

Technical Project Report - Android Module

replace here with your project title

Siga

Cadeira: Computação Móvel

Data: Aveiro, 25/01/2022

Estudantes 88861 Pedro Alves
89293 Rui Santos

Project abstract: A aplicação Siga é um rede social que permite aos utilizadores partilhar momentos com outros utilizadores. Com a funcionalidade inovadora de criar e partilhar eventos. Para além da informação de cada evento, o utilizador tem acesso a um mapa que apresenta os eventos perto de si.

Table 1.

Report contents:

1 Conceito da aplicação

2 Solução implementada

Visão da arquitetura

Interações implementadas

Limitações do projeto

3 Conclusões e recursos do projeto

Lições aprendidas

Recursos do projeto

1 Conceito da aplicação

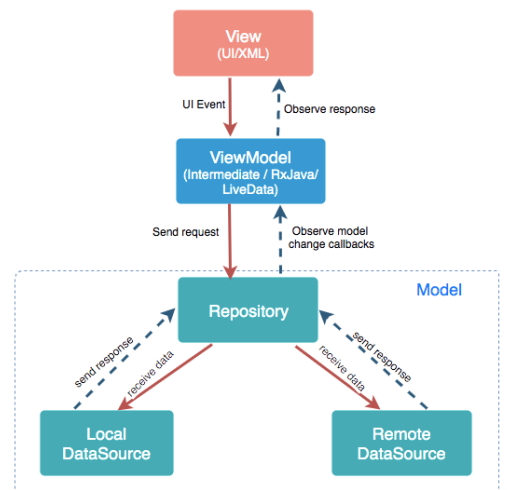
A aplicação Siga surge da necessidade de conhecer os eventos que estão próximos do utilizador de um modo intuitivo e dinâmico, com a vertente de ser também uma rede social, que possibilita partilhar eventos e publicações com amigos. Deste modo, o utilizador alvo é qualquer pessoa que pretenda conhecer e frequentar novos eventos ou dar a conhecer um evento criado por si.

Com esta aplicação, o utilizador irá contar com um mapa que lhe permite ver os eventos em seu redor, uma página que lhe permite criar os seus próprios eventos, que pode partilhar com quem desejar e obter informação da aderência ao mesmo. Para complementar o utilizador pode partilhar fotos e publicações com os outros utilizadores.

2 Solução implementada

Visão da arquitetura

A aplicação Siga está dividida em três camadas. A camada referente à interface gráfica ("view") é onde se encontram todos os fragmentos e atividades que representam as páginas mostradas ao utilizador. O armazenamento de dados é feito na camada "model", nesta camada existem duas fontes de armazenamentos uma na "cloud" e uma "offline", para esse efeito é utilizado o firebase e o SQL lite respetivamente. Para fazer a interligação dos dados com a interface gráfica é utilizada a camada "viewModel". Baseamos-nos na arquitetura MVVM, especialmente no diagrama que é apresentado a seguir.



1

Figura 1.

Como foi referido a arquitetura implementada foi a MVVM e isso levou-nos a organizar o código do projeto (figura 2) respeitando essa arquitetura. Na pasta “mode” encontramos o “backoffice”, utilizamos o DAO para a base de dados local e declaramos para o efeito entidades (pasta “entities”) que correspondem a tabelas na base de dados SQL. Para além do DAO usamos armazenamento remoto que corresponde ao ficheiro “RemoteDatabase.kt”. O ficheiro “AppRepostitory.kt” é equivalente ao “Repository” da figura 1 e possui funções para comunicar tanto com a base de dados local como a na “cloud”.

A pasta “view” é onde se encontram todos os elementos da interface gráfica do projeto, usamos fragmentos e atividades. Por fim, temos a pasta “viewmodel”, que é responsável por interligar a interface gráfica com o repositório. O viewmodel vai ser chamado sempre que for necessário comunicar entre a interface gráfica e o “backoffice”. O objeto “ViewModel” é criado com recurso ao padrão fábrica.

