

PROGRAMAÇÃO IV

Sistema de Controle de Reservas de Espaços

A tarefa é criar um **aplicativo de Sistema de Controle Reservas de Espaços** usando Flutter, Firebase, e Riverpod 2. Este aplicativo permitirá que os usuários visualizem e reservem espaços, como salas de reunião, auditórios ou laboratórios. Cada espaço terá horários e limites configuráveis, e os administradores poderão gerenciar as reservas.

Funcionalidades

1. Autenticação de Usuário

- Usuários autenticados com Firebase Authentication.
- Um nível de permissão:
 - **Usuários:** Podem visualizar e fazer reservas.

2. Visualização de Espaços

- Tela inicial apresenta uma lista de espaços disponíveis, com informações básicas:
 - Nome do espaço.
 - Capacidade.
 - Disponibilidade (horários livres e ocupados).
 - Status (ativo/inativo).
- Os dados dos espaço devem ser cadastrados diretamente no Firebase

3. Sistema de Reservas

- Usuários selecionam um espaço e escolhem um horário disponível para fazer a reserva.
- Reservas incluem:
 - Nome do espaço.
 - Horário de início e término.
 - Nome do usuário.

4. Persistência e Sincronização

- Firebase Realtime Database (REST) para armazenar:
 - Dados dos espaços.
 - Reservas realizadas.
 - Estados gerenciados com Riverpod para garantir atualização em tempo real.

Telas do Aplicativo

1. Tela de Login

- Autenticação por e-mail/senha ou login social (Google).
- Navega para a tela de lista de espaços após o login.

2. Tela de Lista de Espaços

- Apresenta uma lista de espaços com:
 - Nome.
 - Capacidade.
 - Disponibilidade (por exemplo, "3 horários disponíveis hoje").
- Botão para acessar o painel do administrador (se o usuário for admin).

3. Tela de Detalhes do Espaço

- Mostra informações detalhadas sobre o espaço:
 - Capacidade.
 - Horários disponíveis.
- Formulário para selecionar horário e fazer reserva.

4. Tela de Reservas

- Lista todas as reservas feitas pelo usuário.
- Permite cancelar reservas (se permitido pelo status).

Fluxo de Uso

1. Usuário Faz Login

- O usuário é autenticado pelo Firebase Authentication.
- A permissão (usuário ou administrador) é carregada do Firestore.

2. Usuário Visualiza Espaços

- A lista de espaços é carregada do Firestore.
- Espaços inativos são filtrados.

3. Usuário Faz Reserva

- O usuário escolhe um espaço e horário disponível.
- A reserva é criada no Firestore com o status `pending`.

Estrutura de Provedores (Riverpod)

1. Provedor de Autenticação

- Gerencia o estado do usuário logado.

2. Provedor de Espaços

- Gerencia a lista de espaços.

3. Provedor de Reservas

- Gerencia as reservas associadas ao usuário.

Configuração do Firebase

1. Authentication:

- Ativar login por e-mail/senha e Google.

2. Realtime Database:

- Configurar coleções: `spaces` , `reservations` , `users` .