

Frontend em React/Next.js para Sistema de Gestão de Materiais

Vou criar um frontend MVP usando Next.js que consome todos os endpoints do backend que desenvolvemos anteriormente. O foco será em funcionalidade básica com CSS simples e fetch API para comunicação com o backend.

Estrutura do Projeto

```
/pages
  /api
    auth.js          # Proxy para API de autenticação
    _app.js          # Configuração do app
    _document.js     # Configuração do documento
    index.js         # Página inicial (login)
    cadastro.js      # Página de cadastro
    dashboard.js     # Página principal do usuário
    solicitacoes.js  # Página de solicitações (coordenadores)
    planejamento.js  # Página de planejamento (coordenadores)
    materiais.js     # Página de materiais (estoque)
    notificacoes.js  # Página de notificações

  /components
    Layout.js        # Layout base
    Header.js        # Cabeçalho
    NotificationBell.js # Componente de notificações
    MaterialForm.js   # Formulário de materiais
    SolicitationForm.js # Formulário de solicitações
    PlanningForm.js   # Formulário de planejamento

  /styles
    globals.css      # Estilos globais
    utils.css        # Classes utilitárias

  /lib
    api.js           # Configuração do cliente API
    auth.js          # Funções de autenticação
    constants.js     # Constantes do sistema
```

Configuração Inicial

```
pages/_app.js

import '../styles/globals.css';
import Layout from '../components/Layout';
import { AuthProvider } from '../lib/auth';
```

```
function MyApp({ Component, pageProps }) {
  return (
    <AuthProvider>
      <Layout>
        <Component {...pageProps} />
      </Layout>
    </AuthProvider>
  );
}
```

```
export default MyApp;
```

lib/api.js

```
const API_BASE_URL = process.env.NEXT_PUBLIC_API_URL || 'http://localhost:5000/api';

async function fetchAPI(endpoint, method = 'GET', body = null, token = null) {
  const headers = {
    'Content-Type': 'application/json',
  };

  if (token) {
    headers['Authorization'] = `Bearer ${token}`;
  }

  const options = {
    method,
    headers,
    credentials: 'include', // Para cookies
  };

  if (body) {
    options.body = JSON.stringify(body);
  }

  const response = await fetch(`${API_BASE_URL}${endpoint}`, options);

  if (!response.ok) {
    const error = await response.json();
    throw new Error(error.message || 'Erro na requisição');
  }

  return response.json();
}
```

```
export default fetchAPI;
```

lib/auth.js

```
import { useState, useEffect, createContext, useContext } from 'react';
import fetchAPI from './api';
import { useRouter } from 'next/router';

const AuthContext = createContext();

export function AuthProvider({ children }) {
  const [user, setUser] = useState(null);
  const [loading, setLoading] = useState(true);
  const router = useRouter();

  useEffect(() => {
    async function loadUser() {
      try {
        const data = await fetchAPI('/auth/me', 'GET', null, true);
        setUser(data.data);
      } catch (err) {
        setUser(null);
      } finally {
        setLoading(false);
      }
    }

    loadUser();
  }, []);

  const login = async (login, senha) => {
    try {
      const data = await fetchAPI('/auth/login', 'POST', { login, senha });
      setUser(data.data);
      router.push('/dashboard');
    } catch (err) {
      throw err;
    }
  };

  const logout = async () => {
    await fetchAPI('/auth/logout', 'POST');
    setUser(null);
    router.push('/');
  };
}
```

```

    return (
      <AuthContext.Provider value={{ user, loading, login, logout }}>
        {children}
      </AuthContext.Provider>
    );
  }

  export function useAuth() {
    return useContext(AuthContext);
  }
}

