

Resumo das Melhorias nos Formulários de Produtos






Visão Geral

Este documento descreve as melhorias implementadas nas telas de criação e edição de produtos do sistema DELIVEREI, conforme requisitos especificados.



Requisitos Atendidos

1. Campos do Formulário

Todos os campos solicitados foram implementados:

-  **Nome do Produto** - Campo de texto obrigatório
-  **Preço** - Campo numérico formatado para moeda (R\$) com validação
-  **Descrição do Produto** - Campo de texto longo (textarea)
-  **Upload de Imagem** - Componente intuitivo com upload real usando Supabase Storage
-  **Estoque** - Campo numérico para quantidade em estoque

Campos adicionais mantidos do sistema original:

-  **Categoria** - Campo de texto opcional
-  **Produto Ativo** - Checkbox para controlar visibilidade

2. Funcionalidades Implementadas

Upload de Imagem com Supabase Storage

- Upload direto de arquivo do computador/celular (sem necessidade de URL)
- Validação de tipo de arquivo (JPEG, PNG, WebP, GIF)
- Validação de tamanho (máximo 5MB)
- Preview da imagem em tempo real
- Botão para remover imagem
- Loading indicator durante upload
- Mensagens de sucesso/erro
- URLs públicas automáticas

Validação de Formulário

- Nome do produto obrigatório
- Preço deve ser maior que zero
- Estoque não pode ser negativo
- Mensagens de erro em tempo real
- Feedback visual nos campos com erro

Interface Responsiva

- Layout adaptável para desktop e mobile
- Grid responsivo de 2 colunas em desktop, 1 coluna em mobile
- Componentes otimizados para telas pequenas

- Upload de imagem funciona em dispositivos móveis

Arquivos Criados/Modificados

Novos Arquivos

1. **src/lib/supabase.ts**
 - Cliente Supabase configurado
 - Funções de upload e deleção de imagens
 - Gerenciamento de Storage
2. **src/components/common/ImageUpload.tsx**
 - Componente reutilizável de upload
 - Suporte a drag-and-drop visual
 - Preview de imagem
 - Validações integradas
3. **src/components/products/ProductForm.tsx**
 - Formulário reutilizável de produto
 - Validações completas
 - Estados de loading
 - Layout responsivo
4. **src/services/productsApi.ts**
 - API client para produtos
 - CRUD completo
 - Tipagem TypeScript
 - Paginação e filtros
5. **.env.local**
 - Variáveis de ambiente do Supabase
 - Não commitado (está no .gitignore)
6. **SUPABASE_STORAGE_SETUP.md**
 - Documentação completa de setup
 - Políticas de segurança
 - Troubleshooting

Arquivos Modificados

1. **src/pages/admin/store/Products.tsx**
 - Integração com API real
 - Modal melhorado com formulário completo
 - Lista de produtos com thumbnails
 - Indicadores de status
 - Função de exclusão
2. **src/pages/admin/store/ProductEdit.tsx**
 - Integração com API real
 - Formulário completo de edição
 - Loading states
 - Tratamento de erros

3. `package.json`

- Adicionada dependência `@supabase/supabase-js`

Melhorias de UX/UI

Tela de Listagem de Produtos

- **Thumbnails de Imagem:** Miniaturas das fotos dos produtos
- **Badge de Status:** Indicador visual de produto ativo/inativo
- **Coluna de Estoque:** Visualização rápida da quantidade
- **Hover Effects:** Feedback visual ao passar o mouse
- **Loading State:** Spinner durante carregamento

Formulários (Criação e Edição)

- **Layout em Grades:** Organização visual clara
- **Campos Agrupados:** Informações relacionadas juntas
- **Labels Descritivas:** Texto de ajuda e exemplos
- **Validação em Tempo Real:** Feedback imediato
- **Estados de Loading:** Indicadores durante operações assíncronas
- **Preview de Imagem:** Visualização antes de salvar
- **Mensagens Toast:** Feedback de sucesso/erro

Modal de Criação

- **Tamanho Grande:** Modal ocupa 5xl (máx. 1280px) para mostrar todo formulário
- **Scroll Vertical:** Suporte para telas pequenas
- **Botão de Fechar:** X no canto superior direito
- **Backdrop:** Fundo escuro semi-transparente

