

# **RELATORIO DE DESENVOLVIMENTO DO PROJETO:**

## **APLICATIVO “Quiz Master”**

Integrantes: Pedro Emilio Castro Lemos, Paulo Daniel Forti Fonseca, Fabrício Arantes Stecca.

Este relatório documenta as principais dificuldades técnicas encontradas durante o desenvolvimento do aplicativo de Quiz, bem como a divisão de papéis e responsabilidades entre os membros da equipe. O objetivo é registrar os desafios e aprendizados do processo de implementação, focando na arquitetura de dados e na lógica de negócio da aplicação.

### **1. Dificuldades Técnicas Encontradas**

A fase inicial de desenvolvimento, especialmente na comunicação com o banco de dados Firestore, apresentou desafios significativos que exigiram depuração e reestruturação do código.

1.1. Inconsistência na Arquitetura de Dados do Firestore A dificuldade mais crítica foi uma divergência fundamental entre a forma como os dados eram escritos e lidos no banco de dados.

- Problema: As funções de adição de perguntas criavam uma lista "plana" de documentos, enquanto a lógica de leitura esperava uma estrutura aninhada (questions/{categoriaId}/perguntas).
- Impacto: Impedia que qualquer pergunta fosse recuperada do banco de dados, tornando o aplicativo não funcional.
- Solução: Redesenho da estrutura no Firestore e refatoração das funções de escrita para garantir a consistência da base de dados.

1.2. Erros no Mapeamento de Dados (Desserialização) Foram identificadas inconsistências entre os nomes dos campos no banco de dados e as propriedades no modelo de dados (data class Question).

- Problema: Campos como "question" eram usados ao salvar, enquanto a classe Kotlin esperava "questionText".
- Impacto: Resultava em perguntas com texto vazio e respostas incorretas no aplicativo.
- Solução: Padronização dos nomes dos campos em todas as camadas da aplicação, garantindo uma desserialização correta.

1.3. Lógica Frágil na Geração de IDs de Categoria A função para converter nomes de categorias em IDs para documentos do Firestore não tratava adequadamente caracteres especiais e acentuação.

- Problema: A Regex utilizada removia caracteres acentuados em vez de normalizá-los (ex: 'ó' era removido em vez de virar 'o').
- Impacto: Tornava a busca por categorias com nomes acentuados impossível.
- Solução: Implementação de uma função de "slugify" mais robusta para tratar diacríticos corretamente.

1.4. Tratamento de Erros Silencioso O uso de blocos try-catch que retornavam listas vazias sem registrar o erro (log) dificultou a depuração inicial.

- Problema: Em caso de falha, o aplicativo não exibia dados, sem dar nenhuma pista sobre a causa raiz.
- Impacto: Aumentou o tempo necessário para identificar os problemas de arquitetura.
- Solução: Adição de logs de erro para fornecer visibilidade sobre exceções durante o desenvolvimento.

## **2. Divisão de Papéis e Responsabilidades**

A equipe se organizou com focos específicos para construir a aplicação de forma coesa e eficiente:

### **2.1. Pedro Emilio - Foco: Backend & Lógica de Dados**

- Responsabilidades:

- Arquitetura e implementação da camada de dados, utilizando Firebase Firestore como banco de dados.
- Desenvolvimento do FirestoreQuestionRepository, responsável por toda a comunicação com o banco para buscar e gerenciar as perguntas do quiz.
- Identificou e solucionou os desafios críticos de inconsistência de dados, mapeamento e lógica de IDs, que eram impeditivos para o funcionamento do app.
- Criação dos modelos de dados (data class) e da lógica para popular o banco de dados.

## 2.2. Fabricio Arantes - Foco: UI - Fluxo Principal do Quiz com Compose

- Responsabilidades:
  - Desenvolvimento do "coração" da experiência do usuário: o fluxo de jogo do quiz, utilizando Jetpack Compose.
  - Criação da tela de exibição de perguntas e opções, a lógica de seleção de respostas pelo usuário, a navegação entre as questões e a apresentação da tela de resultado final.
  - Integração da sua UI com a camada de dados fornecida por Pedro Emilio para exibir as perguntas e validar as respostas.

## 2.3. Paulo Daniel - Foco: UI - Autenticação & Histórico com Compose

- Responsabilidades:
  - Foco na experiência do usuário relacionada à gestão de contas e visualização de progresso, utilizando Jetpack Compose.
  - Desenvolvimento das telas e da lógica de Autenticação de usuários (telas de login, registro).
  - Implementação da tela de "Histórico", permitindo que os usuários visualizassem seus resultados de quizzes anteriores, consumindo os dados salvos pela lógica de backend.