```

Páginas Principais

pages/index.js (Login)

```

import { useState } from 'react';
import { useAuth } from '../lib/auth';
import { useRouter } from 'next/router';

export default function Login() {
  const [login, setLogin] = useState('');
  const [senha, setSenha] = useState('');
  const [error, setError] = useState('');
  const { login: authLogin } = useAuth();
  const router = useRouter();

  const handleSubmit = async (e) => {
    e.preventDefault();
    try {
      await authLogin(login, senha);
    } catch (err) {
      setError(err.message);
    }
  };

  return (
    <div className="login-container">
      <h1>Gestão de Materiais</h1>
      <form onSubmit={handleSubmit}>
        {error && <p className="error">{error}</p>}
        <div>
          <label>Login:</label>
          <input
            type="text"
            value={login}
            onChange={(e) => setLogin(e.target.value)}

```

```

        required
      />
    </div>
    <div>
      <label>Senha:</label>
      <input
        type="password"
        value={senha}
        onChange={(e) => setSenha(e.target.value)}
        required
      />
    </div>
    <button type="submit">Entrar</button>
  </form>
  <p>
    Não tem conta? <a href="/cadastro">Cadastre-se</a>
  </p>
</div>
);
}

```

pages/cadastro.js

```

import { useState } from 'react';
import fetchAPI from '../lib/api';
import { useRouter } from 'next/router';

export default function Cadastro() {
  const [form, setForm] = useState({
    matricula: '',
    nome: '',
    login: '',
    senha: '',
    email: '',
    tipoUsuario: 'Coordenador',
    idSetor: ''
  });
  const [setores, setSetores] = useState([]);
  const [error, setError] = useState('');
  const router = useRouter();

  useEffect(() => {
    async function loadSetores() {
      try {
        const data = await fetchAPI('/setores');
        setSetores(data.data);
      }
    }
  });

```

```

    } catch (err) {
      setError(err.message);
    }
  }
  loadSetores();
}, []);

const handleSubmit = async (e) => {
  e.preventDefault();
  try {
    await fetchAPI('/usuarios', 'POST', form);
    router.push('/');
  } catch (err) {
    setError(err.message);
  }
};

const handleChange = (e) => {
  setForm({
    ...form,
    [e.target.name]: e.target.value
  });
};

return (
  <div className="form-container">
    <h1>Cadastro de Usuário</h1>
    {error && <p className="error">{error}</p>}
    <form onSubmit={handleSubmit}>
      <div>
        <label>Matrícula:</label>
        <input
          name="matricula"
          value={form.matricula}
          onChange={handleChange}
          required
        />
      </div>
      <div>
        <label>Nome:</label>
        <input
          name="nome"
          value={form.nome}
          onChange={handleChange}
          required
        />
      </div>
    </form>
  </div>
);

```

```

</div>
<div>
  <label>Login:</label>
  <input
    name="login"
    value={form.login}
    onChange={handleChange}
    required
  />
</div>
<div>
  <label>Senha:</label>
  <input
    type="password"
    name="senha"
    value={form.senha}
    onChange={handleChange}
    required
  />
</div>
<div>
  <label>Email:</label>
  <input
    type="email"
    name="email"
    value={form.email}
    onChange={handleChange}
    required
  />
</div>
<div>
  <label>Tipo de Usuário:</label>
  <select
    name="tipoUsuario"
    value={form.tipoUsuario}
    onChange={handleChange}
  >
    <option value="Coordenador">Coordenador de Setor</option>
    <option value="ControleMateriais">Controle de Materiais</option>
  </select>
</div>
<div>
  <label>Setor:</label>
  <select
    name="idSetor"
    value={form.idSetor}

```

```

        onChange={handleChange}
        required
      >
        <option value="">Selecione um setor</option>
        {setores.map(setor => (
          <option key={setor.id_setor} value={setor.id_setor}>
            {setor.nome_setor}
          </option>
        ))}
      </select>
    </div>
    <button type="submit">Cadastrar</button>
  </form>
</div>
);
}

