

## **XpressTrades** Aplicação Android para troca de itens

PROJETO DE MESTRADO

**Eduardo Manuel Vieira Fernandes**

MESTRADO EM ENGENHARIA INFORMÁTICA



UNIVERSIDADE da MADEIRA

*A Nossa Universidade*

[www.uma.pt](http://www.uma.pt)

novembro | 2014



**XpressTrades**  
Aplicação Android para troca de itens  
PROJETO DE MESTRADO

**Eduardo Manuel Vieira Fernandes**  
MESTRADO EM ENGENHARIA INFORMÁTICA

ORIENTADOR  
Karolina Baras

CO-ORIENTADOR  
Lina Maria Pestana Leão de Brito





---

## **XpressTrades – Aplicação Android de troca de itens**

---

**Eduardo Manuel Vieira Fernandes**

(Licenciado em Engenharia Informática)

*Tese Submetida à Universidade da Madeira para a  
Obtenção do Grau de Mestre em Engenharia Informática*

Funchal – Portugal

novembro 2014



## **Agradecimentos**

---

Começo por agradecer à minha orientadora, Professora Doutora Karolina Baras, e à minha co-orientadora, Professora Doutora Lina Brito, por todo o acompanhamento e indicações prestadas, pela partilha de conhecimento, incentivo e disponibilidade incondicional.

Dedico este trabalho especialmente aos meus pais, por todo o acompanhamento e apoio prestados durante toda a minha vida. Agradeço-lhes por sempre terem valorizado a minha formação e me terem fornecido todas as condições necessárias para me sentir confortável e motivado durante todo o meu percurso académico.

Também quero agradecer à minha esposa, Desireé Fernandes, por me ter auxiliado e motivado sempre que precisei.

Aos meus amigos João Bernardo Fernandes Nunes Luís e José Rafael Freitas Olim Marote, que foram um auxílio precioso para a realização deste projeto, sempre disponíveis e empenhados em ajudar-me.

Às pessoas que testaram os protótipos da interface do utilizador e também aos que testaram a aplicação depois de implementada.

A todos expresso o meu reconhecimento pela ajuda concedida.





## Resumo

---

As pessoas que gostam de comprar videojogos e livros, ao final de algum tempo, verificam que têm muitos destes itens armazenados e que já não os utilizam. Se estas pessoas não tiverem o intuito de criar uma coleção desses itens, irão, provavelmente se desfazer deles, por exemplo deitando-os fora. Neste contexto, apresenta-se uma aplicação para dispositivos móveis que possuam o sistema operativo *Android*, designada de *XpressTrades*. Esta aplicação visa resolver o problema descrito acima, tornando as trocas de jogos e de livros mais fácil, ajudando os seus utilizadores a reutilizarem os seus itens e a os utilizarem como moeda de troca.

Juntamente com esta aplicação foi desenvolvida uma Web API, utilizando a *framework* ASP.NET, a qual é utilizada pela aplicação para esta poder funcionar. Embora este projeto de mestrado se tenha focado no desenvolvimento de uma aplicação especificamente para a troca de jogos e de livros, a aplicação foi desenhada e desenvolvida de forma modular e está preparada para ser estendida à troca de qualquer tipo de itens.

A aplicação *XpressTrades* reúne diversas particularidades que tornarão as trocas de itens mais rápidas e eficientes. Algumas delas são: a apresentação da lista de proprietários ordenados por distância em relação ao utilizador e a apresentação de uma lista de itens recomendados com base no histórico de visualizações de itens realizadas pelo utilizador, ou seja, com base nos seus interesses.

Relativamente à metodologia utilizada no desenvolvimento deste projeto, dado que a ideia surgiu do autor deste trabalho, recorreu-se primeiramente a inquéritos para se averiguar se as pessoas realmente revelavam interesse neste projeto e investigou-se também a existência de aplicações semelhantes. Seguidamente, utilizou-se a técnica de *brainstorming* para gerar as ideias e criou-se protótipos de baixa fidelidade para testar a interface de utilizador. Na fase de implementação, seguiu-se o seguinte ciclo para cada funcionalidade: prototipagem, testes com os utilizadores e correções dos erros detetados nos testes.

**Palavras-chave:** Android, Aplicação Móvel, Livros, Videojogos, Trocas.



## Abstract

---

People who like to buy videogames and books, after some time, verify that they have a lot of these old items that they don't use anymore. If these persons aren't collectors, probably they will discard these old items. On this context, it is presented a mobile application for mobile devices that run Android systems, called XpressTrades. This application aims to solve the problem described above, making the exchange of videogames and books easier, helping its users to reuse their items and use them as an exchange currency.

Together with this application, one Web API was developed, using the ASP.NET framework, which is used by the application to run. Even though this master's project has been focused on developing an application specifically for videogames and books exchange, the application was designed and developed in a modular way and it's ready to be extended to exchange any kind of items.

The XpressTrades application combines several characteristics that make the items exchange faster and more efficient. Some of them are: the presentation of the owners list ordered by the user location distance and the presentation of a recommended items list based on the user-made historical views, that is to say, based on their interests.

Regarding the used methodology for the project development, since the idea came from this work author, firstly it was done the inquiry, to ascertain the users interest on this type of mobile application and secondly it was verified the existence of similar applications. The brainstorming technic was used to create the ideas and low fidelity prototypes were used to test the user interface. For each feature, during the implementation phase, it was used the following cycle: prototyping, user tests and errors correction, detected during the test phase.

**Keywords:** Android, Mobile Application, Books, Videogames, Trades.



# Índice

---

Agradecimentos .....	I
Resumo .....	III
Abstract.....	V
Índice .....	VII
Índice de Tabelas .....	X
Índice de Gráficos.....	X
Índice de Figuras .....	X
Lista de Acrónimos.....	XIV
1. Introdução.....	1
1.1. Contextualização.....	1
1.2. Motivação .....	1
1.3. Problema .....	2
1.4. Público-alvo .....	2
1.5. Contribuições .....	2
1.6. Metodologia .....	3
1.7. Organização da Dissertação .....	4
2. Estado da Arte .....	5
2.1. Introdução .....	5
2.2. Trabalhos Relacionados .....	5
2.2.1. Aplicação “TrocaJogo” .....	6
2.2.2. Aplicação “99gamers”.....	7
2.2.3. Aplicação “TrocasOnline” .....	7
2.2.4. Páginas de Facebook destinadas a trocas .....	8
2.2.5. Conclusão .....	9
2.3. Tecnologias Utilizadas.....	9
2.3.1. Smartphones e Tablets .....	10
2.3.2. Plataforma Android .....	11
2.3.3. Ambientes de Desenvolvimento.....	14
2.3.3.1 Aplicação Android.....	14
2.3.3.2. Web API .....	15
2.3.4. Outras Tecnologias Utilizadas (MS SQL, SMTP, FTP, JSON).....	16
2.3.5. Sistemas de Recomendação .....	16
2.4. Padrões de Desenho .....	18

2.4.1. Padrão State .....	19
2.4.2. Padrão Adapter .....	21
2.4.3. Padrão Observer .....	21
2.4.4. Padrão Singleton .....	21
2.5. Conclusão.....	22
3. Desenho do Sistema .....	23
3.1. Introdução .....	23
3.2. Análise do Problema .....	24
3.2.1. Requisitos Funcionais .....	24
3.2.2. Requisitos Não Funcionais .....	27
3.3. Casos de Uso.....	29
3.4. Diagramas de Atividade.....	31
3.5. Modelo Relacional .....	33
3.6. Wireframes/Mockups.....	36
3.7. Arquitetura .....	37
3.7.1. Arquitetura MVC .....	38
3.7.2. Arquitetura Cliente-Servidor (REST) .....	40
3.8. Conclusão.....	43
4. Implementação do Sistema.....	45
4.1. Introdução .....	45
4.2. Integração de vários idiomas .....	45
4.3. Barra de Ação .....	46
4.4. Funcionalidades da aplicação .....	48
4.4.1. Atividade Registo de Utilizador .....	49
4.4.2. Atividade Login.....	52
4.4.3. Atividade Menu Principal (HomeActivity) .....	54
4.4.4. Atividade Jogos/Livros .....	55
4.4.4.1. Separador Pesquisa .....	56
4.4.4.2. Atividade Pedir para Adicionar Novo Item .....	57
4.4.4.3. Separador Géneros.....	58
4.4.4.4. Separador Itens Recomendados.....	59
4.4.4.5. Separadores Top Desejados/Top Oferecidos.....	60
4.4.5. Atividade Perfil de Item .....	62
4.4.5.1. Separador Perfil do Item .....	62
4.4.5.2. Separadores Proprietários/Interessados .....	63

4.4.5.3. Atividade Detalhes do Utilizador .....	65
4.4.5.4. Atividade Enviar Proposta.....	66
4.4.6. Atividade Definições.....	67
4.4.7. Atividade A Minha Conta .....	69
4.4.8. Atividade “Os Meus Jogos” e “Os Meus Livros” .....	70
4.4.9. Atividade Propostas.....	72
4.4.9.1. Atividade Detalhes de uma Proposta.....	74
4.4.9.1.1. Atividade Feedback .....	76
4.4.10. Serviço.....	78
4.5. Ícone da aplicação.....	79
4.6. Conclusão.....	80
5. Testes e Resultados.....	81
5.1. Introdução .....	81
5.2. Utilizadores de Teste.....	81
5.3. Testes de Usabilidade .....	82
5.4. Estudo de Utilização .....	83
5.5. Avaliação do Sistema.....	84
5.6. Discussão .....	90
5.7. Conclusão.....	91
6. Conclusões.....	92
6.1. Resumo e Avaliação do Trabalho Efetuado .....	92
6.2. Trabalho Futuro .....	94
6.3. Considerações Finais .....	95
Referências .....	97
Anexo A – Inquérito inicial para estudo de viabilidade do projeto .....	104
Anexo B – Diagramas de Casos de Utilização (Versão Completa) .....	111
Anexo C – Diagramas de Atividade dos Casos de Utilização.....	113
Anexo D – Explicação das entidades e atributos da base de dados.....	146
Anexo E – Cenários de utilização para teste de protótipos .....	158
Anexo F – Pedidos da Aplicação Android e Ações Executadas na API.....	161
Anexo G – Inquérito de avaliação aos testes de usabilidade .....	169

## Índice de Tabelas

---

Tabela 1 - Worldwide Device Shipments by Segment (Thousands of Units) [3] .....	10
Tabela 2 - Worldwide Device Shipments by Operating System (Thousands of Units) [3] .....	10
Tabela 3 - Lista de versões do OS Android [18].....	12
Tabela 4 - Lista dos utilizadores de teste .....	82
Tabela 5 - Tabela com as ações da API e respetivas Stored Procedures da base de dados.....	161

## Índice de Gráficos

---

Gráfico 1 – Cota de mercado dos dispositivos pela versão do OS Android [17] .....	12
Gráfico 2 – Cota de mercado dos dispositivos com OS Android por dimensão de ecrã [17] .....	13
Gráfico 3 – Cota de mercado de dispositivos com OS Android por densidade de ecrã [17] .....	14
Gráfico 4 – Funcionalidades mais utilizadas na aplicação, pelos utilizadores de teste..	84
Gráfico 5 – Média da classificação dada à utilidade das funcionalidades atuais da aplicação .....	87
Gráfico 6 – Média das respostas da reação geral à aplicação (relativamente à interface de utilizador) .....	88
Gráfico 7 – Média das avaliações dadas ao <i>layout</i> de visualização da aplicação .....	88
Gráfico 8 – Média das avaliações dadas à terminologia e sistema de informação utilizados na aplicação .....	89
Gráfico 9 – Média das avaliações dadas relativamente à facilidade de aprendizagem de utilização da aplicação.....	89
Gráfico 10 – Média das avaliações dadas às capacidades do sistema .....	90

## Índice de Figuras

---

Figura 1 - Ciclo de vida de uma Atividade Android [26].....	20
Figura 2 - Diagrama de Casos de Uso .....	30
Figura 3 - Diagrama de atividade para o caso "Registo de Utilizador" .....	32
Figura 4 - Diagrama de atividade para o caso "Inicio de Sessão (Login)" .....	33
Figura 5 - Modelo Relacional da base de dados da aplicação .....	35
Figura 6 - Exemplo dos wireframes para smartphone .....	36



Figura 7 - Diagrama MVC da aplicação Android .....	39
Figura 8 - Arquitetura MVC da API.....	40
Figura 9 - Arquitetura Cliente-Servidor (com WEB API MVC) .....	41
Figura 10 - Estrutura de pastas do projeto Android (com exemplo de como colocar vários idiomas) .....	46
Figura 11 - Menu principal da aplicação (em inglês e em português) .....	46
Figura 12 - Barra de ação apresentada na atividade Jogos .....	47
Figura 13 - Adicionando biblioteca “v7 appcompat library” ao projeto .....	48
Figura 14 - Processo para escolher a imagem para o perfil de utilizador.....	49
Figura 15 - Final de registo após opções de localização do utilizador .....	51
Figura 16 - Mensagem de <i>e-mail</i> com código de ativação .....	52
Figura 17 - Atividade Login .....	53
Figura 18 - Janela de Ativação de Conta .....	53
Figura 19 - Recuperação de Palavra-Passe.....	54
Figura 20 - Exemplo da mensagem de recuperação de palavra-passe ( <i>password</i> ) .....	54
Figura 21 - Atividade Menu Principal (Home Screen).....	55
Figura 22 - Atividade Item: Separador Pesquisa .....	57
Figura 23 - Adicionar novo item (Categoria Jogos) .....	58
Figura 24 - Separador “Géneros” .....	59
Figura 25 - Separador “Recomendações” (Categoria Jogos) .....	60
Figura 26 - Separador “Top Desejados” (Categoria Jogos) .....	62
Figura 27 - Separador “Top Oferecidos” (Categoria Jogos) .....	62
Figura 28 - Ações para longo clique sobre um item (Jogo/Livro) .....	62
Figura 29 - Atividade “Perfil do Item” .....	63
Figura 30 - Atividade “Detalhes do Utilizador” .....	65
Figura 31 - Atividade “Enviar Proposta” .....	67
Figura 32 - Atividade “Definições” .....	68
Figura 33 - Atividade “A Minha Conta” .....	70
Figura 34 - Lista de ofertas (Categoria Jogos) .....	71
Figura 35 - Lista de desejos (Categoria Jogos).....	72
Figura 36 - Atividade "Detalhes de uma Proposta" (Recebida) .....	75
Figura 37 - Atividade "Detalhes de uma Proposta" (Enviada) .....	75
Figura 38 – Aceder à atividade “Dar Feedback” .....	77
Figura 39 - Atividade "Dar Feedback" .....	78

Figura 40 - Notificação de Propostas ou Mensagens Novas .....	79
Figura 41 - Ícone da aplicação e dimensões padrão do ícone .....	79
Figura 42 - Diagrama de Casos de Uso (Parte 1) .....	111
Figura 43 - Diagrama de Casos de Uso (Parte 2) .....	112
Figura 44 - Diagrama de atividade para o caso "Ativar Opção: Só Receber Propostas com Jogos das Plataformas Seleccionadas" .....	113
Figura 45 - Diagrama de atividade para o caso "Ativar Opção: Quero Receber Propostas Apenas com Livros que Desejo" .....	114
Figura 46 - Diagrama de atividade para o caso "Ativar Opção: Quero Receber Propostas Apenas com Jogos que Desejo" .....	114
Figura 47 - Diagrama de atividade para o caso "Alterar a Foto de Perfil" .....	115
Figura 48 - Diagrama de atividade para o caso "Alterar a Palavra-Passe" .....	116
Figura 49 - Diagrama de atividade para o caso "Alterar morada" .....	117
Figura 50 - Diagrama de atividade para o caso "Alterar cidade" .....	118
Figura 51 - Diagrama de atividade para o caso "Pesquisar um Item pelo Nome" .....	119
Figura 52 - Diagrama de atividade para o caso "Pedir ao Administrador para Adicionar Novo Item" .....	120
Figura 53 - Diagrama de atividade para o caso "Efetuar Pesquisa por Género" .....	121
Figura 54 - Diagrama de atividade para o caso "Visualizar Lista de Itens Recomendados" .....	122
Figura 55 - Diagrama de atividade para o caso "Visualizar Top Desejados" .....	123
Figura 56 - Diagrama de atividade para o caso "Visualizar Top Oferecidos" .....	123
Figura 57 - Diagrama de atividade para o caso "Adicionar item à lista de desejos" ....	124
Figura 58 - Diagrama de atividade para o caso "Adicionar item à lista de ofertas" .....	125
Figura 59 - Diagrama de atividade para o caso "Remover item da lista de desejos" (Modo 1).....	126
Figura 60 - Diagrama de atividade para o caso "Remover item da lista de desejos" (Modo 2).....	127
Figura 61 - Diagrama de atividade para o caso "Remover item da lista de ofertas" (Modo 1).....	128
Figura 62 - Diagrama de atividade para o caso "Remover item da lista de ofertas" (Modo 2).....	128
Figura 63 - Diagrama de atividade para o caso "Editar Estado de Conservação de um Item" .....	129

Figura 64 - Diagrama de atividade para o caso "Editar Nível de Desejo em Relação a um Item" .....	130
Figura 65 - Diagrama de atividade para o caso "Mover Item da Lista de Ofertas para a Lista de Desejos" .....	131
Figura 66 - Diagrama de atividade para o caso "Mover Item da Lista de Desejos para a Lista de Ofertas" .....	132
Figura 67 - Diagrama de atividade para o caso "Visualizar Lista de Interessados em um Item" .....	133
Figura 68 - Diagrama de atividade para o caso "Visualizar Lista de Proprietários de um Item" .....	133
Figura 69 - Diagrama de atividade para o caso "Aceder ao Perfil Público de um Utilizador" .....	134
Figura 70 - Diagrama de atividade para o caso "Enviar Proposta" .....	135
Figura 71 - Diagrama de atividade para o caso "Aceitar Proposta Recebida" .....	137
Figura 72 - Diagrama de atividade para o caso "Rejeitar Proposta Recebida" .....	138
Figura 73 - Diagrama de atividade para o caso "Cancelar Proposta Enviada" .....	139
Figura 74 - Diagrama de atividade para o caso "Dar Feedback (Avaliar)" .....	141
Figura 75 - Diagrama de atividade para o caso "Enviar Mensagem numa Proposta" ..	142
Figura 76 - Diagrama de atividade para o caso "Consultar Pedidos de Adição de Itens" (Administrador) .....	143
Figura 77 - Diagrama de atividade para o caso "Adicionar Novo Item" (Administrador) .....	144
Figura 78 - Diagrama de atividade para o caso "Verificação de Novas Propostas e Mensagens" (Serviço).....	145

## **Lista de Acrónimos**

---

ADT – Android Development Tools

API – Application Programming Interface

APP – Mobile Application

FTP – File Transfer Protocol

GPS – Global Positioning System

HDPI – High Dots Per Inch

HTTP – Hyper Text Transfer Protocol

ID – Identity

IDE – Integrated Development Environment

JSON – JavaScript Object Notation

LDPI – Low Dots Per Inch

MDPI – Medium Dots Per Inch

OS – Operating System

SDK – Software Development Kit

SGBD – Sistema Gestor de Base de Dados

SMTP – Simple Mail Transfer Protocol

SP – Stored Procedure

SPAM – Sending and Posting Advertisement in Mass

SQL – Structured Query Language

URI – Uniform Resource Identifier

XHDPI – Extra High Dots Per Inch

XML – Extensible Markup Language

XXHDPI – Extra Extra High Dots Per Inch

# 1. Introdução

---

## 1.1. Contextualização

Nos últimos anos muitos países têm passado pela tão falada crise financeira. Normalmente, quando as pessoas têm falta de dinheiro, costumam-se privar de efetuar investimentos em itens de entretenimento, como por exemplo videogames ou livros [1] [2].

Note-se que alguns videogames costumam ter uma história e que muitos jogadores após chegarem ao final, normalmente, perdem o interesse em voltar a jogá-lo. Contudo, há outras pessoas que gostariam de jogar esses mesmos videogames e que, por outro lado, podem possuir outros videogames que interessem ao utilizador do primeiro caso. Em relação aos livros, muitas pessoas apenas costumam ter interesse em lê-los uma vez. Posto isto, o livro fica armazenado numa estante ou numa caixa. Saliente-se que este procedimento acontece para muitas categorias de livros. Neste sentido, esta aplicação tem como objetivo facilitar a troca destes itens proporcionando novas experiências aos seus utilizadores e impedindo que estes itens sejam armazenados em casa sem interesse.

Este trabalho concentrou-se em aumentar a facilidade de troca de itens usados de duas categorias, videogames e livros. É importante que as pessoas continuem a obter itens para o seu entretenimento, pois estes itens ajudam-nas a se manterem saudáveis e é de realçar que o lazer, geralmente, ajuda as pessoas a se abstrair do mundo real durante algum tempo, aliviando assim o *stress* sentido no dia-a-dia.

## 1.2. Motivação

Achou-se importante dotar os utilizadores com uma ferramenta que lhes permita a recirculação de itens usados entre os utilizadores, tentando assim, ajudá-los a obterem os itens que desejam sem terem de despendir dinheiro em artigos novos. Através deste processo de trocas de itens, consegue-se também que os utilizadores não armazenem itens já sem interesse e que, de uma forma inteligente e sem custos, os troquem por outros desejados. Além disto, é importante referir que contribuímos para uma melhoria do meio ambiente, pois estes itens estão a ser reutilizados constantemente [3].

### **1.3. Problema**

Nesta tese de mestrado aborda-se o problema da acumulação de itens em casa que já não tem interesse para os seus proprietários e que poderiam ser utilizados como moeda de troca para estes obterem outros itens. Note-se que ao realizar a troca de itens evita-se a poluição ambiental, pois estes itens já não são eliminados. Assim sendo, este projeto visa ser uma solução para esse problema, promovendo a reutilização dos itens.

### **1.4. Público-alvo**

Este projeto destina-se a todas as pessoas com idade igual ou superior a 18 anos. Note-se que, se os indivíduos com idade inferior a 18 anos, pretenderem usufruir desta aplicação deverão solicitar aos seus pais/familiares que se registem e assumam toda a responsabilidade pelos itens que o menor deseja trocar. Esta restrição apenas é colocada de modo a que todos os seus utilizadores sejam responsáveis pelas informações que fornecem sobre os seus itens e pela troca dos mesmos.

Para além do público-alvo, é importante referir que esta aplicação se destina, neste momento, à troca de videojogos e de livros. Como sabemos, os jogos de consolas costumam ser uma área mais apreciada pelos jovens, enquanto que os livros interessam às pessoas de todas as faixas etárias.

É importante mencionar que a aplicação, atualmente, apenas está desenvolvida para o OS Android, o que implica que todos os que queiram utilizá-la tenham de ter um dispositivo com este sistema.

### **1.5. Contribuições**

As principais contribuições desta dissertação são o estudo do interesse que as pessoas têm em relação à troca de itens, o desenvolvimento de uma aplicação para troca de itens de forma a ir ao encontro do interesse da maioria das pessoas e a avaliação da frequência com que as pessoas trocam os seus itens. A aplicação será o maior contributo, pois irá permitir que pessoas consigam trocar os seus videojogos e livros de forma mais rápida, fomentando assim a reutilização destes itens. Desta forma, os utilizadores desta aplicação poderão usufruir de itens que desejem, mas sem fazerem um investimento a fundo perdido.

## 1.6. Metodologia

De forma a se conseguir concretizar os objetivos propostos, começou-se por realizar um inquérito (Ver Anexo A) acerca do que as pessoas achavam sobre uma aplicação para troca de itens e quais os itens que consideravam mais importantes para trocarem. Através da análise dos dados obtidos foi possível compreender melhor que tipo de aplicação deveria ser desenvolvida.

De um modo informal, realizou-se um *brainstorming* com alguns utilizadores. Note-se que os referidos utilizadores acompanharam todo o processo de desenvolvimento do projeto e foram muito úteis a ajudar a tomar decisões relativamente às funcionalidades que deveriam ser implementadas e de como deveriam ser implementadas. Importa ainda referir que estes utilizadores ajudaram a selecionar as funcionalidades mais importantes que deveriam ser implementadas.

Após as funcionalidades terem sido selecionadas, fez-se uma pesquisa das aplicações que implementassem algumas destas funcionalidades e analisou-se o seu funcionamento. Seguidamente, criou-se o primeiro protótipo de baixa fidelidade da interface do utilizador (*wireframes*). Note-se que estes protótipos foram submetidos a testes de usabilidade com os utilizadores e sofreram um processo iterativo de melhorias.

Após concluir-se o passo anterior com um resultado satisfatório, passou-se à implementação da interface de utilizador já na aplicação Android. De um modo informal, testou-se novamente a interface com os utilizadores e procedeu-se a algumas retificações. O método de desenvolvimento utilizado consistiu num processo iterativo, em que a comunicação com os utilizadores foi essencial e basilar.

No que concerne ao método propriamente dito, este consistiu em desenvolver protótipos, testá-los e melhorá-los de acordo com os erros detetados. Saliente-se que todo este processo foi repetido para cada funcionalidade implementada na aplicação e algumas funcionalidades exigiram mais segmentação. De forma adaptada utilizou-se algumas técnicas do método *Agile Development*, nomeadamente os *Sprints* que permitiram prototipar rapidamente, testar e melhorar o protótipo sem se perder demasiado trabalho efetuado [4].

Nesta aplicação algumas das funcionalidades estão muito interligadas, em alguns casos houve a necessidade de seguir uma ordem lógica de implementação, pois algumas das funcionalidades não podiam ser implementadas antes de outras.

Por fim, realizaram-se os testes à aplicação final com vários utilizadores e efetuou-se correções aos erros detetados durante os testes. A aplicação, num futuro próximo, será lançada para o mercado de aplicações Android, para o Google Play.

### **1.7. Organização da Dissertação**

Esta dissertação está estruturada em vários capítulos que descrevem todos os procedimentos efetuados na realização deste projeto. Estes capítulos são os seguintes:

- Introdução – Aqui realiza-se a contextualização do projeto e descreve-se a motivação, o problema, os objetivos e a metodologia utilizada.
- Estado da Arte – Neste capítulo aborda-se, de forma fundamentada, todas as tecnologias que foram utilizadas no desenvolvimento deste projeto e também as aplicações existentes e/ou semelhantes à deste projeto.
- Desenho do Sistema – Nesta secção, estão patentes os diagramas que descrevem o sistema e o comportamento que este deverá ter, também apresenta-se a arquitetura do sistema.
- Implementação do Sistema – Neste capítulo descreve-se como todas as funcionalidades do sistema foram implementadas e interligadas com a aplicação.
- Testes e Resultados – Este capítulo apresenta os testes realizados à aplicação e os respetivos resultados obtidos.
- Conclusões – No final apresentam-se as conclusões que foram obtidas com a realização desta aplicação e projeta-se o trabalho futuro.



## 2. Estado da Arte

---

### 2.1. Introdução

Ao desenvolver-se um projeto desta dimensão é importante estudar os trabalhos já existentes sobre a área científica em que este projeto se insere. Com esse estudo é possível recolher ideias e verificar-se onde é possível inovar, de forma a procurar satisfazer ainda mais as necessidades e as ambições dos utilizadores. Também é importante pesquisar sobre as tecnologias e as técnicas disponíveis atualmente e selecionar as que mais se adequam ao desenvolvimento do projeto que se pretende desenvolver.

Neste sentido, este capítulo abordará os trabalhos existentes que se relacionam com este projeto e também as tecnologias utilizadas para o desenvolvimento desta aplicação.

De modo a verificar a existência de trabalhos idênticos ao que se pretendia desenvolver, decidiu-se pesquisar aplicações em forma de *websites* e de aplicações nativas dos sistemas operativos Android, iOS e Windows Phone que se assemelhassem à ideia deste projeto. Para isso utilizou-se na pesquisa um conjunto de palavras-chave adequadas para localizar estas aplicações. Após obter-se os resultados essas aplicações foram analisadas de forma detalhada.

Saliente-se ainda que este capítulo também aborda, de forma fundamentada, as tecnologias que foram utilizadas no desenvolvimento deste projeto. Note-se que essas tecnologias contemplam os ambientes de desenvolvimento utilizados, a plataforma utilizada para o desenvolvimento deste projeto (neste caso a plataforma Android) e outras tecnologias essenciais ao desenvolvimento deste como por exemplo: padrões de desenho, certas linguagens de programação e determinados protocolos de comunicação.

