



**ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO**

Relatório de Projeto

“FitBuddy”

Unidade Curricular de Computação Móvel e Ubíqua

Grupo 20

Tiago Silva, 8190614

Tiago Coelho, 8190340

Jaques Resende, 8190214

Índice

Índice de Figuras	3
Introdução	4
Visão do produto e análise do problema	5
Decisões Tomadas	5
Requisitos	6
API's Externas Utilizadas	7
Base de dados	8
Funcionamento da Aplicação	10
Funcionalidades Implementadas	12
Obrigatórias.....	12
Bonificação	12
Conclusão	13
Bibliografia	14

Índice de Figuras

Figura 1 - Logótipos das API externas	7
Figura 2 - Versão inicial do diagrama ER da base de dados	8

Introdução

Este projeto foi desenvolvido no âmbito da unidade curricular de computação móvel e ubíqua como forma de avaliação dos conhecimentos lecionados durante as aulas da UC.

Como proposto no enunciado, o projeto teria como objetivo desenvolver uma aplicação em ambiente Android, com a capacidade de servir como um companheiro desportivo, ou ajudante de Fitness. Para tal era necessário que a aplicação implementasse funcionalidades específicas, tais como a monitorização, aconselhamento de locais consoante localização e armazenamento de dados específicos do utilizador.

Como início do desenvolvimento foi feito um levantamento de requisitos específicos para o funcionamento da aplicação, e requisitos que poderiam ser usados também como fator de bonificação. Posteriormente e como forma complementar, foi feito um estudo de como iria ser estruturada a base de dados “Room”.

Após terminado o levantamento de requisitos, o grupo efetuou a divisão de tarefas, horários de reunião para monitorização do desenvolvimento e debates de ideias.

Visão do produto e análise do problema

O desafio lançado para a equipa tratava-se do desenvolvimento de uma aplicação de fitness que permitisse aos utilizadores praticar desporto e registar e monitorizar os seus dados durante a prática do mesmo, bem como, aconselhar ao utilizador, lugares onde este possa também praticar desporto.

De uma forma mais ampla, esta aplicação serviria como um parceiro de treino que regista os dados dos nossos treinos e para além disso esses dados converte em calorias gastas, das quais o utilizador poderá consultar no seu perfil e ter uma ideia da quantidade de calorias que já queimou desde o seu início de uso na aplicação.

Analisando o âmbito do projeto, entendemos que as maiores dificuldades/problemas que enfrentaríamos seria na forma como pensaríamos na nossa base de dados de forma a guardar os dados e o registo dos próprios dados de forma automática e manual, pois iria ser necessário um bom uso dos sensores e funcionalidades do Android.

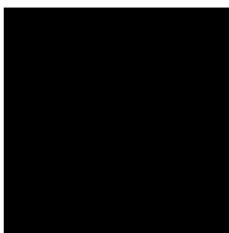
O uso de API's externas também poderia ser uma dificuldade, apesar de o risco não ser muito elevado.

Decisões Tomadas

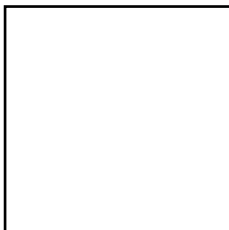
Em termos de design as principais cores utilizadas na aplicação foram as seguintes:



Vermelho
HEX: #E51B23
RGBA(229, 27, 35, 1)



Preto
HEX: #000000
RGBA(0, 0, 0, 1)



Branco
HEX: #FFFFFF
RGBA(255, 255, 255, 1)

Requisitos

Como ponto de partida foi feito uma recolha de requisitos para o planeamento do projeto. A investigação e recolha de requisitos foi baseada num debate de ideias em grupo. Durante a análise do enunciado do trabalho o grupo também foi pesquisando outros modelos de aplicações dentro do mesmo modelo de negócio para ter uma noção mas aprofundada do trabalho.

Após todo o debate foram criados os seguintes requisitos funcionais:

- [RF-101] Como utilizador pretendo fazer a minha autenticação na aplicação (Login e registo);
- [RF-102] Como utilizador pretendo visualizar um mapa que fornece locais de interesse desportivo de acordo com a minha localização atual;
- [RF-103] Como utilizador pretendo selecionar os lugares de interesse/favoritos;
- [RF-104] Como utilizador pretendo dar uma classificação aos meus locais favoritos;
- [RF-105] Como utilizador pretendo ver um local de interesse escrevendo pela rua;
- [RF-106] Como utilizador pretendo visualizar o meu perfil na aplicação;
- [RF-107] Como utilizador pretendo visualizar os locais favoritos da comunidade;
- [RF-201] Como utilizador pretendo selecionar um exercício para treinar;
- [RF-202] Como utilizador pretendo visualizar os exercícios que treinei posteriormente e os dados do treino;
- [RF-203] Sistema deve atualizar dinamicamente os dados do utilizador na aplicação à medida que este vai treinando;
- [RF-204] Sistema deve aceder à localização do utilizador a partir do sensor, de forma a mostrar os locais de interesse;
- [RF-205] Sistema deve fazer uso do sensor de cronómetro do dispositivo como forma de gravar o tempo dos exercícios manuais do utilizador;
- [RF-206] Sistema deve usar o sensor de acelerómetro ou de passos para calcular os passos dados pelo utilizador num exercício automático;
- [RF-207] Sistema deve fazer uso de notificações de forma a manter o utilizador informado do seu progresso ou ações;

Também foram considerados alguns requisitos não funcionais :

- [RNF-301] Otimização da aplicação de resposta rápida e eficiente;
- [RNF-302] Contenção ou minimização de bugs/falhas;
- [RNF-303] Design intuitivo e “User friendly”;

API's Externas Utilizadas

Através da biblioteca Retrofit o grupo implementou com sucesso a ligação de 2 API's externas para a aplicação.

Após alguma consideração e pesquisa por parte do grupo, chegou-se à conclusão que seria mais benéfico utilizar uma biblioteca alternativa ao Google Maps.

A biblioteca selecionada foi a "HereMaps". A mesma permite carregar um mapa com base na localização fornecida pelo GPS do utilizador, mostrar marcadores com pontos de interesse, desenhar linhas com uma trajetória em linha reta entre a localização do utilizador e um ponto de interesse e desenhar um círculo representante do raio de pesquisa da informação.

A recolha de toda a informação dos pontos é feita através da GeoApify. O endpoint utilizado devolve um conjunto de ginásios perto do utilizador.

Como bonificação do trabalho ainda foi utilizado a "Open Weather" API para mostrar a meteorologia consoante a localização atual do utilizador.



