

Relatório Sistema ‘RéServè’

Alunos: Thiago Ribeiro Aragão; Vinícius Toledo do Prado Paes.

Emails: { thiagoaragao; vinicius_toledo }@id.uff.br

Orientador: prof. Dr. Leandro Santiago de Araujo.

Turma: Laboratório de Programação de Dispositivos Móveis (TCC00300 - AA)



2024.2

1. INTRODUÇÃO

O aplicativo 'RéServè' é um sistema de reservas, desenvolvido em Flutter e projetado para atender dois tipos de usuários: anfitriões, que gerenciam propriedades, e hóspedes, que buscam e reservam acomodações. A aplicação segue uma estrutura de dois fluxos, mostrados na tela inicial, e um banco de dados central único para ambos fluxos. Dessa forma, buscamos atender a ambos tipos de usuários, sem a necessidade de realizar comunicação por internet entre duas aplicações. Essa organização foi combinada previamente com o professor. A seguir, explicamos a organização do projeto em três tópicos principais: fluxo do anfitrião, fluxo do hóspede e a estrutura do banco de dados.

2. RÉSERVÈ

2.1. FLUXO DO ANFITRIÃO

No fluxo do anfitrião, o sistema oferece funcionalidades específicas para os usuários que desejam gerenciar propriedades. O diretório *lib/host/* concentra as telas e widgets relacionados a esse fluxo. A tela inicial para o anfitrião é a *host_auth_screen.dart*, onde ele pode realizar login ou se registrar no sistema. Após a autenticação, o usuário é redirecionado para a tela principal, implementada em *host_home_screen.dart*, onde pode visualizar todas as propriedades cadastradas. Cada propriedade é exibida de forma resumida utilizando o widget personalizado *PropertyItem*, localizado em *lib/host/widgets/property_item.dart*. Para adicionar ou editar uma propriedade, o anfitrião utiliza a tela *property_form_screen.dart*, que permite preencher informações detalhadas como título, descrição, preço e capacidade máxima de hóspedes. Além disso, essa tela integra funcionalidades como busca automática de endereços via API ViaCEP e upload de imagens associadas às propriedades.

2.2. FLUXO DO HÓSPEDE

No fluxo do hóspede, o sistema foca em facilitar a busca por acomodações e a realização de reservas. O diretório *lib/guest/* organiza as telas e widgets desse fluxo. A tela inicial para o hóspede é a *guest_auth_screen.dart*, onde ele pode se autenticar, registrar ou entrar como convidado. Após essa etapa, o hóspede é direcionado para a tela de busca (*search_screen.dart*), onde pode filtrar propriedades com base em localização (estado, cidade e bairro), número de hóspedes e intervalo de datas. Os resultados da busca são exibidos em uma lista, permitindo que o hóspede selecione uma propriedade específica para visualizar seus detalhes na tela *property_detail_screen.dart*. Essa tela exibe informações completas sobre a propriedade, como descrição, preço por diária e imagens, além de permitir que o hóspede realize reservas após verificar a disponibilidade nas datas selecionadas. As reservas realizadas podem ser visualizadas na tela *bookings_screen.dart*, que exibe uma lista com todas as reservas feitas pelo

