

Principais estruturas de dados sendo utilizadas no projeto

Complaint.java:

Array (String[] complaintTypes)

- **Como está sendo usado:** É um array de strings que armazena os tipos de denúncias possíveis ("Racismo", "Homofobia", "Abuso Sexual", "Outros"). Ele é utilizado para popular o Spinner (dropdown) com as opções disponíveis.
- **Finalidade:** Fornecer um conjunto fixo de opções para o tipo de denúncia.

ArrayAdapter<String>

- **Como está sendo usado:** Adapta o array de strings complaintTypes para ser exibido no Spinner (complaintTypeDropdown).
- **Finalidade:** Facilitar a exibição dos dados do array no componente visual.

Map<String, String>

- **Como está sendo usado:** É criado dentro do método getParams() da requisição POST. Armazena os dados da denúncia (tipo e relato) já cifrados pela cifra de César, com o horário atual usado como offset.
- **Finalidade:** Enviar os parâmetros da denúncia no corpo da requisição HTTP.

StringBuilder

- **Como está sendo usado:** É utilizado na implementação do método cesarCipher para construir a string cifrada de maneira eficiente, iterando pelos caracteres do texto.
- **Finalidade:** Manipular strings de forma otimizada enquanto implementa a cifra de César.

DataBaseManager.java:

List

- **Como está sendo usado:**

Para armazenar os dados do mapa de calor (WeightedLatLng).

Para armazenar as regiões mais frequentes (String).

- **Finalidade:**

Estruturas dinâmicas para guardar múltiplos elementos, que posteriormente são passados ao callback.

WeightedLatLng

- **Como está sendo usado:**

Cada ponto no mapa de calor é representado como um objeto WeightedLatLng que inclui coordenadas (LatLng) e um peso (1.0 no exemplo).

- **Finalidade:**

Estruturar os dados para o uso em mapas de calor.

JSONArray

- **Como está sendo usado:**

Representa os dados recebidos da API para processar múltiplos objetos JSON.

- **Finalidade:**

Permitir a extração de informações, como nomes de regiões e coordenadas.

GeocodingUtils.java:

LatLng

- **Como está sendo usado:**

Armazena as coordenadas geográficas de uma localização, incluindo latitude e longitude.

- **Finalidade:**

Retornar as coordenadas correspondentes a um nome de localização (locationName).

List<Address>

- **Como está sendo usado:**

Lista retornada pelo método geocoder.getFromLocationName contendo resultados potenciais da geocodificação.

- **Finalidade:**

Armazenar múltiplos endereços encontrados que correspondem ao nome da localização.

O método seleciona apenas o primeiro endereço da lista, que é acessado com addresses.get(0).

Address

- **Como está sendo usado:**

Representa um endereço individual com informações como latitude, longitude, nome da rua, cidade, etc.

- **Finalidade:**

Extrair a latitude e longitude do endereço retornado.

Geocoder

- **Como está sendo usado:**

Classe responsável por traduzir nomes de localização (ex.: "São Paulo") em coordenadas geográficas.

- **Finalidade:**

Interagir com APIs de geocodificação para realizar a conversão de um nome de local para coordenadas geográficas.