### 2.2. Trabalhos Relacionados

Para se verificar a existência de aplicações semelhantes pesquisou-se nos mercados Google Play [5], na Windows Phone Store [6], na Internet (aplicações em forma de *website*) e no iTunes Store [7] de forma a se verificar a existência de aplicações semelhantes à deste projeto. As palavras-chave utilizadas para as pesquisas foram as seguintes: *trade*, *game trade/exchange*, *game trading/exchanging*, *gaming trade/exchange*, *used game market*, *used book market*, *movies trade/exchange*, *music trade/exchange* e também pelos respetivos termos em português.

Através desta pesquisa, efetuada em março de 2014, e posterior análise concluiu-se que nas lojas Itunes Store, Windows Phone Store e Google Play não existem aplicações deste género, no que diz respeito a livros, jogos, música e filmes.

No que concerne à pesquisa realizada na Internet encontrou-se alguns *sites* relacionados com a troca de jogos, filmes, música, livros e outros objetos. Note-se que na apresentação do estudo efetuado considerou-se apenas os resultados mais relevantes. Desta forma, encontrou-se três *websites* dedicados à troca de jogos de consolas videojogos, sendo estes o *website* brasileiro “trocajogo” [8], o *website* norte-americano “99gamers” [9], e o *website* português “trocasonline” [10]. Em relação às redes sociais, mais especificamente no “Facebook”, foi possível encontrar algumas páginas dedicadas à troca e venda de objetos, todavia apenas duas páginas se revelaram as mais importantes, uma para troca de jogos usados [11] e outra para troca de manuais escolares [12], as restantes páginas acabam sendo a repetição dos mesmos conceitos.

De forma a compreender os conceitos mais importantes dos *websites* supramencionados, optou-se por analisá-los com algum pormenor. Neste sentido apresentamo-los em seguida.

### **2.2.1. Aplicação “TrocaJogo”**

A aplicação brasileira “trocajogo” é dedicada à troca de jogos de consolas videojogos mais populares e é uma rede social de troca de jogos. Para os utilizadores usufruírem desta aplicação necessitam efetuar um registo. Cada utilizador tem um perfil que pode completar com uma foto e diversos dados pessoais sendo a morada obrigatória, pois esta serve para ordenar os utilizadores tendo em conta a sua proximidade. Note-se que esta aplicação mostra ainda uma grelha de jogos similares àqueles que se visualizou em detalhe, ou seja, detém uma espécie de recomendação.

No que concerne à usabilidade considera-se que é boa, pois é fácil construir a lista de desejos e a lista dos jogos do qual se é proprietário. A aplicação ainda permite criar comunidades (uma espécie de fóruns) com interesses em comum e contém um blog atualizado regularmente com novidades no mundo dos jogos. A aplicação possibilita também partilhar informação de forma fácil, através das redes sociais mais populares e outros meios.

Quanto aos aspetos que se julga serem mais negativos, a aplicação destina-se somente às pessoas que vivem em território brasileiro, não permite efetuar login com a conta Google ou com a conta do Facebook ou do Twitter e não possui uma versão nativa para dispositivos móveis nem mesmo versão mobile do *website*, dificultando assim a sua utilização em dispositivos móveis.

### 2.2.2. Aplicação “99gamers”

A aplicação “99gamers” é norte-americana e permite fazer *login* através da conta do Facebook, do Twitter ou então através de registo normal, ou seja, utilizando o endereço de *e-mail*. Esta aplicação é uma comunidade constituída apenas por utilizadores residentes nos Estados Unidos da América e no Canadá e que tem como propósito realizar trocas de jogos de consolas.

Saliente-se que antes de fazer o login é apresentada a informação de que não há mensalidades nem custos ocultos, ou seja, que são apenas realizadas trocas. Contudo ao efetuar login verificamos que não é bem assim, isto porque é necessário utilizar dinheiro virtual designado de “*Coins*” (moedas) que após a sua compra não é possível reconverter em dinheiro corrente. Portanto, quando os utilizadores não encontrarem jogos do seu interesse e quiserem desistir da aplicação ficarão com “*coins*” (1 dólar corresponde a 1 “*coin*”) e sem forma de os converter em dinheiro ou jogos desejados. Note-se que desta forma os donos da aplicação *online* ficam com o dinheiro que os utilizadores usam para comprar “*coins*”.

Após fazer login na aplicação pela primeira vez é necessário indicar a morada, criar a lista dos jogos dos quais o utilizador é proprietário e a lista dos jogos que deseja. A aplicação permite a comunicação entre utilizadores e cada jogo tem um perfil com imagem de capa, plataforma, descrição e avaliação do jogo e a lista das pessoas que têm o jogo para “troca” pelas moedas virtuais.

Em suma, como se pode constatar, a ideia da aplicação está interessante e realmente os utilizadores efetuam trocas, mas há um custo associado.

### 2.2.3. Aplicação “TrocasOnline”

A aplicação “TrocasOnline” é portuguesa e só pode ser utilizada por pessoas residentes em Portugal e que tenham idade igual ou superior a 18 anos. Esta permite aos utilizadores

efetuarem trocas de inúmeros artigos, tais como livros, filmes, músicas, jogos. Contudo utiliza a mesma lógica que a aplicação “99gamers” no que diz respeito ao sistema de moedas virtuais e funciona exatamente da mesma forma.

A aplicação apenas permite login se o utilizador se registar na aplicação, não consentindo efetuar login com outras contas, como por exemplo a conta Google ou Facebook. Relativamente às funcionalidades, estas são idênticas as aplicações anteriormente mencionadas, ou seja, possui uma lista de desejos, uma lista de objetos que o utilizador possui para oferecer em troca, um perfil para os objetos e um perfil de utilizador.

O ponto forte desta aplicação é que há um seguro nas transações. Isto é, se algo correr mal e o artigo não chegar, esse valor é reembolsado ao utilizador lesado. Refira-se ainda que a aplicação tem alguma variedade de artigos e é possível trocar muitos tipos de objetos diferentes, o que torna a aplicação mais interessante.

#### **2.2.4. Páginas de Facebook destinadas a trocas**

Na famosa rede social “Facebook” foram encontradas diversas páginas dedicadas a trocas de jogos [11], livros [12], músicas, filmes, etc. As principais vantagens da utilização destas páginas é claramente a liberdade de criação de anúncios e também o número de pessoas que poderão ver esses anúncios. Contudo, apresenta imensas desvantagens, tais como: os utilizadores têm de fazer pesquisas utilizando a própria pesquisa de palavras do *browser* ou do Facebook (mas é pouco eficiente) para verificarem se alguém está a vender ou a trocar algum artigo que lhes interesse; a página torna-se numa espécie de *blog* (o que dificulta a organização eficiente dos artigos que os utilizadores querem trocar ou obter); os anúncios mais recentes ficam em cima e os mais antigos em baixo (dificultando a pesquisa e a visibilidade dos mesmos); o conteúdo dá para filtrar por meses, mas é necessário um esforço acrescido para os utilizadores encontrarem outros utilizadores com um objeto para troca que corresponda ao seu desejo e é também uma página estática, sem algoritmos inteligentes que permitam uma eficiência alta.

Em síntese, apesar de haver muitas pessoas que possam ver os diversos anúncios/propostas, não parece que esta seja a forma mais eficaz de abordar este assunto.

### **2.2.5. Conclusão**

Através deste estudo, constata-se que no mercado atual não há aplicações disponíveis (ou pelo menos não foram encontradas pelos termos pesquisados) para aplicações Android, iOS ou Windows Phone que proporcionem formas rápidas de realizar trocas de objetos, que permitam aos utilizadores visualizarem uma lista de outros utilizadores que possuam o item que desejam e que ordene os utilizadores por menor distância em relação a eles. Assim sendo, podemos assumir que aplicações nativas para este tipo de finalidade são uma área ainda pouco ou nada explorada em qualquer um dos sistemas operativos Android, iOS e Windows Phone.

É importante salientar-se que apesar de existirem aplicações para venda de artigos usados como por exemplo o “OLX” [13] ou o “Coisas” [14], estes não estão direcionados para a troca de artigos, mas sim para a venda. Desta maneira, estes não foram tidos em conta neste estudo sobre trabalhos relacionados. Note-se ainda que estas aplicações tratam os artigos como algo genérico, ou seja, todos os anúncios utilizam uma estrutura padrão, exceto os de automóveis que têm em conta mais algumas especificações. Desta forma, as aplicações supramencionadas fogem do conceito deste projeto que tenta focalizar apenas determinados artigos, neste caso, jogos e livros, permitindo pesquisas mais específicas e onde cada utilizador pode indicar os seus desejos em relação a itens específicos até que a oportunidade de uma proposta interessante apareça.

### **2.3. Tecnologias Utilizadas**

A tecnologia selecionada para o desenvolvimento de um projeto é algo de extrema importância. Como tal, é necessário estudar quais são as melhores tecnologias que existem no momento para o projeto que se pretende desenvolver. Nesse estudo é necessário priorizar e avaliar quais as características mais desejadas para o projeto e cada decisão terá uma grande influência no resultado final do mesmo.

Tenha-se em atenção que as características aludidas poderão ser: a compatibilidade com os dispositivos dos utilizadores; a capacidade de escalabilidade (preparar o sistema para suportar muitos utilizadores em simultâneo sem que se perca desempenho em demasia) e a flexibilidade (programar de forma a minimizar o custo e o esforço na realização de alterações futuras no projeto).

### 2.3.1. Smartphones e Tablets

De acordo com Mark Weiser, o fundador da computação ubíqua, os dispositivos ao longo do tempo tendem a se tornar cada vez mais ubíquos. Por outras palavras, os dispositivos irão realizar muito mais processamento, mas este vai passar cada vez mais despercebido pelo utilizador [15].

Segundo a pesquisa que foi efetuada pela consultora norte-americana Gartner [16], é possível visualizar na Tabela 1 que a maior parte (>75%) dos dispositivos comercializados são *smartphones* (*Mobile Phones*). Na Tabela 2, obtida da mesma fonte, é possível visualizar que de entre os sistemas operativos comercializados, nos dispositivos vendidos, o OS Android apresenta-se em 1º lugar, com uma grande vantagem face aos seus concorrentes e a estimativa para 2015 continua a indicar o crescimento deste. Note-se que em cerca de 2 432 milhões de dispositivos que se espera que sejam comercializados, o OS Android estará presente em cerca de 1 168 milhões (52%).

Tabela 1 - Worldwide Device Shipments by Segment (Thousands of Units) [3]

Worldwide Device Shipments by Segment (Thousands of Units)			
Device Type	2013	2014	2015
Traditional PCs (Desk-Based and Notebook)	296,131	276,221	261,657
Ultramobiles, Premium	21,517	32,251	55,032
<b>PC Market Total</b>	<b>317,648</b>	<b>308,472</b>	<b>316,689</b>
Tablets	206,807	256,308	320,964
Mobile Phones	1,806,964	1,862,766	1,946,456
Other Ultramobiles (Hybrid and Clamshell)	2,981	5,381	7,645
<b>Total</b>	<b>2,334,400</b>	<b>2,432,927</b>	<b>2,591,753</b>

Source: Gartner (June 2014)

Tabela 2 - Worldwide Device Shipments by Operating System (Thousands of Units) [3]

Worldwide Device Shipments by Operating System (Thousands of Units)			
Operating System	2013	2014	2015
Android	898,944	1,168,282	1,370,893
Windows	326,060	333,419	373,694
iOS/Mac OS	236,200	271,115	301,349
Others	873,195	660,112	545,817
<b>Total</b>	<b>2,334,400</b>	<b>2,432,927</b>	<b>2,591,753</b>

Shipments include mobile phones, ultramobiles (including tablets) and PCs

Source: Gartner (June 2014)

Com base nos dados da Tabela 1 e da Tabela 2, podemos afirmar que neste momento o OS Android domina os mercados e tende a continuar durante um longo tempo. Desta forma, tem-se um forte argumento para justificar a opção de desenvolver este projeto para o OS Android, pelo menos na primeira versão, podendo no futuro expandir-se a outros sistemas operativos.

Com o intuito de descobrir se os possíveis utilizadores seguem esta tendência foi realizado um inquérito com vários propósitos, um deles foi o de perceber se as pessoas seguem a tendência do mercado global. De um total de 24 inquiridos verificou-se que 21 deles (88%) possuem, pelo menos um, dispositivo com o OS Android (ver resposta à pergunta P6 do Anexo A), tendo-se verificado uma percentagem muito superior à tendência do mercado global atual para este grupo de inquiridos.

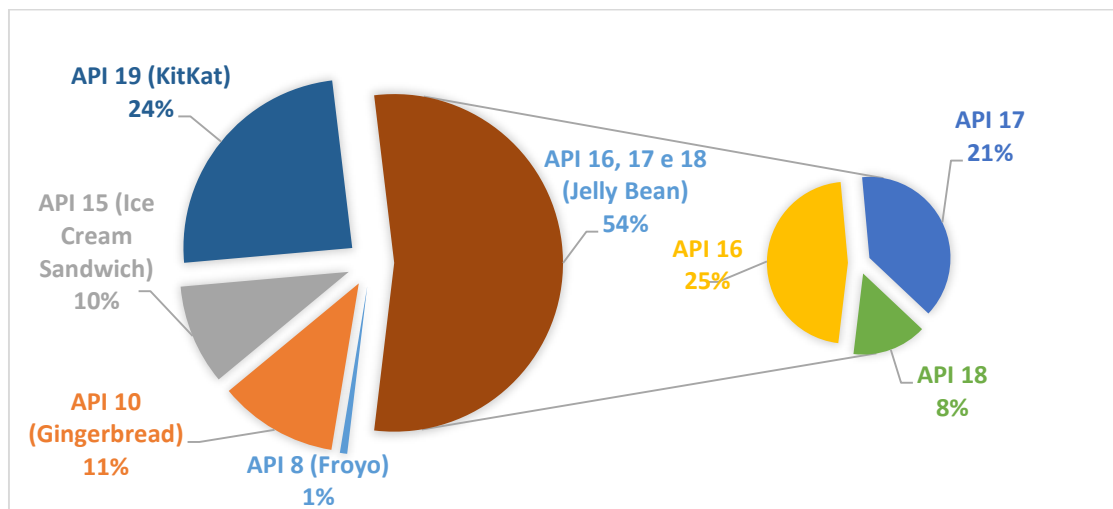
### **2.3.2. Plataforma Android**

Tal como praticamente todos os sistemas operativos conhecidos, o OS Android possui diversas versões. A primeira versão comercial, muito primitiva, foi a versão Android 1.0 Base (lançada em setembro de 2008). A última versão à data de escrita desta dissertação é a versão Android 4.4 (*Application Programming Interface* (API) Level 19). Podemos visualizar na Tabela 3 uma lista de todas as versões do OS Android lançadas até ao momento. Como se poderá deduzir cada nova versão apresenta sempre melhorias em relação às versões anteriores, tanto a nível de funcionalidades como a nível de compatibilidade, ou seja, são cada vez mais compatíveis com os novos dispositivos que vão sendo lançados, suportando as diversas dimensões de ecrã e as diferentes resoluções destes.

Relativamente às versões do OS Android mais utilizadas, como se pode visualizar no Gráfico 1, atualmente a versão Jelly Bean, que engloba as versões 4.1.x (API 16), 4.2.x (API 17) e 4.3 (API 18) é a mais utilizada, seguindo-se imediatamente a versão KitKat que corresponde à versão 4.4 (API 19) que é a última versão Android neste momento [17]. No caso deste projeto, este foi desenvolvido para ser compatível com maioria das versões existentes no mercado atual. Neste sentido, a aplicação é compatível desde a versão Froyo (API18) até à versão KitKat (API19), incluindo todas as versões entre estas duas.

Tabela 3 - Lista de versões do OS Android [18]

Platform Version	API Level	VERSION_CODE
Android 4.4	19	KITKAT
Android 4.3	18	JELLY_BEAN_MR2
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1
Android 4.1, 4.1.1	16	JELLY_BEAN
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH
Android 3.2	13	HONEYCOMB_MR2
Android 3.1.x	12	HONEYCOMB_MR1
Android 3.0.x	11	HONEYCOMB
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD
Android 2.2.x	8	FROYO
Android 2.1.x	7	ECLAIR_MR1
Android 2.0.1	6	ECLAIR_0_1
Android 2.0	5	ECLAIR
Android 1.6	4	DONUT
Android 1.5	3	CUPCAKE
Android 1.1	2	BASE_1_1
Android 1.0	1	BASE

Gráfico 1 – Cota de mercado dos dispositivos pela versão do OS Android<sup>1</sup> [17]

<sup>1</sup> Baseado nos dados (de 9 de setembro de 2014) na tabela presente em “Platform Versions” [5].



Para além da distribuição das versões do OS Android vistas anteriormente, também é interessante observar a distribuição existente no mercado relativamente aos dispositivos com base nas suas resoluções e densidades de ecrã. De acordo com o Gráfico 2 é possível verificar que a resolução mais utilizada é a da categoria *Normal*, esta categoria corresponde a ecrãs com dimensões entre cerca de 3 polegadas a cerca de 5 polegadas (com 82% de cota de mercado), logo após tem-se a resolução da categoria *Large*, mas com apenas 8% de cota de mercado, depois segue-se a *Small* (com 6%) e a menos utilizada *Xlarge* (com 4%).

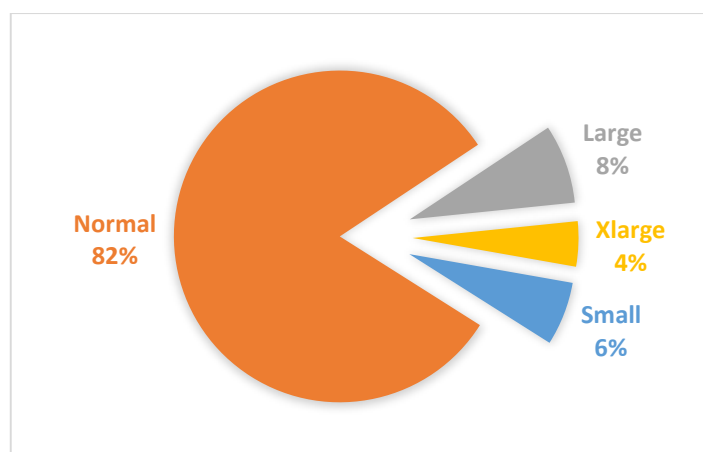


Gráfico 2 – Cota de mercado dos dispositivos com OS Android por dimensão de ecrã<sup>2</sup> [17]

Além da dimensão propriamente dita do ecrã é importante se ter em conta a densidade do ecrã. A densidade de ecrã indica-nos a quantidade de *pixels* por polegada. Se considerarmos dois ecrãs de tamanho igual o que possuir um valor mais elevado de *pixels* por polegada conseguirá apresentar mais itens no ecrã do que o de menor valor. Através do Gráfico 3 consegue-se visualizar as cotas de mercado dos dispositivos com OS Android pela sua densidade de ecrã. Observando o gráfico, conclui-se que a High Dots Per Inch (HDPI) possui a maior cota de mercado, com 37%. Depois tem-se a Extra High Dots Per Inch (XHDPI) com 20%, a Medium Dots Per Inch (MDPI) com 18%, a Extra Extra High Dots Per Inch (XXHDPI) com 16% e a Low Dots Per Inch (LDPI) apenas com 7%. Este projeto foi desenvolvido também de forma a ser o mais flexível possível com as diferentes densidades e dimensões de ecrã, ou seja, a interface do utilizador ajusta-se de acordo com o espaço disponível para apresentar a informação.

---

<sup>2</sup> Baseado nos dados (de 9 de setembro de 2014) na tabela presente em “Screen Sizes and Densities” [5].

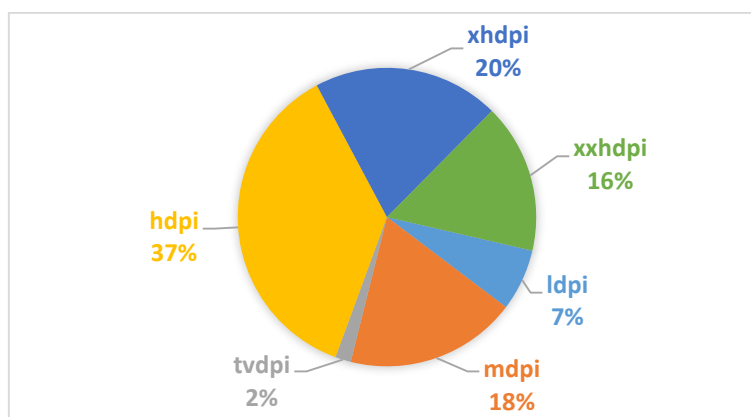


Gráfico 3 – Cota de mercado de dispositivos com OS Android por densidade de ecrã<sup>3</sup> [17]

Após a análise dos três gráficos apresentados anteriormente conclui-se, que no momento em que esta dissertação foi escrita, os dispositivos mais comercializados possuíam a versão do OS Android 4.1.x (Jelly Bean – API 16) com uma densidade de ecrã maioritariamente HDPI e pertencentes à categoria de ecrã de dimensão Normal. Este estudo foi importante para se saber para qual dimensão e densidade de ecrãs nós deveríamos começar por preocupar mais, no que diz respeito à otimização da aparência gráfica da aplicação.

### 2.3.3. Ambientes de Desenvolvimento

#### 2.3.3.1 Aplicação Android

Antes de se pensar em desenvolver um projeto para a plataforma Android é necessário pesquisar-se acerca das opções existentes em termos de *Integrated Development Environment* (IDE). Na internet, encontra-se pelo menos três: NetBeans, Eclipse e o Android Studio (ainda em versão Beta). Tendo em conta a natureza deste projeto preferiu-se não arriscar na utilização da versão Beta do Android Studio, isto porque as versões Beta poderão ainda ter algumas lacunas.

No momento da escolha concluiu-se que a melhor opção seria o Eclipse, apesar de esta informação não estar completamente explícita no *site* oficial de Android Developers [19]. Todavia, após alguma reflexão apercebemo-nos que este *site* recomenda o pacote que inclui o Eclipse *Android Development Tools* (ADT) com o *Android Software Development Kit* (SDK) [20]. Além disso, considera-se que o Eclipse apresenta algumas

<sup>3</sup> Baseado nos dados (de 9 de setembro de 2014) na tabela presente em “Screen Sizes and Densities” [5].

vantagens em relação ao NetBeans relativamente à parte de trabalhar com os *layouts* (interface gráfica da aplicação). Isto porque o *plugin* ADT já instalado no Eclipse do *website* da Android Developers permite construir os *layouts* das atividades através do método de “arrastar e largar” (*Drag and Drop*) e também possibilita alterar todos os atributos de um elemento selecionado através da interface gráfica do próprio Eclipse. Saliente-se que trabalhar nos *layouts* apenas por *tags* XML torna a tarefa mais complicada, principalmente quando se trata de dispositivos de diversas dimensões de ecrã e de resoluções.

Neste sentido, o ambiente de desenvolvimento foi obtido no *website* oficial de Android Developers, onde se escolheu a opção principal de download que é o Eclipse ADT com o Android SDK [21]. Mais detalhadamente este pacote inclui:

- O *software* Eclipse com o módulo de extensão (*plugin*) ADT já instalado;
- O Android SDK Tools é constituído por um conjunto de bibliotecas e de ferramentas que ajudam os desenvolvedores de *software* a implementar as suas aplicações, sendo que existem várias versões. Note-se que é possível instalar outras versões de SDK usando uma aplicação que vem junto, designada de Android SDK Manager. Com a utilização desta é possível fazer *download* de documentação, das imagens de sistema, de exemplos para o SDK, entre outros para diversas versões de Android. Através do SDK Manager ou Eclipse é possível aceder ao AVD Manager. Este permite criar dispositivos virtuais de diversas dimensões e densidades de ecrã, emulando várias versões do OS Android e assim simulando o funcionamento de um dispositivo Android real. Saliente-se que também permite a escolha de periféricos que queremos que funcionem (Bluetooth, Wi-Fi, etc) através de simulação, utilizando para o efeito os equivalentes no computador em que o AVD está a ser executado.

### 2.3.3.2. Web API

Tendo em conta que foi utilizada a tecnologia ASP.NET para o desenvolvimento da Web API, o IDE escolhido foi o Microsoft Visual Studio Express 2013 for Web. Utilizou-se também um repositório para controlo de versões, tendo sido escolhido o Microsoft Team Foundation Version Control, de forma a garantir que a programação realizada não se perdesse e também para se ter possibilidade de voltar a uma versão anterior caso algum erro grave fosse cometido no desenvolvimento da API. Note-se que se conseguiu ter

acesso ao serviço de controlo de versões diretamente do Visual Studio 2013. A versão da .NET Framework utilizada no desenvolvimento da Web API foi a 4.

#### 2.3.4. Outras Tecnologias Utilizadas (MS SQL, SMTP, FTP, JSON)

Utilizou-se os serviços disponibilizados em <https://pt.godaddy.com/> para criar uma conta de acesso a um servidor. Usou-se esse servidor para alojar a API e a base de dados MS SQL. Ainda nesse servidor utilizou-se os serviços *Simple Mail Transfer Protocol* (SMTP) para o envio de *e-mails* a partir da API e o serviço *File Transfer Protocol* (FTP) para guardar as imagens dos utilizadores e dos itens diretamente no servidor. Também solicitou-se o domínio (nome do *site*) [www.xpresstrades.com](http://www.xpresstrades.com) para identificar o servidor.

Para o desenvolvimento da base de dados deste projeto utilizou-se o Sistema Gestor de Base de Dados (SGBD) Microsoft SQL Server. Foi utilizado também o *software* SQL Server Management Studio 2012 para configurar e gerir a base de dados utilizada neste projeto. Na base de dados criada usou-se além das tabelas (entidades e respetivos atributos) as *Stored Procedures* (SP) e *Functions* (funções), onde foram definidos os procedimentos e as funções para ações específicas.

A tecnologia *JavaScript Object Notation* (JSON) foi utilizada para a aplicação Android receber as informações que solicita à API. O OS Android não consegue comunicar diretamente com uma base de dados. O JSON é uma linguagem que facilita o envio e o recebimento de dados, ou seja, é uma linguagem simples de ser entendida, tanto por seres humanos como por máquinas [22]. A API recebe os pedidos da aplicação Android, processa os pedidos comunicando com a base de dados, recebe os resultados fornecidos por ela, lê-os e transforma-os em JSON, depois devolve-os à aplicação Android, a aplicação Android interpreta a informação e realiza ações de acordo com os dados recebidos. O JSON é bastante fácil de ser decodificado pela aplicação Android.

#### 2.3.5. Sistemas de Recomendação

Os sistemas de recomendação utilizam algoritmos específicos, de forma a simular uma inteligência artificial que cria sugestões com base na avaliação de alguns dados. Estes são sistemas que, atualmente, estão presentes na maioria dos *sites* de compras e de vendas de artigos na *Web* (por exemplo: eBay, Amazon, etc). Estes sistemas também são muito utilizados na escolha das publicidades que devem ser apresentadas aos utilizadores, tendo

em conta diversos fatores, tais como a sua localização, a preferência de artigos, etc. Esses dados podem ser obtidos a partir do histórico de pesquisas, de compras e de vendas que são realizadas pelos utilizadores. Também existem outras formas de verificar as preferências dos utilizadores, como por exemplo um sistema explícito de avaliação onde os utilizadores indicam a sua satisfação em relação a um determinado item. [23]

Considerou-se que antes de se avançar para a fase de desenvolvimento do projeto propriamente dito, era interessante e relevante pesquisar e verificar os sistemas de recomendação existentes atualmente. Neste sentido, depois da análise e da pesquisa efetuada e de se ter apurado qual o sistema de recomendação mais adequado para o nosso contexto foi necessário verificar que algoritmos eram utilizados e qual o melhor para o nosso problema.

Para esta aplicação, considerou-se necessário construir um sistema de recomendação que conseguisse sugerir aos utilizadores potenciais itens de interesse, com base no histórico de visualizações de itens. O sistema verifica os géneros preferidos do utilizador e realiza sugestões com base nesses géneros.

