**Interface de Mapeamento PGC-AO**

**Interface de Mapeamento PGC-AO (React)**

**Sistema de revisão do mapeamento automático com:**

* **Visualização de confiança do mapeamento (95%, 75%, etc.)**
* **Edição inline de contas PGC**
* **Status visual (confirmado, em revisão)**
* **Filtros por confiança e pesquisa**
* **Categorias customizadas para baixa confiança**

**VEJA:** [**Interface de Mapeamento PGC-AO | Claude | Claude**](https://claude.ai/public/artifacts/43daeaad-b92b-49a5-8744-b8329655023f)

**CODIGO:**

import React, { useState } from 'react';

import { Search, Edit2, Check, X, AlertTriangle, Info, ArrowRight, Filter, Download } from 'lucide-react';

const PGCMappingInterface = () => {

const [searchTerm, setSearchTerm] = useState('');

const [filterConfidence, setFilterConfidence] = useState('all');

const [editingId, setEditingId] = useState(null);

*// Dados simulados de mapeamento*

const [mappings, setMappings] = useState([

{

id: 1,

originalDescription: 'Venda de Milho',

suggestedPGC: '714',

pgcName: 'Vendas de Produtos Agrícolas',

amount: 350000,

confidence: 95,

customCategory: null,

status: 'confirmed'

},

{

id: 2,

originalDescription: 'Compra de Sementes de Milho',

suggestedPGC: '611',

pgcName: 'Matérias-Primas',

amount: 85000,

confidence: 90,

customCategory: null,

status: 'confirmed'

},

{

id: 3,

originalDescription: 'Serviço de Irrigação Automatizada',

suggestedPGC: '622',

pgcName: 'Serviços Especializados',

amount: 45000,

confidence: 75,

customCategory: 'Irrigação Técnica',

status: 'review'

},

{

id: 4,

originalDescription: 'Salários dos Trabalhadores Rurais',

suggestedPGC: '632',

pgcName: 'Remunerações do Pessoal',

amount: 180000,

confidence: 92,

customCategory: null,

status: 'confirmed'

},

{

id: 5,

originalDescription: 'Combustível para Máquinas Agrícolas',

suggestedPGC: '613',

pgcName: 'Combustíveis e Lubrificantes',

amount: 65000,

confidence: 88,

customCategory: null,

status: 'confirmed'

},

{

id: 6,

originalDescription: 'Consultoria Agronômica',

suggestedPGC: '622',

pgcName: 'Serviços Especializados',

amount: 25000,

confidence: 65,

customCategory: 'Consultoria Agrícola',

status: 'review'

},

{

id: 7,

originalDescription: 'Transporte de Produtos',

suggestedPGC: '625',

pgcName: 'Deslocações, Estadas e Transportes',

amount: 40000,

confidence: 85,

customCategory: null,

status: 'confirmed'

},

{

id: 8,

originalDescription: 'Energia Elétrica - Armazém',

suggestedPGC: '624',

pgcName: 'Energia e Fluidos',

amount: 30000,

confidence: 93,

customCategory: null,

status: 'confirmed'

}

]);

const pgcAccounts = [

{ code: '71', name: 'Vendas', type: 'revenue' },

{ code: '711', name: 'Vendas de Mercadorias', type: 'revenue' },

{ code: '712', name: 'Vendas de Produtos Acabados', type: 'revenue' },

{ code: '714', name: 'Vendas de Produtos Agrícolas', type: 'revenue' },

{ code: '715', name: 'Vendas de Produtos Pecuários', type: 'revenue' },

{ code: '72', name: 'Prestação de Serviços', type: 'revenue' },

{ code: '61', name: 'Custo das Mercadorias Vendidas', type: 'cost' },

{ code: '611', name: 'Matérias-Primas', type: 'cost' },

{ code: '612', name: 'Materiais Diversos', type: 'cost' },

{ code: '613', name: 'Combustíveis e Lubrificantes', type: 'cost' },

{ code: '62', name: 'Fornecimentos e Serviços Externos', type: 'cost' },

{ code: '621', name: 'Subcontratos', type: 'cost' },

{ code: '622', name: 'Serviços Especializados', type: 'cost' },

{ code: '624', name: 'Energia e Fluidos', type: 'cost' },

{ code: '625', name: 'Deslocações, Estadas e Transportes', type: 'cost' },

{ code: '63', name: 'Custos com o Pessoal', type: 'cost' },

{ code: '632', name: 'Remunerações do Pessoal', type: 'cost' },

{ code: '635', name: 'Encargos sobre Remunerações', type: 'cost' },

{ code: '64', name: 'Amortizações e Provisões', type: 'cost' },

{ code: '641', name: 'Amortizações do Exercício', type: 'cost' }

];

const getConfidenceColor = (confidence) => {

if (confidence >= 90) return 'text-green-600 bg-green-50';

if (confidence >= 75) return 'text-yellow-600 bg-yellow-50';

return 'text-red-600 bg-red-50';

