



Principais estruturas de dados utilizadas

CadastroActivity

JSONObject

Utilizado para representar dados em formato JSON, tanto para enviar como para receber informações do servidor. O JSONObject armazena pares chave-valor como usuário e senha e é usado para construir o corpo da requisição e processar a resposta do servidor.

RequestQueue

O RequestQueue do Volley atua como uma fila de requisições HTTP. Ele organiza as requisições e garante que sejam enviadas de forma assíncrona.

CalendarioActivity

ArrayList<Evento>

Usada para armazenar os objetos Evento que são recuperados do servidor e exibidos no ListView.

List<Map<String, String>>

Utilizada para estruturar os dados que são exibidos no ListView. Cada evento é armazenado como um Map contendo as informações (data, título e descrição), que será adaptado para a exibição na UI.

Map<String, String>

Dentro da lista de eventos (data), cada mapa contém as chaves "linha1", "linha2" e "linha3" que representam as diferentes informações sobre o evento (data, título e descrição). O SimpleAdapter usa esses valores para preencher o layout.

JSONObject

Para enviar e receber dados para a API RESTful, o código utiliza JSONObject para estruturar as

informações sobre os eventos (título, descrição e data) antes de enviá-las ou recebê-las. O JSONObject é convertido em uma string JSON ao ser enviado via requisição HTTP.

Array:

String[]

dataSelecionada:

Um array utilizado para armazenar a data selecionada no calendário. Esse valor é utilizado em várias partes do código, como ao cadastrar e excluir eventos.

Descriptografia

Arrays de Bytes (byte[])

byte[] dadosCriptografados

Este array armazena os dados criptografados após serem decodificados da string Base64. O conteúdo da string Base64 é convertido de volta para seu formato binário original, que será manipulado na descriptografia.

byte[] iv

O vetor de inicialização (IV) é extraído dos primeiros 16 bytes dos dados criptografados. O IV é usado para garantir que a criptografia do mesmo dado com a mesma chave resulte em textos cifrados diferentes, prevenindo ataques de repetição.

byte[] dados

Após extrair o IV, o restante dos dados criptografados (a partir do byte 16) são armazenados nesse array. Esses dados serão descriptografados utilizando o algoritmo AES.

byte[] chaveBytes

A chave original derivada do segredo (passado como parâmetro) é gerada usando o algoritmo de hash SHA-256. O array chaveBytes armazena o hash resultante dessa operação.

byte[] chave

Após o cálculo do hash SHA-256, a chave utilizada na criptografia é definida como os primeiros 16 bytes do hash gerado.

LoginActivity

SharedPreferences

Objeto para armazenar e recuperar dados persistentes localmente (como o token de login).

RequestQueue

Fila de requisições do Volley, que gerencia as requisições HTTP assíncronas.

SharedPreferences: Usado para armazenar e recuperar o token de autenticação.

JSON: A requisição POST é enviada como um objeto JSON vazio.

Map<String, String>: Cabeçalhos HTTP para incluir o token de autenticação e o tipo de conteúdo.

RequestQueue: Gerencia as requisições HTTP feitas através da biblioteca **Volley**.

ProvaActivity

List<String> respostas

A lista respostas foi usada para armazenar as respostas da questão. A estrutura de dados permitiu armazenar múltiplas respostas e embaralhá-las para que a ordem das alternativas varie a cada questão. As respostas incluem a resposta correta e outras alternativas (A, B, C).

Array

JSONArray questoes:

O JSONArray é utilizado para armazenar todas as questões da prova que são recebidas do servidor como uma resposta JSON. Cada questão é representada por um objeto JSON, e o array permite navegar por todas as questões de forma sequencial.

JSONObject obj

Dentro do JSONArray, cada item é um JSONObject que contém os dados de uma única questão (como a pergunta, as alternativas e a resposta correta).

Mapa

Map<String, String>:

O mapa foi utilizado para armazenar os cabeçalhos da requisição HTTP, como o token de autenticação.