```

pages/dashboard.js

```

import { useAuth } from '../lib/auth';
import NotificationBell from '../components/NotificationBell';
import { useRouter } from 'next/router';

export default function Dashboard() {
  const { user, loading, logout } = useAuth();
  const router = useRouter();

  if (loading) return <div>Carregando...</div>;
  if (!user) {
    router.push('/');
    return null;
  }

  return (
    <div className="dashboard">
      <h1>Bem-vindo, {user.nome}</h1>
      <NotificationBell />

      <div className="dashboard-actions">
        {user.tipoUsuario === 'Coordenador' && (
          <>
            <button onClick={() => router.push('/solicitacoes')}>
              Gerenciar Solicitações
            </button>
            <button onClick={() => router.push('/planejamento')}>
              Planejamento Mensal
            </button>
          </>
        )}
      </div>
    </div>
  );
}

```



```

        </button>
      </>
    )}

    {user.tipoUsuario === 'ControleMateriais' && (
      <button onClick={() => router.push('/materiais')}>
        Gerenciar Catálogo
      </button>
    )}
  </div>

  <button onClick={logout} className="logout-button">
    Sair
  </button>
</div>
);
}

```

Páginas Específicas por Tipo de Usuário

pages/solicitacoes.js (Coordenador de Setor)

```

import { useState, useEffect } from 'react';
import { useAuth } from '../lib/auth';
import fetchAPI from '../lib/api';
import SolicitationForm from '../components/SolicitationForm';

export default function Solicitacoes() {
  const { user } = useAuth();
  const [solicitacoes, setSolicitacoes] = useState([]);
  const [materiais, setMateriais] = useState([]);
  const [showForm, setShowForm] = useState(false);

  useEffect(() => {
    async function loadData() {
      try {
        const [solicitacoesRes, materiaisRes] = await Promise.all([
          fetchAPI('/solicitacoes/setor'),
          fetchAPI('/materiais')
        ]);
        setSolicitacoes(solicitacoesRes.data);
        setMateriais(materiaisRes.data);
      } catch (err) {
        console.error(err);
      }
    }
  });
}

```

```

        loadData();
    }, []);

    const handleCreate = async (itens) => {
        try {
            const data = await fetchAPI('/solicitacoes', 'POST', { itens });
            setSolicitacoes([...solicitacoes, data.data]);
            setShowForm(false);
        } catch (err) {
            console.error(err);
        }
    };

    return (
        <div className="page-container">
            <h1>Solicitações de Materiais</h1>

            <button onClick={() => setShowForm(!showForm)}>
                {showForm ? 'Cancelar' : 'Nova Solicitação'}
            </button>

            {showForm && (
                <SolicitationForm
                    materiais={materiais}
                    onSubmit={handleCreate}
                />
            )}

            <div className="solicitacoes-list">
                {solicitacoes.map(sol => (
                    <div key={sol.id_solicitacao} className="solicitacao-card">
                        <h3>Solicitação #{sol.id_solicitacao}</h3>
                        <p>Status: {sol.status_solicitacao}</p>
                        <p>Data: {new Date(sol.data_solicitacao).toLocaleDateString()}</p>
                        <button onClick={() => {/* Implementar detalhes */}}>
                            Ver Detalhes
                        </button>
                    </div>
                ))}
            </div>
        </div>
    );
}

