

**Escola Superior de Tecnologia e Gestão**

**CTESP Tecnologias Web e Dispositivos Móveis**

## **Programação de Aplicação Desktop**

**Aplicação de venda de Jogos**

Elaborado por:

Dylan Fialho - 21135

Márcio Martins - 21136

Docentes:

David Fontes

Luís Rosário

Carlos Sancho

03/03/2021



# Índice

Índice de Figura.....	3
Introdução .....	1
Organização do trabalho e ferramentas .....	2
Análise do problema .....	3
Funcionalidades da aplicação.....	4
Desenho da interface da aplicação de baixa fidelidade .....	5
Programação da lógica da aplicação .....	6
Conclusão .....	11

## Índice de Figura

Figura 1 – Entidade utilizador .....	6
Figura 2 -Entidade Jogos .....	6
Figura 3 -Dao utilizadores .....	7
Figura 4 -Dao jogos.....	7
Figura 5 -Item principal de loja .....	8
Figura 6 -Item de cada jogo individual .....	9
Figura 7 – Inserir adaptador .....	9
Figura 8 -Item de Carrinho .....	10

## Introdução

Este trabalho descreve a criação de uma app de em Java no programa Android Studio no âmbito da disciplina de Programação de Aplicação do Lado do Cliente. Esta app tem como objetivo a venda de jogos para os clientes, onde temos os organizados por categorias.

## Organização do trabalho e ferramentas

Após o semestre passado, decidi-mos continuar com o tema de jogos como projeto, tema que foi escolhido pois não havia um tema que nos tivesse chamado a atenção.

Antes de programarmos criamos modelos de baixa fidelidade onde podemos discutir o que iria-mos colocar em cada ecrã. Usámos papel e caneta para os modelos e depois usamos o Adobe XD para fazer uma versão mais próxima do que queríamos criar.

## Análise do problema

Com a criação desta app o objetivo era simples. Quis-se que a app pudesse servir de loja da nossa empresa, onde iremos vender os nossos jogos.

Para isso criamos uma app com vários ecrãs (Register, Login, Store, Highlights, Cart, Payment).

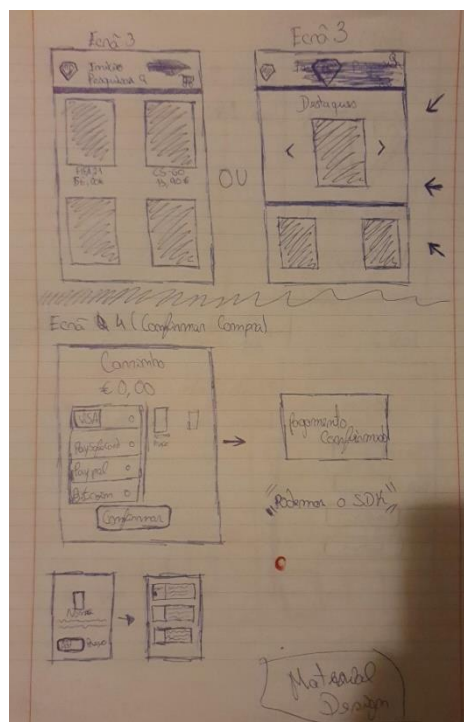
## Funcionalidades da aplicação

Menu de utilizador:

- Ver os jogos;
- Consultar jogos por tipo;
- Adicionar jogos ao carrinho;
- Comprar jogos;



## Desenho da interface da aplicação de baixa fidelidade



## Programação da lógica da aplicação

Nesta aplicação foram usados todos os conceitos aprendidos durante as aulas de SI e de PAC. Tendo isso em conta, ambos os membros quiseram tentar manter a aplicação o mais simples possível, mas também tentar fazer a mesma ter alguma complexidade.

No início da aplicação foi criada uma base de dados em “Room” com 2 entidades, uma de utilizadores e outra de jogos. A tabela dos jogos foi pré-populada com alguns jogos das várias categorias.

```
@PrimaryKey(autoGenerate = true)
private long id;
private String username;
private String password;
private String email;
private String fullName;
private boolean isLoggedIn;
```

Figura 1 – Entidade utilizador

```
@PrimaryKey(autoGenerate = true)
private long id;
private String imgURL;
private String title;
private String description;
private String category;
private float price;
private boolean isInCart;
```

Figura 2 -Entidade Jogos

Apesar de estarem a ser usadas na mesma aplicação, não há uma ligação direta entre estas 2 tabelas.

Com a criação das tabelas veio também a necessidade de conseguir manipular as mesmas. Para isso criou-se 1 “*Dao*” para cada uma das entidades.

```
public interface UsersDao {

    @Query("SELECT * FROM Users WHERE username = :user AND password = :pass")
    Users getByUserAndPass(String user, String pass);

    @Query("SELECT * FROM Users WHERE username = :user AND email = :email")
    List<Users> getUsersByUserAndMail(String user, String email);

    @Insert
    void insertUser(Users users);

    @Update
    void updateUser(Users users);
}
```

Figura 3 -Dao utilizadores

```
public interface GameDao {

    @Query("SELECT * FROM Game")
    List<Game> getAll();

    @Query("SELECT * FROM Game WHERE category LIKE :category")
    List<Game> getAllFromCat(String category);

    @Query("SELECT * FROM Game ORDER BY RANDOM() LIMIT 5")
    List<Game> getHighLights();

    @Query("SELECT * FROM Game WHERE isInCart = 1")
    List<Game> getAllInCart();

    @Update
    public void updateGame(Game game);
}
```

Figura 4 -Dao jogos

Após a criação da base de dados da aplicação, começámos por fazer a maior parte dos ecrãs para conseguirmos ter uma noção do que iria ser necessário em termos de variáveis.