Com o estudo sobre os sistemas de recomendação existentes atualmente, pretendeu-se averiguar os prós e os contras de cada um deles e encontrar aquele que mais se adequasse às necessidades descritas anteriormente.

De acordo com o artigo [23] existem 6 técnicas de recomendação:

- Baseada em conteúdo (*Content-based*) – O sistema aprende a recomendar itens que são semelhantes aos que o utilizador gostou no passado. A semelhança dos itens é calculada com base nas características associadas aos itens comparados;
- Filtragem colaborativa (*Collaborative filtering*) – A implementação mais simples e original desta abordagem consiste na comparação dos gostos do utilizador ativo com outros utilizadores que tenham apresentado gostos semelhantes no passado. A semelhança entre os dois utilizadores é obtida com base no histórico, tendo em conta a semelhança das avaliações realizadas pelos utilizadores;
- Demográfico (*Demographic*) – Esta técnica tem em conta a língua, localização, idade ou outro tipo de informação pessoal do utilizador para fornecer as recomendações;
- Baseada no conhecimento (*Knowledge-based*) – Este sistema de recomendação, baseado num domínio de conhecimento específico, apresenta as recomendações

calculando se certas características de um item vão ao encontro das necessidades e das preferências do utilizador e também, calcula o quanto o item poderá ser útil ao utilizador;

- Baseada na Comunidade (*Community-based*) – Este sistema baseia-se nas preferências dos amigos do utilizador para recomendar itens a este.
- Sistemas de recomendação híbridos (*Hybrid recommender systems*) – Este sistema de recomendação, tal como o nome indica, é um sistema híbrido e como tal poderá combinar duas ou mais técnicas anteriormente descritas;

Para o desenvolvimento do sistema de recomendação desta aplicação decidiu-se que a melhor opção seria a primeira, ou seja, baseada em conteúdo (*content-based*), isto porque para o propósito desta aplicação apenas é pretendido recomendar itens aleatórios que respeitem os géneros preferidos do utilizador. É importante referir que não foi necessário utilizar algoritmos complexos. Para esta aplicação bastou utilizar um algoritmo simples criado pelo programador. Apesar de esta técnica exigir, normalmente, uma classificação explícita realizada pelo utilizador, neste caso as preferências são deduzidas automaticamente pelo sistema, de acordo com os itens que o utilizador visita. Quanto mais o utilizador visualizar itens de um determinado género, mais esse género será considerado preferido pelo utilizador. Assim sendo, as recomendações serão feitas pesquisando um número de itens aleatórios de acordo com os géneros preferidos do utilizador.

## 2.4. Padrões de Desenho

Os padrões de desenho são formas comprovadamente melhores de escrever o código fonte de uma aplicação. O código pode ser escrito de muitas formas, todavia os padrões de desenho ajudam a criar a solução mais adequada. Os padrões visam uma escrita de código otimizada que facilita a sua edição e manutenção no futuro, cabendo ao programador identificar qual o padrão de desenho mais adequado à sua solução. Existem três categorias de padrões de desenho que são [24]:

- **Padrões de Criação (*Creational*)** – Utilizados para controlar a criação de objetos por forma a evitar problemas de desenho ou aumento da complexidade do desenho. Exemplos: Singleton, Prototype, Factory, etc;

- **Padrões Estruturais (*Structural*)** – Facilitam o desenho identificando a maneira mais simples para realizar as relações entre os objetos ou entidades. Exemplos: Adapter, Bridge, Composite, Decorator, Facade, Proxy, etc;
- **Padrões Comportamentais (*Behavioral*)** – Identificam padrões de comunicação comuns entre objetos e implementam esses padrões, de modo a aumentar a flexibilidade na realização da comunicação entre esses objetos. Exemplos: Command, Iterator, Mediator, Observer, State, Strategy, etc.

#### 2.4.1. Padrão State

O padrão State permite que um objeto se comporte de várias maneiras conforme o estado em que este se encontra, pois o objeto possui um conjunto de procedimentos definidos para cada estado. Normalmente, o objeto possui vários estados internos definidos e estes podem ser alterados durante o tempo de execução [25].

Na aplicação Android deste projeto utiliza-se este padrão. O sistema Android por defeito implementa este padrão nas suas atividades fazendo com que estas possuam um ciclo de vida, tal como podemos observar na Figura 1.

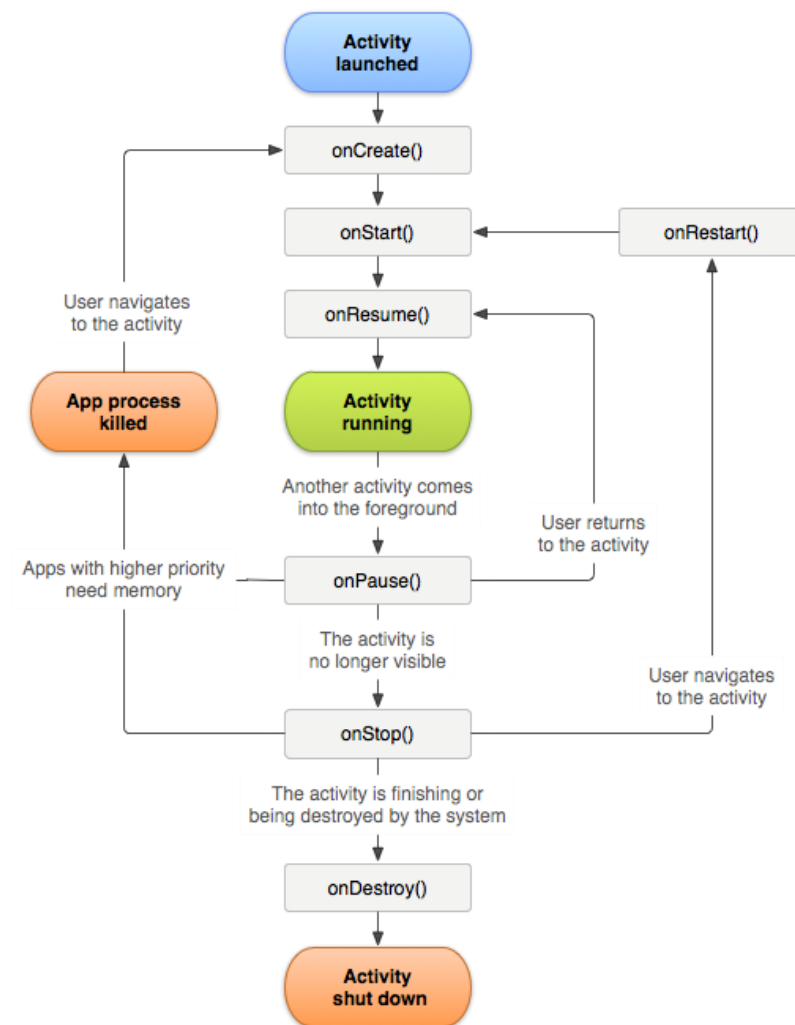


Figura 1 - Ciclo de vida de uma Atividade Android [26]

Em cada atividade de uma aplicação Android podem ser definidos os vários estados presentes na Figura 1. Note-se que não há necessidade de defini-los a todos, contudo alguns são sempre definidos, tal como o `onCreate()`, isto porque este é o estado que é executado apenas uma única vez sempre que uma atividade é lançada.

O estado `onDestroy()` poderá ser utilizado para limpar os recursos que estavam a ser utilizados pela atividade e que já não serão mais necessários. Saliente-se que este estado é chamado quando a atividade está a ser terminada.

Relativamente aos estados intermédios tem-se o `onStart()`, o `onResume()`, o `onPause()` e o `onStop()`. Quando uma nova atividade é lançada, a partir da primeira e, esta não é terminada, a primeira atividade passa pelos estados `onPause()` e logo de seguida para o `onStop()`. Quando a segunda atividade é terminada, o sistema retoma a primeira atividade



executando os seguintes estados pela presente ordem: `onRestart()`, `onStart()` e `onResume()`.

#### 2.4.2. Padrão Adapter

O padrão Adapter tem a finalidade de converter ou de adaptar uma interface de um objeto de forma a poder ser utilizado por outro objeto. Desta forma, é possível colocar objetos a funcionarem em conjunto quando à partida não era possível [25].

Na aplicação Android, uma lista por defeito costuma permitir apenas a representação de objetos do tipo *string*. Com o padrão Adapter consegue-se representar mais do que *strings* na lista, ou seja, consegue-se representar praticamente qualquer tipo de objeto que não *strings*.

#### 2.4.3. Padrão Observer

De forma simples o padrão Observer permite definir uma dependência de um para muitos. Há o objeto observado e os objetos observadores (dependentes) que são notificados e atualizados automaticamente sempre que o objeto observado altera o seu estado [25].

Na aplicação, este padrão foi especialmente útil para fazer uma espera enquanto a vista era carregada e depois carregar a informação para os elementos dessa vista. Isto porque, às vezes a vista leva alguns microssegundos a ser carregada e por conseguinte, pode acontecer que a aplicação tente inserir informação em objetos da vista que ainda não foram criados (estando nulos), desta forma ocorre um erro e a aplicação termina de forma inesperada. Ao aplicar-se o padrão Observer conseguiu-se ultrapassar este problema, pois quando a vista está completamente carregada notifica os objetos que estão dependentes desta e, logo de seguida, estes prosseguem de forma segura (sem haver elementos nulos) ao carregamento da respetiva informação na vista.

#### 2.4.4. Padrão Singleton

O padrão Singleton é um padrão simples, mas muito útil, pois permite que um objeto possa possuir uma única instância [25].

Este padrão foi utilizado na aplicação Android com o intuito de apenas se utilizar uma única instância da classe `HTTPClientManager` (classe responsável pela comunicação), de

forma a se manter um único cliente HTTP. Este padrão também foi utilizado na classe `SharedDataSingleton` da aplicação Android, isto porque se considerou importante manter uma única instância desta classe, já que esta é utilizada para passar informações dentro da aplicação de umas atividades para as outras. Por esta razão, só faz sentido possuir uma instância.

## 2.5. Conclusão

Neste capítulo concluiu-se que a ideia deste projeto ainda é pouco ou nada explorada no que diz respeito a aplicações nativas para sistemas operativos Android, iOS e Windows Phone. Contudo, é de realçar que já existem algumas aplicações *Web* (em forma de *website*) deste género, mas que ainda não estão otimizadas para a sua utilização em dispositivos móveis e também não possuem uma versão *Web* para estes dispositivos. Neste sentido, a ideia deste projeto mostra-se ser interessante e inovadora na área dos dispositivos móveis.

No que diz respeito às tecnologias utilizadas, após o estudo efetuado neste capítulo conseguiu-se selecionar as tecnologias que se considerou serem as mais adequadas para o desenvolvimento deste projeto. Também se estudou e se selecionou os padrões de desenho mais apropriados para a escrita de certas partes do código. A utilização destes é importante, pois a utilização dos mesmos torna as possíveis alterações futuras mais simples e rápidas de se fazer.

Em suma, pode-se afirmar que depois do estudo de aplicações existentes relacionadas com a ideia deste projeto e após se ter selecionado as tecnologias que deverão ser utilizadas para o desenvolvimento da aplicação, encontramo-nos em condições de se avançar para o desenho do sistema, em que o funcionamento do sistema será definido detalhadamente.

### 3. Desenho do Sistema

---

#### 3.1. Introdução

O desenho do sistema é uma das etapas cruciais para o sucesso de uma aplicação que se pretenda desenvolver. É com este que se define praticamente todos os detalhes da aplicação antes de os implementar e esses detalhes incluem a aparência da interface gráfica do utilizador, a interação sistema/utilizador e a arquitetura. Com o desenho do sistema é possível definir exatamente como irá ser o produto final, ou seja, a aplicação depois de implementada. Sendo o desenho do sistema de uma importância extrema, este deverá ser robusto, flexível, atraente, escalável e facilmente compreensível por todos os interessados na aplicação.

No presente capítulo serão apresentados:

- **Análise do Problema:**
  - Requisitos Funcionais – Enumeram as funcionalidades do sistema.
  - Requisitos Não-Funcionais – Estes requisitos são características desejáveis e/ou necessárias para o funcionamento do sistema. Essas características contemplam aspetos de segurança, desempenho, usabilidade, confiabilidade, disponibilidade, facilidade de manutenção e tecnologias necessárias (*hardware*). São aspetos que influenciam definitivamente a qualidade do *software*.
- Diagramas de casos de utilização – Diagramas utilizados para resumir e esquematizar as funcionalidades que o sistema irá implementar, identificando as ações do sistema e os respetivos atores;
- Diagramas de atividade – Detalham os casos de utilização apresentados no diagrama de casos de uso, descrevendo de forma mais pormenorizada o funcionamento do sistema e a interação do utilizador com este. Note-se que se realizará isto para cada caso de utilização.
- Modelo Relacional da base de dados – O sistema requer uma base de dados bem organizada onde se possa armazenar os dados do sistema eficientemente, para isso é necessário construir um modelo relacional bem estruturado.

- Protótipos de baixo nível (*mockups*) da interface de utilizador – São utilizados para planear a interface de utilizador do sistema, com estes é possível testar a usabilidade e melhorá-la rapidamente com pouco esforço.
- Arquitetura que irá ser utilizada na implementação do sistema – Todo o sistema tem uma arquitetura e é essencial pensar nesta de forma adequada para que a solução desenvolvida seja a melhor possível.

### 3.2. Análise do Problema

Uma das primeiras etapas quando se pretende desenvolver um sistema é realizar o levantamento de requisitos. Como é lógico estes requisitos devem ser obtidos consultando os futuros utilizadores do sistema, aqueles que possuem interesse em que o sistema seja desenvolvido. Antes do levantamento de requisitos foi feito um brainstorming com pessoas interessadas neste projeto e em seguida foram selecionadas as ideias mais importantes, neste caso as funcionalidades mais importantes que a aplicação deveria implementar. Este processo foi iterativo e incremental. A análise de requisitos serviu para analisar o problema e, por conseguinte, serviu para definir o rumo deste projeto.

#### 3.2.1. Requisitos Funcionais

Os requisitos funcionais descrevem as funcionalidades que o utilizador irá ter ao seu dispor na utilização do sistema [27]. Durante o levantamento de requisitos obtiveram-se os seguintes requisitos funcionais:

- F1. Registrar novo Utilizador.
- F2. Selecionar foto para perfil.
- F3. Tirar foto de perfil com a câmara.
- F4. Recortar foto para o perfil.
- F5. Selecionar e/ou adicionar novo País.
- F6. Selecionar e/ou adicionar nova Região.
- F7. Selecionar e/ou adicionar nova Cidade.
- F8. Iniciar sessão na aplicação (*Login*).
- F9. Recuperar palavra-passe (*Password*).
- F10. Pedir reenvio do código de ativação.

- F11. Ativar conta de utilizador.
- F12. Selecionar item do menu.
- F13. Selecionar item do submenu.
- F14. Pesquisa de itens por nome e categoria (jogos ou livros).
- F15. Pesquisa de itens da categoria jogos por nome e plataforma.
- F16. Efetuar pedidos para a adição de itens.
- F17. Apresentar lista de itens filtrada por categoria e género.
- F18. Apresentar lista de itens da categoria jogos por género e plataforma.
- F19. Adicionar item à lista de ofertas.
- F20. Adicionar item à lista de desejos.
- F21. Recomendar itens com base no género dos itens mais visualizados.
- F22. Recomendar itens da categoria jogos com base no género dos jogos mais visualizados e plataforma.
- F23. Apresentar a lista dos trinta itens da categoria jogos ou livros mais desejados (de acordo com a categoria que está sendo visualizada) ordenados de forma decrescente.
- F24. Apresentar a lista dos trinta jogos mais desejados filtrados pela plataforma selecionada no momento.
- F25. Apresentar a lista dos trinta itens da categoria jogos ou livros mais oferecidos (de acordo com a categoria que está sendo visualizada) ordenados de forma decrescente.
- F26. Apresentar a lista dos trinta jogos mais oferecidos filtrados pela plataforma selecionada no momento.
- F27. Visualizar o número de proprietários de um item.
- F28. Visualizar o número de interessados num item.
- F29. Ver lista de utilizadores proprietários de um item ordenados por distância (em relação ao utilizador).
- F30. Ver lista de interessados de um item ordenados por distância (em relação ao utilizador).
- F31. Remover item da lista de ofertas.
- F32. Remover item da lista de desejos.

- F33. Visualizar perfil público de um utilizador selecionado (nome de utilizador, data em que se registou na aplicação, idade, país, região, cidade, preferências especiais relativas a propostas e estatísticas de avaliações (*feedback*)).
- F34. Enviar proposta de troca.
- F35. Apresentar lista de itens possíveis para oferecer em troca numa proposta (tendo em conta as preferências do utilizador que irá receber a proposta).
- F36. Permitir definir se o utilizador só deseja receber propostas com ofertas de itens (jogos ou livros) da sua lista desejos.
- F37. Permitir definir se o utilizador pretende receber propostas com oferta de itens da categoria jogos de todas as plataformas ou de apenas uma ou um conjunto específico de plataformas de jogo.
- F38. Visualizar estatísticas de avaliações recebidas (*feedback* recebido).
- F39. Permitir ao utilizador alterar a sua imagem de perfil.
- F40. Permitir ao utilizador visualizar o seu nome de utilizador, *e-mail*, nome verdadeiro e data de nascimento (dados que não podem ser alterados).
- F41. Alterar palavra-passe (*password*).
- F42. Alterar país, região ou cidade.
- F43. Permitir o utilizador adicionar ou modificar o seu endereço postal.
- F44. Apresentar lista de ofertas da categoria selecionada.
- F45. Apresentar lista de desejos da categoria selecionada.
- F46. Editar estado de conservação de um item da lista de ofertas.
- F47. Editar nível de desejo de um item da lista de desejos.
- F48. Mover item da lista de ofertas para a lista de desejos.
- F49. Mover item da lista de desejos para a lista de ofertas.
- F50. Visualizar o perfil de um item selecionado.
- F51. Apresentar lista de géneros da categoria selecionada.
- F52. Apresentar lista de propostas recebidas filtradas por “Com novidades”, “Esperando *Feedback*”, “Em negociação”, “Aceites” e “Rejeitadas”.
- F53. Apresentar lista de propostas enviadas filtradas por “Com novidades”, “Esperando *Feedback*”, “Em negociação”, “Aceites” e “Rejeitadas”.
- F54. Apresentar notificações visuais por cima das propostas listadas (enviadas e recebidas).
- F55. Permitir refrescar manualmente listas de itens ou mensagens.

- F56. Aceitar Proposta Recebida.
- F57. Rejeitar Proposta Recebida.
- F58. Cancelar Proposta Enviada.
- F59. Mostrar proposta em detalhe.
- F60. Permitir envio de mensagens entre utilizadores, no contexto de uma proposta.
- F61. Permitir colar endereço postal no corpo de uma mensagem.
- F62. Permitir dar *feedback* aos outros utilizadores no contexto de uma proposta aceite.
- F63. Aceder ao perfil do utilizador.
- F64. Notificar o utilizador sempre que este receba uma nova proposta.
- F65. Notificar o utilizador sempre que este receba uma nova mensagem, no contexto de uma proposta.

### 3.2.2. Requisitos Não Funcionais

Ao contrário dos requisitos funcionais que descrevem as funcionalidades do sistema, os requisitos não funcionais descrevem aspetos que influenciam a qualidade do sistema [27]. Esses aspetos podem ser de segurança, adaptabilidade, desempenho, confiabilidade, disponibilidade, manutenção e também de *hardware* (componentes físicos que são necessários para o sistema funcionar). Sem dúvida que a qualidade do sistema está dependente dos requisitos não funcionais que são considerados. Neste projeto optou-se por implementar os seguintes requisitos não funcionais:

- NF1. Adaptabilidade/*Hardware* Alvo – A aplicação deve suportar dispositivos Android de diferentes dimensões e resoluções de ecrã, ou seja, *Smartphones* e *Tablets*.
- NF2. Adaptabilidade/*Software* Alvo – A aplicação deve ser compatível com dispositivos que possuam o sistema operativo Android da API8 (versão 2.2) à API 19 (versão 4.4).
- NF3. Adaptabilidade – Deve ter suporte para duas línguas: língua inglesa (Reino Unido) e língua portuguesa (Portugal).
- NF4. Segurança – O sistema deve permitir apenas um registo de utilizador por *e-mail*.
- NF5. Segurança – O sistema deve permitir apenas um “nome de utilizador” por utilizador.

- NF6. Segurança – O utilizador tem de confirmar o *e-mail* utilizado no registo ativando a sua conta com o respetivo código de ativação na primeira utilização.
- NF7. Segurança – O sistema deve eliminar o utilizador se ultrapassar as três tentativas de ativação de conta.
- NF8. Usabilidade – Os teclados que são apresentados aquando da inserção de informação são específicos de acordo com o tipo de inserção desejado (Por exemplo: Se são só números, então o teclado escolhido é o numérico).
- NF9. Usabilidade – Os campos preenchidos pelo utilizador devem ser verificados pela aplicação, de forma a evitar que o utilizador introduza dados errados.
- NF10. Segurança – Os dados de autenticação (*e-mail* e palavra-passe) são enviados para a API, através da rede, codificados em Base64.
- NF11. *Hardware* – O dispositivo deverá possuir sistema de ligação à Internet.
- NF12. *Hardware* – O dispositivo deverá possuir uma câmara digital incorporada.
- NF13. Segurança – O sistema deve enviar o código de ativação de conta para o *e-mail* de registo até três vezes.
- NF14. Desempenho – A aplicação deverá apenas carregar do servidor o que é necessário para cada momento, de forma a não sobrecarregar a ligação do dispositivo à Internet (evitando tornar mais lento o acesso aos dados mais importantes).
- NF15. Usabilidade – Verificar se existem novas plataformas de jogos sempre que a aplicação é iniciada.
- NF16. Segurança – A API recebe sempre os dados de autenticação do utilizador que efetua o pedido de forma a poder validar a operação solicitada.
- NF17. Usabilidade – Na maioria dos botões são utilizados ícones padrão do sistema operativo Android definidos pela Android Developers, tornando o reconhecimento intuitivo e imediato.
- NF18. Usabilidade – As funcionalidades dos botões da barra de ação são descritas numa “mensagem flutuante” (*Toast*) quando o utilizador efetua um clique longo sobre um dos botões.
- NF19. Usabilidade – A descrição de cada campo é apresentada numa “mensagem flutuante” quando o utilizador efetua um clique curto sobre os textos que se encontram no perfil (Detalhes) de um item.



- NF20. Usabilidade – A descrição de cada campo é apresentada numa “mensagem flutuante” quando o utilizador efetua um clique curto sobre os ícones que se encontram no perfil público de um utilizador.
- NF21. Capacidade/Disponibilidade – Utilização da API e da base de dados Microsoft SQL para guardar e/ou apresentar dados dos utilizadores e da aplicação *online*.
- NF22. Desempenho – As imagens deverão ser carregadas à medida que o utilizador for descendo nas listas e estas fiquem visíveis.
- NF23. Segurança – A aplicação deve autenticar o utilizador perante a API para a realização de interações críticas com o sistema.
- NF24. Usabilidade – A aplicação deverá ter uma barra onde as ações disponíveis a cada momento sejam apresentadas de forma a facilitar a realização de tarefas frequentes.
- NF25. Usabilidade – Os ícones são simples e intuitivos.
- NF26. Usabilidade – Apresentar diálogo de confirmação para ações que não podem ser revertidas.
- NF27. A aplicação necessita estar constantemente ligada à Internet para poder funcionar.

### 3.3. Casos de Uso

A primeira vista arquitetural que se utilizou neste projeto foi a vista de Casos de Uso (*Use Cases*).

A vista de casos de uso permite apresentar, de uma forma simples, as ações que o sistema ou a aplicação pode realizar identificando quem irá poder realizar as mesmas, ou seja, possibilita visualizar as relações entre as ações do sistema e os seus atores. Esta vista possui vários componentes que podem ser utilizados, contudo na maior parte dos casos utiliza-se apenas 3 componentes que são: o Ator, a Ação e o Sistema [27].

Para este projeto, identificaram-se dois atores e várias ações. O diagrama está representado na Figura 2. Saliente-se que no Anexo B é possível visualizar um diagrama mais completo, divididos em duas partes e que estão presentes na Figura 42 e na Figura 43.

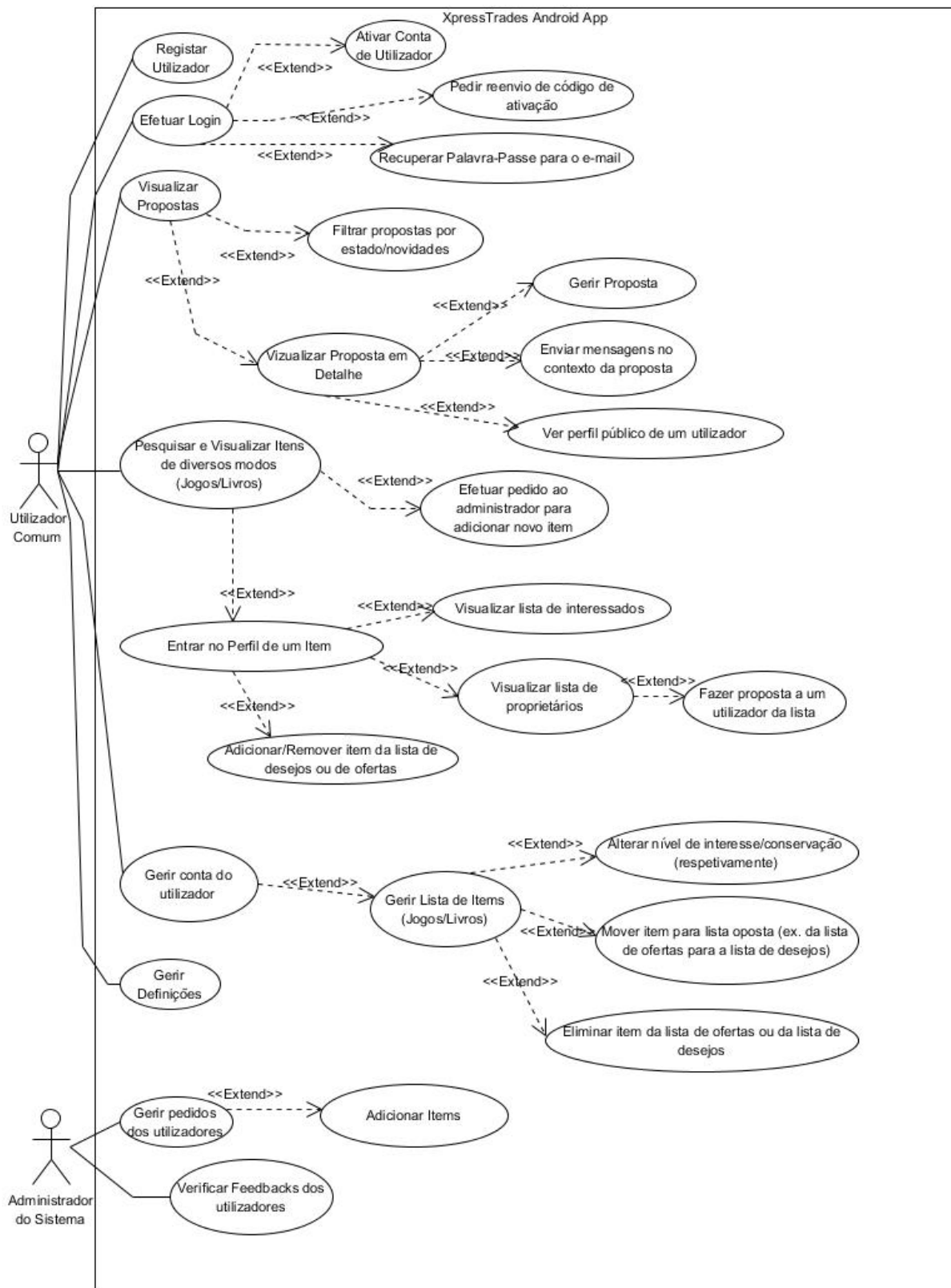


Figura 2 - Diagrama de Casos de Uso

### 3.4. Diagramas de Atividade

Foram desenhados diagramas de atividade, também conhecidos por modelos de tarefa, para muitos dos casos de utilização. Estes têm o intuito de especificar como o sistema irá interagir com o utilizador em diversos casos de uso. A aplicação XpressTrades precisa de uma ligação constante à Internet, contudo se houver uma falha na ligação a aplicação irá solicitar ao utilizador através de uma mensagem numa janela flutuante e/ou uma mensagem de erro, que o utilizador ligue o dispositivo à Internet para poder continuar a efetuar ações. O sistema é constituído pela aplicação Android (APP), pela API (presente no Servidor) e pela base de dados (presente também no servidor) que possui as *Stored Procedures* para cada tipo de ação. Note-se que sem a interligação dos componentes do sistema a aplicação não funciona corretamente.