Figura 1 - Logótipos das API externas

Base de dados

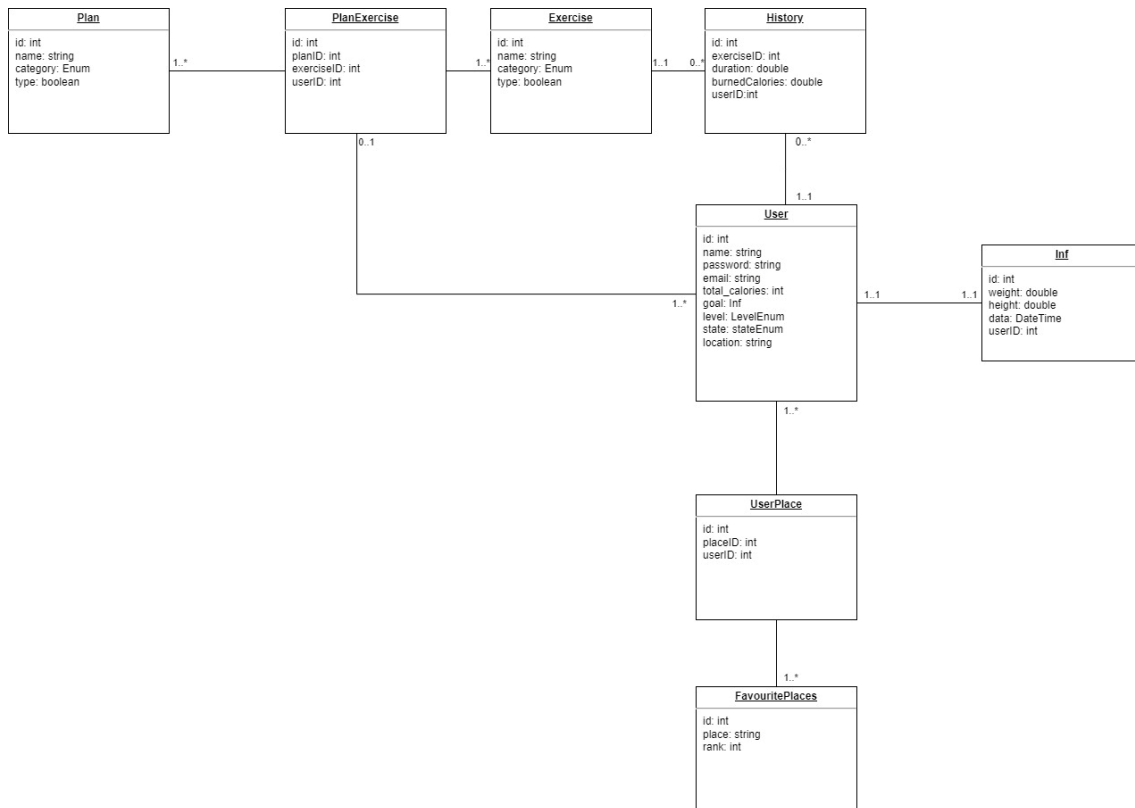


Figura 2 - Versão inicial do diagrama ER da base de dados

A base de dados foi planeada do ponto de vista relacional, resultando nas seguintes tabelas:

- Exercise, para guardar os exercícios disponíveis ao utilizador para ele treinar;
- History, para guardar os dados dos exercícios que o utilizador foi treinando ao longo do tempo;
- FavouritePlaces, para guardar os lugares favoritos que um utilizador avalia na app;
- User, para guardar os dados dos utilizadores da aplicação;

A versão inicial da BD ainda tinha mais entidades, contudo com o decorrer do projeto e devido a restrições de tempo a BD projetada foi simplificada e os extras planeados foram removidos. Essas entidades seriam:

- Information, que guardaria informação muito específica de um utilizador ao longo do tempo, por exemplo, o peso dele ao longo de X meses;
- Plan e PlanExercise, sendo a segunda a tabela intermédia, estas tabelas serviriam para que o utilizador pudesse ter planos de treino personalizados ao seu gosto com os seus exercícios de eleição;

Toda a base de dados é acedida em Threads separadas, não congestionando na Main Thread e é sempre usado o Live Data e o padrão observador para a visualização de todos os dados da aplicação.

Para uso do LiveData e do padrão observador, foi utilizado um repositório para acesso aos Dao's e um SharedViewModel para que a aplicação aceda ao mesmo repositório.

Funcionamento da Aplicação

Funcionamento geral:

Como qualquer aplicação desenvolvida em Android, esta exige que o utilizador tenha uma conta registada na aplicação, caso este não a tenha poderá criar uma nova. Após este estar autenticado na aplicação, irá ser reencaminhado para a página inicial onde lhe serão pedidas permissões de acesso à sua localização para garantir a possibilidade de uso do mapa e da localização de ginásios perto de si. Ainda no componente principal do mapa, o utilizador tem a possibilidade de escolher um dos vários pontos do mapa, visualizar as suas informações específicas e o avaliar, quer seja para ele ou para a comunidade envolvente poder ver a sua classificação, o utilizador pode ainda pesquisar por locais específicos a partir da rua e a aplicação irá mostrar o ponto no mapa e traçar a linha de a nossa posição até ao local. Neste componente é também possível visualizar as condições meteorológicas do local onde nos encontramos.

Na barra de tarefas o utilizador tem ainda a opção de ir à sua área pessoal, nesta área encontrará um pequeno menu com as seguintes opções:

- Profile, fragmento onde se encontrarão as informações pessoais do utilizador, assim como o seu progresso na app;
- Start Training, aqui o utilizador irá encontrar a lista dos seus exercícios e poderá escolher 1 deles para que possa começar um treino. Por defeito as opções que terá sempre disponíveis é a de Alongamentos(manual) e Jogging(automático);
- Training History, aqui serão gravados todos os treinos que o utilizador fez, e os dados de cada treino;
- Community Favourites, aqui poderá ver os favoritos de toda a comunidade, os locais e a sua cotação;

Funcionamento Específico e considerações:

No desenvolvimento deste projeto foram aplicadas certas medidas que faria sentido no contexto do negócio proposto no enunciado e consequentemente aplicadas técnicas de desenvolvimento que fornecessem uma resposta eficaz a cada problema.

Especificamente foram considerados os seguintes pontos:

- Garantir que no registo de um utilizador, o mesmo fosse criado na base de dados Room e no Firebase simultaneamente, pois é importante guardar os registos na base de dados para os poder visualizar e modificar, e no Firebase para fazer a autenticação a qualquer altura e eficientemente.

