

Escola Superior de Tecnologia e Gestão

Licenciatura em Engenharia Informática

Tecnologias de Computadores

Ano Letivo 2025/26

Trabalhos Prático de TAM

Elaborado em: 2025/11/11

Mariana Magalhães a2022147454

André Sequeira2022123691

Índice

List of Figures	ii
1. Introduction.....	1
2. Arquitetura e Estruturas de Dados	2
3. Transição entre Activities e Dados Trocados.....	3
3.1. MainActivity -> AddQuizActivity	3
3.2. MainActivity -> AddQuestionActivity	4
3.3. MainActivity -> QuizActivity	5
3.4. Observações.....	7
4. Conclusion	8
5. References	9

List of Figures

FIGURA 1 - CLASSE QUIZ.....	2
FIGURA 2 - CLASSE QUESTION	2
FIGURA 3 - LISTAS NA MAINACTIVITY	2
FIGURA 4 - ADICIONAR QUIZ (PORTRAIT).....	3
FIGURA 5 - INCREMENTAÇÃO DO NÚMERO DE QUIZZES (PORTRAIT)	3
FIGURA 6 - ADICIONAR UMA QUESTÃO (PORTRAIT).....	4
FIGURA 7 - INCREMENTAÇÃO DO NÚMERO DE QUESTÕES (PORTRAIT)	4
FIGURA 8 - SELECIONAR A PERGUNTA (PORTRAIT).....	5
FIGURA 9- RESPOSTA ERRADA.....	6
FIGURA 10 - RESPOSTA CERTA	6
FIGURA 11 - BOTÃO BACK (PORTRAIT).....	7

1. Introduction

No âmbito da unidade curricular de Tecnologias e Aplicações Móveis, foi proposto o desenvolvimento de uma aplicação móvel para Android, cujo objetivo principal é a criação e execução de jogos de perguntas e respostas (trivia quizzes). Este projeto será desenvolvido em três fases distintas, sendo que este primeiro trabalho prático se foca na implementação da interface do utilizador e na gestão de dados armazenados em memória, sem persistência.

Nesta primeira fase, a aplicação permitirá a inserção de novos quizzes e questões, a visualização do número de elementos introduzidos, e a resolução de questões individuais selecionadas pelo utilizador. Todas as funcionalidades foram implementadas utilizando o ambiente Android Studio e a linguagem de programação Kotlin, com comunicação entre ecrãs (activities) realizada através de Intents.

O presente relatório descreve o processo de desenvolvimento da aplicação, apresentando as interfaces criadas, o fluxo de navegação entre activities, as estruturas de dados utilizadas e as opções de design adotadas para garantir uma experiência de utilizador clara e intuitiva.

2. Arquitetura e Estruturas de Dados

Para o armazenamento dos dados, foram definidas duas classes principais e utilizadas listas para persistência durante a execução da aplicação.

Temos a classe 'Quiz' que como o próprio nome indica representa um quiz com os seguintes atributos:

- **title:** Título do quiz (String)
- **description:** Descrição do quiz (String)
- **time:** Tempo máximo para resolução em segundos (Int)

```
data class Quiz(val title: String, val description: String, val time: Int)
```

Figura 1 - classe Quiz

A seguir temos a classe 'Question' que tal como o próprio nome indica representa uma questão com os seguintes atributos:

- **question:** Texto da pergunta (String)
- **correctAnswer:** Índice da resposta correta (Int)
- **answers:** Lista de 2 a 4 respostas possíveis (List<String?>)

```
data class Question(val question:String, val correctAnswer: Int, val answers:List<String?>)
```

Figura 2 - classe Question

Na 'MainActivity' foram declaradas duas listas que servem para guardar toda a informação das questions e dos quizzes durante a aplicação:

```
val quizzes = ArrayList<Quiz>()  
val questions = ArrayList<Question>()
```

Figura 3 - Listas na MainActivity

Estas listas são armazenadas na 'MainActivity' como propriedades da classe, acessíveis durante todo o ciclo de vida da aplicação, atualizadas através do retorno das activities de inserção ('AddQuestionActivity' e 'AddQuizActivity'), consultadas pela 'MainActivity' para enviar dados para a 'QuizActivity'

A 'QuizActivity' não acede diretamente às listas. Em vez disso, a 'MainActivity' recupera a questão da lista 'questions' com base no índice fornecido pelo utilizador e envia todos os dados necessários (pergunta, respostas e resposta correta) através de uma Intent.

3. Transição entre Activities e Dados Trocados

A aplicação segue um fluxo bem definido de comunicação entre Activities através de Intents. Cada transição possui objetivos e dados específicos.

