

Título do projeto / estágio

Conceção de Interfaces para Aplicações Móveis

Otávio Araújo - A042508



Licenciatura em Informática

19 de Novembro

Conteúdo

1	Definição do conceito	2
2	Metodologia de levantamento de requisitos com os utilizadores	2
2.1	Autocarro	3
2.2	Metro	3
2.3	Outro transporte público	4
3	Criação de mock-ups	4
3.1	Protótipo versão 1	4
3.2	Protótipo versão 2	6
4	Proof of Concept	7
4.1	Google Maps	7
4.2	OpenStreetMap	8
4.3	Base de Dados	8
4.4	Ligaçao com a API	10
4.5	Inserindo informação no mapa	10
4.6	Favoritos	11
4.7	Configurações	11
5	Conclusões	12
6	Referencias	14

1 Definição do conceito

O conceito do aplicativo surgiu após uma palestra apresentada por Pedro Pimenta, profissional da área de Tecnologia da Informação, na qual ele apresentou o projeto BaZe e alguns dados públicos disponíveis. Muitos desses dados estavam relacionados aos transportes públicos, incluindo informações que disponibilizavam a localização de todos os autocarros da STCP em tempo real.

Embora existam aplicativos que utilizam os horários para informar quando será a próxima paragem, nenhum deles disponibiliza a localização exata dos autocarros, o que motivou a concepção deste aplicativo.

2 Metodologia de levantamento de requisitos com os utilizadores

Como metodologia de levantamento de requisitos, foi criado um inquérito utilizando a plataforma Google Forms. Foram obtidas 7 respostas no total.

Section 4 of 8

(Autocarro) Numa escala de 1 a 6, em que 1 é nada interessante e 6 muito interessante, classifica as seguinte funcionalidades:

Description (optional)

Ver Autocarros em tempo real *

1	2	3	4	5	6
<input type="radio"/>					

Ver as paragens de autocarro *

1	2	3	4	5	6
<input type="radio"/>					

Salvar um autocarro em favoritos *

1	2	3	4	5	6
<input type="radio"/>					

Figura 1: Exemplo de uma pergunta feita no inquérito

Com o objetivo de filtrar os possíveis utilizadores, a primeira pergunta do inquérito questiona se o indivíduo utiliza ou não transportes públicos. Caso a

resposta seja negativa, o inquérito é finalizado. Se o utilizador responder afirmativamente, são apresentadas diferentes seções do inquérito, correspondentes a funcionalidades de diferentes meios de transporte público, sempre perguntado primeiro se utiliza ou não o meio de transporte.

2.1 Autocarro

A primeira seção encontrou 5 respostas como utilizadores de autocarro, as seguintes questões são possíveis funcionalidades em que o utilizador foi pedido para avaliar numa escala de 1 a 6, em que 1 é nada interessante e 6 muito interessante, as possíveis funcionalidades que a aplicação pode ou não vir a ter. Dos 5:

- Ver Autocarros em tempo real
 - 4 avaliaram com um 6
 - 1 avaliou com um 5
- Ver as paragens de autocarro
 - 3 avaliaram com um 6
 - 2 avaliaram com um 5
- Salvar um autocarro em favorito
 - 1 avaliou com um 6
 - 4 avaliaram com um 4
 - 1 avaliou com um 3
 - 1 avaliou com um 2
- Receber uma notificação de quando o autocarro está próximo de sua paragem
 - 2 avaliaram com um 6
 - 3 avaliaram com um 3

Nenhum utilizador sugeriu uma funcionalidade.

2.2 Metro

A segunda seção encontrou 4 respostas como utilizadores de metro, dos 4:

- Ver as linhas de metro
 - 3 avaliaram com um 6
 - 1 avaliou com um 5
- Acessar os horários do metro
 - 3 avaliaram com um 6
 - 1 avaliou com um 5

- Salvar uma linha em favoritos

1 avaliaram com um 6
2 avaliaram com um 5
1 avaliou com um 4

- Previsão de chegada

2 avaliaram com um 6
2 avaliaram com um 4

Nenhum utilizador sugeriu uma funcionalidade.