Foram de seguida criados os 3 ecrãs principais, que são:

- Register;
- Login;
- Store.

Foram criados “*layouts*” temporários para estes ecrãs. O ecrã de “*Register*” inclui algumas “*EditTexts*” para o utilizador conseguir escrever os dados que deseja no seu perfil, o ecrã “*Login*” apenas serve para confirmar, através do código, se o utilizador inserido já está criado, e se por acaso o utilizador quiser, pode entrar diretamente no ecrã “*Store*” para efetuar as suas compras sem necessitar conta.

O ecrã “*Store*” foi criado através de uma “*Activity*” já existente dentro do “*Android Studio*” chamada “*Navigation Drawer Activity*”, pois assim apenas foi necessário alterar alguns dados e obteve-se um menu gaveta na aplicação facilmente. Dentro também deste ecrã foi criada uma “*Recycler View*” para obter uma lista de jogos da base de dados e essa lista ser filtrada automaticamente pela categoria que cada jogo tinha. Dentro do item criado para essa base de dados foi passada outra “*Recycler view*” para obter os jogos dependendo da categoria e conseguir demonstrar uma imagem e o preço de cada jogo.

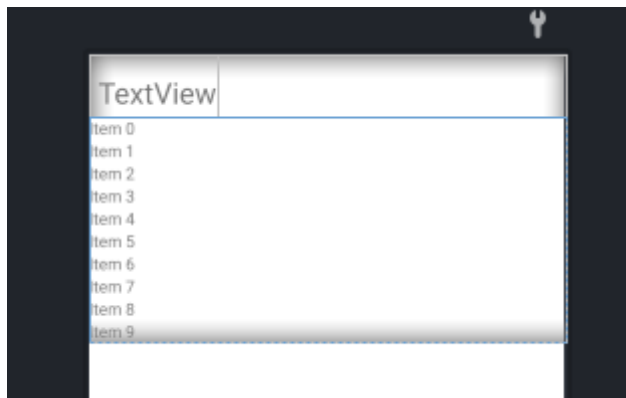


Figura 5 -Item principal de loja

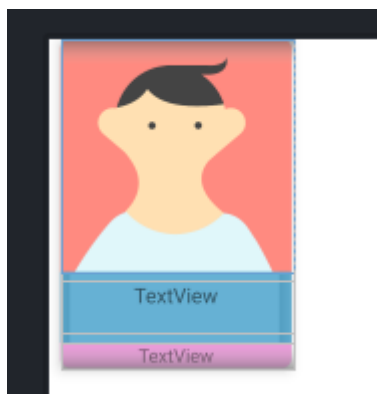


Figura 6 -Item de cada jogo individual

Após criar um adaptador para ambos estes items , foi necessário chamá-lo no fragmento intendido.

```
if(!spinner.getItemAtPosition(position).toString().equals("Todos")){
    recyclerViewMain.setVisibility(View.INVISIBLE);
    recyclerViewCategory.setVisibility(View.VISIBLE);

    GameCategoryAdapter categoryAdapter = new GameCategoryAdapter(getContext(),
        AppDatabase.getInstance(getContext()).getGameDao().getAllFromCat(spinner.getItemAtPosition(position).toString()));

    recyclerViewCategory.setAdapter(categoryAdapter);
    recyclerViewCategory.setLayoutManager(new LinearLayoutManager(getContext()));
}else {
    recyclerViewMain.setVisibility(View.VISIBLE);
    recyclerViewCategory.setVisibility(View.INVISIBLE);
}
```

Figura 7 – Inserir adaptador

Após criar o ecrã “Store” criou-se um “Spinner” que dá a possibilidade ao utilizador de escolher uma categoria e filtrar todos os jogos dessa maneira. Para isso apenas se verifica o texto do spinner e vamos buscar à base de dados os jogos dessa mesma categoria.

A recycler view criada para as categorias foi muito parecida à recycler view criada para os destaques, com a pequena diferença que no ecrã de “Highlights” a recycler view vai buscar 5 jogos ao calhas da base de dados.

Foi criada um ecrã de definições como place holder para ser alterado futuramente se a aplicação tiver continuação.

Após acabar o ecrã “*Store*” começamos o ecrã “*Cart*” que seria o nosso carrinho da aplicação.

Nesse ecrã foi criada uma recycler view que apenas recebe jogos que tenham o boolean “*isInCart*”, que está na entidade de jogos, como verdadeira. Após receber os jogos transforma-os neste item.



*Figura 8 -Item de Carrinho*

O botão deste item serve para remover jogos do carrinho.

Neste mesmo ecrã há um botão para prosseguir para o próximo ecrã que será o ecrã de pagamento, mas neste momento

## Conclusão

Neste projeto foi criada uma aplicação de venda de jogos. Os principais objetivos desta aplicação são a compra de um jogo, e o acesso a informação sobre o jogo que os utilizadores decidirem comprar.

Não foi possível atingir todos os objetivos desta aplicação, mas ambos os alunos aprenderam muito durante o seu desenvolvimento.