De modo a cumprir as boas práticas foi utilizada uma Thread que corre em conjunto na secção da criação de conta pelo Firebase, e esta Thread irá adicionar o utilizador na base de dados Room;

- Após feita a autenticação do utilizador é feita uma pesquisa pelo mesmo utilizador para que os dados do mesmo possam ser visualizados e tratados, maioritariamente na zona de perfil, de histórico e dos favoritos dele;
- Todos os dados são carregados e atualizados automaticamente, dando uso ao LiveData e ao padrão observador;
- Todas as consultas de dados, inserções e atualizações são feitas com o uso de Threads;
- As recycler views também são atualizadas com o uso do LiveData e o padrão observador;
- Para o exercício manual de alongamentos, recorreu-se ao sensor de cronómetro, para contar o tempo, este é iniciado e reinicializado sempre que haja um clique no botão, assim que este é parado e reinicializado, é calculada as calorias fazendo uma multiplicação de um valor constante com o número de tempo gasto no exercício;
- Para o exercício automático de Jogging, recorreu-se ao sensor de acelerómetro, já que o sensor de contador de passos não é suportado por todos os dispositivos, isto é, de forma a garantir que todas as máquinas têm acesso à funcionalidade, os passos são contados de acordo com o movimento do corpo do utilizador, são também calculados os metros percorridos (cada passo equivale a 0.762 metros), e também são calculadas as calorias;
- Foi utilizado o sensor de localização (GPS) para obter a localização do utilizador;
- A API, HereMaps, serviu para mostrar o mapa na aplicação e a OpenWeatherApi para mostrar a meteorologia;
- Em ambas as API's referidas acima são implementas com recurso à biblioteca Retrofit. São criados endpoints e feitos pedidos de acordo com as necessidades da aplicação, por exemplo, um pedido constantemente usado é a visualização de ginásios perto do utilizador.

Funcionalidades Implementadas

Obrigatórias

De acordo com o pedido no enunciado, todos os objetivos e funcionalidades obrigatórias foram desenvolvidas com sucesso, não havendo qualquer falha detetada durante o período de testes.

- Suporte para tablets foi desenvolvido e encontra-se funcional, é usado quando abrimos a app a partir do emulador de tablet;
- Os fragmentos é o que mais usamos no design, apenas temos 3 atividades para carregar fragmentos, todo o resto é fragmentos;
- Usamos as RecyclerView e respetivos adapters em tudo que é listas de visualização, como por exemplo: exercícios, histórico de exercícios e locais favoritos;
- Toda a nossa base de dados Room, que por si está desenvolvido com LiveData, e todos os acessos aos dados são feitos com o padrão observador;
- Usamos Threads sempre que seja preciso interagir com a base de dados, quer seja a adicionar ou atualizar um certo dado num registo;
- Podemos encontrar o uso de notificações em vários locais do programa, até notificações inteligentes, tal como a notificação com o cronómetro no exercício de alongar, que contém ainda um botão para parar o exercício, também aparece notificação ao adicionar locais aos favoritos;
- Usamos os sensores de localização como falado posteriormente;
- Usamos o retrofit na utilização dos pedidos feitos às API's de mapas;
- Toda a autenticação é feita com o Firebase;

Bonificação

- ✓ Utilização de API's de meteorologia;
- ✓ Sistemas de ranking do utilizador;
- ✓ Possibilidade o utilizador armazenar os seus locais favoritos pessoais numa RecyclerView e ao mesmo adicionar para os restantes utilizadores poder ter acesso
- ✓ Suporte de dois idiomas (Inglês e Português)

O sistema de ranking acima referido vai distinguir os utilizadores em diferentes níveis. Os níveis são atualizados consoante as calorias gastas pelo utilizador:

- Beginner, nível inicial;
- Intermediate, assim que o utilizador gastou mais de 4100 calorias;
- Advanced, assim que o utilizador gastou mais de 10000 calorias;
- Superstar, assim que o utilizador gastou mais de 25000 calorias;
- GodLike, assim que o utilizador gastou mais de 50000 calorias;

Conclusão

Com o término deste trabalho a impressão foi que se conclui com sucesso, desenvolveu-se uma aplicação interessante e que cada elemento do grupo conseguiu evoluir os seus conhecimentos da plataforma Android em vários níveis.

Este trabalho serviu como consolidação dos conteúdos lecionados nas aulas da unidade curricular, portanto devido ao trabalho desempenhado o grupo avalia o seu desempenho como positivo.

Em termos de autoavaliação o grupo considera que merece um **17**.

Bibliografia

<https://developer.android.com/studio/intro>

<https://material.io/develop/android>

<https://square.github.io/retrofit/>

<https://apidocs.geoapify.com/#docs>

<https://openweathermap.org/current>

<https://developer.here.com/documentation/android-sdk-explore/>

<https://bumptech.github.io/glide/>

<https://app.diagrams.net/>

<https://www.tailorbrands.com/pt-pt/logo-maker>

<https://docs.oracle.com/javase/8/docs/api/index.html>