Technical Project Report - Android Module

TrackMyRun

|  |  |
| --- | --- |
| Curso: | Introdução à Computação Móvel |
| Data: | Aveiro, 19/06/2024 |
| Estudantes: | 105926: Bernardo Pinto  110555: Joao Santos |
| Resumo do projeto: | TrackMyRun é um aplicativo que transforma suas corridas em experiências memoráveis, permitindo que você salve o histórico de suas corridas e associe fotos a momentos especiais ao longo do percurso. Com a aba "Clima", você pode verificar as condições meteorológicas locais antes de sair, garantindo que esteja sempre preparado. |

[1 Conceito 1](#_Toc512889876)

[2 Solução implementada 2](#_Toc1091790117)

[Visão geral da arquitetura 2](#_Toc1103313266)

[View 3](#_Toc2102567477)

[Viewmodel 3](#_Toc832774687)

[Model 4](#_Toc188649629)

[Armazenamento Local 4](#_Toc2113287890)

[Armazenamento Remoto 4](#_Toc1949086649)

[Autenticação 4](#_Toc1580213098)

[Implemented interactions 4](#_Toc1541552837)

[Login e registo 4](#_Toc332536230)

[Página Principal com mapa e informações de corrida 5](#_Toc1149304124)

[Página de histórico com as corridas registadas 5](#_Toc1280096485)

[Página “Weather” 6](#_Toc2119682645)

[Logout 6](#_Toc48328329)

[Project Limitations 7](#_Toc1638973370)

[3 Conclusão e suporte 7](#_Toc445952197)

[Lições aprendidas 7](#_Toc2086760036)

[Distribuição de trabalho 8](#_Toc1111128325)

[Recursos 8](#_Toc1272537325)

[Referências 8](#_Toc175630145)

# 1 Conceito

A nossa aplicação tem como função registar e mostrar em tempo real informações relativas ao movimento físico do utilizador. A aplicação regista informações como percurso realizado, posição atual, velocidade, distancia, tempo e ritmo do percurso atual. Esta informação pode ser consultada durante a corrida no ecrã ou na aba das notificacoes ou após numa aba separada com as corridas todas do utilizador. Além disso, o utilizador também pode, ainda dentro da aplicação, consultar informações climática antes de iniciar sua próxima corrida e registar fotos ao longo do percurso, que ficam associadas a localização que foi capturada.

Os utilizadores alvo da nossa aplicação seriam as pessoas interessadas em saber informações sobre a sua atividade física em percursos seja para registo, melhora de performance ou por próprio lazer.

# 2 Solução implementada

## Visão geral da arquitetura

A estratégia utilizada para o desenvolvimento do projeto foi a adoção da arquitetura MVVM (Model-View-ViewModel), devido à sua eficiente separação de camadas e à clara definição de interações entre elas.

### View

As telas do sistema foram desenvolvidas utilizando a anotação @Composable do Jetpack Compose, representando a camada de "View" da aplicação. Estas telas são responsáveis apenas pela exibição de elementos visuais ao usuário, mantendo a lógica de negócio separada em outra camada.

### Viewmodel

Para garantir a modularidade e a separação de responsabilidades, foram desenvolvidas ViewModels que contêm toda a lógica por trás das telas. As ViewModels são responsáveis por gerenciar o estado e a lógica de negócio da aplicação, utilizando StateFlow para gerenciar e observar mudanças de estado de forma reativa. Por exemplo, a ViewModel associada à tela do mapa (MapViewModel) contém toda a lógica relacionada à construção e renderização do mapa, bem como a lógica para obter a localização e o movimento do usuário. Esta informação é, então, enviada para a MapScreen, que apresenta os dados ao usuário de forma atualizada.

### Model

A camada "Model" foi desenvolvida com diversas responsabilidades essenciais para o funcionamento e a eficiência da aplicação:

#### Armazenamento Local

Para garantir maior eficiência e a funcionalidade offline, foi implementada uma camada de persistência local utilizando o Room Database. Isso permite que os dados sejam armazenados localmente no dispositivo, proporcionando acesso rápido e confiável mesmo na ausência de conexão com a internet.