No que diz respeito ao armazenamento na “cloud”, são guardados objetos referentes a eventos, publicações e utilizadores. Na base de dados local armazenamos apenas as publicações dos utilizadores. Sempre que é adicionada uma publicação pelo utilizador a mesma é adicionada às duas bases de dados. Quando o utilizador fica sem internet apenas consegue ver as publicações desde a última vez que teve acesso, recorrendo para isso à base de dados local (esta parte de sincronização entre a base de dados local e remota não foi concluída na totalidade).

Para mostrar os eventos que se encontram na zona próxima do utilizador foi usada a API Google Maps.

¹ <https://chintankhetiya.wordpress.com/2021/06/07/mvvm-architecture-all-in-one/>

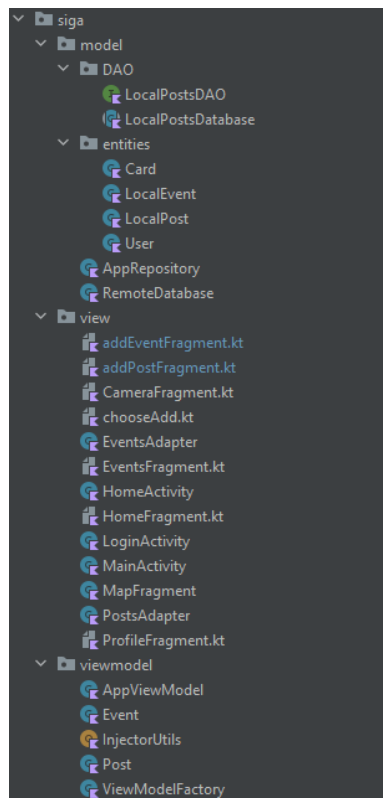


Figura 2.

