// Status de pedido em andamento*

orderCompleted: Boolean, *// Status de pedido finalizado*

orderStatus: String, *// Status do pedido (processing, waiting\_payment, paid, null)*

orderDate: Date *// Data do pedido*

}

#### **CategoryAccess**

javascript

{

customerCode: String, *// Código do cliente*

categoryAccess: [

{

categoryId: String, *// ID da categoria no Google Drive*

enabled: Boolean, *// Estado de acesso (permitido/negado)*

customPrice: Number, *// Preço personalizado para o cliente*

minQuantityForDiscount: Number, *// Quantidade mínima para desconto*

discountPercentage: Number *// Percentual de desconto*

}

]

}

#### **CategoryPrice**

javascript

{

folderId: String, *// ID da pasta no Google Drive*

name: String, *// Nome da categoria*

price: Number, *// Preço padrão para esta categoria*

updatedAt: Date, *// Data da última atualização*

path: String *// Caminho completo da categoria*

}

#### **Order**

javascript

{

customerCode: String, *// Código do cliente*

customerName: String, *// Nome do cliente*

photoIds: [String], *// IDs das fotos selecionadas*

comments: String, *// Comentários adicionais*

status: String, *// Status (processing, waiting\_payment, failed, paid)*

createdAt: Date, *// Data de criação*

processedAt: Date, *// Data de processamento*

folderName: String, *// Nome da pasta criada*

folderId: String, *// ID da pasta no Google Drive*

error: String *// Mensagem de erro (se houver)*

}

#### **Admin**

javascript

{

email: String, *// Email do administrador*

password: String *// Senha criptografada*

}

### **4.2 Estrutura do Google Drive**

* **Pasta Raiz**: Contém todas as categorias de produtos
* **Categorias**: Pastas representando tipos de produtos
* **Waiting Payment**: Pasta para pedidos aguardando pagamento
* **Sold**: Pasta para pedidos concluídos

## **5. Componentes e Módulos Principais**

### **5.1 Módulos do Back-end**

#### **Controladores**

* **adminController.js**: Gerenciamento de acessos, preços e códigos
* **orderController.js**: Processamento de pedidos e imagens
* **photoController.js**: Acesso a fotos e categorias

#### **Serviços**

* **driveService.js**: Interface com Google Drive API
* **emailService.js**: Envio de notificações por email
* **mongoService.js**: Interface com MongoDB

#### **Middleware**

* **auth.js**: Verificação de autenticação
* **errorHandler.js**: Tratamento de erros

### **5.2 Módulos do Front-end**

#### **Interface do Cliente**

* **gallery.js**: Visualização e navegação no catálogo
* **cart.js**: Gerenciamento de seleções do cliente
* **lightbox.js**: Visualização detalhada de imagens
* **auth.js**: Autenticação por código

#### **Interface do Administrador**

* **admin.js**: Painel administrativo
* **price-manager.js**: Gestão de preços
* **folder-manager.js**: Organização de categorias

## **6. Fluxos de Trabalho Principais**

### **6.1 Fluxo do Cliente**

1. **Acesso**: Cliente recebe um código de 4 dígitos
2. **Autenticação**: Insere o código na tela de login
3. **Navegação**: Visualiza categorias e produtos permitidos com preços personalizados
4. **Seleção**: Adiciona produtos ao carrinho
5. **Finalização**: Submete pedido com comentários opcionais
6. **Confirmação**: Recebe confirmação do pedido

### **6.2 Fluxo de Processamento de Pedidos**

1. **Submissão**: Cliente finaliza pedido
2. **Criação de Pasta**: Sistema cria pasta com nome do cliente + data + quantidade
3. **Organização por Categoria**: Dentro da pasta do pedido, sistema cria subpastas por categoria
4. **Movimentação de Arquivos**: Sistema copia as imagens selecionadas para as subpastas
5. **Notificação**: Email enviado ao administrador com resumo e link
6. **Aguardando Pagamento**: Pedido fica na pasta de espera
7. **Finalização**: Administrador marca como pago e pasta move para "Sold"

### **6.3 Fluxo Administrativo**

1. **Login**: Administrador acessa com email e senha
2. **Gerenciamento de Clientes**: Cria/edita/remove códigos de acesso
3. **Configuração de Acesso**: Define quais categorias cada cliente pode ver
4. **Precificação**: Define preços por categoria ou personalizados por cliente
5. **Gestão de Pedidos**: Visualiza, processa e move pedidos entre estados
6. **Manutenção**: Organiza categorias e produtos no Google Drive