3.1. MainActivity -> AddQuizActivity

Ao clicar no botão "Add Quiz" na 'MainActivity', é iniciada a 'AddQuizActivity' através do método 'getQuizz.launch(intent)'.

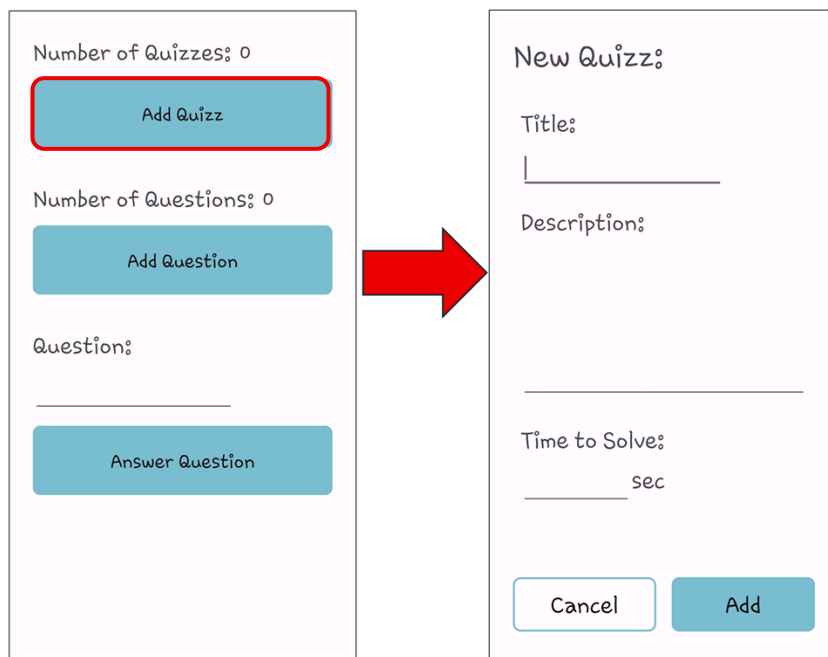


Figura 4 - Adicionar Quis (portrait)

Após o utilizador preencher e validar os dados, a 'AddQuizActivity' termina e retorna à 'MainActivity' com uma Intent contendo os dados do novo quiz. A função "getQuizz" é automaticamente chamada, extrai os dados da Intent e adiciona um novo objeto 'Quis' à lista 'quizzes', atualizando simultaneamente o contador de quizzes na interface.

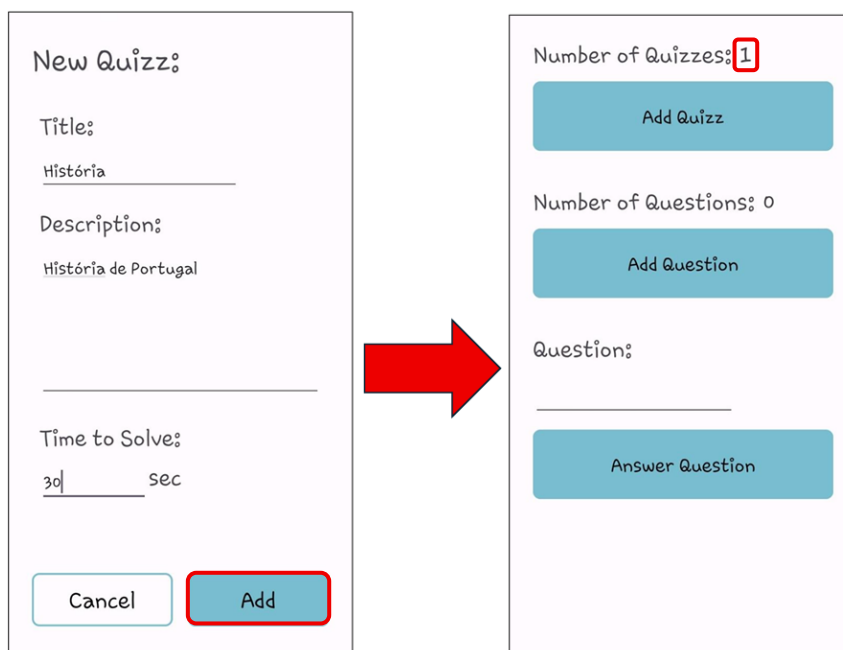


Figura 5 - Incrementação do número de quizzes (portrait)

3.2. MainActivity -> AddQuestionActivity

Ao clicar no botão "Add Question" na 'MainActivity', é iniciada a 'AddQuestionActivity' através do método 'getQuestion.launch(intent)'.