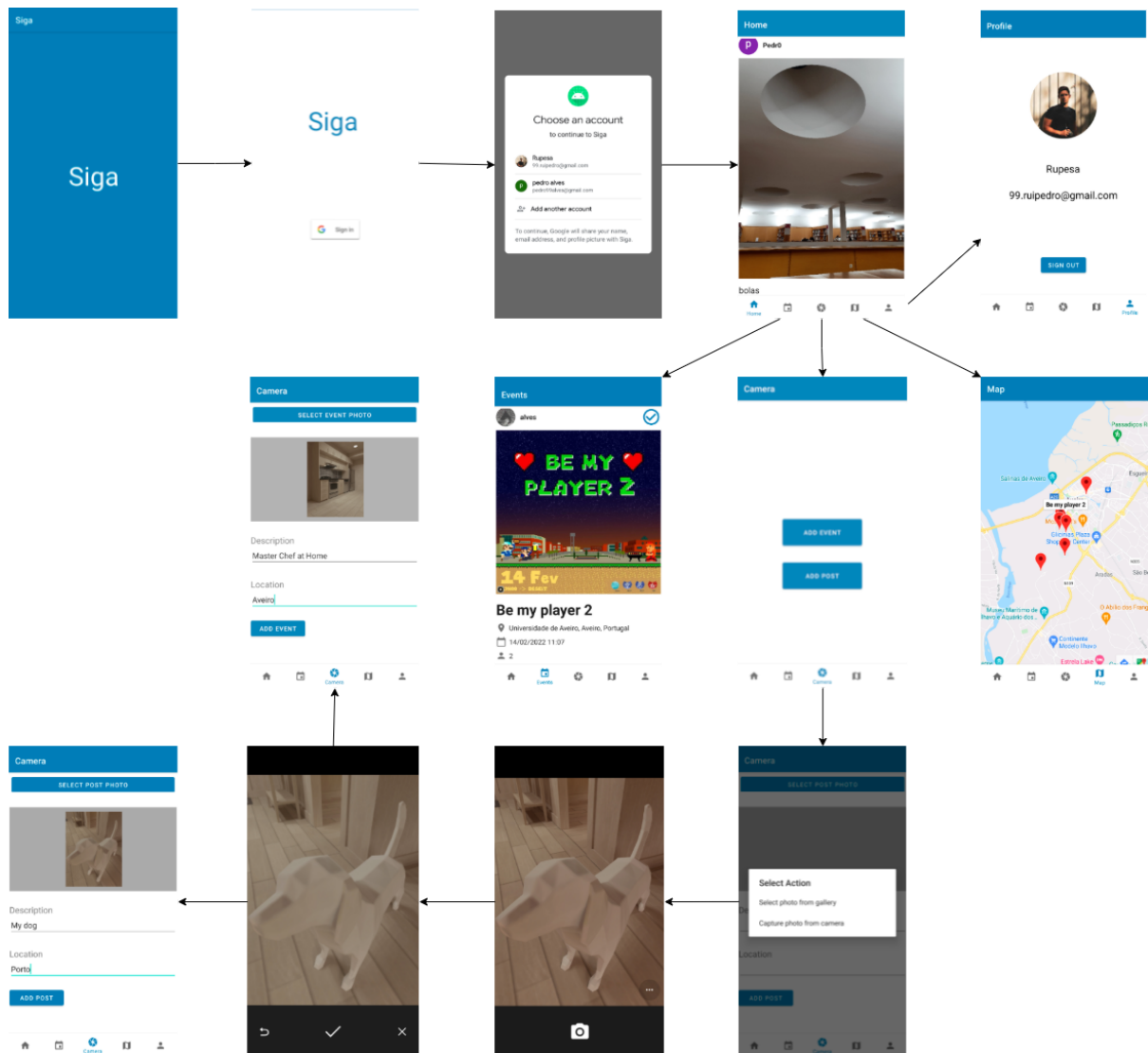
Interações implementadas

Quando o utilizador abre a aplicação Siga pela primeira vez é apresentado o Splash Screen e de seguida a página de autenticação. Aqui, o utilizador poderá autenticar-se com a sua conta Google de uma forma simples e descomplicada.

Após autenticar-se, é apresentado ao utilizador a página Home, onde poderá ver as publicações que os seus amigos fizeram. Através da barra de navegação, o utilizador poderá abrir outras páginas da aplicação, tal como a página Events. Nesta página, é possível conhecer os eventos que irão ocorrer e verificar mais informações sobre os mesmos, como a localização, a sua data e o número de pessoas que confirmaram. O utilizador também poderá confirmar a sua presença no evento clicando no *check* no canto superior direito do evento que irá obter a cor azul e o número de pessoas irá aumentar. Na página Map o utilizador também poderá ver os eventos mais próximos de uma forma simples.

Para fazer um novo *post* ou anunciar um novo evento, o utilizador deve seleccionar o pretendido na página Camera, seleccionar a forma de input da imagem, que pode ser através da sua galeria de imagens ou a partir da câmara, e escrever os detalhes sobre o seu novo *post* ou evento.

Por fim, o utilizador poderá ver as informações acerca do seu perfil e fazer *sign out* se assim o pretender.



Limitações do projeto

No projeto ficaram duas coisas por terminar. A primeira foi a sincronização entre a base de dados local e remota. E a segunda, era usar a API Google Places para recomendar ao utilizador uma localização quando o mesmo criava uma publicação ou evento. Apesar da tentativa de realizar estas duas tarefas o tempo começou a escassear e fomos dando prioridade ao desenvolvimento das funcionalidades base da nossa aplicação.

3 Conclusões e recursos do projeto

Lições Aprendidas

Na totalidade o projeto foi desafiante, a arquitetura mvvm no início foi um pouco difícil de perceber, mas depois de estudarmos o conceito percebemos o seu valor e não hesitamos em implementar. Outro aspecto desafiante foi no desenvolvimento da interface gráfica, começamos por implementar tudo com “Activities”, mas apercebemos-nos que tínhamos problemas devido a isso e passamos a integrar fragmentos.

O grupo achou que o desenvolvimento do layout da aplicação mostrou ser bastante simples e intuitivo, o que contrasta com outras ferramentas como o Flutter. Por outro lado, achámos que o Flutter tem uma abordagem mais simples relativamente à implementação de um *RecyclerView*.

Recursos do projeto

Resource:	Available from:
Code repository:	https://github.com/Rupesa/siga_android
Ready-to-deploy APK:	https://github.com/Rupesa/siga_android/tree/main/siga/app/release

Table 2.

Referências

- <https://developer.android.com/docs>
- <https://chintankhetiya.wordpress.com/2021/06/07/mvvm-architecture-all-in-one>
- <https://medium.com/developer-student-clubs/android-kotlin-camera-using-gallery-ff8591c26c3e>