};

const getStatusIcon = (status) => {

if (status === 'confirmed') return <Check className="w-4 h-4 text-green-600" />;

if (status === 'review') return <AlertTriangle className="w-4 h-4 text-yellow-600" />;

return <X className="w-4 h-4 text-red-600" />;

};

const filteredMappings = mappings.filter(mapping => {

const matchesSearch = mapping.originalDescription.toLowerCase().includes(searchTerm.toLowerCase()) ||

mapping.pgcName.toLowerCase().includes(searchTerm.toLowerCase());

const matchesConfidence = filterConfidence === 'all' ||

(filterConfidence === 'high' && mapping.confidence >= 90) ||

(filterConfidence === 'medium' && mapping.confidence >= 75 && mapping.confidence < 90) ||

(filterConfidence === 'low' && mapping.confidence < 75);

return matchesSearch && matchesConfidence;

});

const handleEdit = (id) => {

setEditingId(id);

};

const handleSave = (id, newPGC) => {

setMappings(prev => prev.map(mapping =>

mapping.id === id

? {

...mapping,

suggestedPGC: newPGC,

pgcName: pgcAccounts.find(acc => acc.code === newPGC)?.name || mapping.pgcName,

status: 'confirmed',

confidence: 100

}

: mapping

));

setEditingId(null);

};

const statistics = {

total: mappings.length,

confirmed: mappings.filter(m => m.status === 'confirmed').length,

review: mappings.filter(m => m.status === 'review').length,

avgConfidence: Math.round(mappings.reduce((sum, m) => sum + m.confidence, 0) / mappings.length)

};

return (

<div className="min-h-screen bg-gray-50 p-6">

<div className="max-w-7xl mx-auto">

{*/\* Header \*/*}

<div className="bg-white rounded-xl shadow-sm p-6 mb-6">

<div className="flex justify-between items-start">

<div>

<h1 className="text-2xl font-bold text-gray-900">Mapeamento PGC-AO</h1>

<p className="text-gray-600 mt-1">Revisão e ajuste do mapeamento automático para o Plano Geral de Contabilidade</p>

</div>

<button className="flex items-center gap-2 px-4 py-2 bg-indigo-600 text-white rounded-lg hover:bg-indigo-700 transition-colors">

<Download className="w-4 h-4" />

Exportar Mapeamento

</button>

</div>

{*/\* Statistics \*/*}

<div className="grid grid-cols-4 gap-4 mt-6">

<div className="bg-gray-50 rounded-lg p-4">

<p className="text-sm text-gray-600">Total de Itens</p>

<p className="text-2xl font-bold text-gray-900">{statistics.total}</p>

</div>

<div className="bg-green-50 rounded-lg p-4">

<p className="text-sm text-green-600">Confirmados</p>

<p className="text-2xl font-bold text-green-700">{statistics.confirmed}</p>

</div>

<div className="bg-yellow-50 rounded-lg p-4">

<p className="text-sm text-yellow-600">Em Revisão</p>

<p className="text-2xl font-bold text-yellow-700">{statistics.review}</p>

</div>

<div className="bg-indigo-50 rounded-lg p-4">

<p className="text-sm text-indigo-600">Confiança Média</p>

<p className="text-2xl font-bold text-indigo-700">{statistics.avgConfidence}%</p>

</div>

</div>

</div>

{*/\* Filters \*/*}

<div className="bg-white rounded-xl shadow-sm p-6 mb-6">

<div className="flex gap-4">

<div className="flex-1">

<div className="relative">

<Search className="absolute left-3 top-1/2 transform -translate-y-1/2 text-gray-400 w-5 h-5" />

<input

type="text"

placeholder="Pesquisar por descrição ou conta PGC..."