#### Armazenamento Remoto

Para garantir a persistência real dos dados e aumentar a segurança, optamos por utilizar o Firestore como banco de dados remoto. Firestore oferece sincronização automática dos dados entre os dispositivos e o servidor, bem como recursos de segurança robustos.

#### Autenticação

O código responsável pela autenticação de usuários está localizado nesta camada, utilizando o Firebase Authentication (FireAuth). Esta escolha garante um processo de autenticação seguro e eficiente, suportando várias formas de login, como email/senha, provedores externos, e permitindo uma integração fluida com o Firestore para gerenciar os dados dos usuários autenticados. Vale ressaltar que para o projeto atual foi implementada apenas login através de email/senha, ficando a utilização de provedores externos para o trabalho futuro.

## Implemented interactions

### Login e registo

Uma imagem com texto, captura de ecrã, Tipo de letra, número

Descrição gerada automaticamente Uma imagem com texto, captura de ecrã, Tipo de letra, file

Descrição gerada automaticamente

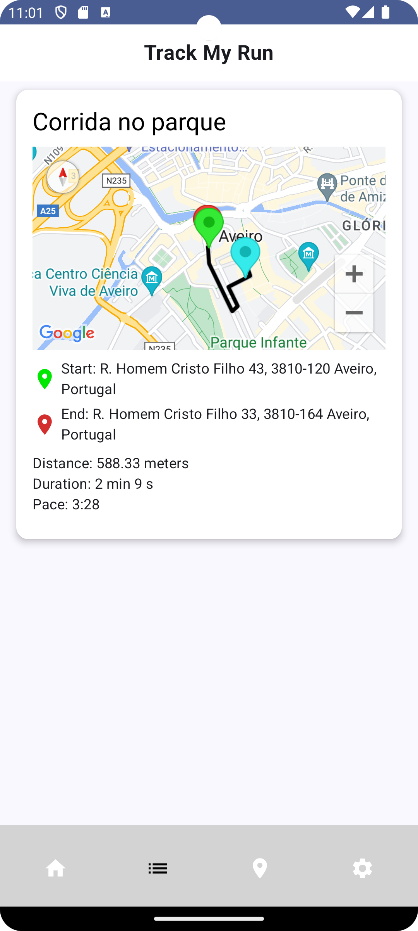
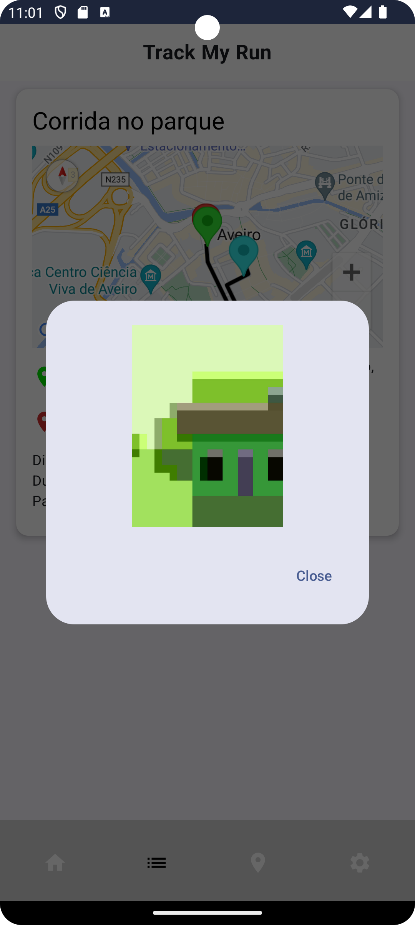
### Página Principal com mapa e informações de corrida

Nesta tela o utilizador pode acompanhar as métricas do seu percurso e também adicionar fotos, que ficarão relacionadas com a localização que foi capturada.