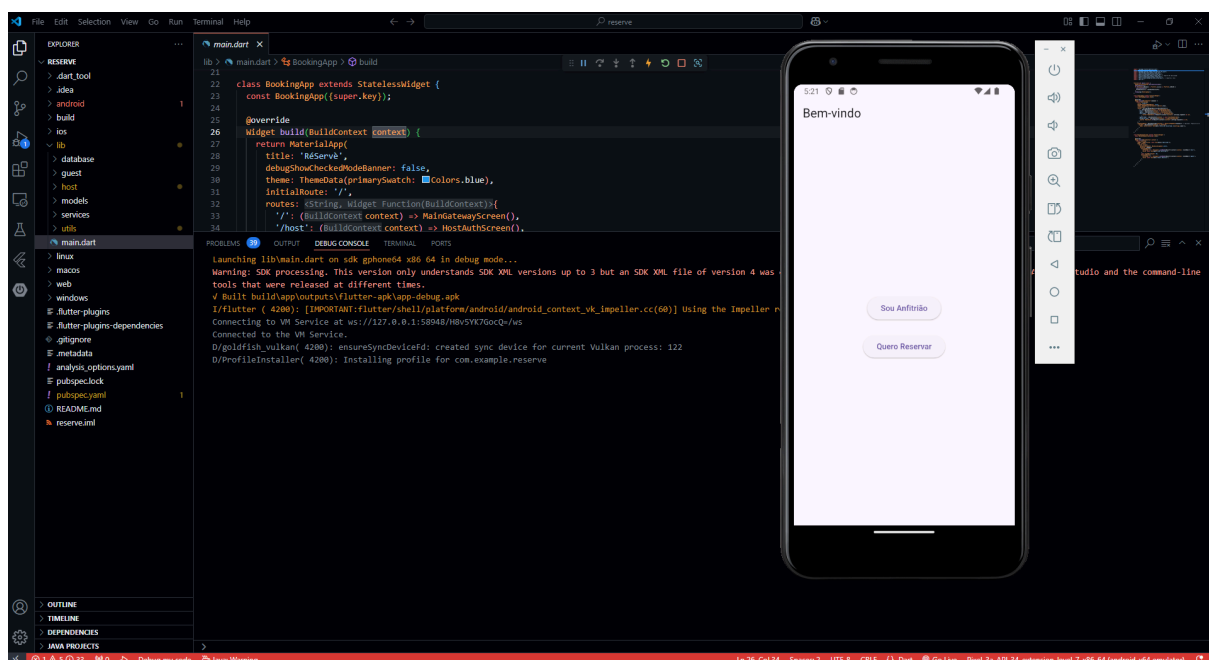
hóspede. Cada reserva é exibida utilizando o widget personalizado *BookingItem*, localizado em *lib/guest/widgets/booking_item.dart*.

2.3. ESTRUTURA DO BANCO DE DADOS

O banco de dados local do sistema foi implementado utilizando SQLite. A lógica de acesso ao banco está centralizada no arquivo *database_helper.dart*, localizado no diretório *lib/database/*. Esse arquivo implementa um padrão singleton para gerenciar conexões com o banco e fornece métodos genéricos para operações como inserção (insert), consulta (query) e exclusão (delete). O banco foi projetado com tabelas específicas para cada tipo de dado: a tabela *user* armazena informações dos usuários (nome, email e senha); a tabela *address* armazena os endereços das propriedades; a tabela *property* contém os dados das propriedades cadastradas pelos anfitriões; a tabela *images* associa imagens às propriedades; e a tabela *booking* registra as reservas feitas pelos hóspedes, incluindo informações como datas de check-in/check-out, valor total da reserva e avaliação da estadia. As operações específicas relacionadas ao gerenciamento de propriedades estão implementadas no arquivo *property_service.dart*, enquanto as operações relacionadas às reservas estão centralizadas no arquivo *booking_service.dart*. Além disso, o arquivo *via_cep_service.dart* integra o sistema à API ViaCEP para buscar endereços automaticamente com base no CEP informado.