## **7. Funcionalidades Detalhadas**

### **7.1 Sistema de Catálogo**

#### **Carregamento Otimizado**

* **Carregamento em Lotes**: Categorias carregadas em sequência para otimizar performance
* **Preview Inicial**: Primeiras 6-12 imagens de cada categoria carregadas imediatamente
* **Lazy Loading**: Carregamento progressivo com IntersectionObserver
* **Carregamento Sob Demanda**: Botão "Ver mais fotos" para carregar lotes adicionais
* **Pré-carregamento Estratégico**: Carregamento antecipado de categorias próximas

#### **Visualização de Imagens**

* **Lightbox**: Visualização ampliada com navegação por teclado
* **Zoom**: Controle de aproximação e navegação em alta resolução
* **Otimização de Imagens**: Processamento para diferentes resoluções
* **Cache de Imagens**: Armazenamento local para reduzir transferências

### **7.2 Sistema de Preços**

#### **Estrutura de Preços em Camadas**

* **Preço Base por Categoria**: Valor padrão para todos os clientes
* **Preços Personalizados**: Valores específicos por cliente e categoria
* **Sistema de Desconto**: Baseado em quantidade mínima de itens
* **Atualização em Lote**: Ferramenta para alterar múltiplos preços simultaneamente

### **7.3 Processamento de Pedidos**

#### **Fluxo Assíncrono**

* **Resposta Rápida**: Confirmação imediata para o cliente
* **Processamento em Segundo Plano**: Movimentação de arquivos sem bloquear a UI
* **Organização Automática**: Criação de estrutura por categorias
* **Confirmação por Email**: Notificação detalhada com agrupamento por categoria

### **7.4 Gerenciamento de Acesso**

#### **Controle Granular**

* **Categorias Personalizadas**: Cada cliente vê apenas categorias autorizadas
* **Preços Personalizados**: Valores específicos por relacionamento comercial
* **Histórico de Acesso**: Registro de datas de último login
* **Inativação**: Capacidade de desativar códigos sem excluir

## **8. Interface do Usuário e Design**

### **8.1 Estilo Visual**

#### **Design de Luxo**

* **Paleta de Cores**: Combinação sofisticada de dourado (#D4AF37), creme (#F8F5F0), e tons escuros (#212529)
* **Tipografia**: Combinação de 'Playfair Display' para títulos e 'Montserrat' para corpo
* **Elementos Visuais**: Sombras suaves, transições elegantes, efeitos sutis de hover
* **Responsividade**: Adaptação elegante para diferentes tamanhos de tela

### **8.2 Componentes de Interface**

#### **Portal do Cliente**

* **Galeria de Produtos**: Layout em grid com visualização por categoria
* **Lightbox**: Visualização detalhada com controles de navegação
* **Carrinho de Seleção**: Painel flutuante com itens selecionados
* **Formulário de Pedido**: Interface para finalização com campo de comentários

#### **Painel Administrativo**

* **Sistema de Abas**: Organização por funções principais
* **Geração de Códigos**: Interface para criar códigos de cliente
* **Editor de Acesso**: Matriz de configuração por categoria
* **Gerenciador de Preços**: Edição individual e em lote
* **Lista de Pedidos**: Organização por status com ações contextuais

## **9. Otimizações de Performance**

### **9.1 Técnicas de Front-end**

* **Lazy Loading**: Carregamento progressivo de imagens
* **Paginação Preguiçosa**: Carregamento de dados sob demanda
* **Carregamento em Lotes**: Priorização de conteúdo crítico
* **Cache Local**: Armazenamento de imagens e dados em memória
* **Pré-carregamento**: Antecipação de necessidades do usuário

### **9.2 Otimizações de Back-end**

* **Processamento Assíncrono**: Operações em segundo plano
* **Sistema de Cache**: Redução de chamadas ao Google Drive
* **ETags**: Prevenção de downloads desnecessários
* **Compressão de Imagens**: Otimização usando Sharp
* **Invalidação Seletiva de Cache**: Limpeza apenas quando necessário