Os diagramas de atividade desenhados e as respetivas descrições são apresentados por uma ordem lógica. Devido à quantidade de diagramas os dois primeiros serão apresentados em seguida e os restantes, não menos importantes, serão apresentados no Anexo C.

Tendo em conta o funcionamento da aplicação, faz sentido começar pelo caso de utilização de **registo de utilizador**. Inicialmente, o utilizador encontra-se na atividade “Login” e deverá escolher a opção de registo de utilizador. Neste instante a atividade de registo é lançada pelo sistema. No ecrã de registo o utilizador escolhe a sua fotografia de perfil (a fotografia pode ser uma já existente ou tirada no momento, a fotografia tem de ser recortada de forma a respeitar um quadrado). Após a fotografia, o utilizador deve preencher os dados que são solicitados nos formulários e no final submetê-los para o sistema. O sistema verifica os dados preenchidos e notifica o utilizador caso existam erros. Esses erros podem surgir no caso de um nome de utilizador já em uso ou de um endereço de correio eletrónico já registado. Caso os dados sejam todos válidos o sistema adiciona o utilizador e envia-lhe um código de ativação de conta para o seu *e-mail*. Este caso de uso é representado pelo diagrama presente na Figura 3.

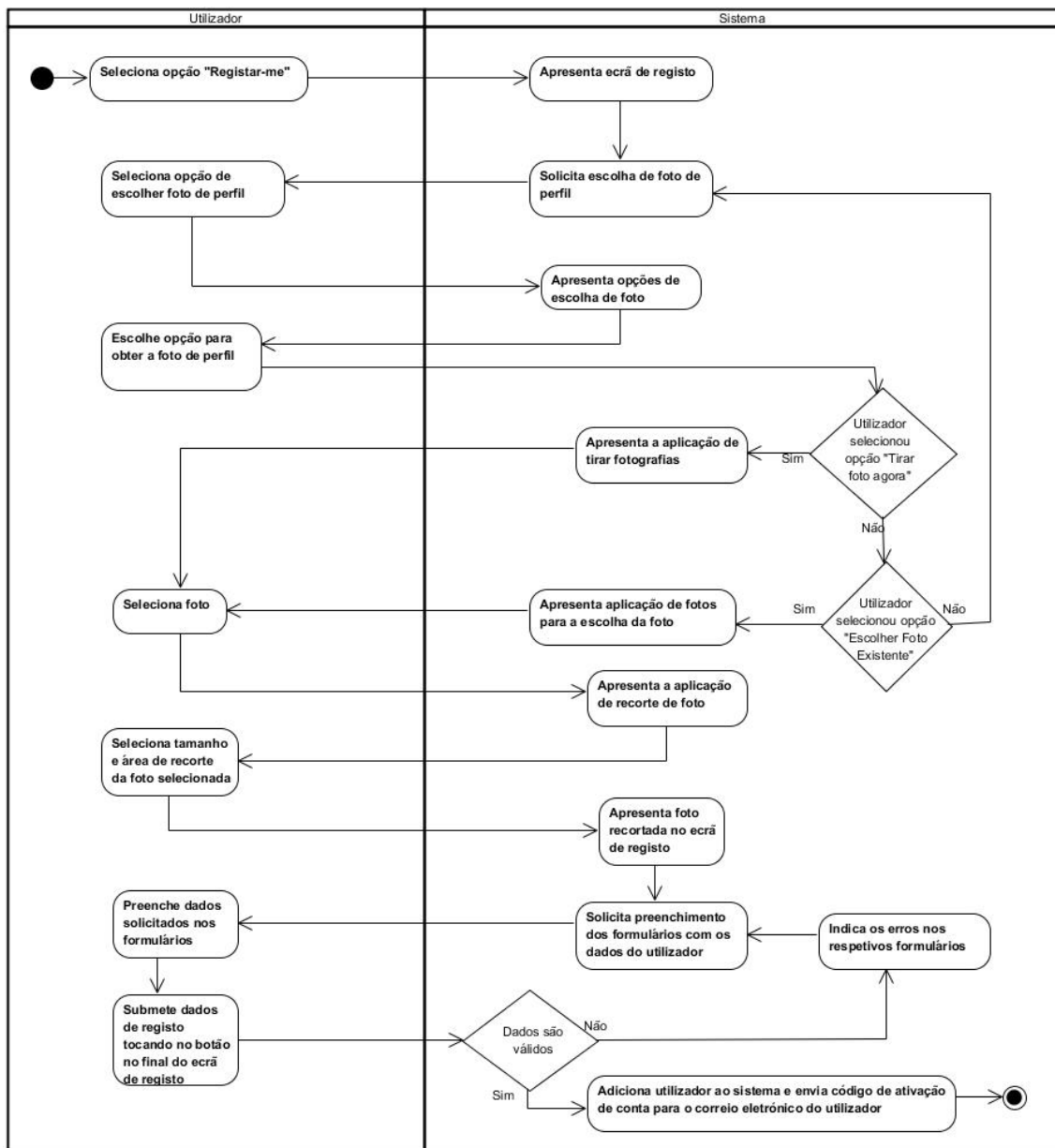


Figura 3 - Diagrama de atividade para o caso "Registo de Utilizador"

Para o utilizador efetuar **início de sessão (login)** na aplicação, tem de aceder à atividade de Login. Caso já tenha a sessão iniciada, o sistema entrará automaticamente para o Menu Principal e neste momento a data e a hora do último acesso (do utilizador à aplicação) são atualizadas. Se o utilizador não tiver sessão iniciada, tem de preencher as credenciais de acesso (endereço de correio eletrónico e palavra-passe). O sistema verifica se estão corretas, se não estiverem indica o erro. Caso contrário, verifica se a conta está ativada e se estiver, o sistema salta o passo de ativação, atualiza a data e hora do último acesso e apresenta a atividade Menu Principal. Note-se que se a conta ainda não estiver ativada, o sistema solicita o código de ativação. Nesta janela o utilizador pode solicitar o envio do

código de ativação para o seu *e-mail* até um máximo de três vezes. O utilizador possui 3 tentativas de ativação, ao final destas a sua conta é eliminada. Saliente-se que se o utilizador introduzir o código correto, a data e a hora do último acesso são atualizadas e a atividade “Menu Principal” é apresentada. Este caso é representado pelo diagrama presente na Figura 4.

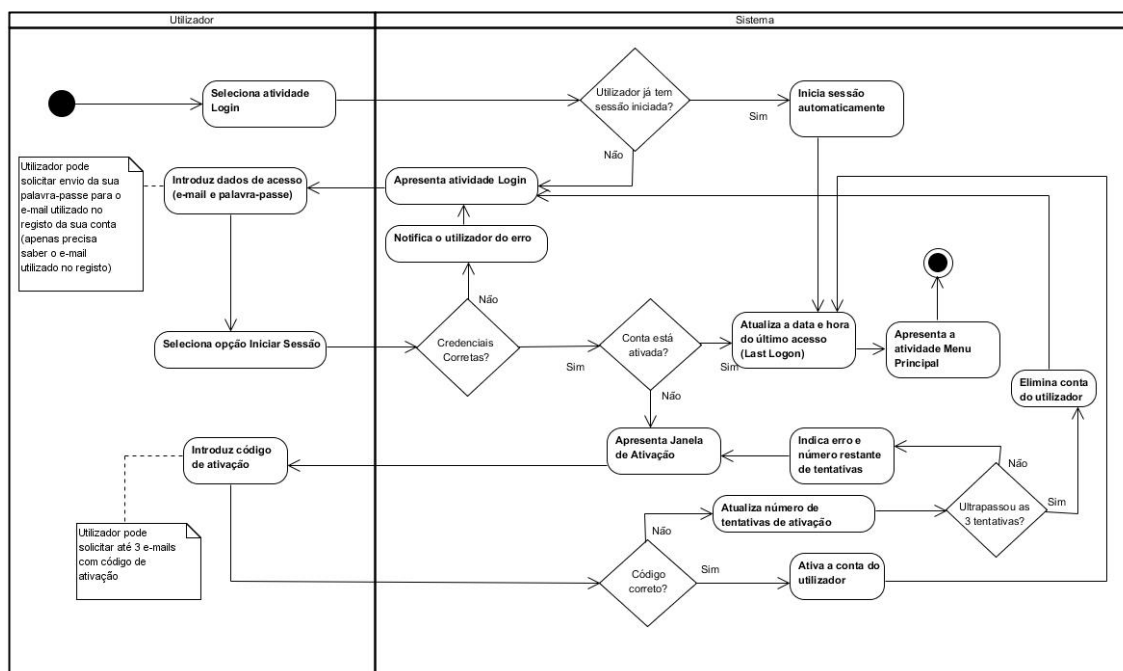


Figura 4 - Diagrama de atividade para o caso "Inicio de Sessão (Login)"

### 3.5. Modelo Relacional

Para desenvolver uma aplicação desta dimensão, foi necessário pensar numa base de dados complexa em que os dados fossem armazenados de forma coerente e sem repetições e que fosse fácil relacionar os dados das várias entidades. Através dos nomes utilizados nos atributos nem sempre é intuitivo entender o seu propósito. Deste modo, considerou-se importante explicar cada entidade e cada atributo dessa entidade de forma a facilitar o entendimento<sup>4</sup>, pois na base de dados existe um campo destinado a um comentário sobre o atributo com vista a esclarecer o propósito do mesmo que apenas é visível no *software* de gestão da base de dados.

Através da observação da Figura 5, é possível visualizar as entidades e os seus respetivos atributos (incluindo os seus tipos). Saliente-se que a explicação presente no Anexo D teve

<sup>4</sup> A explicação de cada entidade e de cada atributo dessa entidade está apresentada de forma pormenorizada no Anexo D

como intuito explicar cada atributo de modo detalhado, tentando-se seguir uma sequência lógica na apresentação das entidades. Note-se, ainda, que os tipos dos atributos não serão elucidados, pois estão explícitos na Figura 5.

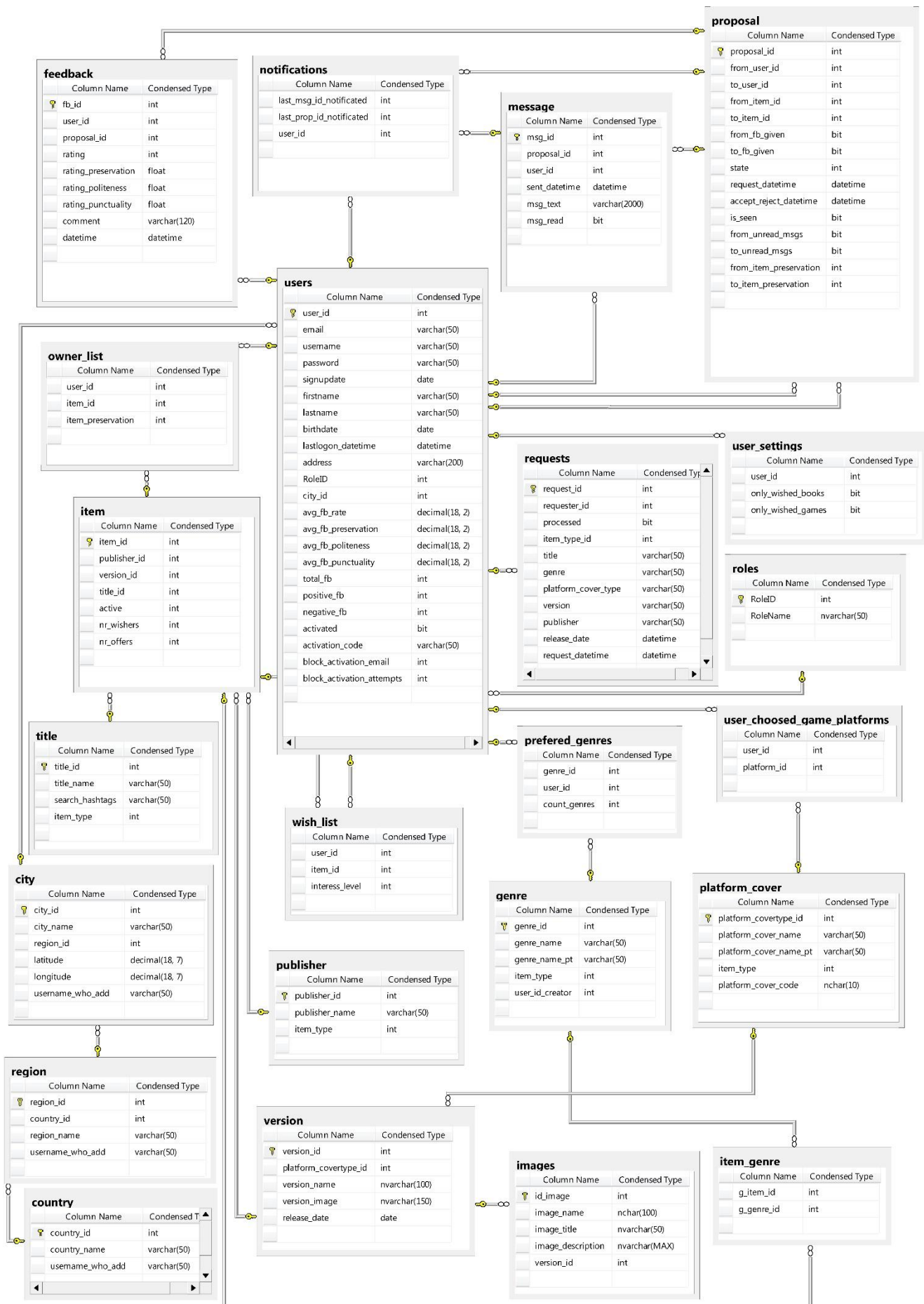


Figura 5 - Modelo Relacional da base de dados da aplicação

### 3.6. Wireframes/Mockups

Para desenvolver uma aplicação com as funcionalidades descritas anteriormente, há a necessidade de realizar protótipos da interface do utilizador que irão ilustrar a forma como a aplicação funcionará e interagirá com este. Existem muitas formas de apresentar uma funcionalidade ao utilizador, mas só algumas seguem a melhor abordagem possível. Ao realizar a prototipagem foram tidos em conta aspetos de *design* e de usabilidade. Desta forma, foram tidas em conta as 10 heurísticas de Nielsen [28] e também os princípios de *design* apresentados pela Android Developers [29].

Assim sendo, utilizou-se o *software* de prototipagem Balsamiq Mockups For Desktop (versão 2.2.3). Este apresenta muitas facilidades na construção dos protótipos, a criação de *wireframes* é muito simples e, basicamente, é possível construí-los arrastando e largando (*drag and drop*) elementos já disponíveis nele. Também permite a importação de novos itens. O *software* mencionado permite ainda criar ligações entre os diversos *wireframes*, de modo que se conseguiu criar uma navegação entre estes, simulando um pouco a navegação que o *software* irá possuir [30]. A criação de protótipos de baixa fidelidade é mais rápida que a criação dos de alta-fidelidade, daí ter-se optado por fazer os testes com estes, pois o objetivo é estudar a melhor interface para a aplicação antes de implementá-la e não criar uma interface já quase final. Na Figura 6 seguem-se alguns exemplos dos *wireframes* criados inicialmente.



Figura 6 - Exemplo dos wireframes para smartphone

Logo após terem sido criados os *wireframes*, solicitou-se a vários utilizadores que os testassem. De forma a não criar nenhum tipo de pressão adicional sobre os utilizadores,



os testes foram feitos de modo informal, contudo tiveram de seguir os cenários de utilização apresentados no Anexo E. Estes cenários englobam as ações mais importantes da aplicação. É de realçar que durante os testes verificou-se os tempos que os utilizadores levaram a realizar determinada ação, o número de erros cometidos e no final anotou-se os comentários dos utilizadores. Após a análise dos dados recolhidos detetou-se falhas de usabilidade.

Seguidamente, os *wireframes* foram melhorados e voltou-se a solicitar aos utilizadores que os testassem e assim, sucessivamente, até se obter uma solução quase ótima. Para estes testes foram escolhidos quatro utilizadores que já utilizavam o sistema operativo Android há algum tempo e um utilizador com pouca experiência, mas com conhecimentos informáticos. De acordo com Nielsen Norman Group, normalmente bastam cinco utilizadores para testarem a usabilidade de um sistema, contudo é assumido que em alguns casos fará sentido mais e noutros menos [31].

Por questões de gestão de tempo a versão dos *wireframes* foi desenhada somente para *smartphones*. Contudo, o sistema funcionará de forma idêntica em *tablets* e outros dispositivos que possam executar o OS Android. Note-se que para as várias dimensões de ecrã, de momento, apenas serão feitos pequenos ajustes de forma a manter uma aparência agradável do visual.

### 3.7. Arquitetura

Uma arquitetura pode ser desenvolvida utilizando apenas um estilo arquitetural ou vários. A arquitetura de um sistema pode ser constituída por diversos tipos de elementos, tais como bases de dados, servidores e clientes, entre outros. O estilo arquitetural é definido pela forma como estes elementos são interligados [32].

A aplicação deste projeto divide-se em três elementos arquiteturais principais: a base de dados e a API que estão hospedadas no servidor e a aplicação Android que é o Cliente. O estilo Cliente-Servidor é um dos estilos deste sistema. Além deste temos presente, tanto na API como da aplicação Android, o estilo MVC (*Model – View – Controller*).

### 3.7.1. Arquitetura MVC

A arquitetura MVC é constituída por três componentes:

- **Modelo (*Model*)** – É responsável por providenciar o núcleo funcional da aplicação. Regista as vistas e os controladores interessados nos dados do Modelo. Notifica os controladores e vistas que estão interessados sempre que existem mudança nos dados [32].
- **Vista (*View*)** – É responsável por criar e inicializar os seus controladores, apresentar a informação ao utilizador, atualizar-se a si própria quando recebe novos dados do modelo e recuperar dados do modelo [32].
- **Controlador (*Controller*)** – É responsável por receber as entradas (*inputs*) dos utilizadores como eventos, traduzir esses eventos em pedidos para o modelo ou apresentar o pedido para a vista, atualizar-se a si próprio quando novos dados chegam do modelo [32].

As vantagens da utilização deste estilo arquitetural são vários, nomeadamente: o desacoplamento dos dados (modelo) da saída/*output* (vista) e das entradas/*inputs* (controlador) permite-nos, por exemplo, utilizar os mesmos dados para múltiplas vistas, alterar os dados do modelo e esta alteração reflete-se automaticamente em todas as vistas que estejam a aceder a estes. Além disto, aumenta a flexibilidade do sistema, isto porque podemos realizar alterações nas vistas e nos controladores mantendo intacto o modelo onde estão os dados e o núcleo da funcionalidade [32].

Como referido anteriormente, na aplicação Android foi utilizado o estilo arquitetural MVC. Por defeito, o SDK da Android já o implementa. A forma como o SDK do Android implementa o MVC torna a programação da aplicação mais flexível e mais fácil para o programador, pois divide a complexidade pelas várias componentes. Na Figura 7 é possível visualizar o diagrama da arquitetura MVC da aplicação Android.

Ao ser criado um novo projeto no SDK da Android, este já apresenta os ficheiros divididos em várias pastas. A organização destas pastas já separa os ficheiros que fazem parte da Vista dos que fazem parte das componentes Modelo e Controlador. Num projeto Android, temos a pasta Src (de *Source*) que armazena todos os ficheiros java e estes podem conter as classes das: atividades, *broadcasts*, serviços, *asynctasks* (tarefas

assíncronas), *handlers*. Na arquitetura MVC, pode-se dizer que esta pasta corresponde aos componentes Modelo e Controlador. Tem-se também a pasta Res (de *Resources*) que possui todos os recursos da aplicação, tais como: imagens, vídeos, sons, *strings*, *layouts*, menus. Esta corresponde à componente Vista, pois são recursos para utilizar na apresentação das vistas ao utilizador. As ligações entre o código e os recursos (*resources*) são feitas pelo ficheiro R.java que se encontra na pasta Gen (*Generated Java Files*). Note-se que este ficheiro é gerado automaticamente e não deve ser alterado pelo programador. Sempre que é adicionado um novo recurso é gerado um identificador único (um número inteiro) para este e, em seguida, é adicionado automaticamente ao ficheiro R.java. Saliente-se que existem outras pastas, todavia estas são as principais.

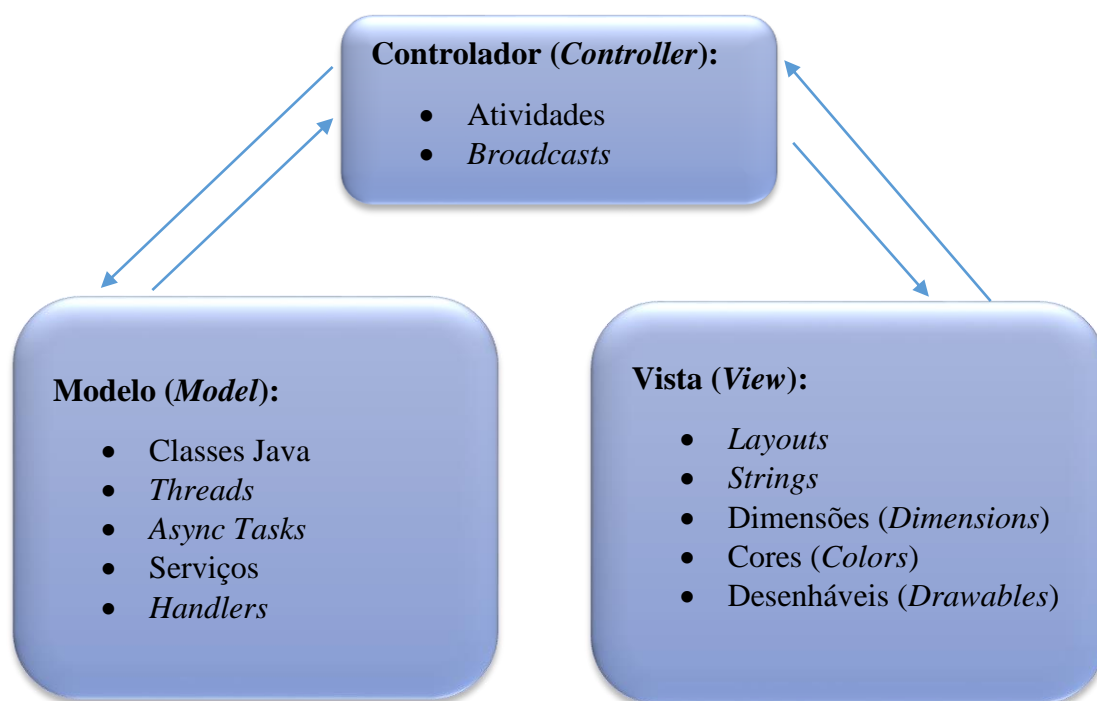


Figura 7 - Diagrama MVC da aplicação Android

É importante mencionar que o projeto tem um ficheiro XML designado de *AndroidManifest* que é o ficheiro que tem de ser utilizado para declarar todas as propriedades da aplicação. Assim, neste ficheiro são declarados os periféricos do dispositivo que precisam de ser utilizados pela aplicação (GPS, WiFi, Câmara, etc), as atividades, serviços e *broadcasts* e, também, as versões de Android que podem executar a aplicação.

Relativamente à *Web API*, esta foi desenvolvida com a tecnologia ASP.NET que por defeito também já traz o MVC implementado. Saliente-se que neste as pastas do Modelo e Controlador têm mesmo o nome *Models* e *Controllers*, respetivamente. Pode-se visualizar na Figura 8 a arquitetura MVC da API e o que está a ser utilizado dentro de cada componente. Note-se que é no controlador que são definidos os endereços das ações da API.

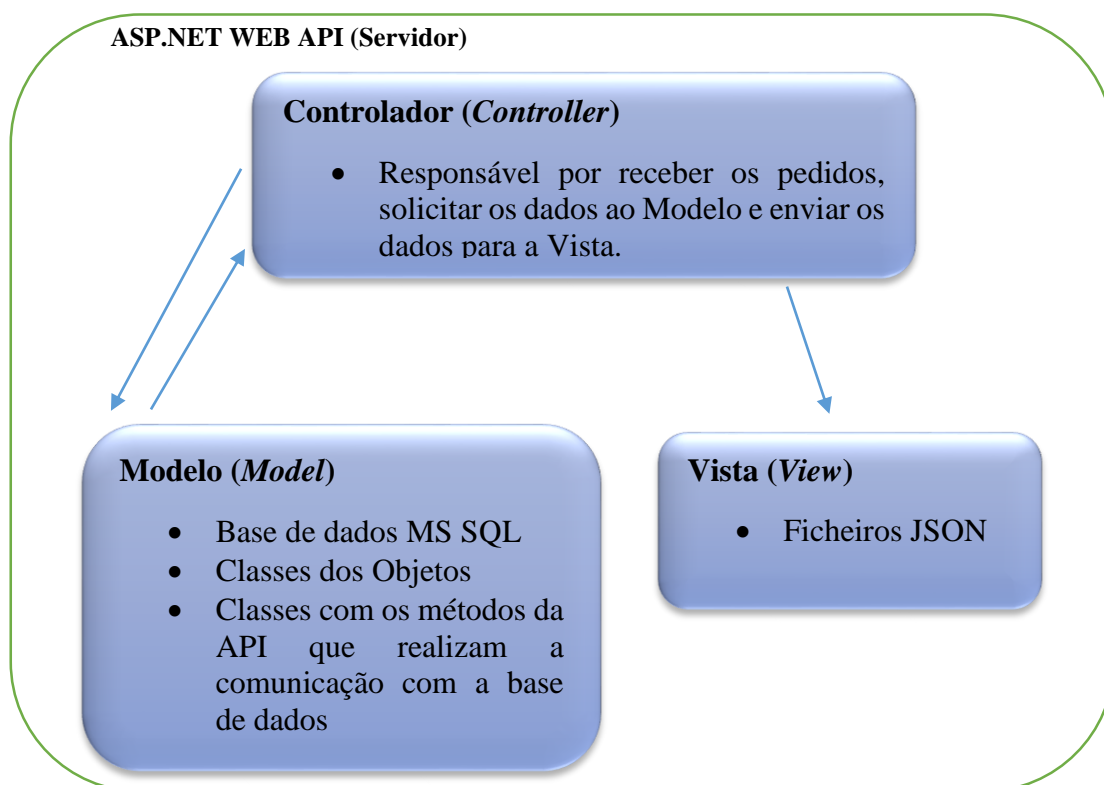


Figura 8 - Arquitetura MVC da API

### 3.7.2. Arquitetura Cliente-Servidor (REST)

A aplicação deste projeto segue a arquitetura Cliente-Servidor REST que permite o endereçamento dos recursos de uma forma padronizada, ou seja, os componentes que agem entre o cliente e o servidor possuem uma visão mais clara do que está a ser enviado e recebido e, também, permite uma menor interligação entre o cliente e o servidor [33]. A aplicação é constituída pelo(s) cliente(s) (Aplicação Android) e servidor (*Web API*, base de dados e serviço de *e-mail*) pelo que necessita de uma ligação constante à Internet para a sua utilização ser possível. Na Figura 9 é possível visualizar a arquitetura da aplicação e na mesma figura está incluída a arquitetura MVC da API, de forma a se perceber que componente recebe os pedidos e através de que componente é apresentada a resposta.

