

Relatório de Desenvolvimento da Aplicação WeatherApp

Aluna: Samara Miranda

1. Funcionalidades Implementadas (e Não Implementadas)

Funcionalidades Implementadas:

- **Visualização da Lista de Climas:** A aplicação permite visualizar uma lista de climas para várias localidades usando WeatherListFragment e WeatherAdapter.
- **Detalhes do Clima:** Ao selecionar uma localidade da lista, o usuário é levado para uma tela de detalhes que mostra a previsão de clima para vários dias usando WeatherDetailFragment.
- **Visualização no Mapa:** O WeatherMapFragment permite visualizar a localização em um mapa e exibe o clima atual para a posição clicada no mapa.
- **Widget de Clima:** A aplicação inclui um widget que mostra a temperatura atual e permite ao usuário configurar a localização através de WeatherWidgetProvider e WidgetConfigureActivity.

Funcionalidades Não Implementadas:

- **Persistência de Dados no Banco:** A integração com o banco de dados Room está implementada parcialmente. As operações de inserção estão comentadas e não são usadas na versão atual do aplicativo.
- **Atualização Automática do Widget:** A lógica para atualizar a frequência do widget está implementada, mas a configuração da frequência ainda não foi concluída.

2. Requisitos/Pressupostos para a Execução da Aplicação

- **Permissões:** A aplicação requer permissões para acesso à internet e localização (se for habilitar funcionalidades futuras que utilizem a localização do usuário).
- **Conexão com a Internet:** A aplicação depende de uma conexão com a internet para buscar dados climáticos da API OpenWeatherMap.
- **API Key:** A chave da API da OpenWeatherMap deve estar configurada corretamente no código para acessar os dados de clima.
- **Configuração do Ambiente de Desenvolvimento:** O desenvolvimento foi realizado usando Android Studio com o SDK do Android.

3. Ambiente de Testes Utilizado

A aplicação foi testada em:

- **Dispositivos Físicos Android:** Realizado em diversos dispositivos com diferentes versões do Android para garantir compatibilidade.
- **Emulador Android:** Utilizado para simulação de diferentes configurações e tamanhos de tela.

4. Estrutura da Aplicação

A aplicação é estruturada da seguinte forma:

- **Camada de Apresentação:** Inclui atividades e fragmentos para exibição de dados (MainActivity, WeatherListFragment, WeatherDetailFragment, WeatherMapFragment).
- **Camada de Dados:** Inclui classes para manipulação e armazenamento de dados (Weather, WeatherEntity, AppDatabase, WeatherDao).
- **Camada de Integração com APIs:** Utiliza Retrofit para comunicação com a API OpenWeatherMap (RetrofitClient, WeatherApiService, WeatherResponse).
- **Camada de Widgets:** Inclui classes para configuração e atualização de widgets (WeatherWidgetProvider, WeatherUpdateService, WidgetConfigureActivity).

5. Aspectos do Código e da Aplicação que Valorizam a Implementação

- **Uso de Retrofit para Integração com API:** A aplicação utiliza Retrofit para simplificar a comunicação com a API, tornando o código mais limpo e gerenciável.
- **Implementação de MVP (Model-View-Presenter):** O uso de AsyncTask para operações em segundo plano mantém a interface do usuário responsiva e melhora a experiência do usuário.
- **Persistência de Dados:** Embora a persistência esteja parcialmente implementada, o uso de Room indica uma estrutura de dados robusta para futuras melhorias.
- **Integração com Google Maps:** O WeatherMapFragment proporciona uma interface interativa e visualmente atraente para o usuário.

6. Conclusões

O trabalho prático foi uma experiência enriquecedora que me proporcionou uma compreensão mais profunda sobre a integração com APIs e a obtenção de dados por meio da leitura de arquivos JSON. Além disso, aprimorou significativamente meu conhecimento em desenvolvimento Android, permitindo-me aplicar conceitos teóricos em um ambiente prático. Esta experiência não só ampliou minha competência técnica, mas também aumentou minha confiança na implementação de soluções inovadoras no desenvolvimento de aplicativos móveis.

A aplicação WeatherApp demonstra um bom nível de funcionalidade e integração com APIs externas. Embora algumas funcionalidades, como a persistência de dados e atualização automática do widget, estejam em progresso, a base estabelecida é sólida. As principais áreas de melhoria incluem:

- **Finalização da Persistência de Dados:** Implementar completamente a persistência e recuperação de dados do banco Room.
- **Aprimoramento da Frequência de Atualização do Widget:** Configurar adequadamente a frequência de atualização do widget para uma melhor experiência do usuário.