className="w-full pl-10 pr-4 py-2 border border-gray-300 rounded-lg focus:ring-2 focus:ring-indigo-500 focus:border-transparent"

value={searchTerm}

onChange={(e) => setSearchTerm(e.target.value)}

/>

</div>

</div>

<div className="flex items-center gap-2">

<Filter className="w-5 h-5 text-gray-500" />

<select

className="px-4 py-2 border border-gray-300 rounded-lg focus:ring-2 focus:ring-indigo-500"

value={filterConfidence}

onChange={(e) => setFilterConfidence(e.target.value)}

>

<option value="all">Todas as Confianças</option>

<option value="high">Alta (≥90%)</option>

<option value="medium">Média (75-89%)</option>

<option value="low">Baixa (&lt;75%)</option>

</select>

</div>

</div>

</div>

{*/\* Mapping Table \*/*}

<div className="bg-white rounded-xl shadow-sm overflow-hidden">

<table className="w-full">

<thead className="bg-gray-50 border-b">

<tr>

<th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

Status

</th>

<th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

Descrição Original

</th>

<th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

Mapeamento PGC

</th>

<th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

Valor (Kz)

</th>

<th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

Confiança

</th>

<th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

Categoria Customizada

</th>

<th className="px-6 py-3 text-left text-xs font-medium text-gray-500 uppercase tracking-wider">

Ações

</th>

</tr>

</thead>

<tbody className="divide-y divide-gray-200">

{filteredMappings.map((mapping) => (

<tr key={mapping.id} className="hover:bg-gray-50 transition-colors">

<td className="px-6 py-4 whitespace-nowrap">

{getStatusIcon(mapping.status)}

</td>

<td className="px-6 py-4">

<div className="text-sm font-medium text-gray-900">{mapping.originalDescription}</div>

</td>

<td className="px-6 py-4">

{editingId === mapping.id ? (

<div className="flex items-center gap-2">

<select

className="px-3 py-1 border border-gray-300 rounded text-sm"

defaultValue={mapping.suggestedPGC}

onChange={(e) => handleSave(mapping.id, e.target.value)}

>

{pgcAccounts.map(acc => (

<option key={acc.code} value={acc.code}>

{acc.code} - {acc.name}

</option>

))}

</select>

<button

onClick={() => setEditingId(null)}

className="p-1 text-gray-500 hover:text-gray-700"

>

<X className="w-4 h-4" />

</button>

</div>

) : (

<div className="flex items-center gap-2">

<span className="text-sm font-medium text-indigo-600">{mapping.suggestedPGC}</span>

<ArrowRight className="w-4 h-4 text-gray-400" />

<span className="text-sm text-gray-600">{mapping.pgcName}</span>

</div>

)}

</td>

<td className="px-6 py-4 whitespace-nowrap">

<span className="text-sm font-medium text-gray-900">

{mapping.amount.toLocaleString('pt-AO')}

</span>

</td>

<td className="px-6 py-4 whitespace-nowrap">

<span className={`px-2 py-1 text-xs font-medium rounded-full ${getConfidenceColor(mapping.confidence)}`}>

{mapping.confidence}%

</span>

</td>

<td className="px-6 py-4">

{mapping.customCategory && (

<span className="px-2 py-1 text-xs bg-purple-100 text-purple-700 rounded">

{mapping.customCategory}

</span>

)}

</td>

<td className="px-6 py-4">

{editingId !== mapping.id && (

<button

onClick={() => handleEdit(mapping.id)}

className="text-indigo-600 hover:text-indigo-900"

>

<Edit2 className="w-4 h-4" />

</button>

)}

</td>

</tr>

))}

</tbody>

</table>

</div>

{*/\* Info Box \*/*}

<div className="mt-6 bg-blue-50 border border-blue-200 rounded-xl p-4">

<div className="flex items-start gap-3">

<Info className="w-5 h-5 text-blue-600 mt-0.5" />

<div>

<p className="text-sm font-medium text-blue-900">Sobre o Mapeamento Automático</p>

<p className="text-sm text-blue-700 mt-1">

O sistema utiliza palavras-chave e padrões para sugerir automaticamente as contas PGC mais apropriadas.

Itens com confiança inferior a 75% são marcados para revisão manual. Você pode ajustar qualquer mapeamento

clicando no ícone de edição.

</p>

</div>

</div>

</div>

</div>

</div>

);

};

export default PGCMappingInterface;