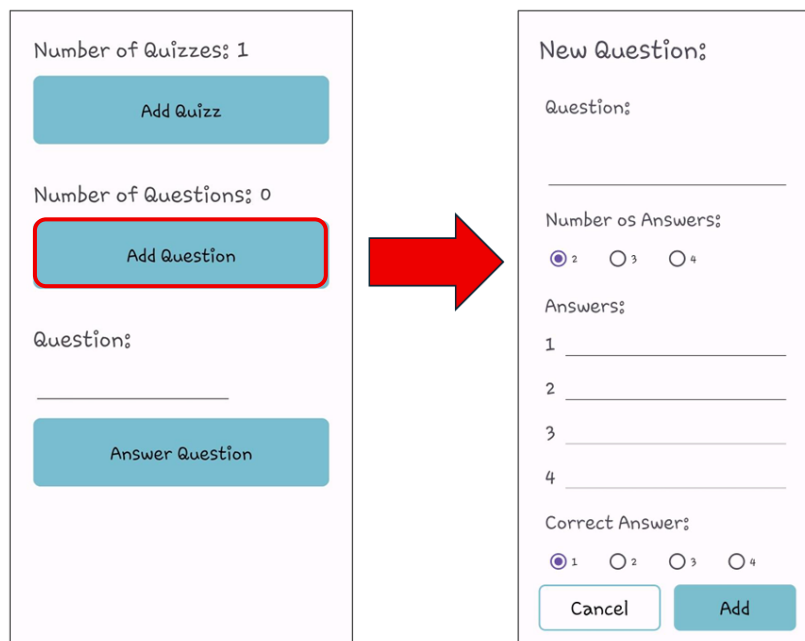


Figura 6 - Adicionar uma questão (portrait)

Esta activity apresenta uma interface dinâmica onde o utilizador pode selecionar o número de respostas (2, 3 ou 4) através de RadioButtons, sendo que a interface se adapta automaticamente, habilitando ou desabilitando os campos de resposta correspondentes. Após preencher a pergunta, as respostas e selecionar a resposta correta, o utilizador pode guardar os dados validados ou cancelar a operação. Ao confirmar, a 'AddQuestionActivity' termina e retorna à 'MainActivity' com uma Intent contendo todos os dados da nova questão. A função 'getQuestion' é automaticamente chamada, extrai os dados da Intent e adiciona um novo objeto Question à lista 'questions', atualizando simultaneamente o contador de questões na interface.

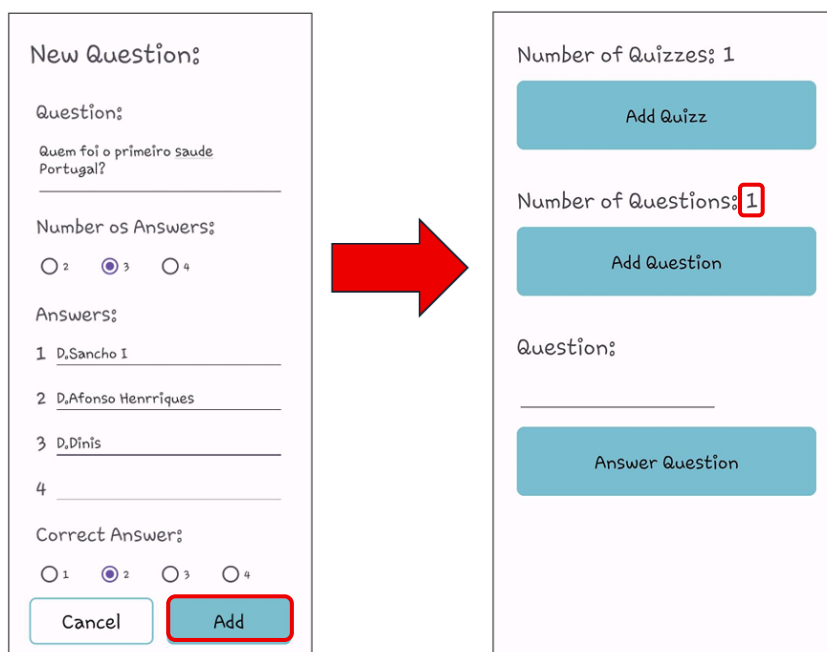


Figura 7 - Incrementação do número de questões (portrait)

3.3. MainActivity -> QuizActivity

Ao clicar no botão "Answer Question" na 'MainActivity', é iniciada a activity 'QuizActivity' através do método 'startActivity(intent)'. No entanto, antes de iniciar a transição, a 'MainActivity' valida o índice da questão inserido pelo utilizador, verifica se é um número válido e se está dentro dos limites da lista de questões existentes. Se a validação for bem-sucedida, a 'MainActivity' recupera a questão correspondente da lista 'questions', extrai todos os seus dados e coloca-os na Intent antes do envio.

