# INSTRUÇÕES RÁPIDAS

Este documento oferece um guia conciso para a instalação e uso do sistema Comercial Alfa. Siga os passos abaixo para colocar o sistema em funcionamento rapidamente.

## 1. Instalação do Banco de Dados

Para começar, configure o banco de dados:

* Instale o MySQL 8+ em seu ambiente.
* Execute o script inicial para criar o banco de dados e popular com dados essenciais:

mysql -u root -p < database/create\_database.sql

## 2. Backend (API)

O backend é a API que alimenta o sistema. Siga estas instruções para iniciá-lo:

* Navegue até o diretório backend:

cd backend

* Instale as dependências do projeto:

npm install

* Inicie o servidor da API:

npm start

Após a inicialização, a API estará acessível em: http://localhost:3000.

## 3. Frontend

O frontend é a interface do usuário. Você pode acessá-lo de duas maneiras:

* Abra diretamente o arquivo index.html em seu navegador.
* Ou, para uma experiência mais completa, inicie um servidor local. Você pode usar Python ou npx http-server:

python -m http.server 8000

ou

npx http-server . -p 8000

O frontend estará disponível em: http://localhost:8000.

## 4. Fluxo Básico de Uso

Com o sistema configurado, você pode começar a utilizá-lo seguindo este fluxo:

1. Cadastre seus fornecedores e produtos no sistema.
2. Crie pedidos de compra para reabastecer seu estoque (o estoque será atualizado automaticamente).
3. Consulte seus pedidos e verifique o status do estoque a qualquer momento.

## 5. Observações Importantes

Alguns pontos cruciais sobre o sistema:

* O sistema utiliza codificação UTF-8, garantindo o correto tratamento de acentos e caracteres especiais.
* Todos os filtros e buscas são processados diretamente pela API, otimizando o desempenho.
* A interface é responsiva, adaptando-se a diferentes tamanhos de tela, e possui um tema escuro moderno com botões intuitivos.

## 6. Solução de Problemas Comuns

Caso encontre alguma dificuldade, consulte estas dicas rápidas:

* **Encoding errado**: Se houver problemas com caracteres, recrie o banco de dados utilizando o script fornecido.
* **Backend não inicia**: Verifique se todas as dependências foram instaladas corretamente e se a porta 3000 está livre.
* **Frontend não carrega dados**: Certifique-se de que o backend esteja em execução antes de acessar o frontend.

Para informações mais detalhadas, consulte a documentação completa localizada no diretório /docs.

## 📋 Funcionalidades Principais

O sistema oferece as seguintes funcionalidades essenciais:

### Gestão de Fornecedores

* **Cadastrar**: Adicione novos fornecedores com todas as informações necessárias.
* **Listar**: Visualize todos os fornecedores em uma tabela organizada.
* **Filtrar**: Pesquise fornecedores por estado utilizando um menu suspenso.
* **Editar**: Modifique os dados de fornecedores existentes.
* **Excluir**: Remova fornecedores (exclusão lógica, marcando como inativo).

### Gestão de Produtos

* **Cadastrar**: Registre produtos com categoria, preço e controle de estoque.
* **Listar**: Visualize produtos em uma tabela com filtros por categoria.
* **Editar**: Atualize as informações dos produtos.
* **Excluir**: Remova produtos (exclusão lógica).
* **Estoque**: O estoque é atualizado automaticamente ao criar novos pedidos de compra.

### Gestão de Pedidos

* **Criar**: Elabore pedidos de compra com múltiplos itens.
* **Listar**: Visualize todos os pedidos realizados.
* **Detalhes**: Acesse informações detalhadas de cada pedido, incluindo os itens.
* **Consultar**: Filtre pedidos por fornecedor e período específico.

## 🔧 Estrutura do Projeto

Para uma melhor compreensão da organização do projeto, observe a estrutura de diretórios:

projeto\_reorganizado/  
├── backend/ # Contém a API Node.js  
│ ├── src/  
│ │ ├── controllers/ # Lógica de controle  
│ │ ├── models/ # Modelos de dados do banco  
│ │ ├── routes/ # Definição das rotas da API  
│ │ └── config/ # Arquivos de configuração  
│ └── package.json  
├── database/ # Scripts SQL para o banco de dados  
│ └── create\_database.sql # Script completo de criação do banco  
├── pages/ # Páginas HTML do frontend  
├── js/ # Arquivos JavaScript do frontend  
├── css/ # Arquivos de estilo CSS  
└── index.html # Página principal do sistema

## 🌐 Endpoints da API

A API expõe os seguintes endpoints para interação com o sistema:

### Fornecedores

* GET /api/fornecedores: Lista todos os fornecedores.
* GET /api/fornecedores/:id: Busca um fornecedor específico por ID.
* POST /api/fornecedores: Cria um novo fornecedor.
* PUT /api/fornecedores/:id: Atualiza os dados de um fornecedor existente.
* DELETE /api/fornecedores/:id: Exclui um fornecedor (exclusão lógica).