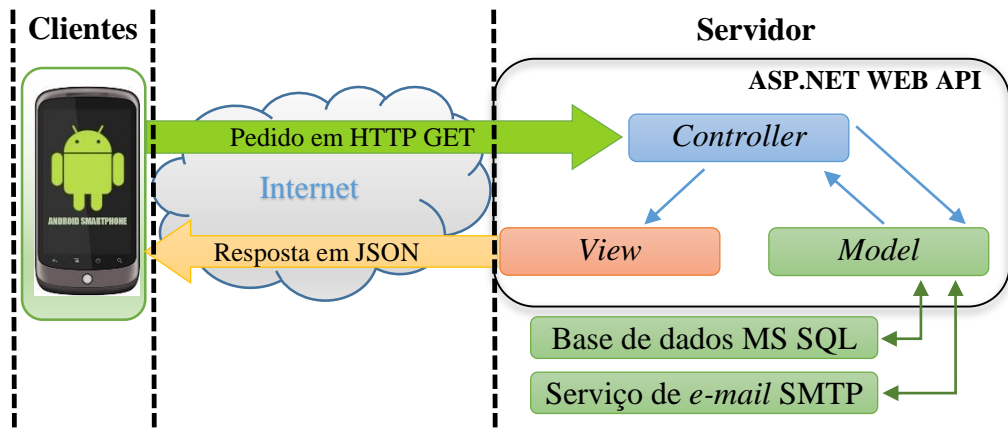


Figura 9 - Arquitetura Cliente-Servidor (com WEB API MVC)

A API contém um conjunto de ações bem definidas que podem ser solicitadas pela aplicação Android através do respetivo *Uniform Resource Identifier* (URI). Cada ação é inequivocamente identificada por um URI específico. Nesse URI são colocados dados necessários à execução da ação na API, conforme o que se pretenda. São apresentados dois exemplos para melhor se perceber:

1. A aplicação Android solicita os dados de um item pelo seu ID (item ID 33):
  - a. O URI que deve ser invocado pela aplicação Android, através do método GET do protocolo *Hypertext Transfer Protocol* (HTTP), deverá ser: `http://xpresstrades.com/api/values/getItem/33`
  - b. A Resposta dada pela API em JSON é a seguinte:

```

- {
  ItemID: 33,
  ItemName: "Battlefield 4",
  VersionID: 13,
  VersionName: "",
  GenreName: "FPS",
  GenreName_PT: "Tiro em Primeira Pessoa",
  PlatformCoverName: "Xbox 360",
  PlatformCoverNamePT: "Xbox 360",
  PlatformCode: "XBOX360 ",
  SearchHashtags: "BF4, BF 4",
  ReleaseDate: "2013-11-01T00:00:00",
  PublisherName: "Electronic Arts",
  ItemPreservation: 0,
  InterestLevel: 0,
  NumberWishers: 0,
  NumberOffers: 3,
  Category: 1
}

```

2. A aplicação Android solicita a lista de jogos (itemType=1) do género Corrida (*Racing*) que neste caso é representado pelo valor 1 (ID do género corrida) antes do carácter “?” do URI apresentado:

- a. O URI que deve ser invocado pela aplicação Android:  
`http://xpresstrades.com/api/values/GetItemsByGenre/1?itemType=1`
- b. Resposta da API (obteve-se a lista de jogos de corrida – apresenta-se apenas um item da lista obtida como exemplo):

```
- {  
  ItemID: 7,  
  ItemName: "Gran Turismo 6",  
  VersionID: 8,  
  VersionName: "",  
  GenreName: "Racing, Sport, Simulator",  
  GenreName_PT: "Corrida, Desporto, Simulador",  
  PlatformCoverName: "Playstation 3",  
  PlatformCoverNamePT: "Playstation 3",  
  PlatformCode: "PS3 ",  
  SearchHashtags: "GT6, GT 6",  
  ReleaseDate: "2013-12-06T00:00:00",  
  PublisherName: "Sony Computer Entertainment",  
  ItemPreservation: 0,  
  InterestLevel: 0,  
  NumberWishers: 1,  
  NumberOffers: 2,  
  Category: 1  
}
```

Tendo em conta a natureza das ações dos exemplos apresentados anteriormente, não existe a necessidade de autenticação para estas, daí ter-se utilizado para a sua execução o *Web Browser* Google Chrome com a extensão JSONView instalada (para se obter os dados JSON com uma aparência organizada). Quando se trata de uma ação que requer autenticação (necessário para todas as ações que efetuem alterações ou os dados devam ser privados), a autenticação é enviada no header do HTTP GET e por padrão é codificada através do método Base64 (codificação MIME).

No seguinte endereço, `http://xpresstrades.com/api/values/1/2?3&4&5(...)`, foram numeradas de 1 a 5 as várias zonas que constituem o URI:

- A zona com o valor 1 corresponde ao nome da ação da API. As ações possíveis estão descritas na Tabela 5 do Anexo F, na coluna Ação API.
- As zonas com os valores 2, 3, 4 e 5 são os parâmetros para a ação solicitada. Note-se que o número de parâmetros dependerá da ação solicitada, tal como descrito na

Tabela 5 do Anexo F, na coluna Parâmetros recebidos pela API. Aqui são apresentados os parâmetros que precisam de ser enviados juntos no URI de acordo com a ação. O parâmetro 2 tem de ser declarado após o caracter “?” e os restantes (podem ser mais do que cinco) são sempre declarados após o caracter “&”.

- Ainda poderá haver informação adicional que vai no HEADER do pedido feito através do HTTP GET, tais como as credencias de autenticação dos utilizadores.

A gravação das imagens de perfil dos utilizadores é guardada diretamente no servidor utilizando o protocolo FTP. Note-se que não há comunicação com a API, mas existe comunicação com o servidor na mesma.

### 3.8. Conclusão

Neste capítulo descreveu-se, com algum detalhe, todas as funcionalidades do sistema XpressTrades e os estilos arquiteturais que serão utilizados no sistema.

Como se pode verificar, a arquitetura geral do sistema será constituída pela aplicação Android (cliente) e pelo servidor, onde irão estar hospedados a API e a base de dados MS SQL. Note-se que para esta já se desenhou o modelo relacional.

Saliente-se que além destes aspetos importantes também se planeou a interface de utilizador que por ventura também já foi testada e melhorada de forma a otimizar a sua usabilidade para os utilizadores.

Em suma, após o desenho do sistema encontramos-nos aptos à passar à fase de implementação que será abordada no capítulo seguinte.





## 4. Implementação do Sistema

---

### 4.1. Introdução

Após o desenho do sistema passou-se à fase da implementação do mesmo. Este capítulo destina-se a apresentar o processo de implementação do sistema, explicando com algum detalhe como foram implementadas as funcionalidades e também como estas funcionam. Além disso, serão apresentados *screenshots* da interface do utilizador de forma a facilitar as explicações. Também se considerou ser uma mais-valia identificar os requisitos funcionais e não funcionais que cada funcionalidade satisfaz.

De forma a se conseguir localizar o código de programação correspondente às ações que serão referidas nas explicações das funcionalidades, o leitor deve consultar o Anexo F, mais precisamente a Tabela 5, que faz a correspondência entre as operações solicitadas pela aplicação Android e os métodos que são executados na API e, por sua vez, as *Stored Procedures* que são executadas pela API para realizarem o processamento dos pedidos. Tenha-se em atenção que os tipos de dados não são especificados, mas podem ser facilmente visualizados no código fonte.

Refira-se ainda que, relativamente à implementação da API, esta foi acompanhando as necessidades da aplicação. Todas as funcionalidades da aplicação são realizadas uma parte na aplicação Android e outra grande parte na API e na base de dados. Importa também mencionar que neste capítulo será abordada a forma como foi criado o ícone da aplicação e como foi feita a integração dos vários idiomas.

### 4.2. Integração de vários idiomas

A aplicação foi desenvolvida primeiramente apenas em inglês. Numa fase posterior, graças à forma como os projetos das aplicações Android são organizados (Ver Figura 10), facilmente se traduz a aplicação para outros idiomas. O idioma padrão fica armazenado num ficheiro designado de *strings* e é do tipo *eXtensible Markup Language* (XML). Este, por sua vez, encontra-se dentro do projeto numa pasta com o nome *values*, que se encontra dentro de uma outra pasta chamada *res*. Na pasta *res* só se tem de criar uma nova pasta utilizando o nome *values* com as siglas do país à frente, separado por um hífen e sem espaços. Deste modo, para traduzir a aplicação para a língua portuguesa bastou criar uma pasta com o nome *values-pt* (as letras deverão estar em minúsculas) em que *pt* indica o

país - Portugal. Depois, fez-se uma cópia do ficheiro *strings* da pasta *values* para a pasta *values-pt* e a partir desse momento bastou editar a cópia do ficheiro, mantendo os IDs das *strings*, mas colocando o texto em português [34]. Neste sentido, a aplicação está implementada em dois idiomas, inglês (Ver Figura 11 (a)) e português (Ver Figura 11 (b)), todavia apenas será utilizado o português quando o idioma do sistema operativo estiver configurado para este idioma. Quando não for possível, utilizará o inglês por predefinição. Saliente-se que a escolha da língua é feita automaticamente pelo sistema de acordo com o idioma em utilização pelo mesmo. Desta forma, o requisito não funcional NF3 foi satisfeito.

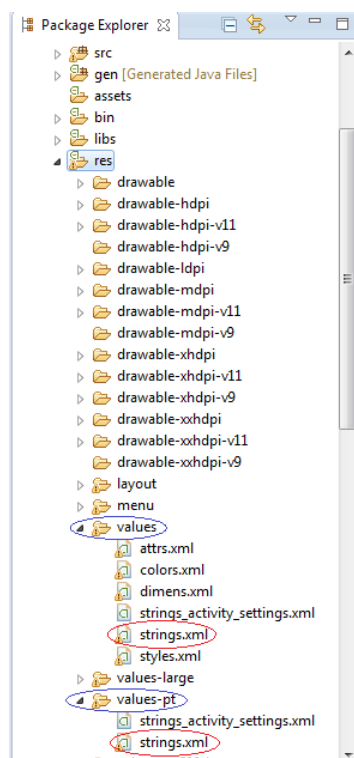
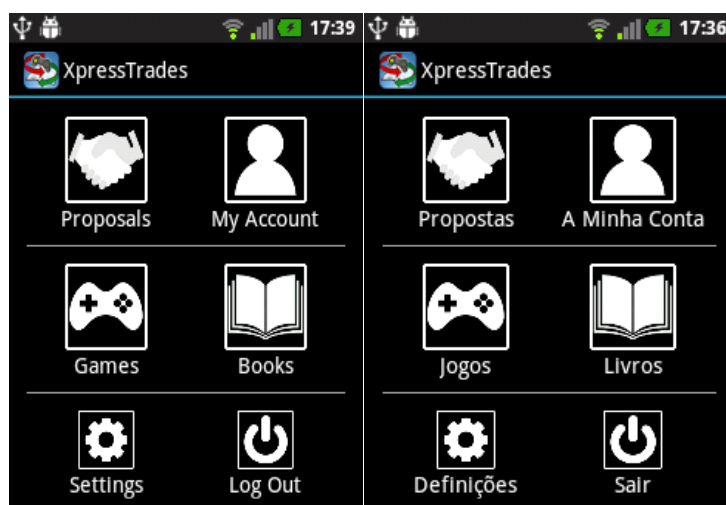


Figura 10 - Estrutura de pastas do projeto Android (com exemplo de como colocar vários idiomas)



(a) (b)  
Figura 11 - Menu principal da aplicação  
(em inglês e em português)

### 4.3. Barra de Ação

A barra de ação, conhecida também como *Action Bar*, é atualmente muito utilizada na maioria das aplicações desenvolvidas. Esta apresenta os botões de acordo com as ações disponíveis a cada momento durante a execução da aplicação [35]. Saliente-se que a barra de ação pode também incluir separadores, conhecidos também por *Tabs*. Na Figura 12 é

possível observar os separadores que, neste caso, representam os vários fragmentos da atividade “Jogos”.

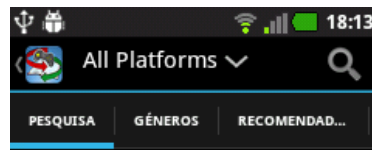


Figura 12 - Barra de ação apresentada na atividade Jogos

Tendo em conta o objetivo de desenvolver uma aplicação que cobrisse praticamente todas as versões Android (a partir da versão 2.2), surgiu a necessidade de utilizar uma biblioteca de compatibilidade disponibilizada pela comunidade de desenvolvimento de aplicações Android [36], de forma a se conseguir ter uma barra de ação funcional nas versões mais antigas. Note-se que havia ainda a possibilidade de criar uma barra de ação com o recurso a um objeto com um *layout*, que depois seria incluída em todas as atividades, contudo iria ser um processo mais moroso e o resultado final seria o mesmo ou inferior [36].

Relativamente à barra de ação foi utilizada uma biblioteca auxiliar designada de “v7 appcompat library” que é disponibilizada no SDK e que ao instalarmos o ADT fica localizada em (`<sdk>\extras\android\support\v7\appcompat`). Para adicioná-la ao projeto é necessário importá-la e, de seguida, alterar as propriedades do projeto da aplicação para incluir esta biblioteca, tal como ilustra a Figura 13.

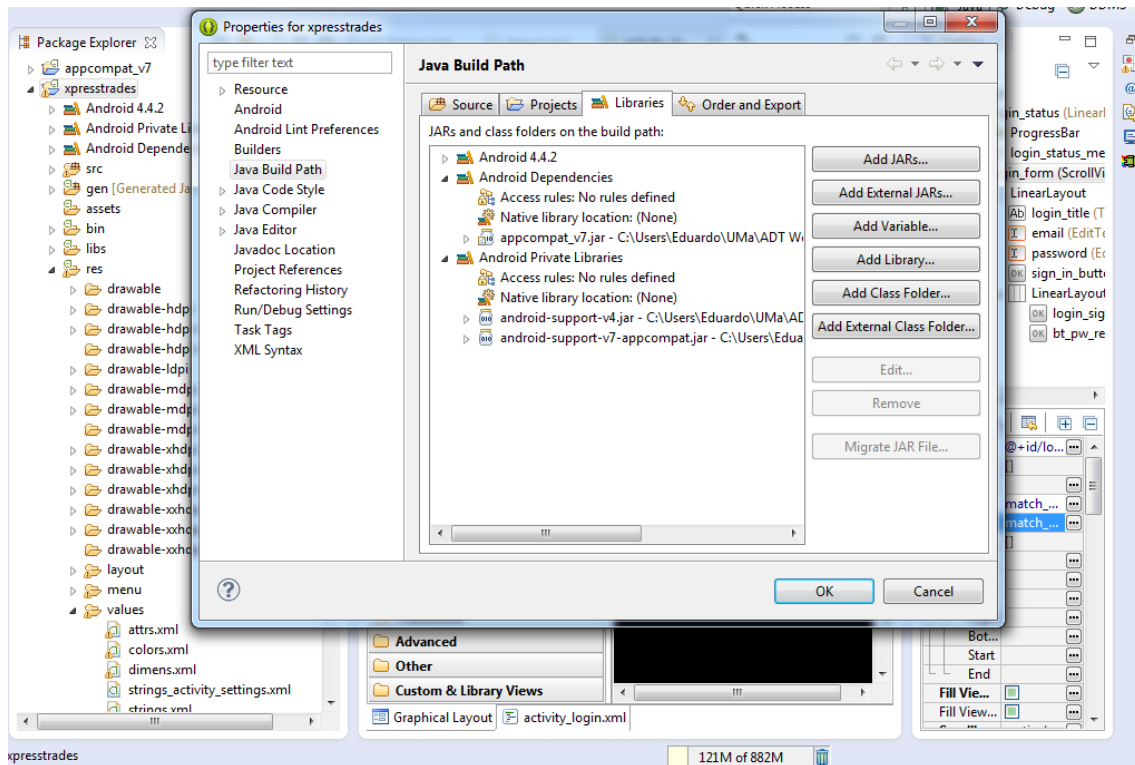


Figura 13 - Adicionando biblioteca “v7 appcompat library” ao projeto

#### 4.4. Funcionalidades da aplicação

Para implementar as funcionalidades foram criadas várias atividades. Cada uma das atividades carrega um *layout* principal e outros secundários. Tenha-se em atenção que muitas das atividades possuem fragmentos, cada fragmento tem o seu próprio *layout*, mas apesar disso nada invalida a utilização do mesmo *layout* para dois fragmentos ou mais, isto se utilizarem a mesma estrutura visual. Nos pontos seguintes as funcionalidades da aplicação serão descritas utilizando a versão *Smartphone* e o sistema está em português; como tal, a aplicação também será apresentada na versão portuguesa.

De forma a facilitar a explicação das funcionalidades, é importante referir que todas as ações necessitam de comunicação com o servidor *online* para serem efetuadas, o que torna esta uma aplicação *online*. Isto significa que as ações efetuadas são realizadas através da comunicação da aplicação Android com a API *online*. Ressalve-se que grande parte da funcionalidade está na API e nas *Stored Procedures* (SP) armazenadas na própria base de dados MS SQL utilizada. A aplicação Android no fundo é a “ponta do iceberg”, ou seja, é o que está visível ao utilizador e é a partir dela que os utilizadores conseguem realizar as suas interações com o sistema.

#### 4.4.1. Atividade Registo de Utilizador

No início do registo é solicitado que o utilizador escolha uma imagem para o seu perfil. O utilizador pode escolher tirar uma foto no momento, utilizando a câmara do dispositivo, ou seleccionar uma imagem já existente no dispositivo (Ver Figura 14 (a)). Após a escolha da imagem, esta passa para o modo de recorte (Ver Figura 14 (b)) onde o utilizador pode cortar a imagem (na forma de um quadrado) podendo aumentar ou diminuir o tamanho desse quadrado sobre a imagem e pode ainda movê-lo para a zona que pretende seleccionar. Desta forma, garante-se que a imagem irá respeitar a proporção pretendida e o utilizador terá maior liberdade para escolher apenas partes de imagens de tamanho superior para a sua imagem de perfil. A proporção é importante para que as imagens dos utilizadores sejam apresentadas com a melhor aparência possível na aplicação. O resultado final fica como ilustra a Figura 14 (c).

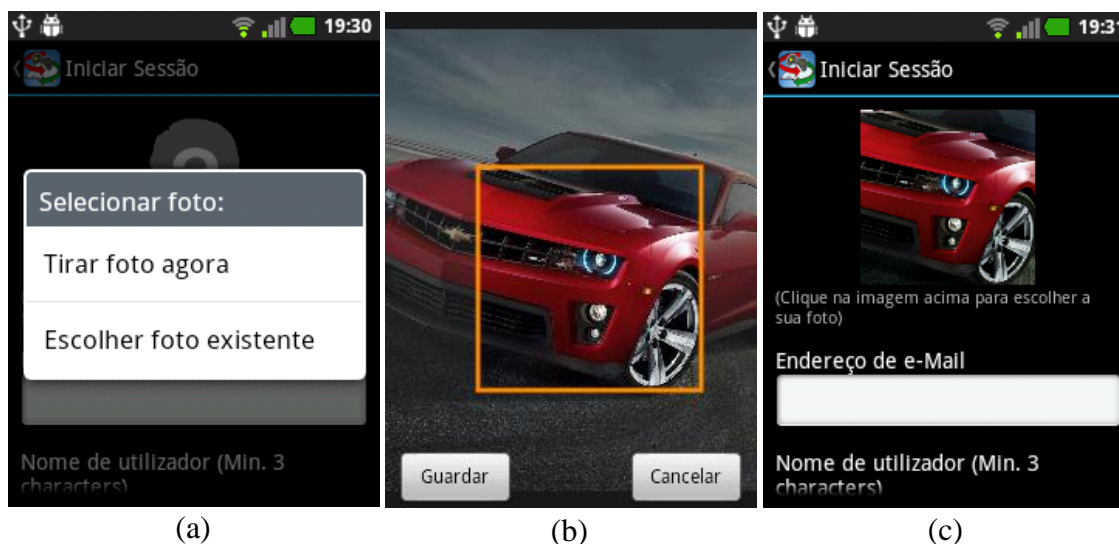


Figura 14 - Processo para escolher a imagem para o perfil de utilizador

Quando é seleccionada a atividade “Registo”, é carregada a lista de todos os nomes de utilizador e *e-mails* registados até ao momento (Ver Ação 1 da Tabela 5 do Anexo F) para depois ser possível verificar rapidamente se o nome de utilizador e *e-mail* que o utilizador introduziu já existem (a operação é repetida na base de dados, pois pode acontecer, por exemplo, que outra pessoa esteja a registar-se no mesmo momento com um mesmo nome de utilizador).

Para o utilizador indicar em que cidade vive, primeiro tem de seleccionar o seu País (ou adicioná-lo se este ainda não existir). A lista com os nomes dos países é carregada (Ver Ação 2 da Tabela 5 do Anexo F) logo após a entrada na atividade “Registo de Utilizador”

(SignUpActivity). Após a escolha do País, é carregada a lista de regiões existentes para o país selecionado (Ver Ação 3 da Tabela 5 do Anexo F), enviando o ID do país selecionado como parâmetro e o utilizador neste momento escolhe a sua região. Logo depois, a lista de cidades é carregada (Ver Ação 4 da Tabela 5 do Anexo F) seguindo o mesmo procedimento da lista de regiões, com a diferença de que o parâmetro enviado é o ID da região selecionada e, caso existam, são carregadas apenas as cidades da região selecionada. Desta forma, obriga-se o utilizador a seguir uma ordem lógica, indo do mais geral para o mais específico, tornando mais fácil a seleção da cidade em que este vive.

Após o utilizador preencher os formulários e clicar no botão “Registar-me” (Ver Figura 15), é feita uma verificação na aplicação se os campos estão corretamente preenchidos e algumas mensagens de erro serão apresentadas se os campos obrigatórios não estiverem totalmente preenchidos ou se estiverem incorretos. No caso do nome de utilizador e *e-mail* são verificados se já foram registados e se já estiverem, é apresentada uma mensagem de erro também. Como é óbvio, a verificação mais profunda é feita aquando da execução da *Stored Procedure* do registo de utilizador na base de dados. Quando o utilizador clica em “Registar-me”, toda a informação preenchida é enviada através do método GET do protocolo de comunicação HTTP para a API (Ver Ação 5 da Tabela 5 do Anexo F). Na *Stored Procedure* desta ação é feita novamente a verificação se o nome de utilizador ainda não está a ser utilizado e se o *e-mail* ainda não está registado. Caso estejam registados, é devolvido o número do erro para a API e a API devolve esse resultado no formato JSON para a aplicação. A aplicação, por sua vez, de acordo com o número sabe qual o erro ocorrido e notifica o utilizador para proceder a correção do mesmo. Caso não ocorram erros, o utilizador está em condições de ser adicionado à base de dados, ainda assim algumas verificações são feitas, pois durante o registo, se o utilizador não encontrar o seu país, região ou cidade poderá acrescentá-lo. Neste sentido, a *Stored Procedure* verifica se foram adicionados alguns deles e, se sim, adiciona-os à base de dados antes de acrescentar o utilizador, isto porque é necessário saber-se o ID da cidade para atribuir esse ID no campo de cidade do utilizador na tabela utilizadores (*users*). Após este processo, o utilizador é adicionado à base de dados e, ao mesmo tempo, é gerado um código de ativação para a conta do mesmo. Esse código é gerado internamente na base de dados através de uma função que constrói uma combinação aleatória de oito caracteres alfanuméricos.



Figura 15 - Final de registo após opções de localização do utilizador

Relativamente ao código de ativação, é importante referir que este nunca é enviado para a aplicação, mas sim para o *e-mail* do utilizador. Desta forma, garante-se que o mesmo não é enviado na comunicação com a aplicação, protegendo-o de ataques por *Sniffing* (processo que consiste em obter os dados quando estes estão a passar pela rede, na comunicação), por parte de um utilizador que se esteja a tentar registar com o endereço de *e-mail* de outra pessoa. O objetivo do código de ativação é garantir que as pessoas que se registam estejam realmente a utilizar o seu *e-mail*, ou pelo menos um *e-mail* ao qual têm acesso. Evita-se assim que os endereços de correio eletrónicos de outras pessoas sejam utilizados indevidamente.

Após o registo com sucesso na base de dados, a *Stored Procedure* devolve o código de ativação gerado para esse utilizador à API. Esta envia um *e-mail* (Ver Figura 16) para o endereço de correio eletrónico fornecido no registo, que contém o código de ativação da conta. Com recurso a este processo simples, obriga-se os utilizadores a fornecerem um *e-mail* verdadeiro, isto porque o utilizador apenas pode tentar ativar a sua conta três vezes. Depois dessas tentativas infrutíferas, a conta é eliminada. Nesta atividade foram implementados vários requisitos funcionais tais como F1, F2, F3, F4, F5, F6, F7 e F43 e também alguns requisitos não funcionais como o requisito NF4, NF5, NF8, NF9, NF10, NF11, NF12, NF14 e NF27.

Relativamente ao código de ativação, é possível pedir o reenvio do código para o *e-mail* até três vezes. Desta forma evita-se que a aplicação seja utilizada como ferramenta de

envio de *e-mails* indesejados (conhecidos por SPAM). A estrutura do *e-mail* enviado com o código é apresentado na Figura 16.

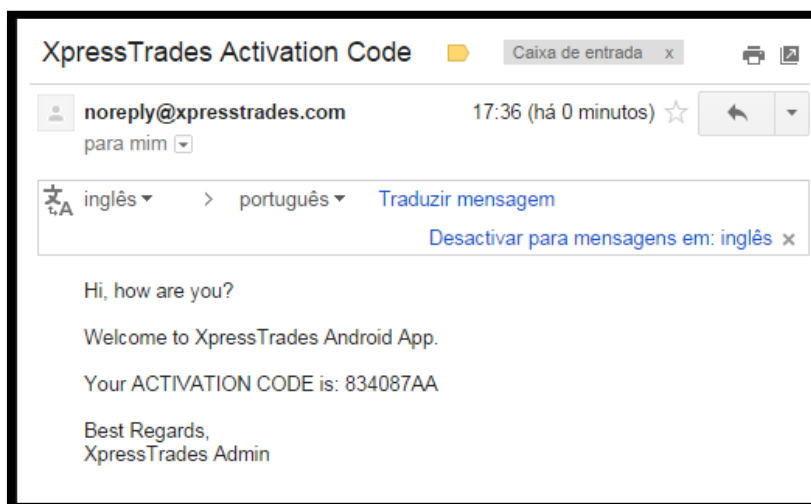


Figura 16 - Mensagem de *e-mail* com código de ativação

Após o registo ser efetuado com êxito, é enviado o resultado para aplicação no formato JSON que indica que o utilizador foi registado com sucesso. Após receber este resultado, a imagem do perfil do utilizador é gravada diretamente no servidor, numa pasta específica. Através do protocolo FTP são gravados dois ficheiros de imagem do tipo JPG, um numa dimensão superior e outro numa dimensão inferior. O nome dos ficheiros é construído utilizando uma letra como prefixo. Essa letra indica se é a imagem de dimensão superior (letra *b* de *big*) ou se é a de dimensão inferior (letra *s* de *small*), seguida do nome de utilizador e a respetiva extensão “.jpg”. Note-se que o nome de utilizador nunca pode ser alterado e é único (cada utilizador tem de ter um nome distinto). O objetivo de ter duas imagens de perfil de utilizador em dimensões diferentes é a otimização do tempo que é necessário para carregar as fotos dos utilizadores na lista de proprietários e na lista de interessados que iremos ver mais à frente neste capítulo.

#### 4.4.2. Atividade Login

De seguida, apresenta-se um resumo das ações que podem ser efetuadas nesta atividade:

- Aceder à atividade “Registo de Utilizador”;
- Ativar conta de utilizador;
- Pedir reenvio de código de ativação;



- Recuperar palavra-passe (*password*);
- Iniciar sessão.

Quando o utilizador se regista na aplicação, este tem de ativar a sua conta apenas da primeira vez que inicia a sessão (Ver Figura 17). Como é possível visualizar na Figura 18, é apresentada uma janela para introduzir o código de ativação. Nessa mesma janela também é possível solicitar (até três vezes) o código de ativação (para que este seja enviado para o *e-mail* de registo novamente) (Ver Ação 7 da Tabela 5 do Anexo F). A ativação da conta é feita através da Ação 8 da Tabela 5 do Anexo F, em que o utilizador tem de colocar o código de ativação que recebeu e clicar em Ativar. Se o código for o correto, a conta fica ativada; se não, o número de tentativas (inicialmente três) vai diminuindo e quando se esgotarem, a conta é eliminada. Com este número de tentativas evitamos que o código tente ser descoberto à força (*Brute Force*) por tentativa e erro.

