

# Purchasing Controller



Relatório Prático para a disciplina:

Desenvolvimento de Jogos para Plataformas Móveis

Trabalho realizado por:

Fábio Lopes

## Índice

<b>I. Introdução.....</b>	<b>3</b>
<b>II. Funcionalidade da aplicação .....</b>	<b>3</b>
<b>III. Desenhos, esquemas e protótipos da aplicação .....</b>	<b>3</b>
<b>IV. Modelo de dados .....</b>	<b>4</b>
<b>V. Conclusão e dificuldades .....</b>	<b>5</b>

## I. Introdução

Em resumo, o objetivo deste projeto era criar uma aplicação de gestão de estudos de produtos e despesas chamado PurchasingController. Usando o Android Studio e a linguagem Kotlin, implementei recursos para criar e listar estudos de produtos e grupos de despesas, bem como a funcionalidade de login, logout e criação de conta.

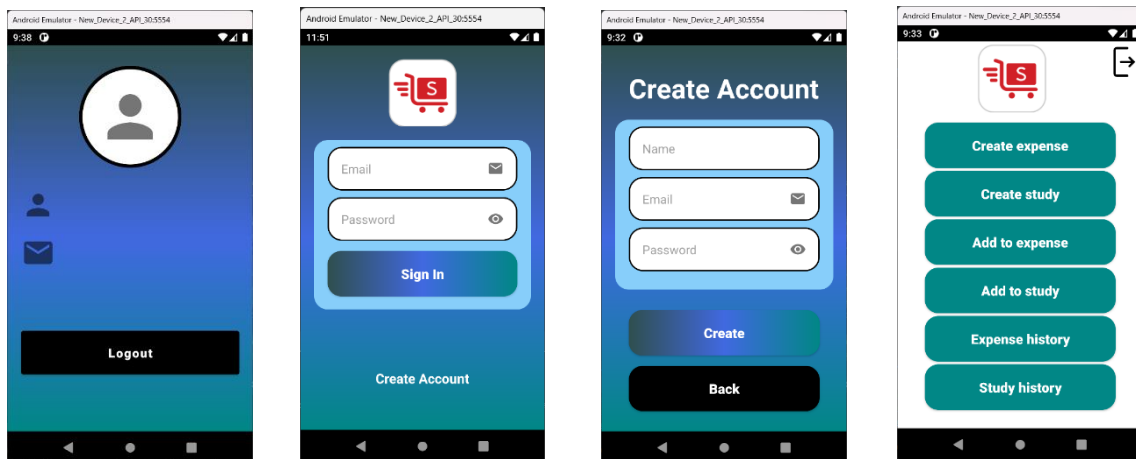
O aplicativo permite aos usuários criar e listar estudos de produtos e grupos de despesas, e também permite acessar essas informações com autenticação de usuário.

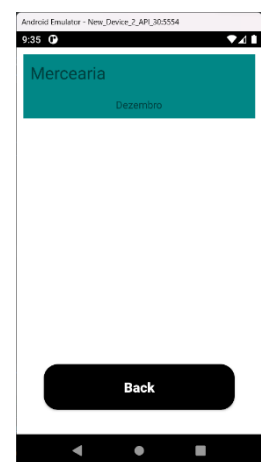
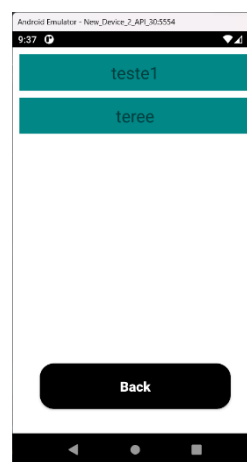
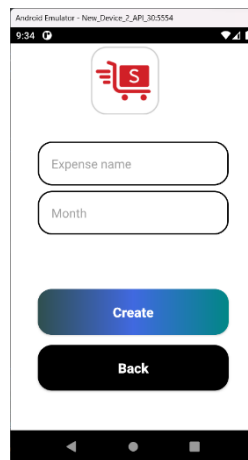
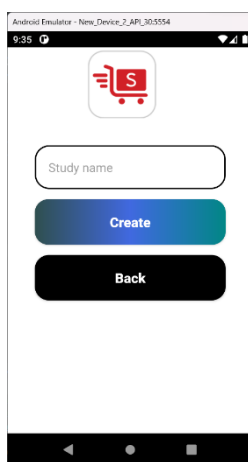
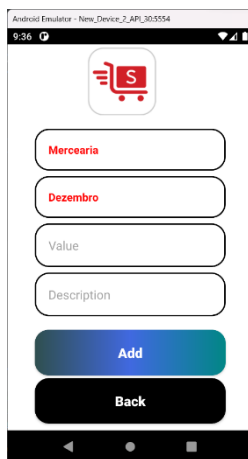
## II. Funcionalidade da aplicação