2.3 Outro transporte público

Nenhum utilizador respondeu positivamente a outro transporte público.

3 Criação de mock-ups

Primeiro foram criado alguns esboços da interface

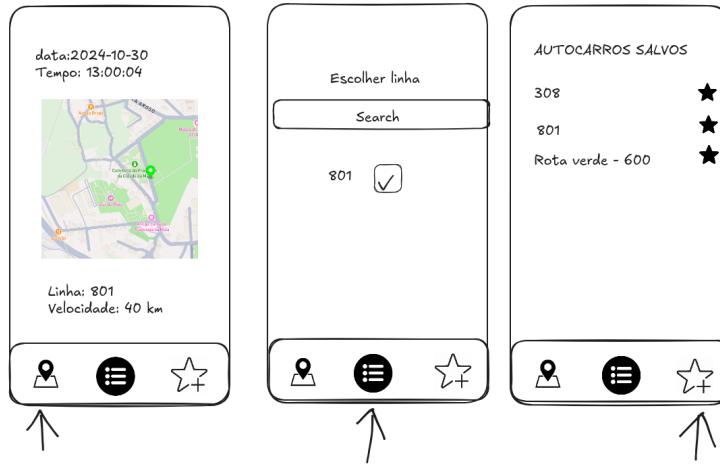


Figura 2: Esboço da interface

Para navegar a App o utilizador tem acesso a 3 botões no canto inferior do ecrã, o primeiro representa o mapa, o segundo uma pesquisa e o terceiro os favoritos.

3.1 Protótipo versão 1

Após isso foi utilizado a plataforma figma para recriar o mock-up e também criar um protótipo da app. Uma primeira versão da interface foi criada juntando a

pesquisa com os favoritos, e no lugar da antiga pagina de favoritos, foi criado uma pagina para aceder as definições.



Figura 3: Primeira versão da mockup

Link com um protótipo da primeira versão da mock-up [aqui](#)

3.2 Protótipo versão 2

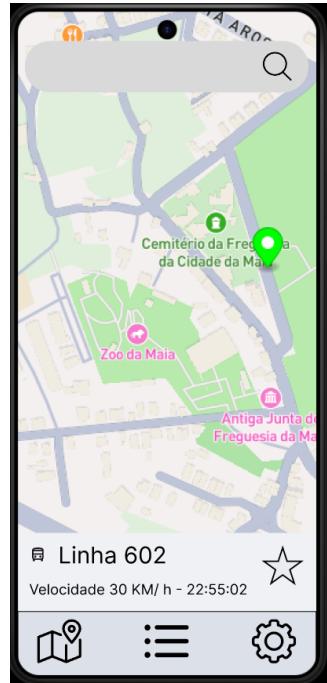


Figura 4: segunda versão da mockup

Após apresentar o protótipo ao docente, a até então versão final foi criada. Agora a pesquisa é feita no mesmo ecrã do mapa, tornando o segundo ecrã como apenas uma lista de favoritos. Para marcar uma linha como favorito basta carregar na estrela ao lado do nome da atual linha. Ao pesquisar por uma linha um novo ecrã aparece com algumas das pesquisas recentes feitas, caso o utilizador queira rapidamente consultar linhas previamente selecionadas.

Link com o protótipo da segunda versão da mock-up [aqui](#)

4 Proof of Concept

Durante a criação do protótipo, a primeira questão foi escolher como expor o mapa

4.1 Google Maps

Originalmente, o mapa de escolha foi o google maps. A maior vantagem de se utilizar o google maps para expor o mapa é a rica documentação, incluindo um tutorial bastante intuitivo, e a fácil sua facil integração com o Jetpack Compose.



Figura 5: App utilizando o Google Maps

Porém a ideia foi abandonada por ser obrigatório o uso de um cartão de ter um cartão de credito para se obter uma API Key. Parte do código feito para o google maps ainda existe em uma branch *Google Maps* no git do projeto porém não deveria ser mais funcional. A figura 5 demonstra o projeto com o código da branch *Google Maps*.

