Trabalho Prático Android Kotlin

Relatório técnico

Arquiteturas Móveis

João Vieira | nº 2019127230

Pedro Vieira | nº 2019127241

Sérgio Soares | nº 2016014425

Licenciatura em Engenharia Informática

Ramo de Desenvolvimento de Aplicações

Instituto Superior de Engenharia de Coimbra

Instituto Politécnico de Coimbra

Janeiro 2025

Índice

[Índice de figuras 4](#_Toc185543783)

[Índice de tabelas 5](#_Toc185543784)

[Acrónimos 6](#_Toc185543785)

[1 Introdução 7](#_Toc185543786)

[2 Requisitos da aplicação 7](#_Toc185543787)

[2.1 Requisitos funcionais 7](#_Toc185543788)

[2.2 Requisitos não funcionais 8](#_Toc185543789)

[3 Arquitetura da aplicação 8](#_Toc185543790)

[3.1 Camada de Apresentação 8](#_Toc185543791)

[3.2 Camada de dados 9](#_Toc185543792)

[4 Implementação 9](#_Toc185543793)

[5 Conclusão 9](#_Toc185543794)

[6 Referências 9](#_Toc185543795)

# Índice de figuras

[Figura 1.Arquitetura da aplicação 7](#_Toc185542993)

[Figura 2. Fluxo de dados unidirecional (UDF) 8](#_Toc185542994)

# Índice de tabelas

[Tabela 1. Requisitos funcionais da aplicação 4](#_Toc185458296)

[Tabela 2. Requisitos não funcionais da aplicação 5](#_Toc185458297)

# Acrónimos

|  |  |
| --- | --- |
| Acrónimo | Definição |
| MVVM | *Model-View-ViewModel* |
| UDF | *Unidirectional data flow* |

# Introdução

O presente relatório descreve o desenvolvimento da aplicação *Quizec*, projetada para criar e gerir questionários. Este trabalho foi desenvolvido no âmbito da unidade curricular Arquiteturas Móveis, utilizando Kotlin, Jetpack Compose e Firebase.

A aplicação permite que utilizadores registados criem, editem e respondam a questionários permitindo ainda a criação de perguntas com os seguintes formatos diferentes:

* Perguntas de Sim/não
* Perguntas de escolha múltipla com uma resposta certa
* Perguntas de escolha múltipla com várias respostas certas
* Perguntas de correspondência
* Perguntas de ordenação
* Perguntas de preenchimento de espaços em branco
* Perguntas de associação
* Perguntas com resposta baseada na indicação de palavras

Este relatório documenta os principais passos do projeto, incluindo a análise de requisitos, arquitetura e implementação técnica.

# Requisitos da aplicação

Este capítulo apresenta os requisitos essenciais para o desenvolvimento da aplicação *Quizec.* Os requisitos foram identificados a partir da análise detalhada do enunciado do projeto e têm como objetivo assegurar a funcionalidade e usabilidade da aplicação. Estes foram organizados em duas categorias principais, Requisitos funcionais que descrevem as funcionalidades que a aplicação deve possuir, e os Requisitos não funcionais, que especificam características de qualidade e restrições técnicas.

## Requisitos funcionais

Os requisitos funcionais descrevem as operações e funcionalidades que a aplicação deve oferecer para atender às necessidades dos utilizadores. A Tabela 1 lista os requisitos funcionais identificados assim como o seu estado, implementado ou não implementado

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Estado** |
| RF1 | Gestão de questionários (Criação, edição, remoção) |  |
| RF2 | Gestão de perguntas (Criação, edição, remoção) |  |
| RF3 | Suporta diferentes tipos de perguntas |  |
| RF4 | Reutilização de perguntas |  |
| RF5 | Permite acesso ao questionário imediato ou apenas quando criador autoriza |  |
| RF6 | Ecrã de espera para utilizadores conectados antes da autorização |  |
| RF7 | Suporta restrição de localização |  |
| RF8 | Permite definir se resultados são apresentados imediatamente ou após final do tempo |  |
| RF9 | Criador pode encerrar questionário a qualquer momento |  |
| RF10 | Todos os utilizadores devem estar autenticados |  |
| RF11 | Contém histórico de questionários participados |  |
| RF12 | Permite duplicação de perguntas de questionários participados |  |

Tabela . Requisitos funcionais da aplicação

## Requisitos não funcionais

Os requisitos não funcionais definem os tributos de qualidade, como desempenho e usabilidade.

|  |  |  |
| --- | --- | --- |
| **ID** | **Descrição** | **Estado** |
| RNF1 | Desenvolvimento usando *jetpack Compose* |  |
| RNF2 | Informação armazenada usando serviços (*Firebase*) |  |
| RNF3 | Requer registo e autenticação dos utilizadores |  |
| RNF4 | Idioma principal inglês e suporta português |  |
| RNF5 | Suporta *Portrait* e *Landscape* |  |

Tabela . Requisitos não funcionais da aplicação

# Arquitetura da aplicação

A aplicação *Quizec* segue a arquitetura *Model-View-ViewModel* (MVVM) para garantir a modularidade, escalabilidade facilidade de manutenção. A arquitetura está organizada por duas camadas principais: a camada de dados (*Data layer*) e a camada de apresentação dos dados (*UI layer*) como ilustrado na Figura 1 [1].

A green and blue rectangular object with white text

Description automatically generated

Figura .Arquitetura da aplicação

## Camada de Apresentação

A camada de apresentação é composta por *Views* criadas com *jetpack Compose*, responsáveis por renderizar a interface gráfica e interagir diretamente com o utilizador. Por sua vez, *os ViewModels* são responsáveis por armazenar e gerir o estado das *Views,* permitindo que os dados persistam durante mudanças de configuração.

## Camada de dados

A camada de dados é composta por Repositórios, que encapsulam a lógica de acesso aos dados. As fontes de dados, são responsáveis por lidar diretamente com a origem dos dados, como serviços (*Firebase*) ou bases de dados locais. Os modelos representam os objetos manipulados pela aplicação, assegurando consistência na troca de informações entre camadas.

A aplicação adota o padrão de *design Unidirectional data flow* (UDF) [2] onde os eventos são gerados pela interface de utilizador e são enviados para os *ViewModels.* Os *ViewModels* processam os eventos, atualizam os estados relevantes e notificam as *Views* de qualquer alteração. Por sua vez, as *Views*  exibem os dados atualizados de forma reativa. A Figura 2 ilustra o ciclo de interação entra os *ViewModels* e as *Views* .

A screen shot of a cell phone

Description automatically generated

Figura . Fluxo de dados unidirecional (UDF)

# Implementação

# Conclusão

# Referências

|  |  |
| --- | --- |
| [1] | “Recommended app architecture,” Google, [Online]. Available: https://developer.android.com/topic/architecture#recommended-app-arch. [Acedido em 19 12 2024]. |
| [2] | “Unidirectional data flow,” Google, [Online]. Available: https://developer.android.com/develop/ui/compose/architecture#udf. [Acedido em 19 12 2024]. |