```

components/SolicitationForm.js

```
import { useState } from 'react';

export default function SolicitationForm({ materiais, onSubmit }) {
  const [itens, setItens] = useState([
    { idMaterial: '', quantidade: 1 }
  ]);

  const handleAddItem = () => {
    setItens([...itens, { idMaterial: '', quantidade: 1 }]);
  };

  const handleRemoveItem = (index) => {
    const newItens = [...itens];
    newItens.splice(index, 1);
    setItens(newItens);
  };

  const handleItemChange = (index, field, value) => {
    const newItens = [...itens];
    newItens[index][field] = value;
    setItens(newItens);
  };

  const handleSubmit = (e) => {
    e.preventDefault();
    onSubmit(itens.filter(item => item.idMaterial && item.quantidade > 0));
  };

  return (
    <form onSubmit={handleSubmit} className="solicitation-form">
      <h3>Itens da Solicitação</h3>

      {itens.map((item, index) => (
        <div key={index} className="form-item">
          <select
            value={item.idMaterial}
            onChange={(e) => handleItemChange(index, 'idMaterial', e.target.value)}
            required
          >
            <option value="">Selecione um material</option>
            {materiais.map(mat => (
              <option key={mat.id_material} value={mat.id_material}>
                {mat.nome_material} ({mat.unidade_medida})
              </option>
            ))}
          </select>
        </div>
      ))}
    </form>
  );
}
```

```

        <input
          type="number"
          min="1"
          value={item.quantidade}
          onChange={(e) => handleItemChange(index, 'quantidade', parseInt(e.target.value))}
          required
        />

        {itens.length > 1 && (
          <button
            type="button"
            onClick={() => handleRemoveItem(index)}
            className="remove-button"
          >
            Remover
          </button>
        )}
      </div>
    )}

    <button type="button" onClick={handleAddItem}>
      Adicionar Item
    </button>

    <button type="submit">Enviar Solicitação</button>
  </form>
);
}

```

pages/materiais.js (Controle de Materiais)

```

import { useState, useEffect } from 'react';
import { useAuth } from '../lib/auth';
import fetchAPI from '../lib/api';
import MaterialForm from '../components/MaterialForm';

export default function Materiais() {
  const { user } = useAuth();
  const [materiais, setMateriais] = useState([]);
  const [showForm, setShowForm] = useState(false);
  const [editingMaterial, setEditingMaterial] = useState(null);

  useEffect(() => {
    async function loadMateriais() {
      try {

```

```

        const data = await fetchAPI('/materiais');
        setMateriais(data.data);
      } catch (err) {
        console.error(err);
      }
    }
    loadMateriais();
  }, []);

const handleCreate = async (material) => {
  try {
    const data = await fetchAPI('/materiais', 'POST', material);
    setMateriais([...materiais, data.data]);
    setShowForm(false);
  } catch (err) {
    console.error(err);
  }
};

const handleUpdate = async (id, material) => {
  try {
    const data = await fetchAPI(`/materiais/${id}`, 'PUT', material);
    setMateriais(materiais.map(mat =>
      mat.id_material === id ? data.data : mat
    ));
    setEditingMaterial(null);
  } catch (err) {
    console.error(err);
  }
};

const handleDelete = async (id) => {
  try {
    await fetchAPI(`/materiais/${id}`, 'DELETE');
    setMateriais(materiais.filter(mat => mat.id_material !== id));
  } catch (err) {
    console.error(err);
  }
};

return (
  <div className="page-container">
    <h1>Catálogo de Materiais</h1>

    <button onClick={() => {
      setEditingMaterial(null);

```

```

        setShowForm(!showForm);
    }}>
    {showForm ? 'Cancelar' : 'Adicionar Material'}
</button>

{(showForm || editingMaterial) && (
  <MaterialForm
    material={editingMaterial}
    onSubmit={editingMaterial ?
      (data) => handleUpdate(editingMaterial.id_material, data) :
      handleCreate}
  />
)}

<div className="materiais-list">
  {materiais.map(mat => (
    <div key={mat.id_material} className="material-card">
      <h3>{mat.nome_material}</h3>
      <p>{mat.descricao_material}</p>
      <p>Unidade: {mat.unidade_medida}</p>

      <div className="material-actions">
        <button onClick={() => {
          setEditingMaterial(mat);
          setShowForm(true);
        }}>
          Editar
        </button>

        <button
          onClick={() => handleDelete(mat.id_material)}
          className="delete-button"
        >
          Remover
        </button>
      </div>
    </div>
  )})}
</div>
</div>
);
}