4.2 OpenStreetMap

O próprio projeto BaZe utiliza o [OpenStreetMap](#) (OSM) para expor seu mapa, que o torna uma escolha quase óbvia para o projeto. Incluir o OSM foi um pouco difícil já que o [OSMDROID](#) incluía instruções para Kotlin porém não para Jetpack compose, e o projeto havia sido arquivado no dia 20 de novembro do ano de 2024, fato que não impedia o projeto de ser utilizado mas garantia que não iria existir uma integração **oficial** com o Jetpack Compose. Como o projeto é *open source* um outro projeto chamado [osm-android-compose](#) que já havia conseguido adaptar o OSMDROID para o Jetpack Compose foi utilizado como guia para conseguir expor o mapa com sucesso e foi a forma com que o Proof of concept foi finalizado.

Com o mapa em estado funcional o próximo desafio foi criar a base de dados

4.3 Base de Dados

Para evitar a necessidade de criar uma conta para se utilizar a aplicação, foi utilizado uma base de dados usando a biblioteca de persistência *Room*. Foi utilizado como referência a Wiki [ccti.ismai.pt](#) disponibilizada pelo docente Alexandre Valente para criação da base de dados e de outros aspectos da aplicação.

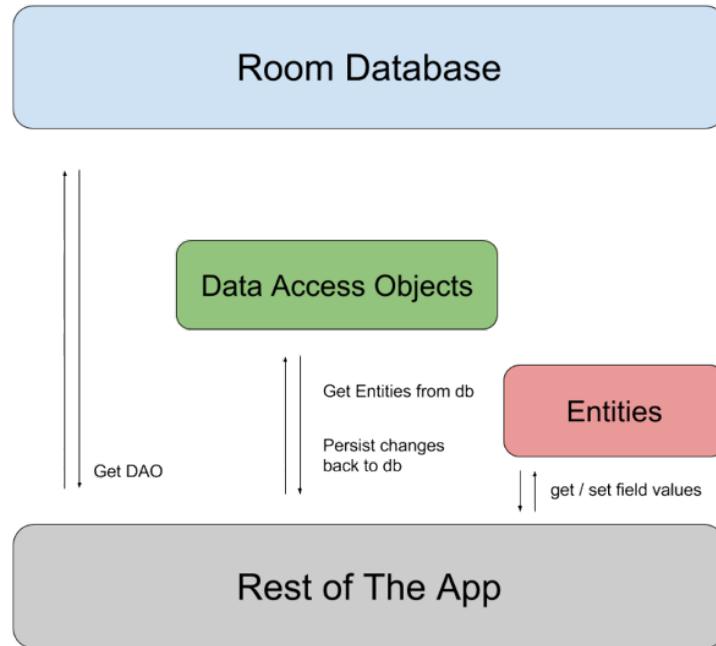


Figura 6: Diagrama da database

A base de dados serve para guardar uma lista de favoritos do usuário, cada item da base de dados iria precisar de uma classe para armazenar informação obtida pela API. Exemplo de um Autocarro na API:

```
"type": "FeatureCollection",
"features": [
{
    "type": "Feature",
    "properties": {
        "icon": "./pics/bus6.png",
        "marker-color": "lime",
        "popupContent": "<div style=width:200px;><h1>Linha 701</h1><hr c
0 km/h<br>data:2025-01-05 20:35:02 </div>"
    },
    "geometry": {
        "coordinates": [
            -8.59382,
            41.1672
        ],
        "type": "Point"
    }
}]
```

Apesar de toda a informação obtida ter sido armazenada nem toda ela foi utilizada. Caso o projeto seja continuado futuramente, a *marker-color*, que é retornada em duas cores, lima caso o autocarro esteja sendo em serviço ou vermelho caso esteja fora de serviço poderia ser integrada no mapa.

Como se pretende que o utilizador possa adicionar Autocarros e Metros como favoritos, ambas classes foram criadas. Para a criação de entidades na base de dados a única informação pertinente é o nome da linha, é suposto a aplicação disponibilizar todos os autocarros a linha selecionada, justificando a escolha de tornar o nome da linha como Chave primaria. Toda outra informação não é relevante para o autocarro já que esses dados estão sendo alterados em tempo real (Velocidade, cor, data...), o mesmo é valido para o metro.