Segurança

Upload de Imagens

- Validação de tipo MIME no frontend
- Validação de tamanho (5MB max)
- Nomes de arquivo únicos (timestamp + random)
- Autenticação necessária para upload
- Políticas RLS no Supabase

API

- Autenticação JWT obrigatória
- Multi-tenancy: empresalid isolado
- Validação de dados no backend
- Rate limiting (via Supabase)

Configuração Necessária

1. Instalar Dependências

```
npm install
```

2. Configurar Supabase Storage

Siga as instruções em `SUPABASE_STORAGE_SETUP.md` para:

1. Criar o bucket `produtos`
2. Configurar políticas de acesso
3. Testar upload

3. Configurar Variáveis de Ambiente

O arquivo `.env.local` já foi criado com as credenciais fornecidas:

```
VITE_SUPABASE_URL=https://yfqxqxqxqxqxqxqxqx.supabase.co  
VITE_SUPABASE_ANON_KEY=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```



4. Executar o Projeto

```
# Desenvolvimento  
npm run dev  
  
# Build para produção  
npm run build
```



Responsividade

O sistema foi testado e funciona corretamente em:

-  Desktop (1920x1080+)
-  Laptop (1366x768)
-  Tablet (768x1024)
-  Mobile (375x667)

Breakpoints

- **md (768px+)**: Layout de 2-3 colunas
- **< 768px**: Layout de 1 coluna



Testando as Funcionalidades

1. Criar Produto

1. Acesse `/admin/store/products`
2. Clique em “Novo Produto”
3. Preencha os campos obrigatórios
4. Faça upload de uma imagem
5. Clique em “Criar Produto”

6. Verifique o toast de sucesso
7. Produto deve aparecer na lista

2. Editar Produto

1. Na lista de produtos, clique em “Editar”
2. Modifique os campos desejados
3. Troque a imagem (opcional)
4. Clique em “Atualizar Produto”
5. Verifique o toast de sucesso
6. Verifique as mudanças na lista

3. Excluir Produto

1. Na lista de produtos, clique em “Excluir”
2. Confirme a exclusão
3. Produto é marcado como inativo (soft delete)

Backend

O backend **NÃO** precisou de modificações, pois:

- O modelo `Produto` já tinha o campo `imagem` (String)
- O DTO já aceitava imagem como string (URL)
- A validação já estava implementada
- O sistema já tinha autenticação e multi-tenancy

As imagens são armazenadas no Supabase Storage e apenas a URL pública é salva no banco de dados PostgreSQL.

Comparação Antes/Depois

Antes

- ❌ Upload de imagem pedia URL manual
- ❌ Sem preview de imagem
- ❌ Sem validação visual
- ❌ Modal pequeno com campos básicos
- ❌ Sem indicador de loading
- ❌ Sem gestão de estoque
- ❌ Dados mockados

Depois

- ✅ Upload direto de arquivo
- ✅ Preview em tempo real
- ✅ Validação completa com feedback visual
- ✅ Modal grande com todos os campos
- ✅ Loading states em todas operações
- ✅ Campo de estoque funcional
- ✅ Integração com API real

Próximos Passos Sugeridos

1. **Múltiplas Imagens:** Permitir galeria de fotos por produto
2. **Crop de Imagem:** Editor para ajustar imagens antes do upload
3. **Categorias Predefinidas:** Select ao invés de texto livre
4. **Variações de Produto:** Tamanhos, cores, etc.
5. **Importação em Massa:** CSV de produtos
6. **Analytics:** Produtos mais vendidos, estoque baixo
7. **Otimização de Imagens:** Redimensionamento automático

Suporte

Para dúvidas ou problemas:

1. Consulte `SUPABASE_STORAGE_SETUP.md` para configuração do Storage
2. Verifique os logs do console do navegador
3. Verifique os logs do backend
4. Consulte a documentação do Supabase

Notas Importantes

- As imagens antigas com URLs externas continuam funcionando
- O sistema suporta tanto URLs do Supabase quanto URLs externas
- O campo de imagem é opcional
- Produtos sem imagem aparecem sem thumbnail na lista
- O soft delete mantém os dados no banco
- As credenciais do Supabase devem ser tratadas como secretas

Desenvolvido por: DeepAgent - Abacus.AI

Data: Outubro 2025

Versão: 1.0.0