

# Frontend Updates - App Fiscal

---

## Resumo das Alterações

Todas as interfaces frontend React/TypeScript foram criadas/atualizadas para consumir os endpoints implementados no backend.

## Novos Módulos Implementados

---

### 1. CADASTROS GERAIS ( /cadastros )

#### Completamente implementado e funcional

##### Funcionalidades:

- ⚡ Página principal com listagem de cadastros em tabela responsiva
- 🔎 Filtros por tipo (fornecedor/cliente/usuário/outros) e status (ativo/inativo)
- 🔎 Busca por nome ou documento
- ✚ Botão “Novo Cadastro” com modal de criação
- 🖊️ Modal/Formulário para editar cadastro existente
- 🔍 Ações: editar, inativar/ativar
- 🎨 Indicadores visuais de status com cores
- 📊 Dashboard com contagem por tipo de cadastro (4 cards com estatísticas)
- 🎨 Design moderno com gradientes e sombras

##### Endpoints consumidos:

- GET /api/cadastros-gerais (listar com filtros)
- POST /api/cadastros-gerais (criar)
- GET /api/cadastros-gerais/{id} (buscar por ID)
- PUT /api/cadastros-gerais/{id} (atualizar)
- DELETE /api/cadastros-gerais/{id} (inativar)
- GET /api/cadastros-gerais/count-by-type (contagem por tipo)

---

### 2. CONTAS A PAGAR ( /contas-pagar )

#### Completamente atualizado com novas funcionalidades

##### Funcionalidades:

- 📋 Listagem atualizada mostrando **nome do fornecedor** (não apenas ID)
- 🔎 Campo de seleção de fornecedor (dropdown com cadastros do tipo “fornecedor”)
- ✅ Validação: só permite criar se fornecedor existir
- 📦 Formulário de parcelamento com campo “quantidade de parcelas”
- 📁 Visualização expandível (accordion/collapse) para mostrar todas as parcelas de um grupo
- 🖊️ Cada parcela mostra: número, valor, vencimento, status (paga/pendente)
- 🖊️ Modal para editar parcela individual (valor, vencimento, data pagamento)

- ⚡ Botão de “baixa antecipada” para marcar parcelas futuras como pagas
- 📊 Indicador visual de progresso das parcelas (ex: 3/10 pagas)
- 🎨 Interface agrupada por `codgrp` com expansão de detalhes

#### **Endpoints consumidos:**

- GET /api/contas-pagar (listar com filtros)
  - POST /api/contas-pagar/parcelar/{codfor} (criar parcelas)
  - GET /api/contas-pagar/grupo/{codgrp} (listar parcelas do grupo)
  - PUT /api/contas-pagar/{id} (editar parcela)
  - POST /api/contas-pagar/{id}/baixar (dar baixa)
  - POST /api/contas-pagar/{id}/cancelar (cancelar)
  - DELETE /api/contas-pagar/{id} (excluir)
- 

### **3. CONTAS A RECEBER ( /contas-receber )**

#### **✓ Completamente atualizado com novas funcionalidades**

#### **Funcionalidades:**

- 📋 Listagem atualizada mostrando **nome do cliente** (não apenas ID)
- ⚠ Campo de seleção de cliente (dropdown com cadastros do tipo “cliente”)
- ✓ Validação: só permite criar se cliente existir
- 📦 Formulário de parcelamento com campo “quantidade de parcelas”
- 📂 Visualização expandível (accordion/collapse) para mostrar todas as parcelas de um grupo
- 📝 Cada parcela mostra: número, valor, vencimento, status (recebida/pendente)
- 📝 Modal para editar parcela individual
- ⚡ Botão de “baixa antecipada”
- 📊 Indicador visual de progresso das parcelas
- 🎨 Interface similar ao Contas a Pagar, mas com tema verde

#### **Endpoints consumidos:**

- GET /api/contas-receber (listar com filtros)
  - POST /api/contas-receber/parcelar/{codcli} (criar parcelas)
  - GET /api/contas-receber/grupo/{codgrp} (listar parcelas do grupo)
  - PUT /api/contas-receber/{id} (editar parcela)
  - POST /api/contas-receber/{id}/baixar (dar baixa)
  - POST /api/contas-receber/{id}/cancelar (cancelar)
  - DELETE /api/contas-receber/{id} (excluir)
- 

### **4. GESTÃO DE LICENÇAS ( /superadmin/licencias )**

#### **✓ Novo módulo criado - SuperAdmin only**

#### **Funcionalidades:**

- 🔒 Página restrita ao SuperAdmin
- 📊 Dashboard com cards de estatísticas:

- Total de licenças
- Ativas
- Vencidas
- A vencer (próximos 30 dias)
- Inadimplentes
- 📋 Listagem de licenças em tabela com filtros
- ✎ Formulário para cadastrar nova empresa/filial
- 🔑 Gerador automático de chave de licença (exibir chave gerada)
- 🖊 Campos: empresa/filial, CNPJ, data início, data encerramento, status pagamento
- 🖍 Modal para editar licença
- 🔍 Ações: renovar (por X meses), ativar, desativar, excluir
- 🌈 Indicadores visuais de status (ativa/vencida/inativa)
- 💳 Badge de status de pagamento (pago/pendente)
- 📋 Botão para copiar chave de licença
- 🕒 Alerta visual para licenças a vencer em menos de 30 dias
- 🎨 Design moderno com gradientes específicos por card