### Produtos

* GET /api/produtos: Lista todos os produtos.
* GET /api/produtos/:id: Busca um produto específico por ID.
* POST /api/produtos: Cria um novo produto.
* PUT /api/produtos/:id: Atualiza os dados de um produto existente.
* DELETE /api/produtos/:id: Exclui um produto (exclusão lógica).

### Categorias

* GET /api/categorias: Lista todas as categorias de produtos.

### Pedidos

* GET /api/pedidos: Lista todos os pedidos.
* GET /api/pedidos/:id: Busca um pedido específico por ID.
* POST /api/pedidos: Cria um novo pedido.
* DELETE /api/pedidos/:id: Exclui um pedido.

## 🗄️ Banco de Dados

Detalhes sobre a estrutura do banco de dados:

### Tabelas Principais

* fornecedores: Armazena os dados dos fornecedores.
* categorias: Contém as categorias de produtos.
* produtos: Guarda o cadastro de todos os produtos.
* pedidos: Armazena o cabeçalho dos pedidos de compra.
* itens\_pedido: Detalha os itens de cada pedido.

### Configuração

* **Charset**: utf8mb4
* **Collation**: utf8mb4\_unicode\_ci
* **Engine**: InnoDB

## 🔍 Funcionalidades Especiais

O sistema inclui recursos avançados para otimizar a gestão:

### Filtros Dinâmicos

* **Fornecedores**: Filtragem por estado via API.
* **Produtos**: Filtragem por categoria via API.
* **Pedidos**: Filtragem por fornecedor e período.

### Validações

* **CNPJ único**: Garante que cada fornecedor tenha um CNPJ exclusivo.
* **Código único**: Assegura que cada produto tenha um código exclusivo.
* **Estoque mínimo**: Permite configurar um nível mínimo de estoque para produtos.
* **Preços**: Formatação de preços com duas casas decimais.

### Relatórios

* **View vw\_pedidos\_detalhes**: Uma view otimizada para consultas complexas e geração de relatórios.
* **Dados agregados**: Fornece dados consolidados de pedidos e itens para análise.

## 🛠️ Manutenção

Orientações para a manutenção do sistema:

### Reiniciar Backend

Para reiniciar o backend, siga estes comandos:

cd backend  
npm start

### Limpar Cache do Navegador

Se o frontend não estiver atualizando, tente limpar o cache do navegador (Ctrl+F5).

### Verificar Logs

Monitore o sistema através dos logs:

* **Console do navegador**: Acesse via F12 para verificar erros no frontend.
* **Terminal do backend**: Observe o terminal onde o backend está em execução para logs da API.

## 📝 Notas Importantes

Considerações adicionais sobre o desenvolvimento e comportamento do sistema:

### Encoding

* O sistema é configurado para trabalhar com UTF-8, garantindo compatibilidade.
* Suporte completo a caracteres especiais, como acentos e cedilha.
* Os dados no banco já foram corrigidos para este padrão.

### Performance

* Índices foram criados nas tabelas para otimizar consultas frequentes.
* A view de relatórios foi otimizada para garantir rapidez na recuperação de dados.
* Os filtros são processados via API, evitando sobrecarga no lado do cliente.

### Segurança

* Validação de dados implementada tanto no frontend quanto no backend.
* Sanitização de inputs para prevenir ataques de injeção.
* Uso de prepared statements no banco de dados para maior segurança.

## 🆘 Solução de Problemas

Um guia detalhado para resolver problemas comuns:

### Erro de Conexão com o Banco de Dados

1. Verifique se o serviço MySQL está em execução.
2. Confirme as credenciais de acesso no arquivo backend/src/config/database.js.
3. Teste a conexão diretamente pelo terminal: mysql -u root -p.

### Problemas com Caracteres Especiais

1. Verifique o charset do banco de dados: SHOW CREATE DATABASE comercial\_alfa;.
2. Confirme a collation das tabelas para utf8mb4\_unicode\_ci.
3. Reinicie o backend após realizar quaisquer correções.

### Frontend Não Carrega

1. Assegure-se de que o backend esteja rodando na porta 3000.
2. Abra o console do navegador (F12) para identificar erros de JavaScript ou rede.
3. Verifique se há erros de CORS ou problemas de conexão com a API.

## 📞 Suporte

Para assistência adicional:

* Consulte os logs do console para mensagens de erro.
* Acesse a documentação completa no diretório docs/.
* Revise a estrutura do projeto em ESTRUTURA.md para entender a organização dos arquivos.

Esperamos que este guia rápido seja útil para você! Em caso de dúvidas, não hesite em consultar a documentação completa.