## **10. Segurança**

### **10.1 Modelo de Autenticação**

* **Códigos Numéricos**: Acesso de cliente por código de 4 dígitos
* **Credenciais Administrativas**: Email e senha com bcrypt
* **Verificação de Sessão**: Estado mantido localmente

### **10.2 Controle de Acesso**

* **Permissões Granulares**: Acesso específico a categorias por cliente
* **Service Account**: Uso de credenciais seguras para Google Drive
* **Headers de Segurança**: Configuração de cabeçalhos HTTP adequados
* **Validação de Entrada**: Sanitização de parâmetros de requisição

## **11. Integração com Serviços Externos**

### **11.1 Google Drive API**

* **Autenticação**: Service Account com escopo limitado
* **Organização**: Estrutura hierárquica para categorias e pedidos
* **Manipulação de Arquivos**: Movimentação e cópia de imagens
* **Permissões**: Acesso controlado por API

### **11.2 Serviço de Email**

* **Nodemailer**: Envio de notificações via SMTP
* **Templates HTML**: Emails formatados com detalhes do pedido
* **Links Diretos**: Acesso facilitado às pastas relevantes

## **12. Manutenção e Monitoramento**

### **12.1 Logging**

* **Console Logs**: Registro detalhado de operações
* **Tratamento de Erros**: Captura e registro estruturado
* **Telemetria**: Acompanhamento de tempos de resposta

### **12.2 Backups e Recuperação**

* **MongoDB Backup**: Rotina de backup para dados estruturados
* **Google Drive**: Redundância nativa para arquivos

### **12.3 Troubleshooting**

* **Ferramentas de Diagnóstico**: Endpoints de teste
* **Verificação de Status**: Monitoramento de serviços externos
* **Limpeza de Cache**: Ferramentas manuais e automáticas

## **13. Implantação**

### **13.1 Requisitos de Servidor**

* **Ambiente Node.js**: v14+ com NPM
* **Memória**: Mínimo de 1GB RAM
* **Armazenamento**: 10GB+ para cache de imagens
* **Largura de Banda**: Suficiente para tráfego de imagens

### **13.2 Variáveis de Ambiente**

Configurações necessárias:

PORT=3000

DRIVE\_FOLDER\_ID=id\_da\_pasta\_raiz

WAITING\_PAYMENT\_FOLDER\_ID=id\_da\_pasta\_aguardando\_pagamento

SOLD\_FOLDER\_ID=id\_da\_pasta\_vendidos

ADMIN\_EMAIL=email\_do\_administrador

ADMIN\_PASSWORD=senha\_do\_administrador

EMAIL\_USER=email\_para\_notificacoes

EMAIL\_PASSWORD=senha\_de\_app\_do\_email

MONGODB\_URI=uri\_de\_conexao\_do\_mongodb

### **13.3 Arquivos de Credenciais**

* **credentials.json**: Configuração do Google Drive API
* **firebase-credentials.json**: Configuração do Firebase (se usado)

## **14. Futuras Expansões e Melhorias**

### **14.1 Melhorias Planejadas**

* **CDN para Imagens**: Implementação de rede de distribuição de conteúdo
* **PWA**: Transformação em Progressive Web App
* **Sistema de Pedidos Recorrentes**: Facilitar recompras
* **Dashboard Analítico**: Métricas de vendas e comportamento
* **Multi-idioma**: Suporte a diferentes línguas

### **14.2 Escalabilidade**

* **Arquitetura de Microserviços**: Separação em componentes independentes
* **Balanceamento de Carga**: Distribuição de tráfego
* **Otimização de Banco de Dados**: Índices e sharding para crescimento