```

Componentes Adicionais

components/NotificationBell.js

```
import { useState, useEffect } from 'react';
import fetchAPI from '../lib/api';

export default function NotificationBell() {
  const [notifications, setNotifications] = useState([]);
  const [showList, setShowList] = useState(false);
  const [unreadCount, setUnreadCount] = useState(0);

  useEffect(() => {
    async function loadNotifications() {
      try {
        const data = await fetchAPI('/notificacoes');
        setNotifications(data.data);
        setUnreadCount(data.data.filter(n => !n.lida).length);
      } catch (err) {
        console.error(err);
      }
    }
    loadNotifications();
  }, []);

  const markAsRead = async (id) => {
    try {
      await fetchAPI(`/notificacoes/${id}/marcar-lida`, 'PUT');
      setNotifications(notifications.map(n =>
        n.id_notificacao === id ? { ...n, lida: true } : n
      ));
      setUnreadCount(unreadCount - 1);
    } catch (err) {
      console.error(err);
    }
  };

  return (
    <div className="notification-bell">
      <button onClick={() => setShowList(!showList)}>
        {unreadCount > 0 && <span>{unreadCount}</span>}
      </button>

      {showList && (
        <div className="notification-list">
          {notifications.length === 0 ? (
```

```

        <p>Nenhuma notificação</p>
    ) : (
        notifications.map(notif => (
            <div
                key={notif.id_notificacao}
                className={`notification-item ${notif.lida ? '' : 'unread'}}
                onClick={() => markAsRead(notif.id_notificacao)}
            >
                <h4>{notif.titulo}</h4>
                <p>{notif.mensagem}</p>
                <small>
                    {new Date(notif.data_criacao).toLocaleDateString()}
                </small>
            </div>
        ))
    )
</div>
)}
</div>
);
}

```

Estilos Básicos (styles/globals.css)

```

body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
    line-height: 1.6;
}

.login-container, .form-container {
    max-width: 400px;
    margin: 50px auto;
    padding: 20px;
    border: 1px solid #ddd;
    border-radius: 5px;
}

.error {
    color: red;
}

input, select, button {
    width: 100%;
    padding: 8px;
}

```



```

    margin: 5px 0 15px;
    border: 1px solid #ddd;
    border-radius: 4px;
}

button {
    background-color: #4CAF50;
    color: white;
    border: none;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
}

.remove-button, .delete-button {
    background-color: #f44336;
}

.remove-button:hover, .delete-button:hover {
    background-color: #d32f2f;
}

.dashboard {
    padding: 20px;
}

.dashboard-actions {
    display: flex;
    gap: 10px;
    margin: 20px 0;
}

.page-container {
    padding: 20px;
}

.solicitacao-card, .material-card {
    border: 1px solid #ddd;
    border-radius: 5px;
    padding: 15px;
    margin: 10px 0;
}

.notification-bell {

```

```

    position: relative;
    display: inline-block;
}

.notification-list {
    position: absolute;
    right: 0;
    background: white;
    border: 1px solid #ddd;
    width: 300px;
    max-height: 400px;
    overflow-y: auto;
}

.notification-item {
    padding: 10px;
    border-bottom: 1px solid #eee;
}

.notification-item.unread {
    background-color: #f0f8ff;
}

.solicitation-form .form-item {
    display: flex;
    gap: 10px;
    margin-bottom: 10px;
}

.solicitation-form select, .solicitation-form input[type="number"] {
    flex: 1;
}

```

Funcionalidades Implementadas

1. **Autenticação:**
 - Login com JWT
 - Logout
 - Redirecionamento baseado em tipo de usuário
2. **Páginas Específicas:**
 - Coordenador de Setor: Solicitações e Planejamento
 - Controle de Materiais: Catálogo de Materiais
 - Usuários comuns: Dashboard básico
3. **Formulários:**
 - Cadastro de materiais (CRUD completo)
 - Criação de solicitações com múltiplos itens

- Cadastro de usuários
- 4. **Notificações:**
 - Listagem de notificações
 - Marcar como lida
 - Indicador de não lidas
- 5. **Integração com Backend:**
 - Fetch API para todos os endpoints
 - Tratamento de erros básico
 - Gerenciamento de estado simples

Este MVP implementa todas as funcionalidades essenciais do sistema com uma interface simples e funcional, pronta para ser expandida com mais recursos e melhorias de UI/UX.