Com a classe criada, um Data Access Object (DAO) foi criado para ambos Autocarro e Metro, onde são definidas as funções **Insert**, **Delete** e **Query** e por fim um repositorio chamado *LinhaRepository* onde corotinas são utilizadas para de uma forma assíncrona inserir ou apagar linhas de metro ou autocarro.

Originalmente se pensou em exibir o nome da linha e a velocidade na parte inferior do mapa. Isso foi alterado já que existem diversos autocarros de uma só linha dificultando demonstrar a qual autocarro aquela informação se refere.

4.4 Ligação com a API

Para efetuar a ligação a API foi utilizado a biblioteca *retrofit* que facilitou bastante a interação com a API.

Três *Endpoints* foram criados, um para Autocarros, outro para linhas de metro e finalmente um para paragens de autocarro STCP na maia.

4.5 Inserindo informação no mapa

Em geral, inserir marcadores funciona exatamente como instruído na [wiki do projeto](#). Para cada item (Autocarro, Paragem, Metro...) inserir um marcador com uma posição, titulo e ícone. O título expõe informação relevante sobre o item quando pressionado, com exceção do metro, que ao ser pressionado não apresenta nenhuma informação apesar de contar com um título.

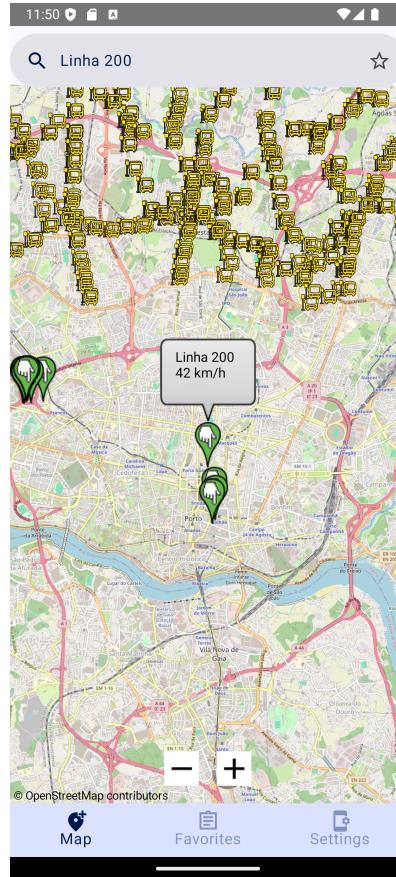


Figura 7: Linha 200 vista do mapa

Ao utilizar a barra de pesquisa, sugestões são oferecidas para indicar o utilizador como fazer uma pesquisa. Quando uma pesquisa é feita, o texto se torna a linha atual (*currentLinha*) e uma pesquisa na base de dados é feita. Caso um Metro ou Autocarro seja encontrado ele é exposto no mapa. Além disto, o usuário tem a opção de adicionar a linha atual como um favorito através de um ícone de uma estrela não preenchida na barra de pesquisa. Caso o texto na barra de pesquisa esteja presente na base de dados o ícone da estrela se torna preenchido, e ao pressionar o ícone a linha atual é removida dos favoritos. Note que também é possível remover um item no ecrã de favoritos.

Existem dois elementos no mapa que são expostos independente da barra de pesquisa, a localização do utilizador e paragens de autocarro. As paragens rapidamente se tornaram um problema por serem muito abundantes. Para mitigar isto, foi criado uma configuração para ligar ou desligar as paragens. Futuramente também deve se criar ícones para melhorar a visibilidade do mapa.

Já a localização do usuário, se apresentou um desafio não em se obter a localização mas sim em adquirir permissão do utilizador para conseguir utilizar o GPS, algo que é obrigatório por ser uma permissão considerada perigosa. Foi necessário algum esforço para descobrir como pedir permissão ao utilizador, a solução foi um *ActivityResultContracts.RequestPermission* que faz o pedido ao utilizador, que caso confirme, o mapa cria um marcador para o utilizador e foca o mapa nele. Caso o utilizador rejeite, o mapa foca no centro da Maia.

