

Sistemas Móveis e Aplicações

Relatório do Trabalho Prático

Ano Letivo 2022/2023

StudyPlanTasker



Trabalho realizado por:

- Diogo Mestre, nº 48973
- Rodrigo Alves, nº 48681

1. Introdução

Este relatório tem como objetivo descrever o trabalho prático que foi desenvolvido no âmbito da cadeira de Sistemas Móveis e Aplicações. O projeto foi escolhido e pensado pelo grupo e posteriormente foi apresentada esta proposta ao docente responsável da cadeira. Este trabalho tem como finalidade pôr em prática os conteúdos lecionados ao longo do semestre.

1.1. StudyPlanTasker

A StudyPlanTasker foi desenvolvida com o propósito de simplificar a vida tanto dos estudantes como de qualquer pessoa que precise de realizar múltiplas tarefas. Esta aplicação permite que o utilizador insira as suas tarefas de acordo com o estado em que estas se encontram:

- TO DO → Neste estado estão presentes todas as tarefas que ainda não foram realizadas pelo utilizador, ou seja, encontram-se por implementar.
- DOING → Neste estado encontram-se as tarefas que estão a ser realizadas pelo utilizador.
- DONE → Estado que representa as tarefas que já foram concluídas.

2. Funcionalidades

A aplicação possui as seguintes características:

1. Visualização dos detalhes de cada tarefas, selecionada pelo utilizador na interface da app;
2. Criação de novas tarefas, com todas as informações relativas inseridas pelo utilizador;
3. Atualização de uma tarefa, incluindo o título, a data, a descrição e o estado;
4. Eliminação de uma tarefa;

2.1. Funcionamento da aplicação

No menu principal, as tarefas são exibidas de acordo com seu estado, facilitando a sua visualização e organização. De forma a melhorar a perceção das tarefas na aplicação, decidimos adicionar um símbolo de aviso quando o prazo da mesma estiver próximo de terminar.

3. Conceitos utilizados

Neste ponto vamos mencionar os conteúdos que foram mencionados em sala de aula e que ajudaram no desenvolvimento da aplicação.

- RecyclerView → É uma componente da interface do Android que atualmente faz parte do Android Jetpack. É usada para exibir grandes conjuntos de dados de forma eficiente numa lista. A RecyclerView oferece um melhor desempenho, flexibilidade e funcionalidade comparada com outras componentes como por

exemplo a ListView. A principal vantagem da RecyclerView é a capacidade de reutilizar e reciclar os itens da lista conforme necessário, em vez de os estar a criar ou eliminar constantemente, permitindo economizar recursos e melhorar a eficiência.

- **CardView** → É uma componente da interface do Android que atualmente faz parte do Android Jetpack. Fornece uma maneira fácil e flexível de exibir conteúdo em forma de cartão, que utiliza certas formas de design. O CardView é usado para exibir informações ou conteúdo num formato visualmente agradável, semelhante a um cartão físico. Possui sombras subtis, cantos arredondados e pode ter uma cor de fundo personalizada. O CardView é frequentemente usado em listas ou em layouts onde várias informações devem ser exibidas de maneira organizada. A principal vantagem do CardView é a sua capacidade de fornecer uma aparência consistente e agradável.
- **TextView** → É uma das componentes da interface do Android, sendo que é uma das mais utilizadas pelo facto de ser muito simples de utilizar. É responsável por exibir e manipular o texto permitindo apresentá-lo de forma estática e dinâmica.
- **EditText** → É uma das componentes da interface do Android, que permite ao utilizador inserir e editar texto. É semelhante à TextView, mas com a capacidade adicional de entrada de texto.
- **Bases de dados SQL locais no Android**, uma para os utilizadores e a outra para as tarefas.
- **Fragments** → São componentes modulares e reutilizáveis da interface do Android. Podem ser considerados como "sub-ecrãs" ou "blocos de construção" de uma atividade, permitindo criar interfaces flexíveis e dinâmicas. Cada fragment possui um ciclo de vida próprio e a sua gestão é feita por uma atividade. Os Fragments podem ser alterados dinamicamente durante a execução da aplicação.
- **Intent** → É uma componente fundamental no desenvolvimento de aplicações Android uma vez que permite a comunicação e a interação entre diferentes componentes do sistema, como por exemplo atividades e serviços.
- **Binding** → É um recurso do Android que permite estabelecer conexões diretas entre os componentes da interface e as fontes de dados no código. Permite simplificar o processo de atualização da interface, reduzindo a quantidade de código necessário, tornando-o mais legível e fácil de manter.
- **View, ViewModel** → É uma classe fundamental que representa os componentes básicos da interface de uma aplicação. É usada para exibir e interagir com os elementos visuais do ecrã. É responsável por renderizar os elementos da interface no ecrã e capturar as interações do utilizador. Desta forma, é uma classe abstrata, ou seja, não pode ser diretamente instanciada.
- **Adapter** → É responsável por fornecer os dados e criar os elementos visuais no ecrã para cada item de uma lista. É responsável por preencher os layouts dos itens e associar-lhes os seus dados.

