

Open Air Running Tracker (OART)

Grupo 07

Roberto Graça, 93020

Eduardo Almeida, 93148

Projeto pode ser encontrado em: [Github link](#)

Motivação

O projeto OART foi desenvolvido no âmbito do módulo de Flutter da disciplina de Computação Móvel. Este projeto tem como objetivo manipular e apresentar informação essencial que ajuda o utilizador a entender o desenvolvimento do seu treino ao longo do tempo, assim como o dos seus amigos. Centrada num estilo minimalista, esta aplicação permite ao utilizador realizar um treino, guardar a informação desse treino e analisar a informação num momento futuro.

Solução

A solução tem como funcionalidades:

1. Utilização de GPS e exibição da localização num mapa
2. Obter a distância percorrida assim como o tempo passado
3. Utilizar text-to-speech para avisar o utilizador em intervalos de distância regulares (1Km)
4. Ter acesso à informação de treinos realizados
5. Poder tirar fotografias durante um treino e armazená-las
6. Adicionar amigos e poder ter acesso à informação dos seus treinos
7. Saber quais os recordes pessoais

A solução final consiste em três elementos diferentes:

- Um elemento onde se possa criar um treino, onde os campos do treino possam ser visualizados em tempo-real (localização, tempo, distância, percurso percorrido) e onde as fotografias possam ser tiradas;
- Um elemento onde se possa observar as informações relativas aos treinos passados;
- Um elemento onde se possa observar estatísticas gerais.

Arquitetura e Elementos da Aplicação

A arquitetura é dividida entre 2 elementos, a API e a aplicação. A API foi construída usando FastAPI. Na API, está uma base de dados que contém dados relacionados com todos os utilizadores, os seus treinos e a sua lista de amigos. Esta informação é colocada na API e recebida da API utilizando métodos tradicionais de comunicação com REST API em

servidores web. Na aplicação, a informação retirada da API é guardada numa base de dados local escrita em sqflite, uma extensão de SQLite para Flutter. A informação guardada é relativa ao utilizador que inseriu as credenciais na aplicação.

A aplicação tem 4 Bloc: login_bloc, para controlar os valores do campo de login e decidir se o utilizador pode entrar na aplicação; sign_up_bloc, para controlar os valores de sign up e decidir se o utilizador pode ser registado na aplicação; map_bloc para controlar os vários estados da página do mapa; e feed_bloc para controlar os vários estados da página do feed e fornecer informação à página de perfil.

A aplicação tem 6 Cubit e 5 Navigator: session_cubit é usado para efeitos de autenticação e de cancelamento, juntamente com app_navigator e o auth_repository; auth_cubit serve para distinguir entre login e sign up, juntamente com o auth_navigator, e, se as credenciais forem autorizadas, lança a sessão no session_cubit; bottom_nav_bar_cubit indica ao bottomNavigationBar qual a página de destino; map_navigator_cubit, juntamente com o map_navigator fazem a navegação entre a página do mapa, página de definições e a página de guardar o treino; feed_navigator_cubit, juntamente com o feed_navigator fazem a navegação entre a página do feed, a página do QR Code e a página de um treino individual; profile_navigator_cubit, juntamente com profile_navigator fazem a navegação entre a página de perfil e a página de um treino individual.

Na página do maps, está implementado um mapa (Google Maps) usado para exibir a localização do utilizador e o percurso percorrido, a localização é obtida através do pacote Location. Durante a realização de um treino, é possível pausar e tirar fotografias usando ImagePicker. E também é possível controlar do tempo do treino usando um StopwatchTimer. Para o cálculo da distância, é usado o array de coordenadas obtido pelo Location e a cada coordenada nova é aplicada a fórmula de Haversine para calcular a distância à última coordenada obtida. Por último, no decorrer do treino é possível ativar a funcionalidade de Text-To-Speech em diferentes idiomas, que estão listados na aba de “Definições”; esta funcionalidade avisa o utilizador em intervalos de 1 quilómetro, sobre a distância percorrida e tempo passado e a sua configuração é guardada usando SharedPreferences.

Na página do feed, estão listados os treinos do utilizador e dos seus amigos e em cada treino são exibidas as suas estatísticas, descrição e imagens, sendo as últimas exibidas num CarouselSlider. Nesta página, no canto superior direito, é possível aceder à página de adicionar amigos, onde está implementado um leitor de QR Code (QR Code scanner) e gerador de QR Code (QR flutter).

Na página de perfil, estão expostas as estatísticas globais do utilizador, assim como os seus recordes pessoais. Aqui estão apresentados 2 gráficos que representam os últimos 7 treinos (syncfusion_flutter_chart). O botão de sign out pode ser encontrado no canto superior direito.

