

Documentação do Projeto "Cabeleleila Leila"

Visão Geral

O "Cabeleleila Leila" é um sistema web desenvolvido para gerenciar agendamentos de serviços de salão de beleza, com interfaces específicas para clientes e administradores. Construído em Next.js com TypeScript e estilizado com Tailwind CSS, o projeto foi concluído em 4 dias, priorizando modularidade, reutilização de código, e uma experiência de usuário consistente com as cores rose e pink.

Objetivo

Permitir que clientes agendem serviços, visualizem seus agendamentos, editem e cancele dentro do prazo.

Oferecer aos administradores um painel com estatísticas de agendamentos (total, concluídos, pendentes, cancelados) e valor total, filtrados por dia, semana, mês, ou todos.

Tecnologias Utilizadas

Frontend: Next.js (App Router), TypeScript, Tailwind CSS..

Gerenciamento de Estado: React Context (AuthProvider, AppointmentsProvider)

API: Chamadas assíncronas via lib/api.ts ao backend em <http://localhost:8080/api>.

Funcionalidades

Autenticação

Login e Registro:

Usuários (clientes e admins) fazem login ou registro via `/auth/login` e `/auth/register/client`.

Dados salvos no `localStorage` e cookies (token, role, email, id, name).

Proteção de Rotas:

Middleware (`middleware.ts`) verifica o token no header `Authorization` para rotas `/dashboard` e `/admin`.

Verificação adicional no cliente via `useAuth` redireciona usuários não autenticados.

Painel do Cliente (`/(client)/dashboard`)

Agendamento:

Formulário em `AppointmentForm.tsx` para escolher serviço e data, com sugestão inteligente (modal) se houver conflito na mesma semana.

Feedback de sucesso ("Agendamento concluído!") antes de trocar para "Meus Agendamentos".

Meus Agendamentos:

Lista em `AppointmentsList.tsx` mostrando agendamentos com filtros (dia, semana, mês, todos).

Edição e cancelamento via `AppointmentCard.tsx` (dentro do prazo de 2 dias).

Painel do Admin (/admin/dashboard)

Estatísticas:

Total, concluídos, pendentes e cancelados exibidos em cards

Valor Total:

Calculado para agendamentos concluídos (appt.service.price),
formatado em reais (ex.: R\$150,00).

Filtros:

Dia, semana, mês, todos, aplicados às estatísticas (sem renderizar
cards).

Detalhes Técnicos

Autenticação

AuthProvider: Gerencia estado de usuário e token, salvando em
localStorage e cookies com expiração de 6 horas
(TOKEN_EXPIRATION).

Middleware: Verifica Authorization no servidor, mas depende de
proteção no cliente para requisições iniciais.

Agendamentos

AppointmentsProvider: Centraliza o estado appointments e ações
(createAppointment, cancelAppointment, editAppointment).

API: Integração com backend em /auth/login, /client/appointments,
/admin/appointments, etc.

Lógica baseada em `startOfWeek`, `endOfWeek`, `startOfMonth`, e `endOfMonth`, usando `toISOString().split("T")[0]` para comparações em UTC.

Reutilização: Aplicado em `AppointmentsList` para listar cards e em `AdminDashboard` para estatísticas.

Fluxo do Sistema

Cliente

Login/Registro: Usuário acessa `/login` ou `/register`, autentica e é redirecionado para `/dashboard`.

Agendamento: Preenche serviço e data, confirma sugestão (se aplicável), vê feedback e troca para lista.

Gestão: Visualiza, edita ou cancela agendamentos em "Meus Agendamentos" (dentro do prazo).

Admin

Login: Admin acessa `/login` e é redirecionado para `/admin/dashboard`.

Painel: Veja estatísticas e valor total, filtrados por dia, semana, ou mês.

Testes

Cliente

Agendamento: Teste com data na mesma semana → Modal aparece → Confirme/Mantenha → Lista atualiza.

Filtros: Teste "Dia", "Semana", "Mês", "Todos" → Veja os cards corretos.

Edição/Cancelamento: Edite/cancele dentro do prazo → Lista reflete mudanças.

Admin

Estatísticas: Teste filtros → Confirme total, concluídos, pendentes, cancelados e valor total.

Melhorias Futuras:

Proteger rotas iniciais com cookies HTTP-only no backend.

Adicionar mais estatísticas (ex.: por serviço) ou ações admin (ex.: alterar status).