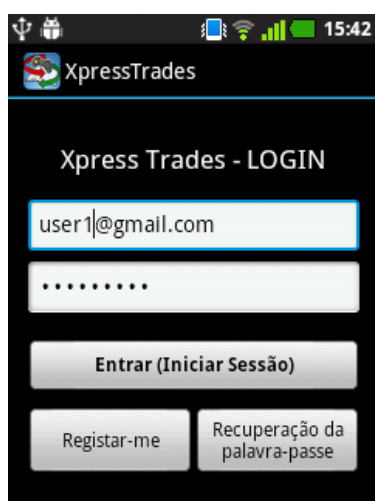


Figura 17 - Atividade Login

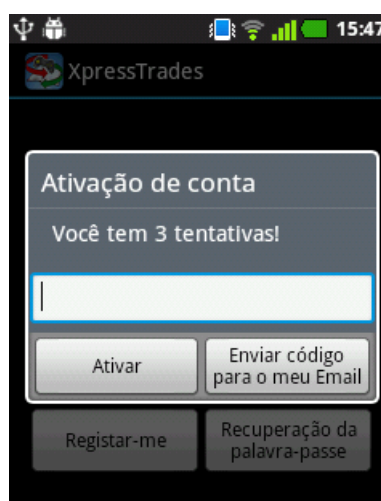


Figura 18 - Janela de Ativação de Conta

Finalmente, se o utilizador já tiver a sua conta ativada e efetuar início de sessão fornecendo as respetivas credenciais, este irá iniciar a sessão (Ver Ação 6 da Tabela 5 do Anexo F) e será enviado para a atividade “Menu Principal” (HomeActivity) da aplicação. Se o utilizador não efetuar o término da sessão (*Logout*), esta ficará sempre iniciada, não sendo necessário realizar o *login* quando regressar à aplicação. Note-se que os dados de sessão ficam guardados localmente até que a aplicação seja desinstalada, os dados sejam limpos ou então o utilizador termine a sessão. Sempre que o utilizador acede à aplicação já com a sessão iniciada, a aplicação atualiza na base de dados a data e hora do último acesso à mesma (Ver Ação 13 da Tabela 5 do Anexo F), de forma a mostrar essa informação no perfil público do utilizador. Esta informação permite aos outros

utilizadores saberem qual foi a última vez que este esteve ativo na aplicação. Além disso, as plataformas de jogos também são carregadas (Ver Ação 15 da Tabela 5 do Anexo F) quando o utilizador acede à aplicação.

Hoje em dia, sabemos que, por vezes, as pessoas têm dificuldade em ter todas as *passwords* memorizadas, isto porque é recomendado ter uma *password* diferente para cada conta [37]. Todavia, tendo em consideração a quantidade de contas que os utilizadores têm, acabam por esquecer algumas. Como tal, é essencial um sistema de recuperação de *passwords*. Neste sentido, no ecrã de login existe uma opção para recuperar a *password* (é utilizada a Ação 9 da Tabela 5 do Anexo F). Tudo o que o utilizador precisa de fazer é fornecer o seu endereço de *e-mail* (Ver Figura 19), com o qual efetuou o registo na aplicação. Podemos visualizar um exemplo de uma mensagem de *e-mail* recebida pelo utilizador que solicitou a recuperação de *password* na Figura 20.

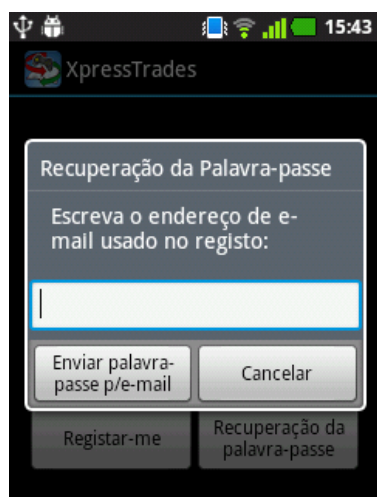


Figura 19 - Recuperação de Palavra-Passe

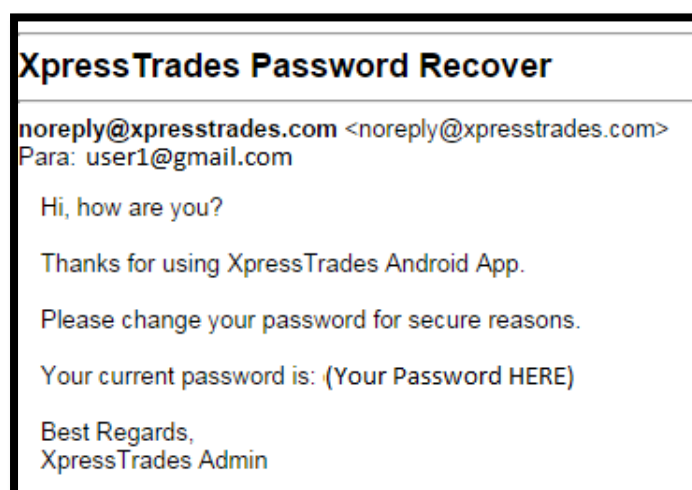


Figura 20 - Exemplo da mensagem de recuperação de palavra-passe (*password*)

Nesta atividade foram implementados vários requisitos funcionais tais como F8, F9, F10 e o F11 e também alguns requisitos não funcionais como o requisito NF6, NF7, NF8, NF10, NF11, NF14, NF15 e NF27.

#### 4.4.3. Atividade Menu Principal (HomeActivity)

De forma a implementar o menu principal da aplicação (requisito funcional F1 e requisito não funcional NF25), criou-se um *layout* com botões-imagem acompanhados da sua designação. Cada um desses botões permite ao utilizador navegar para atividades distintas, de forma a satisfazer as várias opções disponíveis.

Os botões-imagem foram ordenados da seguinte forma: daqueles que os utilizadores mais usam para os que são menos usados, pois os testes com os utilizadores levaram à conclusão de que as propostas são a opção mais acedida na utilização a médio-prazo. As imagens foram montadas com recurso ao *software* Photoshop. Quando são premidas, mudam de cor, pois cada botão possui duas imagens, a imagem do botão pressionado e a do botão não seleccionado. Através de um ficheiro do tipo XML, localizado em “<Pasta do Projeto>\res\drawable”, é definido qual a imagem que deve ser apresentada quando o botão-imagem não é seleccionado e quando é seleccionado. Este processo foi utilizado para os seis botões-imagem que constituem o menu principal (Ver Figura 21).

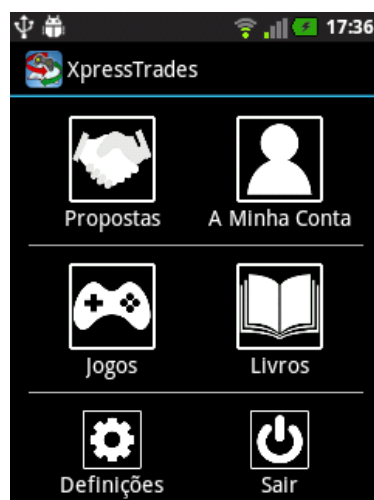


Figura 21 - Atividade Menu Principal (Home Screen)

#### 4.4.4. Atividade Jogos/Livros

Antes de se avançar com a explicação da implementação da atividade “Jogos”/“Livros”, é importante referir que os itens da categoria Livros e da categoria Jogos partilham dados análogos, os Jogos possuem plataformas específicas (cada jogo só funciona numa máquina específica, cada máquina é designada de plataforma de jogo) enquanto os Livros possuem tipos de capas diferentes (duras, moles, etc.), ambos têm um título, os Jogos possuem um publicador e os Livros um editor, ambos têm géneros, ambos possuem versões, ambos têm datas de lançamento e uma imagem de capa. Desta forma, no futuro, é possível adicionar novas categorias de itens com pouco esforço, pois grande parte do código já está implementado.

O código foi reutilizado para as duas categorias de itens, pois na realidade são a mesma atividade, mas ao seleccionar o botão-imagem “Jogos” a atividade recebe um valor a

indicar que os itens são da categoria Jogos e, desde então, tudo fica restringido a itens desta categoria. O mesmo procedimento acontece ao premir o botão-imagem “Livros”. As pequenas variações entre as duas categorias foram facilmente tratadas. Os Jogos têm uma complexidade maior que os Livros devido às plataformas, e por isso, foram criadas ações específicas para os Jogos.

Esta atividade “Jogos”/“Livros” está subdividida em vários fragmentos, cada fragmento corresponde a um separador. A navegação entre separadores poderá ser feita tocando por cima do nome do separador ou deslizando o dedo sobre o ecrã do dispositivo, para a esquerda ou para a direita. Os separadores podem ser deslocados horizontalmente de forma a ser possível visualizar todos, mas sem ter de os selecionar imediatamente.

#### **4.4.4.1. Separador Pesquisa**

O separador inicialmente selecionado é o “Pesquisa” (Ver Figura 22 (a)). Neste separador é possível aceder à atividade onde o utilizador pode enviar um pedido ao administrador para adicionar um novo item, através do botão apresentado na parte inferior da Figura 22 (a). A pesquisa pode ser acedida clicando sobre o ícone da lupa (observar o canto superior direito da Figura 22 (a)) a partir de qualquer um dos separadores desta atividade. No separador “Pesquisa” são apresentados os resultados de uma pesquisa, este é automaticamente selecionado quando a lupa é selecionada. Outro aspeto importante é que, no caso dos Jogos, a pesquisa será limitada à plataforma que estiver selecionada no momento (Ver Ação 18 da Tabela 5 do Anexo F), como ilustra a Figura 22 (b). Se não tiver nenhuma plataforma escolhida, então o sistema assume uma pesquisa em todas as plataformas, tendo apenas em conta o nome fornecido pelo utilizador e a categoria Jogos (Ver Figura 22 (c)). É importante frisar que, como na categoria Livros não existem plataformas, a pesquisa é sempre feita de forma global (Ver Ação 17 da Tabela 5 do Anexo F). Este separador satisfaz os requisitos funcionais F14 e F15 e os requisitos não funcionais NF8, NF14, NF17, NF21, NF22, NF24 e NF27.



Figura 22 - Atividade Item: Separador Pesquisa

#### 4.4.4.2. Atividade Pedir para Adicionar Novo Item

A partir do separador de pesquisa mencionado anteriormente, é possível aceder à atividade que permite ao utilizador enviar um pedido ao administrador para adicionar o seu item (livro ou jogo, de acordo com a categoria escolhida no menu principal). Esta atividade tem a particularidade de ir buscar (Ver Ação 19 da Tabela 5 do Anexo F) a lista de todos os títulos, géneros, publicadores (no caso dos Jogos) ou editores (no caso dos Livros), e plataformas (no caso dos Jogos) ou capas de livros (no caso dos Livros) existentes na base de dados que pertençam à categoria escolhida. No caso da Figura 23, a categoria é Jogos, então a atividade vai buscar todos os títulos de jogos existentes na base de dados. Deste modo, quando o utilizador começa a escrever o título do item que quer adicionar, ao fim da terceira letra o sistema apresenta uma lista de sugestões. O mesmo foi feito para os géneros, plataformas e publicadores, mas nestes basta uma letra para começarem as sugestões. A intenção é facilitar o preenchimento do formulário de pedido de itens que já existam mas que tenham apenas alguma diferença, por exemplo na versão. Cada item é uma combinação única de atributos como o título, os géneros, a versão, etc.

Não é permitido o utilizador adicionar diretamente os itens na aplicação sem antes passar por uma verificação. Neste caso, é utilizada uma tabela na base de dados onde ficam registados os pedidos (*requests*). Quando o utilizador termina o preenchimento do formulário (que não obriga ao preenchimento de todos os dados), submete-o no sistema (através da Ação 20 da Tabela 5 do Anexo F). O administrador depois gere estes pedidos

verificando se os dados estão corretos e adiciona os itens que realmente ainda não existem. Foi definido um período de 24 horas como limite para serem adicionados os itens pedidos. Esta atividade satisfaz o requisito funcional F16 e os requisitos não funcionais NF8, NF14, NF21 e NF27.

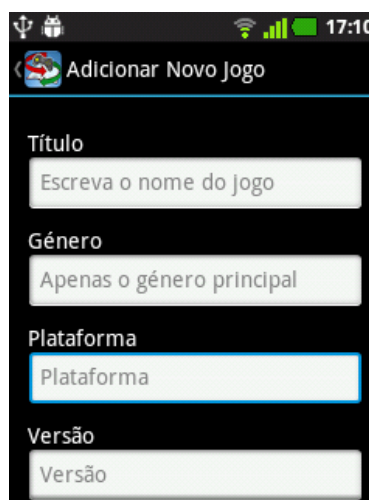


Figura 23 - Adicionar novo item (Categoria Jogos)

#### 4.4.4.3. Separador Géneros

Quando o utilizador seleciona o separador “Géneros” (*Genres*), como mostram as Figuras 24 (a) e 24 (b), todos os géneros desta categoria são carregados (Ver Ação 21 da Tabela 5 do Anexo F) numa nova atividade que recebe o título do género selecionado (Ver Figura 24 (c)). Essa atividade satisfaz os requisitos funcionais F17, F18, F19, F20 e F50. Mais uma vez, de acordo com a plataforma selecionada, as pesquisas pelo género serão filtradas pela mesma (Ver Ação 22 da Tabela 5 do Anexo F). Se a opção selecionada nos Jogos for a de Plataformas, então a pesquisa pela plataforma (Ver Figura 24a) será análoga à dos Livros (Ver Figura 24 (b)), pois só é necessário ter em conta o género e a categoria de itens (Ver Ação 23 da Tabela 5 do Anexo F). Este separador satisfaz o requisito funcional F51. Quanto aos requisitos não funcionais, tanto no separador como na atividade do género selecionado foram satisfeitos os seguintes: NF14, NF17, NF21, NF22, NF24 e NF27.

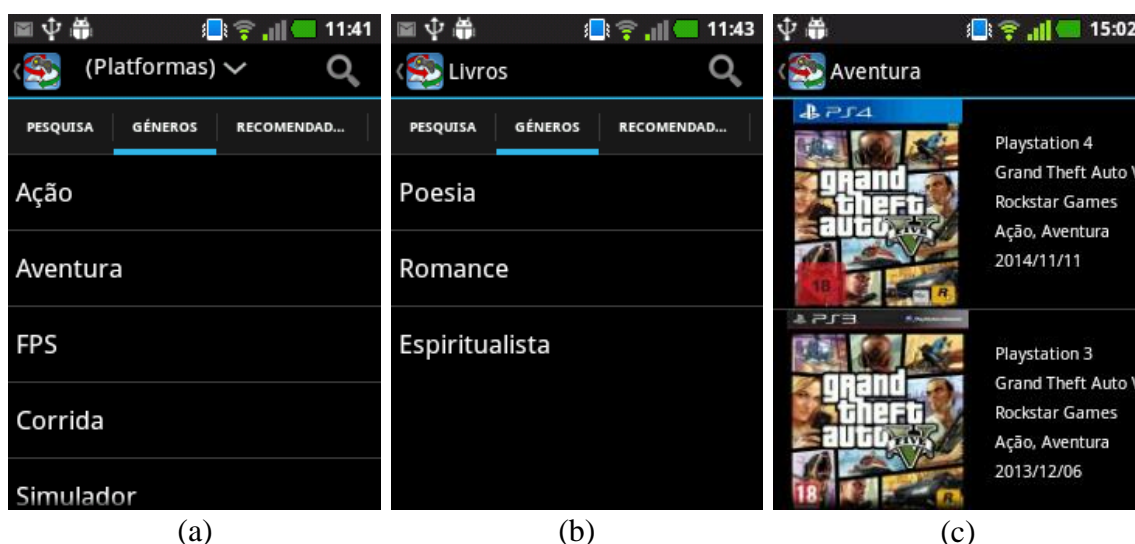


Figura 24 - Separador “Gêneros”

#### 4.4.4.4. Separador Itens Recomendados

Antes de passar à explicação do funcionamento do separador de itens recomendados, é importante esclarecer o funcionamento do sistema de recomendação. Para o pretendido não foi necessário utilizar sistemas de recomendação complexos. Este sistema foi desenvolvido sem recurso a algoritmos existentes e funciona da seguinte forma (Ver diagrama da Figura 54 do Anexo C): existe na base de dados uma tabela que regista os géneros preferidos do utilizador (*preferred\_genres*); quando o utilizador visualiza o perfil de determinado item, é executada uma ação (Ver Ação 14 da Tabela 5 do Anexo F) que consiste em verificar os géneros a que esse item pertence e incrementar o contador respetivo. Por exemplo, se o utilizador acede ao perfil de um jogo de ação e aventura, então o valor de contagem dos géneros ação e aventura são incrementados.

Para obtermos a lista dos itens recomendados (Ver Ação 25 da Tabela 5 do Anexo F), é feita a seleção dos cinco géneros preferidos do utilizador, ordenados por ordem decrescente, pelo número de contagens. Após isto, são escolhidos aleatoriamente dois itens de cada género preferido, seguindo a mesma ordem dos géneros selecionados (Ver Figura 25 (a)). No total, são dez itens recomendados. No caso dos Jogos, se houver uma plataforma específica selecionada a recomendação respeitará também a plataforma (Ver Ação 24 da Tabela 5 do Anexo F), como demonstram as Figuras 25 (b) e 25 (c). No caso do utilizador ainda não ter visitado itens suficientes para criar uma lista de cinco géneros preferidos, são utilizados os géneros que já existirem e as restantes recomendações serão feitas de forma aleatória com itens de outros géneros. Nesta lista de 10 itens recomendados os itens nunca são repetidos. A ideia de apresentar itens aleatórios quando



não há informação suficiente no histórico de visualizações é para evitar que se apresente um separador vazio, prejudicando assim a aparência da aplicação.

No que concerne ao separador dos itens recomendados, estes são obtidos pela Ação 24 ou 25 da Tabela 5 do Anexo F que implementa o sistema de recomendação e é apresentada na Figura 25. É importante mencionar que todas as listas apresentadas com itens ou propostas respeitam um critério no carregamento das imagens, isto porque as imagens são os dados que mais demoram a ser carregados na aplicação. De forma a otimizar este processo de carregamento, as imagens são carregadas apenas para os itens que ficam visíveis no ecrã a cada momento, ou seja, o utilizador navega na lista e, quando pára numa determinada posição, são verificados os itens da lista que estão visíveis ao utilizador, sendo carregadas apenas as respetivas imagens.

Este separador satisfaz os requisitos funcionais F19, F20, F21, F22 e F50. Quanto aos requisitos não funcionais, foram satisfeitos os seguintes: NF14, NF21, NF22, NF24 e NF27.



Figura 25 - Separador “Recomendações” (Categoria Jogos)

#### 4.4.4.5. Separadores Top Desejados/Top Oferecidos

Os separadores “Top Desejados” (Ver Figura 26) e “Top Oferecidos” (Ver Figura 27) funcionam de forma similar, a diferença é que o Top Desejados apresenta os trinta itens que mais pessoas desejam e o Top Oferecidos apresenta os trinta itens que mais pessoas possuem para oferecer em troca. Na base de dados (consultar Figura 5 do Cap.4) podemos ver que, na tabela item, temos um atributo para a contagem do número de interessados



(nr\_wishers) e outro para o número de proprietários (nr\_offers). Estes atributos são atualizados sempre que os utilizadores adicionam ou removem os itens das suas listas de ofertas (atualizando o nr\_offers) ou das suas listas de desejos (atualizando o nr\_wishers), de forma a fazer corresponder ao número exato dos utilizadores que desejam e dos que têm o item para oferecer. Ou seja, os itens são ordenados de forma decrescente de acordo com o número de interessados no caso do Top Desejados (Ver Ação 26 da Tabela 5 do Anexo F). Em relação ao Top Oferecidos (Ver Ação 28 da Tabela 5 do Anexo F), os itens são ordenados também de forma decrescente, mas tendo em conta o número de proprietários (número de ofertas).

Importa mencionar que, na categoria Jogos, estes Tops podem ser filtrados por plataformas de jogo (Ver Ação 27 para Top Desejados filtrados por plataforma e ver Ação 29 para Top Oferecidos filtrados por plataforma, da Tabela 5 do Anexo F). Note-se que os Tops são sempre filtrados por categoria, nunca há mistura de itens da categoria Jogos com a categoria Livros quando são apresentadas estas listas. É possível ainda adicionar qualquer item à lista de ofertas ou à lista de desejos a partir de qualquer lista de qualquer separador desta atividade (Ver Figura 28). Para isso, apenas é necessário que o utilizador faça um toque longo sobre o item que deseja adicionar e, em seguida, são apresentadas duas opções: adicionar à lista de desejos ou adicionar à lista de ofertas. Se o item já existir numa das listas, aparece uma mensagem a indicar que o item não pode ser adicionado. Quando o item é adicionado com sucesso, é também apresentada uma mensagem por breves segundos.

Relativamente aos requisitos funcionais, o separador “Top Desejados” satisfaz os F23, F24, F50 e o separador “Top Interessados” satisfaz os F25, F26 e F50 e ambos satisfazem os F19, F20. Quanto aos requisitos não funcionais, os dois incluem os NF14, NF21, NF22, NF24 e NF27.



Figura 26 - Separador “Top Desejados” (Categoria Jogos)



Figura 27 - Separador “Top Oferecidos” (Categoria Jogos)

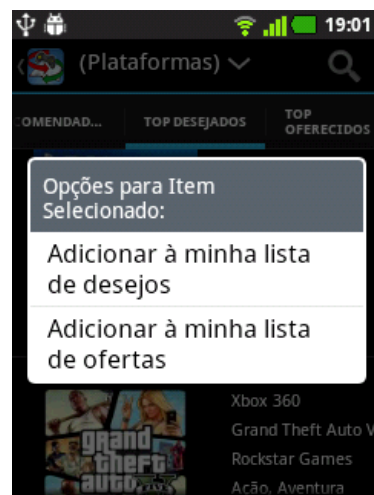


Figura 28 - Ações para longo clique sobre um item (Jogo/Livro)

#### 4.4.5. Atividade Perfil de Item

Depois da explicação de como pesquisar itens, faz sentido explicar a atividade “Perfil de Item”, que é acessado através do toque num item de uma das listas mencionadas na atividade anterior (Jogos ou Livros). Esta atividade é subdividida em três fragmentos, que correspondem a três separadores diferentes: “Perfil”, “Proprietários” e “Interessados”.

Esta atividade satisfaz os seguintes requisitos funcionais:

- No separador “Perfil”: F19, F20, F27, F28, F31 e F32;
- No separador “Proprietários”: F29;
- No separador “Interessados”: F30.

Os requisitos não funcionais presentes nesta atividade, mais precisamente nos três separadores, são: NF14, NF17, NF18, NF21, NF22, NF24, NF25 e NF27. O requisito não funcional NF19 está presente apenas no separador “Perfil”.

##### 4.4.5.1. Separador Perfil do Item

É no separador “Perfil” que são apresentadas as informações do item, como se observa na Figura 29 (a). Aqui o utilizador pode clicar em cima da informação do item para saber de que informação se essa (ex.: título, ano de lançamento, etc.). É também possível visualizar o número de utilizadores que possuem o item e o número de utilizadores que estão interessados nele. Além disto, neste separador o utilizador pode adicionar um item

à sua lista de ofertas (Ver ação 31) ou de desejos (Ver Ação 38 da Tabela 5 do Anexo F), isto porque são apresentados botões que permitem estas ações. Tenha-se presente que estes botões só aparecem se o utilizador ainda não possuir o item em nenhuma das suas listas, pois caso contrário, aparecerá um botão com a opção de remover o item da lista de ofertas (Ver Ação 33 da Tabela 5 do Anexo F) ou da lista de desejos (Ver Ação 40 da Tabela 5 do Anexo F) de acordo com a lista onde este estiver. Outra funcionalidade neste separador é a apresentação da imagem do item em ecrã inteiro. Para tal, é necessário tocar em cima da imagem do item e é aberta uma nova atividade que vai buscar a imagem de dimensão superior.



Figura 29 - Atividade “Perfil do Item”

#### 4.4.5.2. Separadores Proprietários/Interessados

No separador “Proprietários” encontra-se uma lista de utilizadores (Ver Ação 36 da Tabela 5 do Anexo F), através da qual é possível ver o estado de conservação do item de cada proprietário, a classificação (*feedback*) que estes utilizadores receberam (Ver Figura 29 (b)) e ainda a Cidade/Concelho dos utilizadores. Outro aspeto, que é um dos grandes pontos fortes desta aplicação, corresponde exatamente à ordenação dos utilizadores por distância, ou seja, os utilizadores são apresentados dos mais próximos para os mais distantes de modo que os utilizadores possam fazer trocas com os utilizadores mais próximos. O serviço *Web* da API do Google Maps poderia ter sido facilmente utilizado para obter as distâncias entre os utilizadores. Contudo, após alguma pesquisa, verificou-se que nos limites de uso desta API [38], não é permitido o seu uso se a informação obtida não for apresentada num mapa Google e, além disso, tem um limite diário de consultas.

Depois de se ter verificado estes aspetos, outra solução foi pensada e desenvolvida, pois só se pretendia ordenar os utilizadores da lista por distância. Neste sentido, implementou-se uma função na base de dados, designada de *GetDistance*, recorrendo à fórmula de Haversine (1) [39]. Esta função recebe 4 coordenadas, a latitude e a longitude da cidade a que o utilizador pertence e a latitude e longitude da cidade do outro utilizador e devolve a distância entre estes dois pontos, tendo em conta a curvatura do globo. Quando é solicitada a lista de proprietários (Ver Ação 36 da Tabela 5 do Anexo F), esta utiliza a função *GetDistance* para obter a distância que separa o utilizador dos outros utilizadores. As coordenadas estão na tabela *city* da base de dados.

Na *Stored Procedure* utilizada (Ver Ação 36 da Tabela 5 do Anexo F), é feita uma pesquisa dos utilizadores que possuem o jogo juntamente com a sua localização (coordenadas das cidades a que pertencem). Saliente-se que, como a distância é sempre em relação ao utilizador, o cálculo desta vai receber sempre as mesmas coordenadas da parte do utilizador e as outras duas coordenadas é que irão variar, pois são as que correspondem à localização dos proprietários dos itens. No final, a lista é ordenada de acordo com a distância, do mais próximo para o mais distante e devolvida em JSON à aplicação Android.

$$\text{haversin}\left(\frac{d}{R}\right) = \text{haversin}(\Delta\phi) + \cos(\phi_1) \cos(\phi_2) \text{haversin}(\Delta\lambda) \quad (1)$$

De forma a proteger a privacidade dos utilizadores, a distância calculada é tendo sempre em conta as coordenadas do centro das cidades/concelhos. Se o utilizador quiser indicar o seu endereço postal, poderá enviá-lo numa mensagem pré-feita quando se encontrar dentro de uma proposta.

O utilizador pode, a qualquer momento, clicar no botão presente no canto superior direito (na barra de ação) para atualizar as listas (Ver Figura 29). Quando se clica na imagem de um utilizador, na lista, é aberta uma nova atividade – “Perfil Público do Utilizador” – em que são mostrados alguns dados sobre esse utilizador (Ver Ação 12 da Tabela 5 do Anexo F). Se o utilizador selecionar o botão “Enviar Proposta”, é aberta a atividade enviar proposta onde o utilizador pode selecionar um item para oferta (se tiver alguma opção válida).

Relativamente ao separador “Interessados” (Ver Figura 29 (c)) funciona de forma análoga ao separador “Proprietários”. A única diferença é que, em vez do estado de conservação, são indicados os níveis de desejo/interesse e, neste momento, não é permitido realizar

propostas aos utilizadores interessados. Para obter a lista de interessados é utilizada a Ação 41 da Tabela 5 do Anexo F.

#### 4.4.5.3. Atividade Detalhes do Utilizador

A atividade “Detalhes do Utilizador” (Ver Figuras 30 (a) e 30 (b)) permite aceder ao perfil público de um utilizador. Normalmente, é acedido através do toque na fotografia de um utilizador da lista de ofertas ou da lista de desejos de um item. Para tal, é apresentada a fotografia numa dimensão superior, a data em que o utilizador se registou, a última vez que este esteve online, a sua idade e a sua localização (Cidade, Região e País). Saliente-se que, se o utilizador que está a visualizar o perfil tiver dúvidas em relação ao significado de cada ícone, basta tocar em cima do mesmo para receber a sua descrição numa janela flutuante. Por exemplo, se o utilizador clica sobre o primeiro ícone, a mensagem que será apresentada por breves segundos será “Data de registo na aplicação”.