Objetivos alcançados e problemas encontrados

Das funcionalidades que nos propusemos a desenvolver com a entrega da proposta do projeto, as que foram alcançadas são:

- Utilização de GPS e exibição da localização num mapa
- Utilizar text-to-speech para avisar o utilizador
- Ter acesso à informação de treinos realizados
- Poder tirar fotografias durante um treino e armazená-las

- Adicionar amigos
- Observar recordes pessoais

As que não foram alcançadas são:

1. Obter distância percorrida e velocidade em tempo-real usando acelerómetros e temporizadores. Esta funcionalidade foi alterada para obter a distância percorrida determinando a distância entre duas coordenadas, usando a fórmula de Haversine, e obter tempo passado usando temporizadores. A velocidade é calculada no final do treino.
2. Partilhar informação dos treinos usando QR Code ou numa página com formato fórum. Esta funcionalidade foi alterada para partilhar informação dos treinos numa página com formato fórum com os treinos do utilizador e dos seus amigos.
- 3.

Objetivos extra:

- Sincronização entre a base de dados local e a API, dando a possibilidade de utilizar a aplicação offline.

Problemas encontrados:

- Quando se liga a internet durante a utilização da aplicação é necessário esperar alguns momentos ou reiniciar a aplicação.

Percentagem de trabalho

Roberto Graça - 50%

Eduardo Almeida - 50%

Tutorial/Manual/Ecrãs

Adicionar um treino



No final, o treino pode ser observado no feed.

Observar Feed



Nota: Ao criar novos treinos, estes não são apresentados, por defeito, no feed. É preciso atualizar manualmente.

Adicionar amigos

Na página do feed, no canto superior direito, tem um ícone que leva para a página de adicionar amigos.

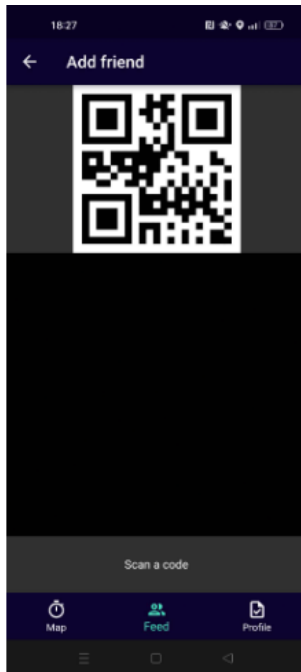


Imagem 1: Antes de detetar um QR Code na câmara

Deteção de um QR Code

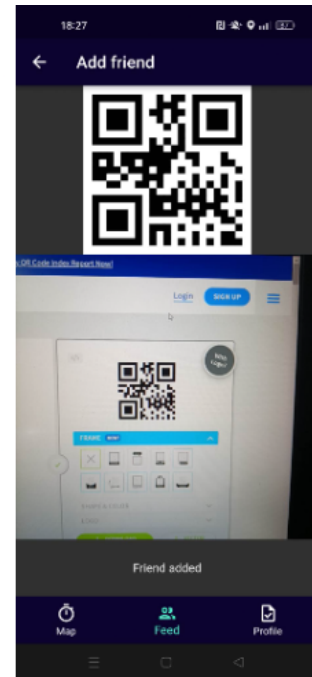
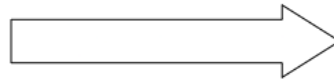


Imagem 2: Depois de detetar um QR Code na câmara

Nota: É preciso internet para realizar esta ação

Nota2: É preciso carregar na zona negra do ecrã (Imagem 1) para iniciar a câmara.

Observar o perfil

Aqui estão presentes as estatísticas e recordes pessoais do utilizador até ao momento



Imagem 1: Estatísticas do utilizador

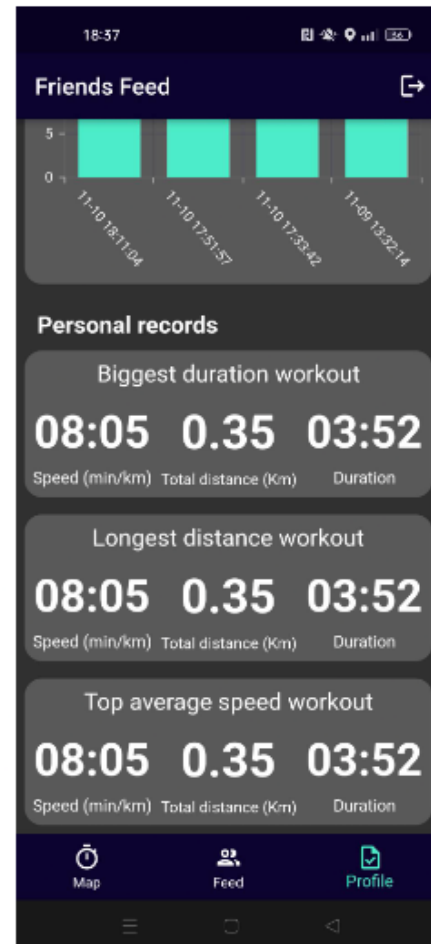


Imagem 2: Recordes pessoais do utilizador