3. OBSERVAÇÕES

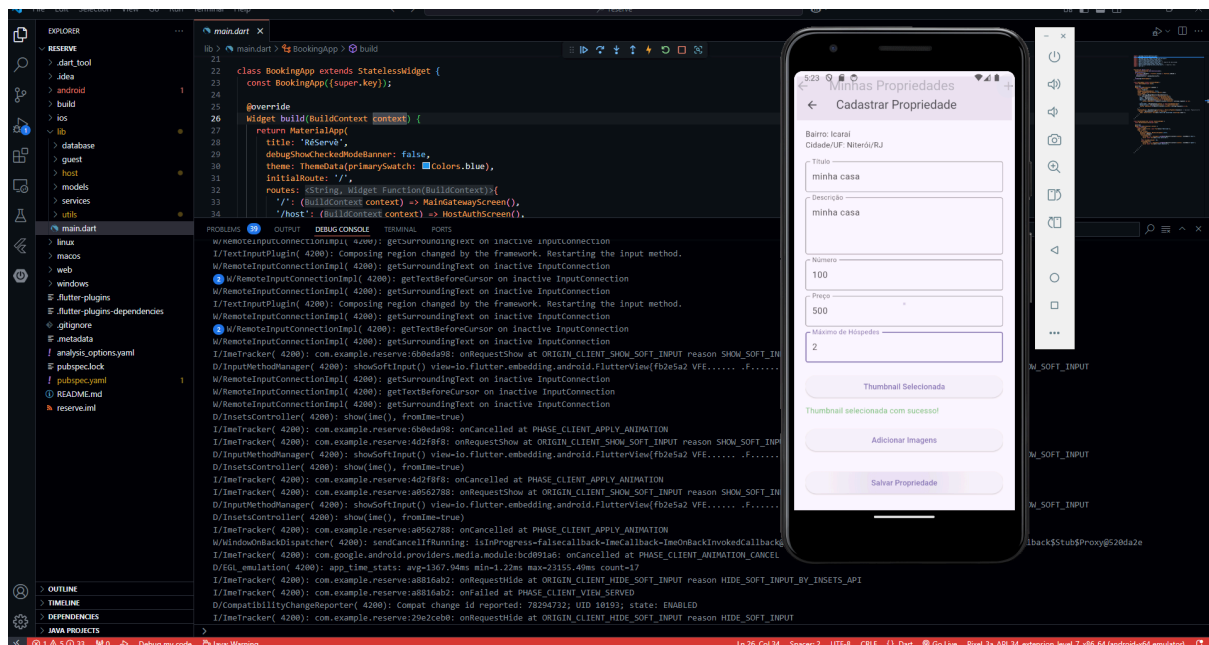
3.1. PROVA DE FUNCIONAMENTO



3.2. GARGALO ENFRENTADO

Durante o desenvolvimento do projeto, enfrentamos um problema relacionado ao congelamento do aplicativo ao alternar entre telas, especialmente no fluxo do anfitrião ao retornar para a tela principal após adicionar uma nova propriedade. Acreditamos que esse gargalo foi causado por operações pesadas de consulta ao banco de dados sendo executadas diretamente na thread principal, bloqueando temporariamente a interface do usuário. Como o Flutter utiliza uma única thread para renderizar a interface e processar eventos, qualquer operação síncrona ou demorada pode impactar negativamente no processamento do aplicativo. No caso específico, a consulta para carregar todas as propriedades cadastradas pelo anfitrião não estava sendo tratada de forma assíncrona, o que resultava no travamento da tela até que a operação fosse concluída. Para tentar resolver esse problema, foi necessário reestruturar o código utilizando o widget *FutureBuilder*, para garantir que as consultas ao banco fossem realizadas de forma assíncrona e que a interface permanecesse responsiva enquanto os dados eram carregados em segundo plano. Essa abordagem melhorou a experiência do usuário durante a navegação entre as telas, porém, não foi o suficiente para solucionar completamente.

Após, muito tempo de tentativa de reparar totalmente o erro, decidimos não continuar com a investigação do problema, tendo em vista que o erro é intermitente, estava tomando tempo demais de nós, durante este fim de período, e não nos parecia solucionável e nem algo dentro do escopo desta atividade, tendo em vista que estávamos já esbarrando em problemas relacionados aos sistemas operacionais, e máquinas virtuais, onde estávamos executando a aplicação.



4. CONCLUSÃO

A aplicação RéServè apresenta uma ótima estrutura e boa modularização, permitindo a implementação das funcionalidades para atender tanto aos anfitriões quanto aos hóspedes. O projeto foi organizado de forma a garantir escalabilidade e manutenção, tendo em vista boas práticas de engenharia de software, com fluxos bem definidos para cada tipo de usuário e integração eficiente com a API ViaCEP. No entanto, durante o processo de desenvolvimento, enfrentamos desafios técnicos significativos, como o congelamento do aplicativo ao alternar entre telas, cujo problema não foi completamente solucionado devido à sua natureza intermitente e complexidade, envolvendo possíveis interações entre o ambiente de execução e o SQLite. Apesar dos desafios enfrentados, acreditamos que o projeto alcançou seus objetivos principais.