## **15. Apêndices**

### **15.1 Estrutura de Arquivos**

fotosystem/

├── public/ # Arquivos estáticos para o frontend

│ ├── css/ # Folhas de estilo

│ │ └── main.css # Estilos principais

│ ├── js/ # Scripts do cliente

│ │ ├── admin.js # Funcionalidades administativas

│ │ ├── api-client.js # Cliente de API

│ │ ├── auth.js # Autenticação

│ │ ├── cart.js # Gerenciamento do carrinho

│ │ ├── database-service.js # Serviço de banco de dados

│ │ ├── folder-manager.js # Gerenciamento de pastas

│ │ ├── gallery.js # Exibição da galeria

│ │ ├── lightbox.js # Visualização de imagens

│ │ ├── price-manager.js # Gerenciamento de preços

│ │ └── utils.js # Utilitários

│ └── index.html # Página principal

│

├── src/ # Código-fonte do backend

│ ├── config/ # Configurações

│ │ ├── database.js # Configuração do MongoDB

│ │ └── google.drive.js # Configuração do Google Drive

│ │

│ ├── controllers/ # Controladores para lógica de negócios

│ │ ├── adminController.js

│ │ ├── orderController.js

│ │ └── photoController.js

│ │

│ ├── models/ # Modelos de dados

│ │ ├── admin.js

│ │ ├── categoryAccess.js

│ │ ├── categoryPrice.js

│ │ ├── customerCode.js

│ │ └── order.js

│ │

│ ├── routes/ # Definição de rotas da API

│ │ ├── admin.js

│ │ ├── client.js

│ │ ├── config.js

│ │ ├── db.js

│ │ ├── orders.js

│ │ └── photos.js

│ │

│ ├── services/ # Serviços para operações complexas

│ │ ├── driveService.js

│ │ ├── emailService.js

│ │ └── mongoService.js

│ │

│ └── server.js # Ponto de entrada do aplicativo

│

├── cache/ # Diretório de cache local

│ └── optimized/ # Cache de imagens otimizadas

│

├── .env # Variáveis de ambiente

├── credentials.json # Credenciais do Google Drive

└── package.json # Dependências e scripts

### **15.2 API Endpoints**

#### **Endpoints de Fotos**

* GET /api/photos - Obter fotos com filtros e paginação
* GET /api/photos/categories - Obter categorias disponíveis

#### **Endpoints de Pedidos**

* POST /api/orders - Criar novo pedido
* GET /api/orders/folders - Listar pastas de pedidos
* PUT /api/orders/status - Atualizar status do pedido
* GET /api/orders/details - Obter detalhes de um pedido
* GET /api/orders/thumbnail/:fileId - Servir thumbnail otimizado
* GET /api/orders/highres-image/:fileId - Servir imagem em alta resolução

#### **Endpoints de Administração**

* POST /api/admin/login - Autenticar administrador
* POST /api/admin/code - Gerar código de cliente
* GET /api/admin/codes - Listar códigos ativos
* DELETE /api/admin/code/:code - Excluir código de cliente
* GET /api/admin/folders/leaf - Listar pastas folha
* GET /api/admin/categories/prices - Obter preços de categorias
* POST /api/admin/categories/:folderId/price - Definir preço
* POST /api/admin/categories/batch-update - Atualização em lote
* GET /api/admin/customers/:code/category-access - Obter configurações
* POST /api/admin/customers/:code/category-access - Salvar configurações

#### **Endpoints de Cliente**

* GET /api/client/initial-data - Obter dados iniciais do cliente
* POST /api/client/clear-cache - Limpar cache do servidor
* POST /api/client/selections - Salvar seleções do cliente

### **15.3 Componentes Visuais e UI Kit**

#### **Design System**

* **Cores Primárias**:
* --color-dark: #212529
* --color-gold: #D4AF37
* --color-cream: #F8F5F0
* --color-taupe: #483C32
* --color-charcoal: #36454F
* **Tipografia**:
* Títulos: 'Playfair Display', serif
* Corpo: 'Montserrat', sans-serif
* **Espaçamento**:
* Base: 8px
* Escala: 8, 16, 24, 32, 40px
* **Sombras**:
* Leve: 0 10px 30px rgba(0, 0, 0, 0.08)
* Média: 0 25px 50px rgba(0, 0, 0, 0.2)
* **Transições**:
* Padrão: all 0.3s cubic-bezier(0.25, 0.8, 0.25, 1)

#### **Componentes**

* **Botões**: Primário (Dourado), Secundário (Marrom), Perigo (Vermelho)
* **Cards**: Para exibição de produtos
* **Modais**: Para detalhes e confirmações
* **Formulários**: Campos de entrada estilizados
* **Notificações**: Sistema de toast para feedback

Esta documentação técnica completa fornece uma visão abrangente do Sistema de Gerenciamento e Venda de Couro Bovino (Sunshine Cowhides), cobrindo todos os aspectos técnicos, funcionais e de design. A documentação serve como referência para desenvolvedores, administradores de sistema e stakeholders para entendimento, manutenção e evolução da plataforma.