

Sistema de Aprovação de Clientes Atacadistas

1. INTRODUÇÃO

Este documento especifica os requisitos funcionais e técnicos para implementação do sistema de aprovação de clientes atacadistas, incluindo modificações no back-end (Express + Prisma) e front-end (Next.js).

2. OBJETIVO

Desenvolver funcionalidades que permitam aos clientes solicitarem acesso a preços atacadistas e aos administradores gerenciarem essas solicitações através de um painel de controle dedicado.

3. REQUISITOS FUNCIONAIS

3.1 Funcionalidades do Cliente

- **RF-01:** O cliente deve poder solicitar acesso como atacadista durante o registro
- **RF-02:** O cliente deve poder solicitar acesso atacadista através de sua página de perfil
- **RF-03:** O sistema deve registrar automaticamente a data da solicitação
- **RF-04:** O cliente deve ter acesso a preços diferenciados após aprovação

3.2 Funcionalidades do Administrador

- **RF-05:** O administrador deve visualizar lista de solicitações pendentes
- **RF-06:** O administrador deve poder aprovar solicitações de atacado
- **RF-07:** O administrador deve poder rejeitar solicitações de atacado
- **RF-08:** O sistema deve registrar data e responsável pela aprovação/rejeição
- **RF-09:** O administrador deve acessar painel através de autenticação específica

4. REQUISITOS TÉCNICOS

4.1 Banco de Dados (Prisma Schema)

4.1.1 Modificações na Tabela Cliente

Novos campos a serem adicionados:

Campo	Tipo	Descrição	Padrão
solicitouAtacado	Boolean	Indica se o cliente solicitou acesso atacadista	false
statusAtacado	String	Status da solicitação	"pendente"
dataSolicitacaoAtacado	DateTime	Data da solicitação	null
dataAprovacaoAtacado	DateTime	Data da aprovação/rejeição	null
aprovadoPorAdminId	Int	ID do administrador responsável	null

4.1.2 Estados do Status Atacado

- **"pendente"**: Solicitação aguardando análise
- **"aprovado"**: Cliente aprovado para preços atacadistas
- **"rejeitado"**: Solicitação rejeitada

4.2 Back-end (Express + Prisma)

4.2.1 Endpoints da API

Endpoints Administrativos

- **GET** /admin/wholesale-requests
 - **Função**: Listar solicitações pendentes de atacado
 - **Autenticação**: Requerida (administrador)
 - **Resposta**: Lista de clientes com solicitações pendentes
- **PATCH** /admin/wholesale-requests/:clientId
 - **Função**: Aprovar ou rejeitar solicitação
 - **Parâmetros**: clientId, status ("aprovado" ou "rejeitado")
 - **Autenticação**: Requerida (administrador)
 - **Resposta**: Confirmação da operação

4.2.2 Middleware de Segurança

- **isAdmin**: Middleware para verificação de permissões administrativas
- Validação de token de acesso administrativo
- Controle de acesso baseado em roles

4.3 Front-end (Next.js)

4.3.1 Páginas a Desenvolver

Página de Aprovações Atacadistas

- **Rota:** /admin/wholesale-approvals
- **Funcionalidades:**
 - Listagem de solicitações pendentes
 - Botões de aprovação/rejeição
 - Filtros e ordenação por data
 - Feedback visual das ações

Interface do Cliente

- **Registro:** Checkbox para solicitar acesso atacadista
- **Perfil:** Opção para solicitar acesso (se não solicitado anteriormente)
- **Status:** Visualização do status da solicitação

4.3.2 Componentes de Interface

Tabela de Solicitações

- Colunas: Nome, Email, CPF, Data Solicitação, Ações
- Ações: Botões "Aprovar" e "Rejeitar"
- Responsividade para dispositivos móveis

Notificações

- Toast notifications para feedback das ações
- Confirmações de sucesso/erro
- Integração com ToastContext existente

5. ARQUITETURA DO SISTEMA

5.1 Fluxo de Dados

Processo de Solicitação

1. Cliente solicita acesso atacadista (registro ou perfil)
2. Sistema atualiza campos: solicitouAtacado = true, statusAtacado = 'pendente'
3. Data da solicitação é registrada automaticamente

Processo de Aprovação

1. Administrador acessa painel de aprovações
2. Sistema busca solicitações com status "pendente"

3. Administrador aprova/rejeita solicitações
4. Sistema atualiza status e registra data da decisão

5.2 Segurança

Autenticação

- Tokens JWT para autenticação de administradores
- Middleware de verificação de permissões
- Proteção de rotas administrativas

Autorização

- Controle de acesso baseado em níveis (admin/cliente)
 - Validação de permissões em endpoints sensíveis
 - Proteção contra acesso não autorizado
-

6. IMPLEMENTAÇÃO

6.1 Etapas de Desenvolvimento

Fase 1: Back-end

1. Atualizar schema do Prisma
2. Executar migração de banco de dados
3. Implementar endpoints administrativos
4. Desenvolver middleware de autenticação
5. Testes das APIs

Fase 2: Front-end

1. Criar página de aprovações administrativas
2. Implementar interface de solicitação para clientes
3. Desenvolver componentes de UI
4. Integrar com APIs do back-end
5. Testes de interface

Fase 3: Integração

1. Testes end-to-end
2. Validação de fluxos completos
3. Otimização de performance
4. Documentação final

6.2 Comandos de Migração

```
cd server
```

```
npx prisma migrate dev --name add_wholesale_status_to_cliente
```

7. CONSIDERAÇÕES TÉCNICAS

7.1 Performance

- Indexação adequada para consultas de status
- Paginação para listas grandes de solicitações
- Cache de dados administrativos

7.2 Manutenibilidade

- Código modular e reutilizável
- Documentação de APIs
- Padrões de nomenclatura consistentes

7.3 Escalabilidade

- Estrutura preparada para múltiplos níveis de administrador
 - Sistema de notificações expansível
 - Arquitetura modular para futuras funcionalidades
-

8. VALIDAÇÃO E TESTES

8.1 Testes Funcionais

- Solicitação de acesso atacadista por clientes
- Aprovação/rejeição por administradores
- Verificação de permissões de acesso
- Validação de preços diferenciados

8.2 Testes de Segurança

- Verificação de autenticação administrativa
- Teste de autorização de endpoints
- Validação de tokens e sessões
- Prevenção de acesso não autorizado

8.3 Testes de Interface

- Responsividade em diferentes dispositivos
- Usabilidade do painel administrativo
- Feedback visual das ações
- Navegação e fluxos de usuário

9. CONCLUSÃO

Este documento estabelece os requisitos completos para implementação do sistema de aprovação de clientes atacadistas, garantindo funcionalidades robustas tanto para clientes quanto para administradores, com foco em segurança, usabilidade e escalabilidade.