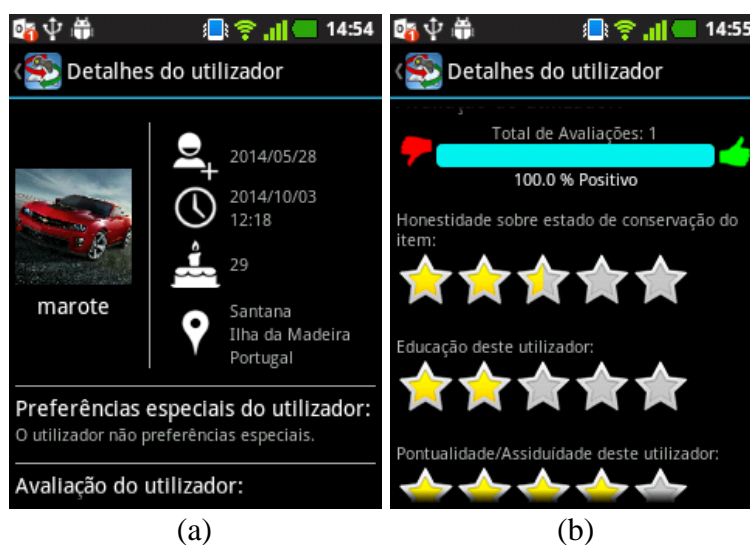


Figura 30 - Atividade “Detalhes do Utilizador”

Neste perfil também é possível visualizar, nas preferências especiais do utilizador, que opções tem em relação às propostas que lhe podem ser feitas, por exemplo: se só aceita livros ou jogos desejados ou se só aceita jogos de determinadas plataformas. Além disto, também é possível visualizar informação mais detalhada sobre a avaliação (*feedback*) que este utilizador tem recebido. Assim sendo, é possível ver quantas avaliações recebeu e a percentagem de avaliações positivas e negativas, ou ainda como este é classificado quanto à honestidade da descrição do estado de conservação dos itens, em relação à pontualidade

e assiduidade e em relação ao seu nível de educação (se o utilizador trata os outros utilizadores com o devido respeito).

Esta atividade cumpre o requisito funcional F33 e os requisitos não funcionais: NF17, NF20, NF25 e NF27.

#### 4.4.5.4. Atividade Enviar Proposta

É importante frisar que, no contexto desta aplicação, uma proposta consiste em um utilizador propor uma troca de um item que possui por outro item que outro utilizador detém.

A atividade “Enviar Proposta” (Ver Figura 31) é acedida através do clique no botão “Enviar Proposta” presente na lista “Proprietários”. Esta atividade permite o envio de uma proposta a outro utilizador (Ação 44 da Tabela 5 do Anexo F). Contudo, não é garantido que o utilizador que pretende fazer a proposta tenha itens para poder oferecer em troca, pois a lista de ofertas apresentada irá ter em conta as preferências especiais do outro utilizador e só os itens que satisfazem essas preferências serão apresentados como opção. Como se pode ver na Figura 31 (a), o utilizador tem de clicar em “Selecionar o meu Item” e, neste momento, é apresentada a lista dos itens (Ver Figura 31 (b)) que possui e que podem ser oferecidos em troca (Ver Ação 34 da Tabela 5 do Anexo F). Assim sendo, são feitas algumas verificações na *Stored Procedure* desta ação para filtrar os itens.

Antes de passar à explicação desta *Stored Procedure*, vamos assumir que o utilizador F (de *From*) é aquele que vai fazer a proposta e o utilizador T (de *To*) é aquele que irá receber a proposta. É importante mencionar que o utilizador F nunca pode utilizar itens da sua lista de ofertas que já estejam numa proposta em negociação. Posto isto, suponha-se o caso de uma proposta de troca de livros. Quando o utilizador F entra na atividade “Enviar Proposta”, é verificado pela *Stored Procedure* se o utilizador T só deseja receber propostas com livros desejados; se for o caso, a lista de ofertas possíveis do utilizador F (se existir alguma correspondência) conterá os itens da lista de ofertas do utilizador F que existirem também na lista de desejos do utilizador T. Caso o utilizador aceite receber propostas com qualquer Livro, então a lista terá todos os itens da lista de ofertas do utilizador F que não estejam envolvidos em propostas em negociação.

No caso dos Jogos, é um pouco mais complexo, isto porque se o utilizador T só quiser receber propostas com Jogos da sua lista de desejos, o raciocínio é análogo ao caso descrito anteriormente para os Livros. Mas, se o utilizador T não tiver esta opção ativa, ainda há mais uma filtragem – as plataformas. Se o utilizador T não tiver todas as plataformas selecionadas nas definições, então o utilizador F só poderá fazer propostas com jogos que pertençam às plataformas selecionadas pelo utilizador T. No caso do utilizador T ter todas as plataformas selecionadas e não ter a opção de receber apenas jogos desejados, então é apresentada a lista das ofertas do utilizador F (excluindo os que já estão em propostas em negociação). Se a lista apresentar itens para oferta, então o utilizador escolhe um item da sua lista para oferecer em troca. Neste momento, o botão “Selecionar o meu Item” é substituído pelo item selecionado (Ver Figura 31 (c)). De seguida, o utilizador só tem de selecionar o botão “Enviar Proposta” (poderá ter de deslizar o ecrã para o encontrar). Saliente-se que é possível o utilizador alterar o item que está oferecer, bastando tocar no item selecionado e a lista de ofertas é apresentada novamente.

Esta atividade satisfaz os requisitos funcionais F34 e F35. Relativamente aos requisitos não funcionais, satisfaz os N14, NF21, NF22 e NF27.



Figura 31 - Atividade “Enviar Proposta”

#### 4.4.6. Atividade Definições

Como mencionado anteriormente, o utilizador pode selecionar certas preferências para filtrar as propostas que quer receber. Essas opções são configuradas na atividade “Definições” (*Settings*) (Ver Figura 32) e as opções são guardadas através da Ação 11



presente da Tabela 5 do Anexo F. Existem duas tabelas na base de dados que guardam estas definições, a tabela `user_choosed_game_platforms` onde são guardadas as plataformas de jogo para as quais o utilizador aceita receber propostas (se não tiver registo de nenhuma, é porque o utilizador aceita todas) e a tabela `user_settings` onde ficam registadas as opções, nomeadamente se o utilizador aceita apenas propostas com jogos desejados ou apenas com livros desejados.

No que concerne à seleção das plataformas, se o utilizador carregar por cima da opção “Plataformas”, esta seleciona/desseleciona todas as plataformas consoante as opções escolhidas anteriormente. Todavia, é obrigatório o utilizador selecionar pelo menos uma plataforma, indicando assim a sua preferência pelos jogos apenas dessa plataforma de jogo. Se o utilizador não tiver preferências específicas, deverá manter todas as plataformas selecionadas, pois assim irá receber propostas com jogos de qualquer plataforma. Note-se que o sistema não aceita a condição de nenhuma plataforma estar selecionada. Se, por alguma razão, não for possível gravar as alterações, é apresentada uma janela a perguntar se quer tentar guardar novamente ou desistir das alterações (Ver Figura 32 (c)).

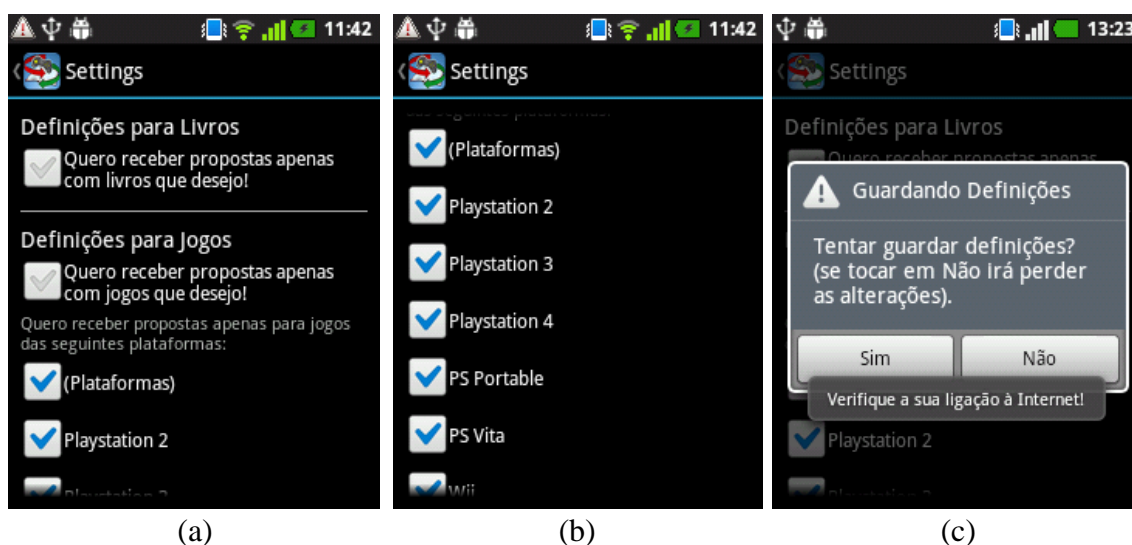


Figura 32 - Atividade “Definições”

Esta atividade satisfaz os requisitos funcionais F36 e F37 e os requisitos não funcionais NF9 (caso o utilizador não selecione nenhuma plataforma de jogo), NF14, NF21 e NF27.



#### 4.4.7. Atividade A Minha Conta

Na atividade “A Minha Conta” o utilizador pode gerir certos aspetos da sua conta, visualizar as estatísticas das avaliações (*feedbacks*) que tem recebido e ainda aceder às listas de desejos e de ofertas de Livros e Jogos (Ver Figura 33 (a)). Refira-se que as estatísticas das avaliações são análogas ao já explicado na atividade “Detalhes do Utilizador” (Secção 4.4.5.3).

Relativamente aos aspetos de gestão permitidos neste momento são: alteração da foto de perfil (similar à forma como o utilizador escolhe a imagem do perfil aquando do registo) (Ver Figura 33 (b)), alteração da palavra-passe (*password*), alteração da localização (País, Região e/ou Cidade) e alteração do seu endereço postal (aquele que só é partilhado por vontade do utilizador e para um utilizador em específico). Em relação à alteração da foto, esta é guardada de imediato, substituindo a existente, e é utilizado o mesmo processo (gravação através de FTP), tal como explicado na atividade “Registo de Utilizador” (Secção 4.4.1).

Nesta atividade, o utilizador ainda poderá visualizar dados da sua conta que não pode editar, tais como o seu nome de utilizador, o seu *e-mail*, o seu nome verdadeiro e a sua data de nascimento.

Após a alteração de dados, ao sair da atividade (voltar para trás), é verificado se os dados foram alterados e, caso tenham sido alterados, é apresentada uma janela (Ver Figura 33 (c)) a perguntar se o utilizador pretende guardar as alterações (Ver Ação 10 da Tabela 5 do Anexo F); Caso contrário, sai sem perguntar nada. O utilizador também pode mandar gravar os dados através do botão “Guardar Alterações” (poderá ser necessário deslizar o ecrã para baixo (*scroll down*) até o botão ficar visível). Se ocorrer algum erro de comunicação, é aberta uma janela a perguntar se o utilizador quer tentar novamente ou desistir das alterações.



Figura 33 - Atividade “A Minha Conta”

Esta atividade satisfaz os requisitos funcionais F2, F3, F4, F5, F6, F7, F38, F39, F40, F41, F42 e F43 e os requisitos não funcionais NF8, NF9, NF14, NF16, NF21, NF23, NF25 e NF27.

#### 4.4.8. Atividade “Os Meus Jogos” e “Os Meus Livros”

Nesta atividade é apresentada a lista de itens que o utilizador possui e a lista de itens que deseja obter. Mais uma vez, temos a mesma atividade para itens de categorias diferentes. Assim, quando se escolhe “Os Meus Jogos”, acedemos à lista de ofertas e à lista de desejos apenas para itens da categoria Jogos. Para a categoria Livros, procede-se da mesma forma. Deste modo, explicar-se-á apenas a atividade da categoria Jogos, visto que a outra funciona de forma análoga.

Neste sentido, quando acedemos à atividade “Os Meus Jogos”, existem dois separadores que possuem a designação de “Eu Tenho” (Ver Figura 34 (a)) e “Eu desejo” (Ver Figura 35 (a)). Desta forma, usa-se uma linguagem muito simples mas intuitiva para qualquer utilizador. Como se pode deduzir o separador “Eu Tenho”, apresenta uma lista dos itens (Ver Figura 34 (a)) que o utilizador possui (obtém-se através da Ação 37 da Tabela 5 do Anexo F), enquanto no separador “Eu desejo” é apresentada a lista de desejos do utilizador (Ver Figura 35 (a)), ou seja, itens que pretende obter no futuro (obtida através da Ação 43 da Tabela 5 do Anexo F).

Para interagir com um item de qualquer lista, utiliza-se o toque longo por cima do item que se quer gerir. Como resultado, é aberta uma janela com as opções possíveis. Relativamente aos itens da lista de ofertas, temos as seguintes opções (Ver Figura 34 (b)):

- Mover para a lista de desejos – remove um item da lista de ofertas e adiciona-o à lista de desejos. Neste processo é necessário o utilizador indicar o nível de desejo (Ver ações 33 e 38 da Tabela 5 do Anexo F);
- Alterar o estado de conservação (Ver Figura 34 (c)) – O utilizador poderá querer alterar o estado de conservação de um dos seus itens da sua lista de ofertas (Ver ação 32 da Tabela 5 do Anexo F);
- Remover item da lista de ofertas – O utilizador pode remover o item da sua lista de ofertas (Ver ação 33 da Tabela 5 do Anexo F).



Figura 34 - Lista de ofertas (Categoria Jogos)

Em relação às opções possíveis para itens da lista de desejos, é possível:

- Mover para a lista de ofertas – remove um item da lista de desejos (Ver Ação 40 da Tabela 5 do Anexo F) e adiciona-o à lista de ofertas (Ver Ação 31 da Tabela 5 do Anexo F). Neste processo é necessário o utilizador indicar o nível de conservação do item. Este processo é especialmente útil quando um utilizador adquire um item que deseja e quer colocá-lo já na sua lista de ofertas;
- Editar o nível de desejo (Ver Figura 35 (c)) – Alterar o nível de desejo de um item selecionado na lista de desejos (Ver Ação 39 da Tabela 5 do Anexo F);

- Remover item da lista de desejos – Remove o item selecionado da lista de desejos (através da ação 40 da Tabela 5 do Anexo F).

Ao tocar sobre a barra (verde ou vermelha, conforme se esteja na lista de desejos ou na lista de ofertas), aparece uma descrição numa janela flutuante a indicar o que representam estas barras. A verde indica o nível de conservação do item (Ver na Figura 34 (a)) e a vermelha indica o nível de desejo pelo item (Ver Figura 35 (a)). O processo para adicionar itens à lista de ofertas ou de desejos já foi explicado anteriormente na secção 4.4.5.1 (Separador “Perfil de Item”).

Nesta atividade são satisfeitos os seguintes requisitos funcionais:

- No separador “Eu Tenho”: F31, F44, F46, F48, F50
- No separador “Eu desejo”: F32, F45, F47, F49, F50

Relativamente aos requisitos não funcionais, nesta atividade foram satisfeitos os seguintes: NF14, NF21, NF22, NF23 e NF24.



Figura 35 - Lista de desejos (Categoria Jogos)

#### 4.4.9. Atividade Propostas

Para gerir as propostas temos a atividade denominada “Propostas”. Neste sentido, as propostas estão divididas em dois separadores distintos, propostas recebidas e propostas enviadas. As propostas são apresentadas em listas ordenadas por data, das mais recentes (início da lista) para as mais antigas (fim da lista) e podem ser filtradas de cinco formas distintas (Ver Ação 45 da Tabela 5 do Anexo F):

- Propostas com novidades – Apresentará as propostas que são novas (ainda não abertas pelo utilizador) ou então propostas já visualizadas mas que possuem mensagens por ler.
- Propostas que estão à espera de *Feedback* – Apresenta todas as propostas aceites que ainda não foram avaliadas. Note-se que esta é uma forma de incentivar e de facilitar os utilizadores a darem o seu *feedback*, pois sabem exatamente a que propostas ainda não deram avaliação.
- Propostas em Negociação – Esta mostrará propostas que ainda estão em negociação, ou seja, propostas recebidas sobre as quais os utilizadores ainda não se decidiram (Aceitar ou Rejeitar para os que receberam e Cancelar para o utilizador que enviou a proposta. Saliente-se que este apenas pode cancelar se a proposta ainda não tiver sido aceite).
- Propostas Aceites – Este apresenta a lista das propostas que foram aceites, ou seja, que os utilizadores chegaram a um acordo (na fase de negociação) e que decidiram proceder à troca (conceito central deste projeto). Tenha-se em atenção que a avaliação só será dada quando se trata de propostas aceites, pois se não houver troca, não faz sentido haver avaliação.
- Propostas Rejeitadas – Apresenta propostas que foram rejeitadas pelos utilizadores que as receberam.

Em qualquer estado (Aceite/Rejeitada/Em Negociação) que as propostas se encontrem, é sempre possível os utilizadores comunicarem no contexto dessas propostas, excepto quando a proposta passou para o estado cancelada.

Se o utilizador cancelou uma determinada proposta e decidir voltar a realizar a mesma proposta, o sistema automaticamente relembrará a situação e vai colocar essa proposta no estado “Em Negociação”. Quando isto acontece, ao abrir a proposta, o utilizador irá tornar a visualizar as mensagens já trocadas anteriormente no contexto dessa proposta.

Esta atividade cumpre os requisitos funcionais F52, F53 e F54 e os requisitos não funcionais NF14, NF16, NF21, NF22, NF25 e NF27.

#### 4.4.9.1. Atividade Detalhes de uma Proposta

Um utilizador acede aos Detalhes de uma Proposta quando selecciona uma das listas da atividade anterior. Quando o utilizador entra nesta atividade, as mensagens da proposta são carregadas (se existirem) (Ver Ação 46 da Tabela 5 do Anexo F) e é verificado o estado da mesma periodicamente, para o caso desta mudar enquanto o utilizador está a visualizá-la (Ver Ação 49 da Tabela 5 do Anexo F). Dentro de uma proposta é possível realizar várias ações:

- Trocar mensagens com o outro utilizador envolvido na proposta – É possível enviar mensagens ao outro utilizador envolvido na proposta (Ver Ação 47 da Tabela 5 do Anexo F). Em cada mensagem aparece a foto e o nome do utilizador que enviou a mensagem, além da data e da hora da mesma. Note-se que as mensagens ficam com a hora e a data do servidor, contudo a diferença do fuso horário em relação ao servidor é feita na aplicação, ficando as mensagens com a hora e a data do sítio em que o utilizador se encontra;
- Colar o endereço postal no corpo da mensagem – O utilizador se pretender efetuar uma troca pelo correio, terá a necessidade de enviar a sua morada ao outro utilizador. De modo a facilitar esta ação (que poderá ocorrer com frequência), o utilizador poderá colar o seu endereço postal acedendo a esta opção através do toque no botão de menu (o botão do próprio dispositivo) e depois escolhendo esta opção na lista;
- Aceitar uma proposta recebida (que ainda esteja no estado “Em negociação”) – Após a negociação através do sistema de mensagens da aplicação, o utilizador que recebeu a proposta deverá decidir se aceita a proposta. Se aceitar, deverá clicar no ícone presente na barra de ação (canto superior direito – ícone da mão com o polegar virado para cima – Ver Figura 36 (a)). Na sequência desta aceitação, é apresentada uma janela (Ver Figura 36 (b)) a confirmar se o utilizador realmente aceita a proposta, avisando que a partir desse momento se compromete a realizar a troca. Se der seguimento, a ação é feita (Ver Ação 48 da Tabela 5 do Anexo F);
- Rejeitar uma proposta recebida (que ainda esteja no estado “Em negociação”) – O utilizador também poderá rejeitar a proposta (clicando no ícone da mão com o polegar virado para baixo – Ver Figura 36 (a)). Quando é escolhida esta opção, é aberta uma janela de confirmação (de forma a evitar ações involuntárias – Ver

Figura 36 (c)), pois tanto a ação Aceitar e como a Rejeitar não são reversíveis. Se confirmar, a ação é efetuada (Ver Ação 48 da Tabela 5 do Anexo F);



Figura 36 - Atividade "Detalhes de uma Proposta" (Recebida)

- Cancelar uma proposta enviada (que ainda esteja no estado “Em negociação”) – O utilizador que fez a proposta pode cancela-la através do ícone na forma de X a qualquer momento (Ver Figura 37) desde que esta ainda não tenha sido aceite ou rejeitada pelo outro utilizador (Ver Ação 48 da Tabela 5 do Anexo F). Quando o utilizador que recebeu uma proposta está a visualizar os detalhes desta e a mesma é cancelada nesse momento pelo outro utilizador, é apresentada uma mensagem a informar que a proposta foi cancelada e de seguida a proposta é fechada (Aplicação recua para a atividade anterior);
- Avaliar/Dar *Feedback* ao outro utilizador quando a proposta já foi aceite – O utilizador pode aceder à atividade que permite fazer a avaliação.



Figura 37 - Atividade "Detalhes de uma Proposta" (Enviada)

Saliente-se que as mensagens apresentadas quando o utilizador se encontra a visualizar os detalhes de uma propostas são, atualizadas de trinta em trinta segundos. Desta forma, um utilizador pode comunicar quase em tempo real com outro utilizador.

É importante referir que o utilizador pode fazer um toque longo sobre determinados botões, de forma a obter uma descrição da funcionalidade deste. Se o utilizador não souber para que servem os botões apresentados na barra de ação, basta fazer um toque longo sobre estes para receber a descrição. Da mesma forma, poderá tocar nas barras que indicam os estados de conservação dos itens para ficar a saber do que se trata.

Note-se que o utilizador pode aceder ao perfil público do outro utilizador clicando sobre a foto deste (que aparece sempre ao lado de uma mensagem enviada ou recebida). Se o utilizador clicar sobre a sua própria foto, acederá à gestão do seu perfil. Quando o utilizador quiser aceder ao perfil de um item envolvido na proposta, basta clicar na imagem do mesmo. É possível ainda o utilizador atualizar as mensagens manualmente através de uma ação do menu (acedido através do toque no botão menu do dispositivo).

Realce-se também que esta atividade apresenta dois botões que têm o intuito de deslizar o ecrã para o topo ou para o fundo. Estes botões são úteis sobretudo quando existem muitas mensagens e é necessário um deslizar extenso para ir ao topo ver os itens ou para aceder ao botão de *feedback*. Note-se que os botões podem estar escondidos no menu, pois só irão aparecer se existir espaço para estes na barra de ação.

Esta atividade cumpre os requisitos funcionais F33, F50, F55, F56, F57, F58, F59, F60 e F63. Relativamente aos requisitos não funcionais esta cumpre os seguintes: NF14, NF17, NF21, NF22, NF25, NF26 e NF27.

#### **4.4.9.1.1. Atividade Feedback**

Tendo em conta a natureza deste projeto não podia faltar um sistema de *feedback* (avaliações de utilizadores). O sistema de *feedback* ajuda, neste caso, aos utilizadores a se avaliarem entre si e, com o passar do tempo, de acordo com as estatísticas das avaliações recebidas, poderem deduzir com qual dos utilizadores será mais fiável negociar [40]. O *feedback* pode ser dado apenas quando uma proposta já está no estado Aceite. Para o utilizador dar o seu *feedback*, tem de entrar nos detalhes de uma proposta



Aceite que ainda não tenha sido avaliada e clicar no botão “Dar *Feedback*” que aparece logo abaixo dos itens da proposta (Ver Figuras 38 (a) e 38 (b)).

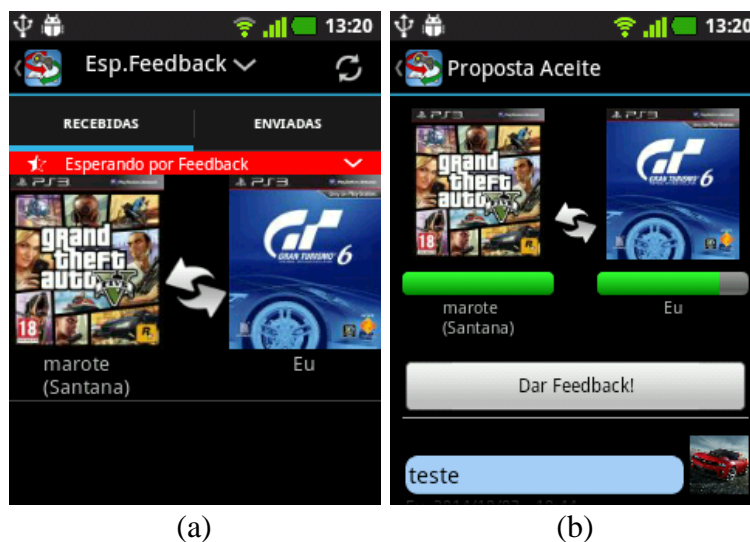


Figura 38 – Aceder à atividade “Dar Feedback”

Tudo o que o utilizador tem de fazer é preencher o formulário de *feedback* (Ver Figuras 39 (a) e 39 (b)) de acordo com a sua satisfação em relação ao outro utilizador. O comentário tem um contador de caracteres e não permite mensagens superiores a 100 caracteres, obrigando os utilizadores a serem objetivos. No final, o utilizador submete o seu *feedback*, mas antes é aberta uma janela (Ver Figura 39 (c)) a confirmar se realmente tem a certeza do mesmo. Quando o utilizador confirma, então o *feedback* é enviado (Ver Ação 50 da Tabela 5 do Anexo F). Note-se que sempre que um utilizador recebe um novo *feedback*, as suas estatísticas são recalculadas de forma a ficarem atualizadas. Esta atividade satisfaz o requisito funcional F62 e os requisitos não funcionais NF16, NF17, NF21, NF23, NF25, NF26 e NF27.

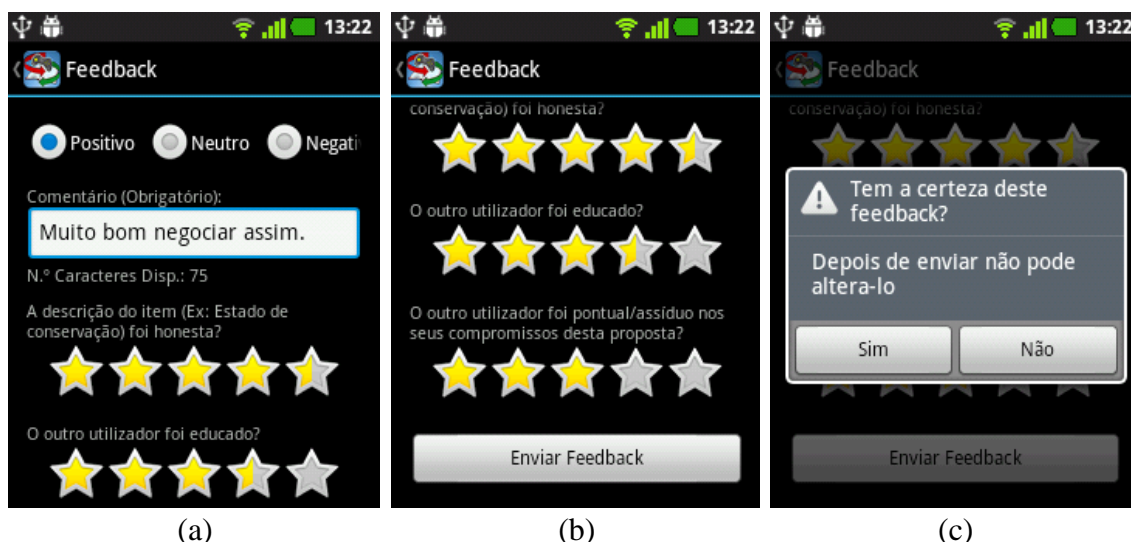


Figura 39 - Atividade "Dar Feedback"

#### 4.4.10. Serviço

O serviço é algo que é executado em *Background*, ou seja, a aplicação pode estar fechada, mas o serviço continua em execução. Na tabela *notifications* da base de dados são guardados os IDs da última mensagem notificada e da última proposta notificada. O serviço entra em execução desde o momento em que a aplicação é instalada (e a sessão é iniciada) ou no arranque do dispositivo. Se houver ligação à internet, o serviço verifica se há novidades (mensagens ou propostas novas) a cada trinta segundos. De forma a otimizar o consumo de energia, os trinta segundos não são exatos, poderá demorar alguns segundos mais. Cada vez que é feita uma verificação, o serviço executa a Ação 51 da Tabela 5 do Anexo F, porém apenas recebe um valor a indicar se precisa notificar ou não o utilizador e internamente atualiza a tabela *notifications* para saber qual foi o ID da última verificação. Desta forma, notifica-se o utilizador apenas uma vez relativamente a alguma mensagem ou proposta nova recebida. A notificação de mensagens ou propostas novas fica localizada na barra de notificações (Ver Figura 40) e ao clicar nela o utilizador é encaminhado para a parte das propostas com novidades, dentro da aplicação.