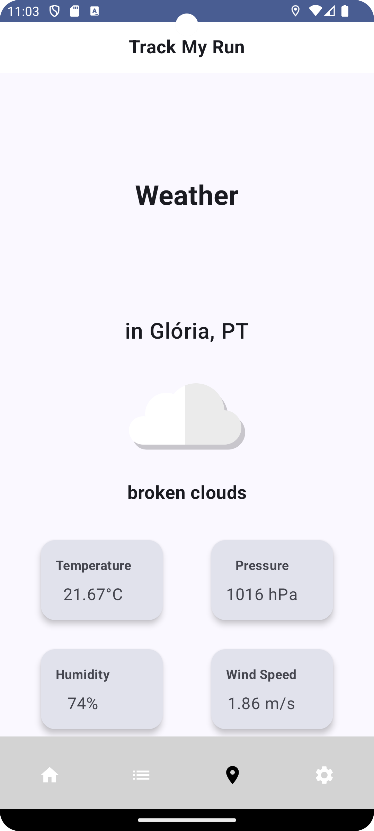
 

### Página de histórico com as corridas registadas

Os marcadores azuis representam as fotos tiradas ao longo do percurso

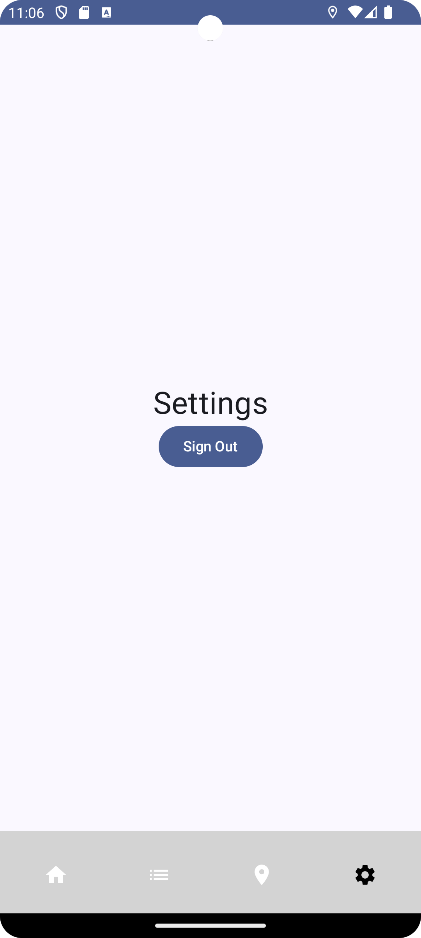
 

### Página “Weather”



Apresenta informação sobre as condições meteorológicas sobre a localização atual

### Logout



### Notificações na corrida

Uma imagem com texto, captura de ecrã, cartão de visita, Tipo de letra

Descrição gerada automaticamente

## Project Limitations

* Rede social: Uma feature que ficou por implementar foi a funcionalidade da rede social em que os utilizadores partilhavam as suas corridas com outros utilizadores com os mesmos interesses.
* Autenticação: Outra feature que ficou para trabalho futuro foi a autenticação a partir de provedores externos
* UI: No seguimento do projeto, uma das prioridades será melhorar a capada de visualização

# 3 Conclusão e suporte

## Lições aprendidas

Relativamente ao android em comparação com o flutter sentimos que o android é bem mais complexo, mas em android aprendemos a organizar as camadas de uma aplicação relativamente às funcionalidades da mesma. A interação com API’s externas é simples de executar. A lógica por detrás da maneira de funcionar do android com os composables, suspend functions, mutableStatesOf, no início como são conceitos novos não são muito fáceis de adaptar, mas depois fazem sentido e são uteis no desenvolvimento das funcionalidades.

## Distribuição de trabalho

* Bernardo Pinto – 60%
* João Santos – 40%

## Recursos

|  |  |
| --- | --- |
| **Recurso:** | **Disponível em:** |
| Code repository: | https://github.com/beernardoc/ICM\_AndroidProject |
| Ready-to-deploy APK: | https://github.com/beernardoc/ICM\_AndroidProject/blob/main/app/app-debug.apk |
| Demo video: | https://github.com/beernardoc/ICM\_AndroidProject/blob/main/demo/demo.mp4 |

## Referências

<https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderClient>

<https://firebase.google.com/?hl=pt>

https://developer.android.com/develop?hl=pt-br