A aplicação tem uma tela de login como a primeira coisa que o utilizador vê quando abre a aplicação. Se o usuário não tiver uma conta, pode clicar no botão "criar conta" que os levará para uma tela de registo. Depois de criar uma conta, eles serão redirecionados de volta para a tela de login onde podem inserir suas credenciais para se conectar.

Uma vez ligados, os usuários serão apresentados com uma tela com 6 opções: "criar estudo", "criar despesa", "adicionar estudo", "adicionar despesa", "histórico estudo" e "histórico despesa". As opções "criar estudo" e "criar despesa" permitem que os usuários criem estudos e despesas, respetivamente. As opções "adicionar estudo" e "adicionar despesa" permitem que os usuários adicionem novas entradas a estudos e despesas existentes. As opções "histórico estudo" e "histórico despesa" permitem que os usuários visualizem o histórico das suas entradas tanto de estudos como de despesas. Essas funcionalidades são implementadas com a ajuda do Firebase.

## III. Desenhos, esquemas e protótipos da aplicação





#### IV. Modelo de dados

```
data class Study(val name: String = "") {
    fun toHashMap(): HashMap<String, Comparable<Any>?> {
        val study = hashMapOf1(
            "name" to name
        )
        return study as HashMap<String, Comparable<Any>?>
    }

    companion object {
        fun fromHashMap(hash: Map<String, Any>): Study = Study(
            name = hash["name"] as String
        )
    }
}
```

```
data class ExpenseEntry(val name: String = "", val value: String = "", val description: String = "", val month: String = "") {
    fun toHashMap(): HashMap<String, Comparable<Any>?> {
        val ExpenseEntry = hashMapOf(
            "name" to name,
            "value" to value,
            "description" to description,
            "month" to month
        )
        return ExpenseEntry as HashMap<String, Comparable<Any>?>
    }

    companion object {
        fun fromHashMap(hash: Map<String, Any>): ExpenseEntry = ExpenseEntry(
            name = hash["name"] as String,
            value = hash["value"] as String,
            description = hash["description"] as String,
            month = hash["month"] as String
        )
    }
}
```

## **V. Conclusão e dificuldades**

Os resultados deste projeto mostram que é possível criar um aplicativo de gerenciamento de estudos de produtos e despesas usando o Android Studio e a linguagem Kotlin. Além disso, o uso do Firebase como banco de dados tornou o aplicativo escalável e fácil de manter.

Em termos de implicações, acredito que este aplicativo pode ser útil para pessoas que precisam gerenciar suas despesas e estudos de produtos de forma eficiente. Além disso, este projeto contribui para a área de desenvolvimento de aplicativos para dispositivos móveis, mostrando como é possível criar aplicativos úteis e fáceis de usar.

Em termos de metodologia, acredito que a escolha do Android Studio e do Kotlin foi apropriada, e o uso do Firebase como banco de dados foi eficaz. No entanto, encontrei algumas dificuldades com a manipulação de dados do Firebase e necessito de mais pesquisas para tornar esse processo mais fácil. Enfrentei também algumas dificuldades e trabalhar com a linguagem Kotlin, mas consegui superá-las e criar um aplicativo funcional e fácil de usar.

Em geral, penso ter cumprido com o objetivo e acredito que a App PurchasingController com algumas melhorias e outras funcionalidades pode vir a ser útil.