

Análise Detalhada da Implementação Flutter

1. Estrutura do Aplicativo:

O aplicativo Flutter possui uma estrutura básica com um ponto de entrada (main) que chama o widget MyApp. A classe MyApp configura o tema e define a página inicial como MyHomePage.

2. Página Inicial (My Home Page):

MyHomePage é um StatefulWidget que gerencia o estado da página. A página possui um CustomScrollView contendo SliverAppBar e SliverList. No SliverAppBar, há um cabeçalho expandido com saudações e ícones de ação no topo direito. A lista de slivers inclui um SliverList vazio, mas este pode ser um espaço para conteúdo futuro.

3. Dados da UF:

As informações dos estados brasileiros (UF) estão representadas na lista ufs

4. Captura da Localização do Navegador:

A implementação da captura da localização do navegador é feita através do uso da biblioteca geolocator. Geolocator.getCurrentPosition é utilizado para obter a posição atual do dispositivo. Os dados de latitude e longitude são então utilizados para chamar a função getAddressFromCoordinates que retorna um mapa contendo UF, Capital e Bairro com base nessas coordenadas

5. Construção da Tabela Paginada:

A tabela paginada é construída no método buildTable, que recebe uma lista de dados e utiliza um PaginatedDataTable para exibição. Os dados são fornecidos por uma classe MyDataTableSource que implementa DataTableSource. Os nomes das colunas e os dados são exibidos de acordo.

6. Requisições Assíncronas e Velocidade:

As requisições HTTP assíncronas são feitas usando o pacote http. Para otimizar a velocidade das requisições, o método getDadosTable é assíncrono e utiliza async/await. Caso o status da resposta seja 200, os dados da tabela são construídos chamando o método buildTable com os dados recebidos do servidor.

7. Lógica de Filtragem e Atualização:

Os botões "FILTRAR" e "ATUALIZAR" têm lógica associada para buscar dados com base na UF e no bairro selecionados. A localização atual também é usada caso UF e bairro não sejam especificados. A lógica está associada aos métodos getDadosTable e ao tratamento do botão "FILTRAR".

8. Construção da Tabela Dinâmica:

A tabela dinâmica é construída com o uso do `PaginatedDataTable`, que proporciona uma exibição paginada dos dados. A classe `MyDataTableSource` implementa a interface `DataTableSource` e é responsável por fornecer os dados necessários para a tabela.

9. Menu Lateral Ocultável:

Apesar de o menu lateral não estar completamente construído, a lógica de ocultação e revelação é mencionada. Isso é feito através de um `Drawer` que contém uma coluna de elementos. O menu é expandido/recolhido quando o usuário interage com a barra superior.

10. Interface do Usuário:

Há uma barra de ações (`SliverAppBar`) com saudações, um espaço delimitado por uma linha cinza. O conteúdo principal está contido em um `CustomScrollView`. A parte inferior contém um botão de ação flutuante (`floatingActionButton`) com opções para filtrar, atualizar, cadastrar e fazer download.

11. Aprimoramento no Layout:

Adição de espaçamentos e alinhamentos para melhorar a estética do aplicativo. Alterações nos tamanhos dos botões e layouts para melhor se adequarem à interface do usuário.

12. Gerenciamento de Estado:

A classe `MyHomePage` mantém o estado de variáveis como UF selecionada, bairro, tabela de dados, etc. O estado é atualizado ao interagir com os elementos da interface.

13. Remoção de Caracteres Especiais:

Há uma função `removeSpecialCharacters` para remover acentos e caracteres especiais de uma string.

14. Utilização de Packages Externos:

O código faz uso de pacotes externos como `http`, `geolocator`, e `collection`.

15. Responsividade e Web:

Há verificações para a plataforma web (`kisWeb`) para garantir o bom funcionamento em diferentes plataformas.