A 'QuizActivity', ao ser iniciada, recebe automaticamente esta Intent através do sistema Android e extrai todos os dados necessários para preencher a interface. A activity adapta-se dinamicamente ao número de respostas da questão, tornando invisíveis os botões correspondentes a respostas nulas.

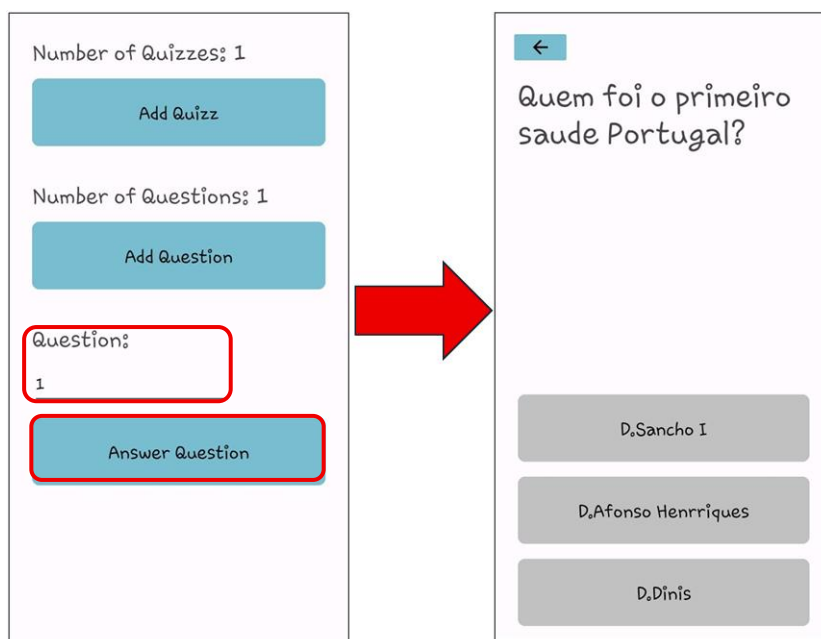


Figura 8 - Selecionar a pergunta (portrait)

Quando o utilizador seleciona uma resposta, a 'QuizActivity' fornece feedback visual imediato. Caso o utilizador selecione a resposta errada ele coloca o botão vermelho, mostra a resposta correta a verde.

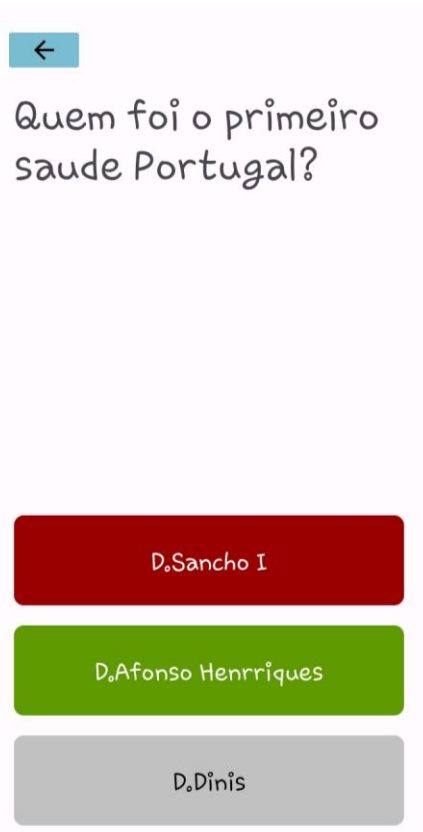


Figura 9- resposta errada

Caso ele selecione a resposta correta vai aparecer o botão a verde



Figura 10 - resposta certa

Ao seleccionar qualquer resposta ele desativa todos os botões para prevenir tentativas adicionais. O utilizador pode regressar à MainActivity a qualquer momento através do botão de retrocesso.