### **Endpoints consumidos:**

- GET /api/licencias (listar com filtros)
- POST /api/licencias (criar com geração automática de chave)
- GET /api/licencias/{id} (buscar)
- PUT /api/licencias/{id} (atualizar)
- DELETE /api/licencias/{id} (deletar)
- GET /api/licencias/dashboard (estatísticas)
- POST /api/licencias/{id}/renovar (renovar licença)
- POST /api/licencias/{id}/ativar (ativar)
- POST /api/licencias/{id}/desativar (desativar)

## **5. INTEGRAÇÃO E NAVEGAÇÃO**

### **✓ Sistema de rotas e menu atualizado**

#### **Alterações:**

- ✓ Adicionada rota /superadmin/licencias protegida por guard (SuperAdmin only)
- ✓ Item de menu “🔑 Gestão de Licenças” adicionado na seção SuperAdmin
- ✓ Guard RequireSuperAdmin protege rotas de acesso não autorizado
- ✓ Importação da LicenciasPage no DashboardLayout
- ✓ Tratamento de erros e loading states em todas as páginas
- ✓ Mensagens de sucesso/erro (toasts/notifications) padronizadas
- ✓ Validações de formulário em todos os módulos
- ✓ Responsividade mobile com TailwindCSS

## Design e UX

---

### Padrões Implementados:

-  **Gradientes modernos** em headers e cards
-  **Animações suaves** em hover e transições
-  **Design responsivo** com breakpoints mobile/tablet/desktop
-  **Cores consistentes** usando paleta do TailwindCSS
-  **Loading states** com spinners animados
-  **Alertas visuais** com cores semânticas (verde=sucesso, vermelho=erro)
-  **Indicadores de status** com badges coloridos
-  **Barras de progresso** para parcelas
-  **Dropdowns estilizados** para seleção
-  **Modais elegantes** com sombras e overlay

### Cores por Módulo:

-  **Cadastros Gerais**: Azul/Verde/Roxo/Cinza (por tipo)
  -  **Contas a Pagar**: Azul
  -  **Contas a Receber**: Verde
  -  **Licenças**: Azul/Verde/Vermelho/Amarelo/Laranja (por status)
- 

## Arquivos Criados/Modificados

---

### Novos arquivos:

1. frontend/src/pages/LicenciasPage.tsx  **NOVO**

### Arquivos modificados:

1. frontend/src/pages/CadastrosPage.tsx  **REFATORADO**
  2. frontend/src/pages/ContasPagarPage.tsx  **ATUALIZADO**
  3. frontend/src/pages/ContasReceberPage.tsx  **ATUALIZADO**
  4. frontend/src/components/DashboardLayout.tsx  **ATUALIZADO**
- 

## Como Executar

---

### Desenvolvimento:

```
cd frontend
npm install
npm run dev
```

### Build de Produção:

```
cd frontend
npm run build
```

## Preview de Produção:

```
cd frontend
npm run preview
```

## Checklist de Funcionalidades

### Cadastros Gerais:

- [x] Listagem com tabela responsiva
- [x] Filtros por tipo e status
- [x] Busca por nome/documento
- [x] Modal de criação
- [x] Modal de edição
- [x] Ação de inativar/ativar
- [x] Dashboard com estatísticas
- [x] Validações de formulário

### Contas a Pagar:

- [x] Mostrar nome do fornecedor
- [x] Dropdown de seleção de fornecedor
- [x] Validação de fornecedor existente
- [x] Formulário de parcelamento
- [x] Visualização expandível de parcelas
- [x] Edição de parcela individual
- [x] Baixa antecipada
- [x] Indicador de progresso
- [x] Agrupamento por codgrp

### Contas a Receber:

- [x] Mostrar nome do cliente
- [x] Dropdown de seleção de cliente
- [x] Validação de cliente existente
- [x] Formulário de parcelamento
- [x] Visualização expandível de parcelas
- [x] Edição de parcela individual
- [x] Baixa antecipada
- [x] Indicador de progresso
- [x] Agrupamento por codgrp

### Gestão de Licenças:

- [x] Dashboard com 5 cards de estatísticas
- [x] Listagem com filtros
- [x] Gerador de chave automático
- [x] Modal de criação

- [x] Modal de edição
- [x] Renovar licença
- [x] Ativar/Desativar
- [x] Copiar chave
- [x] Indicadores visuais
- [x] Restrição SuperAdmin

## Navegação:

- [x] Rotas configuradas
- [x] Menu atualizado
- [x] Guards de permissão
- [x] Responsividade

## Próximos Passos Sugeridos

1. **Testes de integração:** Testar todas as funcionalidades com o backend rodando
2. **Melhorias de UX:** Adicionar mais animações e transições suaves
3. **Acessibilidade:** Implementar labels ARIA e melhorar navegação por teclado
4. **Internacionalização:** Adicionar suporte a múltiplos idiomas
5. **Testes unitários:** Implementar testes com React Testing Library
6. **Otimização:** Code splitting e lazy loading para melhor performance



## Notas Importantes

-  Todo o código segue os padrões estabelecidos no projeto
-  TypeScript configurado corretamente sem erros de compilação
-  TailwindCSS usado para estilização consistente
-  API client ( api.ts ) usado para todas as requisições
-  Tratamento de erros implementado em todos os módulos
-  Loading states e feedback visual adequados
-  Responsividade garantida em todos os módulos



## Build Status

 **Build bem-sucedido** - Sem erros de TypeScript ou compilação

- ```
✓ 99 modules transformed.
✓ built in 4.31s
```

## **Supporte**

---

Para dúvidas ou problemas, consulte a documentação do backend ou entre em contato com a equipe de desenvolvimento.

---

**Data de criação:** 19/11/2025

**Versão:** 1.0.0

**Status:**  Pronto para produção