4.6 Favoritos

Toda vez que um transporte é adicionado como favorito simultaneamente ele é adicionado a uma *LiveData<List<_>* no *MainViewModel*. Permitindo que, em outro ecrã seja possível disponibilizar a lista. Que é exatamente o que acontece no ecrã de favoritos. Cada item da lista é exposto permitindo que o utilizador ou pressione no item, o direcionando para o mapa, ou remova o item da lista ao pressionar no ícone "X".

4.7 Configurações

Após algumas considerações, três possíveis configurações poderiam ser implementadas. Delas, uma definição que permitisse alterar o tema e outra que alterasse a visibilidade das paragens de autocarro foram implementada, deixando apenas a terceira configuração, que permitia que o utilizador mudasse o idioma da aplicação como não implementado.

Das configurações que foram implementadas, alterar a visibilidade já foi mencionado previamente, é um *boolean* na *MainViewModel* que verifica e altera a visibilidade das paragens no Ecrã mapa. Alterar temas permite que o utilizador alterne entre *Dark* e *Light modes*, alterando as cores para dar um aspeto claro

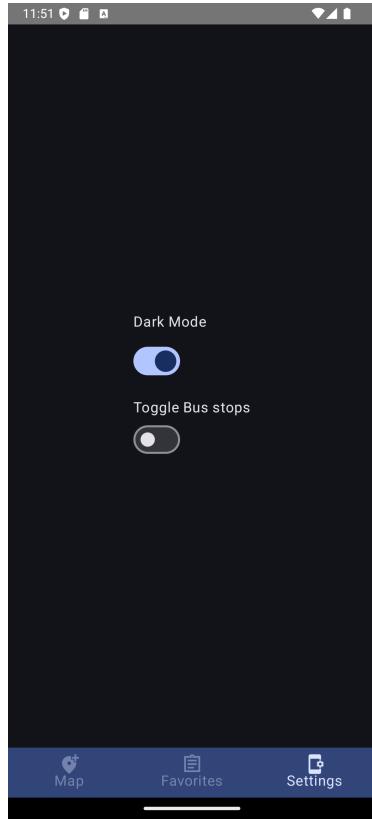


Figura 8: Configurações

ou escuro a aplicação. As cores foram criadas utilizando o website [Material Theme Builder](#), porém muitas cores foram editadas visto que os temas criados não seguiam um tema monocromático. Os temas foram definidos no *Theme.kt*.

Mudar o idioma da aplicação foi uma funcionalidade que teve de ser descartada. Apesar de algumas tentativas em geral o resultado fracassou, porém, caso o telemóvel do utilizador esteja em Inglês ou Português, a aplicação vai expor o texto no idioma correto.

5 Conclusões

Em geral muitas das dificuldades encontradas no projeto foram devidas a pouca familiaridade com a linguagem de programação. Alguns métodos foram criados de forma mais orientada a objetos que certamente afetou a legibilidade em alguns momentos, isto afetou o ficheiro *Ecras.kt* que deverá ser refatorado no futuro para melhor legibilidade, muitos métodos e variáveis podiam ser passadas para outras classes.

Quase todas as funcionalidades levantadas nos inquéritos foram cumpridas. Das que não foram implementadas, a Aplicação não envia uma notificação quando o autocarro está próximo de sua paragem, não disponibiliza os horários do metro e não oferece uma previsão de chegada.

O sucesso da aplicação dependeu muito da API do projeto BaZe que atendeu quase todas as necessidades, e em geral se provou fácil de integrar com o projeto.

6 Referencias

- Add a map to your Android app
- UMAIA Wiki Android - Kotlin.
- Material components in Compose
- composables Searchbar
- Material 3 Search Bar with Jetpack Compose - Easy Tutorial
- osmdroid
- OSM-Android-Compose
- Material Theme Builder
- BaZe
- Request runtime permissions