4. Activitys implementadas

Foram criadas várias activitys (Classes), sendo que as principais são:

- **MainActivity** → É aqui que se encontra o coração da aplicação, pois nesta são apresentadas todas as tarefas, ou seja, consoante o estado que esteja selecionado pelo utilizador ser-lhe à mostrado todas as tarefas do mesmo. Nesta

activity existem três fragmentos que representam os três estados possíveis já mencionados anteriormente. A seleção do estado é feita na barra no topo do ecrã a partir de botões. Cada fragment é composto por uma RecyclerView, facilitando a apresentação das mesmas. No final do layout encontra-se um menu, que permite sair da app, adicionar uma nova tarefa e ir para a página principal, caso não estejamos nela.

- LoginActivity → Esta é a primeira activity que surge ao iniciar a aplicação e serve para o utilizador conseguir autenticar-se, esta autenticação é validada através da pesquisa na base de dados e verificada se o username e password coincidem com algum dos valores da mesma.
- RegisterActivity → Nesta activity é possível criar uma conta na nossa app, onde o username não pode ser igual a um já existente. Após o registo o utilizador é então direcionado para a página de Login, onde já consegue realizar o mesmo, uma vez que os seus dados já foram inseridos na base de dados.
- Add_Task → Aqui é possível criar tarefas, cada uma tem de possuir o título da mesma, um prazo de realização, uma descrição e um estado, sendo que todos estes valores têm de ser preenchidos, de forma a não causar problemas na aplicação. Também todos os títulos das tarefas têm de ser diferentes, uma vez que a app realiza certas pesquisas com base nos títulos das mesmas. O processo de criação só é finalizado quando se clica no botão representado por um certo.
- TaskInfo → Ao clicar em cada tarefa é apresentado um novo layout com as suas informações, aqui é possível alterar os seus parâmetros e ainda é possível eliminá-la. Para cada uma destas funcionalidades existe um botão para a execução da mesma. Em ambas é necessário realizar alterações à base de dados.
- TaskAdapter → Foi aqui que implementámos como é que a tarefa seria apresentada e que valores iria possuir. Em cada tarefa é mostrado o título, a data final da mesma e os dias até ao final do prazo, estes dias são apresentados com diferentes cores consoante o dia atual, ou seja, se o prazo da mesma for superior à data atual, então é apresentado a verde, caso contrário é apresentado a vermelho. Ainda foi inserido na mesma uma imagem de “aviso”, se o número de dias até ao prazo for inferior a cinco. No caso da mesma se encontrar no estado DONE, então são apenas mostrados o título e o seu prazo.

No desenvolvimento da aplicação apercebemo-nos que ao transitarmos entre activitys era necessário guardar as informações de autenticação do utilizador, de forma a ser possível atualizar/apresentar todas as tarefas do mesmo.

4.1 Layouts/Interfaces



Figure 1 - Login



Figure 2 - Register



Figure 3 - Main



Figure 4 – Task Info



Figure 5 – New Task

5. Objetivos Atingidos

No início do projeto foram colocados certos objetivos que esperávamos desenvolver na aplicação, no entanto quase todos foram elaborados ao longo da criação da app. O objetivo principal da aplicação é ajudar e facilitar a organização na vida das pessoas, sendo que esta app permite gerir e planear a execução das tarefas do dia a dia, tanto profissionais, educacionais ou mesmo pessoais. Desta forma, tentámos construir tanto o design como o modo de utilização da mesma o mais intuitivo possível. Os objetivos que ficaram por concluir foram: Uma página apenas acessível pelo administrador, de forma a gerir os utilizadores, que os mesmos consigam alterar os seus dados de login e ainda o login através da conta google, e assim guardar todas as tarefas criadas na app, no calendário da Google. Atualmente não realizamos a verificação da diferenciabilidade dos usernames, nem dos títulos das tarefas, uma vez que se trata de um protótipo, mas seria implementado se esta fosse lançada no mercado.

6. Obstáculos encontrados

Grande parte dos problemas com os quais nos deparamos foram no nosso conhecimento de Kotlin e com o gradle. Tínhamos inicialmente pensado em utilizar uma firebase para o login, permitindo guardar os dados dos utilizadores e ainda realizar o mesmo através de uma conta da Google. No entanto, após várias tentativas falhadas decidimos não utilizar uma firebase, mas usar uma base de dados SQL local do android para a funcionalidade em questão. O maior problema que enfrentamos no desenvolvimento da firebase foi nas dependências do Gradle, que sempre que as alterávamos o nosso programa encontrávamos um erro fatal. Assim não foi possível implementar uma autenticação com base numa conta da Google.

Tanto a linguagem Kotlin como o Android Studio, mostraram ser obstáculos ao desenvolvimento da app, uma vez que como não tínhamos muitos conhecimentos em relação a estes temas, tivemos de perder mais tempo a realizar um trabalho de pesquisa. O IDE apresentou ter uma elevada complexidade o que também não favoreceu o desenvolvimento da mesma.

7. Conclusão

Em suma no decorrer do semestre foram-nos providenciadas as bases para desenvolver uma aplicação. No entanto foi no decorrer desta que notámos e aprendemos verdadeiramente a trabalhar com o Kotlin e com o Android Studio. Foi com o auxílio do curso apresentado nas aulas, o Android Training Program, e com a pesquisa extensiva realizada por nós em diversos assuntos do Kotlin e em diversas funcionalidades do mesmo que nos foi possível elaborar esta aplicação. Posteriormente será possível introduzir novas funcionalidades, tanto as que não conseguimos desenvolver como novas sugestões/ideias que nos surjam, nomeadamente o envio de notificações quando o prazo de uma tarefa está próximo do fim.