

PROGRAMAÇÃO ORIENTADA A OBJETOS E ESTRUTURA DE DADOS

ESTRUTURA DE DADOS UTILIZADAS

Membros:

Pedro Della Rosa Antônio

Felipe Vallim Soares

Lucas Silva Maciel

Leonardo Santos Da Silva

INTRODUÇÃO:

Estrutura de dados são métodos organizacionais para armazenar e manipular dados, facilitando o acesso e a manipulação de informações. No nosso aplicativo foi necessário utilizar a estrutura de dados **Lista** para permitir a adição e remoção de elementos sem a necessidade de um tamanho fixo, neste caso, para retornar uma lista de endereços que correspondem ao nome do local fornecido pelo usuário.

Exemplos:

```
private void searchLocation() {  
    String destination = destinationEditText.getText().toString();  
    if (destination.isEmpty()) {  
        Toast.makeText(this, "Digite um destino para pesquisar", Toast.LENGTH_SHORT).show();  
        return;  
    }  
  
    new Thread(() -> {  
        try {  
            Geocoder geocoder = new Geocoder(this, Locale.getDefault());  
            List<Address> addresses = geocoder.getFromLocationName(destination, 1);  
  
            if (addresses != null && !addresses.isEmpty()) {  
                Address address = addresses.get(0);  
                LatLng destLatLng = new LatLng(address.getLatitude(), address.getLongitude());  
            }  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
    }).start();  
}
```

```

private void updateCurrentAddress(Location location) {
    new Thread(() -> {
        try {
            Geocoder geocoder = new Geocoder(this, Locale.getDefault());
            List<Address> addresses = geocoder.getFromLocation(
                location.getLatitude(), location.getLongitude(), 1);

            if (!addresses.isEmpty()) {
                String address = addresses.get(0).getAddressLine(0);
                runOnUiThread(() -> originEditText.setText(address));
            }
        } catch (IOException e) {
            Log.e("Geocoder", "Erro ao obter endereço", e);
        }
    }).start();
}

```

A **Lista** é uma estrutura de dados linear que armazena elementos em sequência, que no caso armazena resultados de geocodificação (Address), sendo necessária mesmo quando limitamos o resultado a 1, já que o Geocoder retorna múltiplos endereços para uma coordenada.

Importação:

```

import java.io.IOException;
import java.util.List;
import java.util.Locale;
import java.util.concurrent.TimeUnit;

```