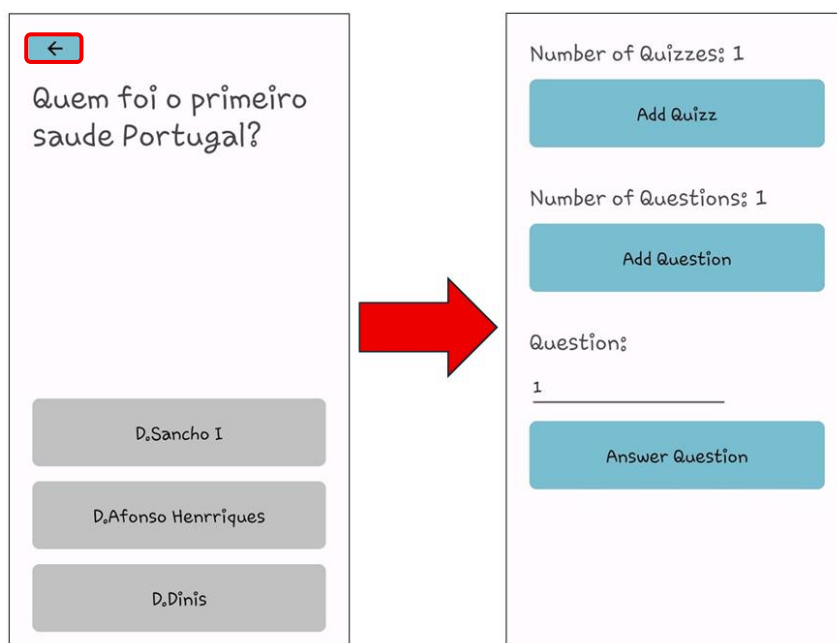


Figura 11 - Botão back (portrait)

3.4. Observações

Durante o desenvolvimento e testes da aplicação, foi observado um comportamento relacionado com a rotação do dispositivo, ou seja, quando o telemóvel está na orientação vertical (portrait) e se adiciona uma questão, esta é guardada corretamente, mas ao rodar o telemóvel para horizontal (landscape), a activity da 'AddQuestionActivity' sofre um reset, os campos ficam vazios e a seleção de número de respostas volta ao estado inicial.

Este comportamento é normal no Android, quando a orientação muda, a activity é destruída e recriada.

Apesar deste comportamento, as funcionalidades principais (adicionar quizzes, questões e resolver questões) continuam a funcionar perfeitamente em landscape.

4. Conclusion

Sendo assim conclui-se que o trabalho prático 1 foi desenvolvido com sucesso, tendo sido atingidos todos os objetivos definidos no enunciado. A aplicação implementa todas as funcionalidades requeridas, inserção de novos quizzes, criação de questões com 2 a 4 respostas, apresentação dos contadores de quizzes e questões introduzidas, e resolução de questões selecionadas pelo utilizador.

As principais dificuldades encontradas durante o desenvolvimento estiveram relacionadas com a curva de aprendizagem de Android Studio e Kotlin, tecnologias novas. Especificamente, a compreensão do conceito de Intents e da comunicação entre Activities representou um desafio significativo inicialmente, mas que foi superado com a implementação bem-sucedida do fluxo de dados entre as diferentes componentes da aplicação.

Em termos de avaliação geral, a aplicação encontra-se funcional e operacional, cumprindo com todos os requisitos solicitados. As interfaces desenvolvidas são intuitivas, permitindo uma experiência de utilizador fluida.

A aplicação representa assim uma base sólida para as fases seguintes do projeto, onde serão adicionadas persistência de dados e funcionalidades de rede.

5. References

Afonso, Francisco. *Tecnologias E Aplicações Móveis (LEI) Kotlin Programming Language*.

2025.

---. *Tecnologias E Aplicações Móveis Android User Interface*. 2025.

“Android Getting Value from Selected Radiobutton.” *Stack Overflow*, Aug. 2013,

stackoverflow.com/questions/18179124/android-getting-value-from-selected-radiobutton/37362772#37362772. Accessed 11 Nov. 2025.

GeeksforGeeks. “How to Change the Background Color of Button in Android Using

ColorStateList?” *GeeksforGeeks*, 21 Oct. 2020, www.geeksforgeeks.org/android/how-to-change-the-background-color-of-button-in-android-using-colorstatelist/. Accessed 12

Nov. 2025.

---. “How to Get Extra Data from Intent in Android?” *GeeksforGeeks*, 15 July 2022,

www.geeksforgeeks.org/android/how-to-get-extra-data-from-intent-in-android/.

Accessed 12 Nov. 2025.

---. “How to Make a Button Invisible in Android?” *GeeksforGeeks*, 31 Dec. 2021,

www.geeksforgeeks.org/kotlin/how-to-make-a-button-invisible-in-android/. Accessed 11

Nov. 2025.

“Kotlin Docs | Kotlin.” *Kotlin Help*, kotlinlang.org/docs/home.html. Accessed 11 Nov. 2025.