Figura 40 - Notificação de Propostas ou Mensagens Novas

Relativamente ao serviço, este satisfaz os seguintes requisitos funcionais F64 e F65. Em relação aos requisitos não funcionais este satisfaz o requisito não funcional NF14.

#### 4.5. Ícone da aplicação

O ícone da aplicação foi criado com recurso a ilustrações existentes na biblioteca de imagens do *software* Microsoft Word 2013. Após o ícone ter sido montado, foi aprimorado no *software* Adobe Photoshop CS6 e obteve-se como resultado o ícone apresentado na Figura 41 (a). Este ícone foi redimensionado de acordo com o critério de multiplicação apresentado na Figura 41 (b) e colocado nas respetivas pastas do projeto. Por exemplo, o ícone na dimensão HDPI é colocado na pasta “res\drawable-hdpi” do projeto, o ícone da dimensão XXHDPI é colocado na pasta “res\drawable-xxhdpi”, ou seja, através deste modo colocou-se o ícone nas restantes pastas, tendo sempre em conta as dimensões de ecrã. Na Figura 41 (b) o tamanho do MDPI é de 48x48px (pixels) e os restantes são multiplicações do tamanho deste [41].

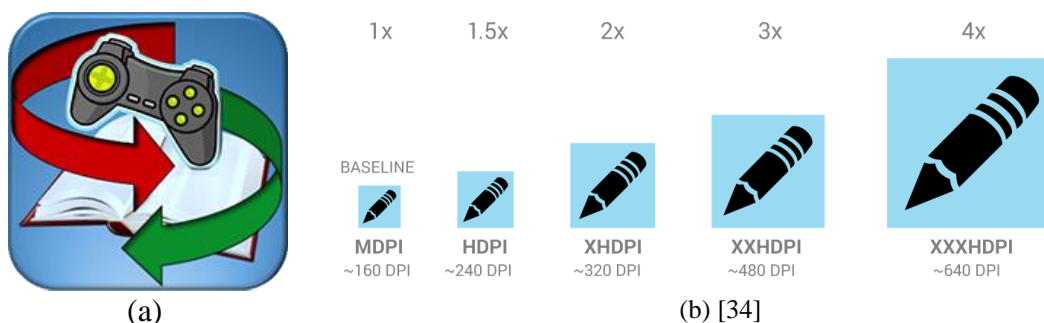


Figura 41 - Ícone da aplicação e dimensões padrão do ícone

#### 4.6. Conclusão

Neste capítulo explicou-se de que modo as funcionalidades foram implementadas, mencionando detalhes acerca de como estas devem ser utilizadas. Além disto, mencionou-se como se criou o ícone da aplicação, como se integrou a língua inglesa e portuguesa e também se mencionou a biblioteca utilizada para se obter uma barra de ação na aplicação nas versões mais antigas do OS Android.

Neste momento faz sentido avançar-se para a fase de testes e de resultados da aplicação, de forma a se poder avaliar o trabalho efetuado e a corrigir eventuais *bugs* que, por vezes, são tão difíceis de detetar. Neste sentido, estes assuntos serão tema do próximo capítulo.

## **5. Testes e Resultados**

---

### **5.1. Introdução**

Antes de se ter terminado a fase de implementação iniciou-se a fase de testes. Esta é uma fase essencial no desenvolvimento e no aprimoramento de qualquer sistema. Apesar do programador verificar constantemente o funcionamento da aplicação ao estar a implementá-la, muitas das vezes só quando a aplicação é testada pelos utilizadores é que é possível detetar certos erros, principalmente os de usabilidade.

Este capítulo apresenta os resultados dos testes de usabilidade (secção 5.3), do estudo de utilização (secção 5.4) e a avaliação geral efetuada pelos utilizadores à aplicação. Os testes realizados à aplicação foram efetuados à maioria das funcionalidades implementadas, descritas no capítulo anterior. Os testes de usabilidade serviram para testar se a interface de utilizador estava fácil e intuitiva de ser utilizada e o estudo de utilização serviu principalmente para se verificar se os utilizadores consideravam úteis as funcionalidades implementadas.

### **5.2. Utilizadores de Teste**

Para a realização do estudo de utilização, dos testes de usabilidade e da recolha das opiniões/avaliações da aplicação Android, selecionou-se cinco voluntários, todos com experiência na utilização do OS Android, pois considerou-se importante os utilizadores possuírem as noções básicas do funcionamento das aplicações Android e se focarem apenas no teste da aplicação.

Note-se que os cinco utilizadores são identificados por um ID, pois os seus nomes não serão divulgados por questões de confidencialidade. As informações dos utilizadores estão presentes na Tabela 4.

Tabela 4 - Lista dos utilizadores de teste

<b>ID do Utilizador</b>	<b>Género</b>	<b>Idade</b>	<b>Tempo de Experiência com o OS Android (Anos)</b>
1	Feminino	26	1
2	Feminino	27	3
3	Masculino	28	4
4	Masculino	29	5
5	Masculino	59	1

### 5.3. Testes de Usabilidade

Os utilizadores seleccionados realizaram os testes de usabilidade utilizando os cenários de utilização presentes no Anexo E. Após a realização desses testes e da utilização da aplicação de forma livre durante uma semana (simulando uma utilização real), estes preencheram o inquérito (online) presente no Anexo G. Nesse inquérito os utilizadores avaliaram e deram a sua opinião relativamente à usabilidade da aplicação. Estes testes possibilitaram a deteção de alguns problemas na aplicação, que durante o desenvolvimento da mesma não tinham sido detetados. Os problemas apontados pelos utilizadores foram os seguintes:

- Na atividade “Propostas”, os elementos que constituem a lista de filtros deveriam ter uma dimensão vertical superior, de forma a facilitar a seleção do filtro desejado. A cor da letra dos itens dessa lista deveria ser branca, de forma a ser mais visível.
- O botão de pesquisa, presente na barra de ação das atividade “Jogos” e “Livros”, deveria estar mais visível.
- A aplicação, por vezes, teve alguma dificuldade em efetuar o carregamento das imagens.
- A aplicação encerrou inesperadamente ao regressar ao menu anterior, quando estava a retornar ao menu principal, a partir da atividade “Jogos”.
- Na aplicação, quando se terminava a sessão e se iniciava uma nova sessão com outra conta de utilizador, o sistema apresentava falhas nas listas de jogos ou livros desejados, tendo de se limpar os dados da aplicação (no gestor de aplicações do OS Android) para esta voltar a funcionar bem.

- A aplicação em inglês possuía alguns erros linguísticos.
- Ao aceder ao perfil de um item, a partir de uma lista, a imagem da capa reduzia ligeiramente; era preferível manter a dimensão.
- Todas as imagens das capas de itens deveriam manter as mesmas dimensões entre elas de forma a dar uma aparência mais uniforme.
- Na atividade “Jogos”, no filtro da plataforma, o nome do primeiro elemento da lista de plataformas de jogo deveria ser “Todas as plataformas”, a cor da letra da lista de plataformas deveria ser branca e os elementos dessa lista deveriam ter uma dimensão vertical maior.

Através da recolha destes dados conseguiu-se detetar alguns problemas presentes na aplicação e resolvê-los.

#### **5.4. Estudo de Utilização**

Após terem realizado os cenários de utilização e terem percebido como a aplicação funcionava, os utilizadores utilizaram-na durante uma semana, de forma a simular a sua utilização no dia-a-dia. Após esse período responderam ao inquérito presente no Anexo G. Os cinco utilizadores manifestaram o desejo de descarregar e utilizar a aplicação quando for lançada no mercado Google Play. Dois dos utilizadores preferem a utilização da aplicação num *Tablet* e três deles preferem a sua utilização em *Smartphones*.

Relativamente às funcionalidades mais utilizadas por estes utilizadores, obteve-se o seguinte resultado:

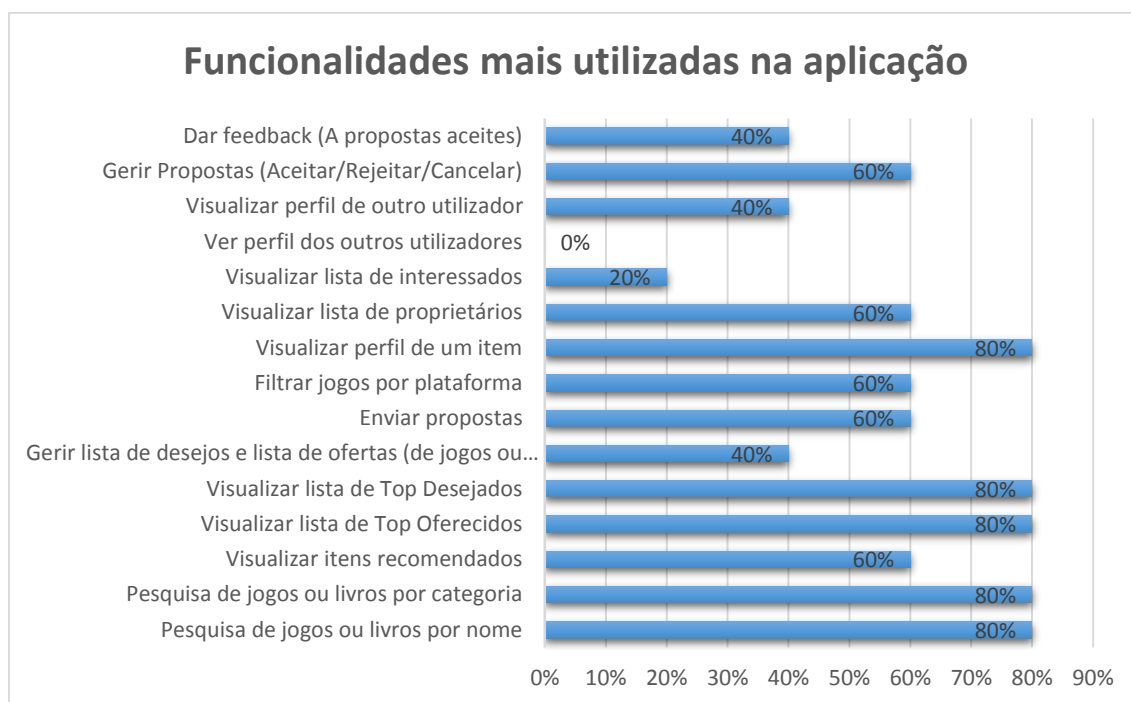


Gráfico 4 – Funcionalidades mais utilizadas na aplicação, pelos utilizadores de teste.

Importa referir que o Gráfico 4 foi construído com base nas respostas dadas à pergunta “Quais as funcionalidades que mais utilizaste na aplicação”, presente no inquérito no Anexo G. Ao visualizar o Gráfico 4, constatamos que os utilizadores não utilizaram muito as funcionalidades “Ver perfil de outros utilizadores” e “Visualizar a lista de interessados”, podendo indicar que estas são pouco úteis. As restantes funcionalidades foram consideradas como “mais utilizadas” por 2 (40%) ou mais utilizadores.

## 5.5. Avaliação do Sistema

Parte do inquérito presente no Anexo G foi utilizada para os utilizadores avaliarem a aplicação relativamente aos seguintes aspetos:

- Funcionalidades que deveriam ser adicionadas;
- Funcionalidades que deveriam ser removidas;
- Classificação da utilidade das funcionalidades atuais da aplicação;
- Avaliação da interface do utilizador relativo aos seguintes pontos:
  - Reação geral à aplicação;
  - *Layout* de visualização;
  - Terminologia e sistema de informação;
  - Facilidade de aprendizagem;



- Capacidades do sistema.

Relativamente à questão “Que funcionalidades deveriam ser adicionadas à aplicação?”, os utilizadores indicaram as seguintes:

- A aplicação deveria possuir uma variedade maior de artigos para troca, além dos jogos e livros.
- Quando nos encontramos a visualizar o perfil de um utilizador, deveria ser possível visualizar as suas listas de desejos e listas de ofertas.
- Deveria ser possível efetuar login através da conta Facebook e/ou Google.
- A aplicação deveria possuir um tutorial para as primeiras utilizações, a explicar o seu funcionamento, pois há certos aspetos que podem passar despercebidos ao utilizador, tais como, por exemplo, a seleção de itens com toques longos.

Neste sentido, a questão mencionada anteriormente, trouxe respostas que serão tidas em conta aquando da melhoria da aplicação num futuro próximo. Note-se que a aplicação encontra-se na sua primeira versão, mas tenderá a evoluir por forma a ir cada vez mais ao encontro dos interesses dos utilizadores. Para isso, ao longo do tempo, serão implementadas novas funcionalidades, de forma priorizada, tendo sempre em conta a relevância das mesmas.

No que concerne à questão “Que funcionalidades achas que deveriam ser retiradas da aplicação?”, praticamente todos os utilizadores acharam que se deveria manter todas as funcionalidades já implementadas. Contudo, houve um utilizador que mencionou o seguinte: *“A troca de livros deveria ser removida da aplicação, esta deveria ser dedicada apenas a jogos, de forma a ser possível ser feito marketing dirigido ao público dos jogos. São mercados e públicos muito distintos.”*

Relativamente à questão que solicitava que os utilizadores classificassem a utilidade das funcionalidades atuais da aplicação, estes consideraram que todas as funcionalidades eram úteis. Para esta classificação utilizou-se uma escala de 0 a 10, onde 0 significa inútil e 10 muito útil.

Através da análise do Gráfico 5 pode-se observar a média das classificações dadas pelos utilizadores. Neste sentido, constatamos que todas as funcionalidades foram avaliadas como úteis, isto porque todas tiveram uma classificação positiva. Note-se que a avaliação das funcionalidades está ordenada no gráfico da melhor classificação para a pior.

Importa referir que a apresentação da data de registo dos utilizadores, a idade e a data e hora do último acesso aplicação foram as funcionalidades que menos classificação obtiveram, revelando-se assim, menos úteis que as restantes.

Relativamente às questões relacionadas com a interface do utilizador obtiveram-se bons resultados, indicando por parte dos utilizadores a satisfação com a mesma.

O Gráfico 6 apresenta as médias das respostas relativamente à reação geral dos utilizadores ao utilizarem a interface gráfica da aplicação. Como se pode visualizar através do gráfico mencionado, considera-se que a reação geral dos utilizadores foi boa, isto porque deram uma classificação mínima de 8,8 e uma máxima de 9,2.

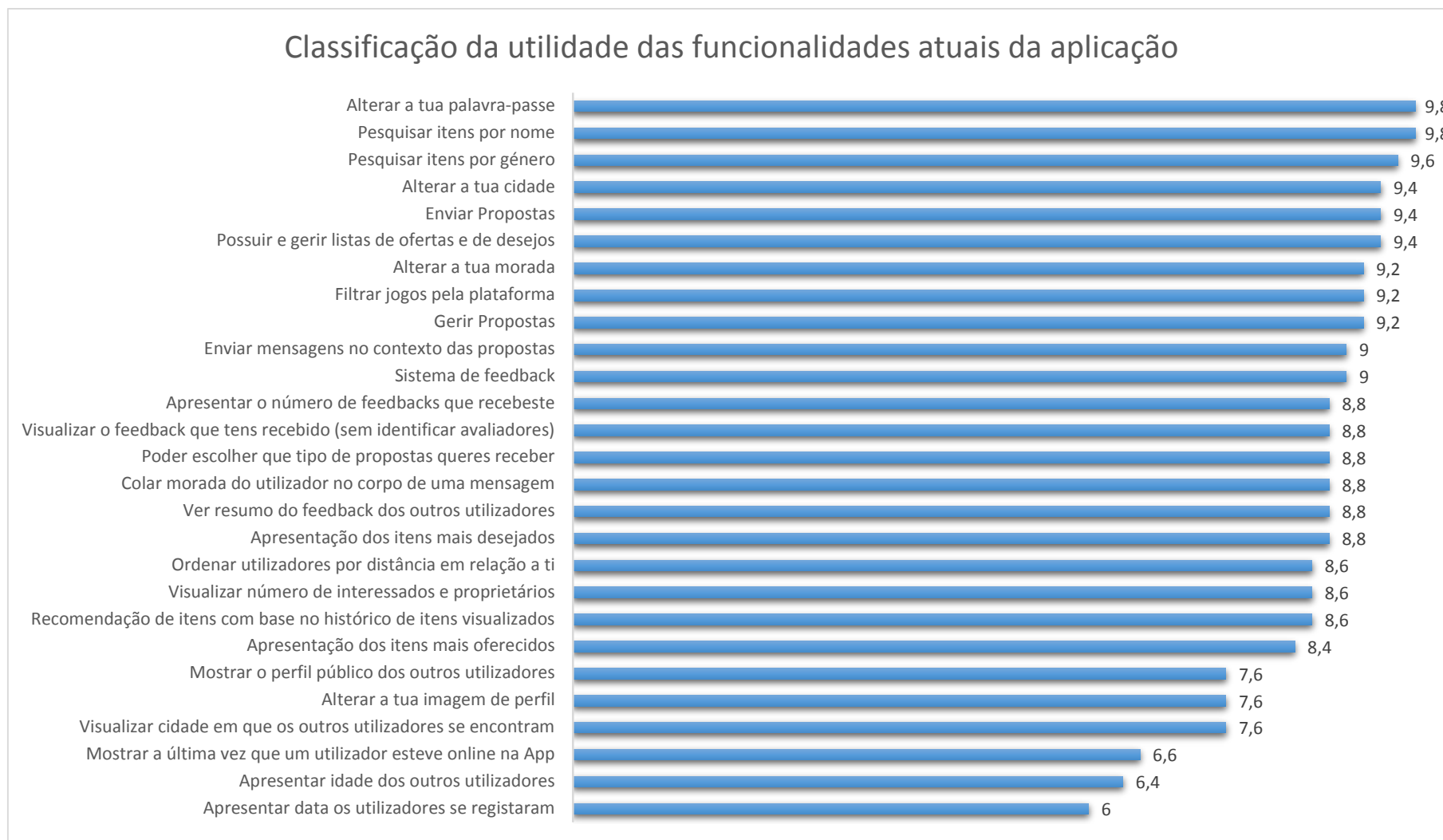


Gráfico 5 – Média da classificação dada à utilidade das funcionalidades atuais da aplicação

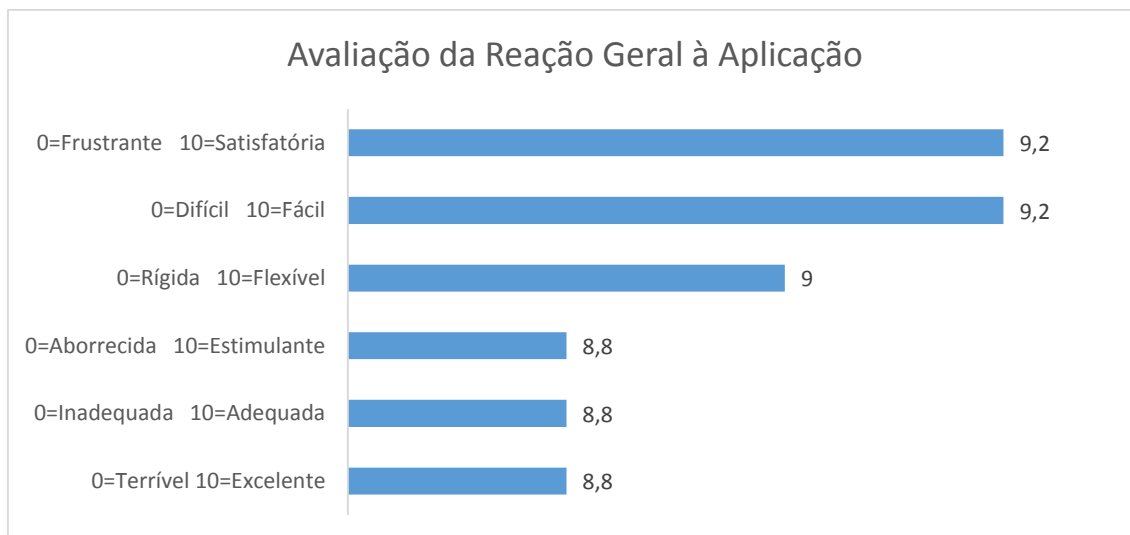


Gráfico 6 – Média das respostas da reação geral à aplicação (relativamente à interface de utilizador)

Como se pode visualizar no Gráfico 7, os utilizadores consideraram que o *Layout* está bom. Escrutinando um pouco o gráfico, verifica-se que a classificação média mínima foi de 9 relativamente às animações nas tarefas e a melhor classificação foi conseguida nos negritos e nas cores utilizadas nos *layouts*, obtendo assim uma classificação de 9,6.

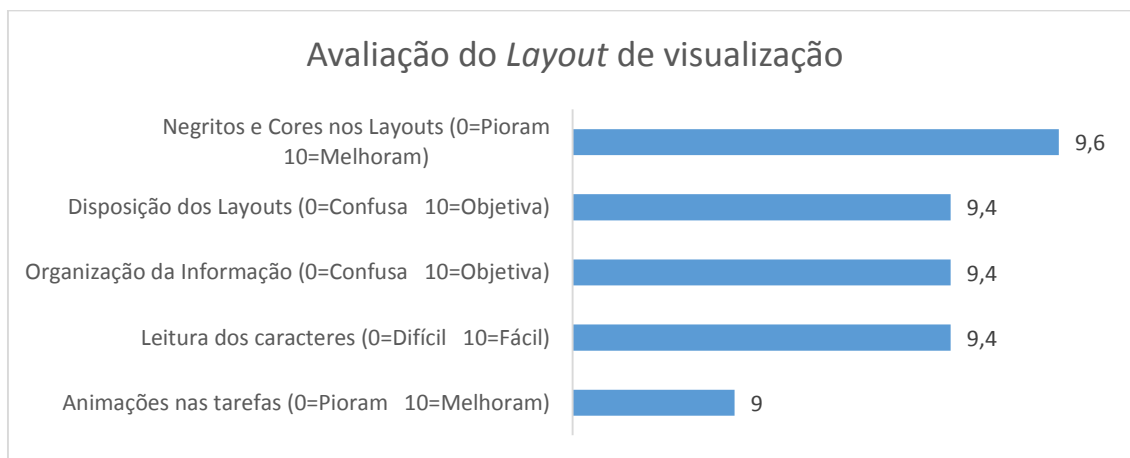


Gráfico 7 – Média das avaliações dadas ao *layout* de visualização da aplicação

Relativamente à terminologia e sistema de informação utilizados na aplicação tem-se as respostas resumidas no Gráfico 8. Deste modo, obtiveram-se classificações médias entre 9 e 9,4 (numa escala da 0 a 10). Estes valores indicam que os utilizadores entendem bem a linguagem utilizada, as mensagens de erro, a indicação do progresso e concordam com as terminologias utilizadas durante uma tarefa.

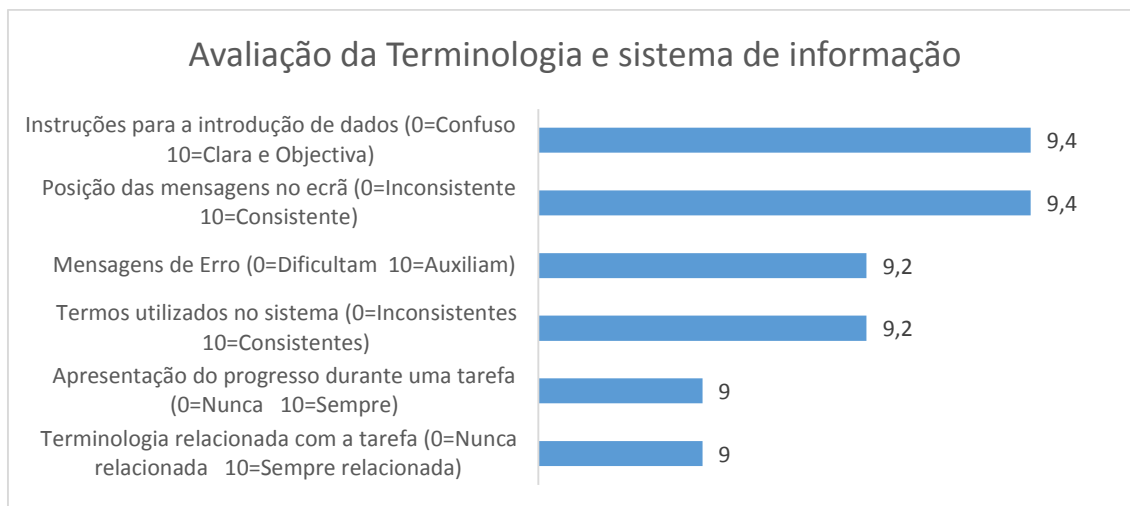


Gráfico 8 – Média das avaliações dadas à terminologia e sistema de informação utilizados na aplicação

O Gráfico 9 apresenta as classificações médias no que concerne à aprendizagem da utilização da aplicação. Aqui verifica-se que os utilizadores consideram que as mensagens de ajuda estão apropriadas (classificação 9) e que realmente ajudam na aprendizagem do uso da aplicação. A facilidade de aprendizagem da utilização da aplicação e a facilidade em explorar as funcionalidades da mesma também foram bem classificadas (classificação de 8,8). Note-se que o ponto menos bem classificado foi o da facilidade de lembrar os nomes e os comandos da aplicação (com uma classificação de 8,2). Em suma, através desta análise é possível inferir que os utilizadores facilmente aprendem a utilizar a aplicação.

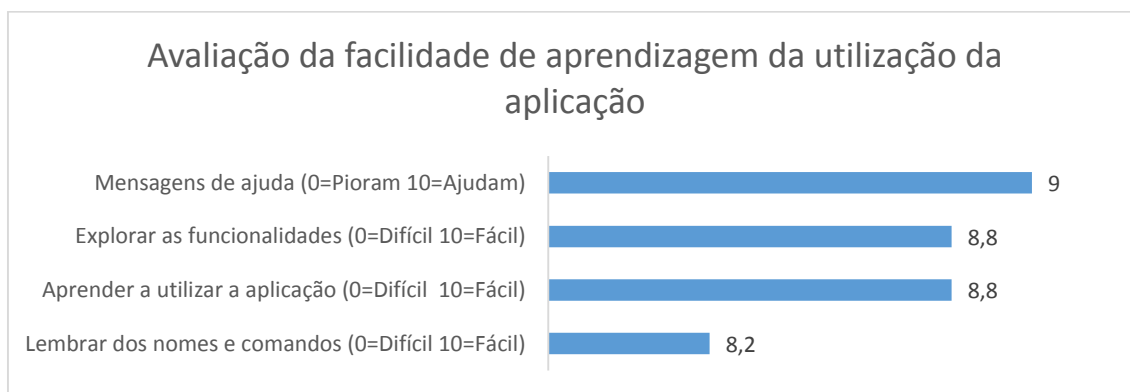


Gráfico 9 – Média das avaliações dadas relativamente à facilidade de aprendizagem de utilização da aplicação

Por fim, no que concerne à avaliação das capacidades do sistema, é possível verificar-se no Gráfico 10 que se obteve também boas classificações. “As formas análogas de executar

uma mesma ação” foi o pior classificado (7,4), mas os restantes tiveram uma avaliação superior a 8,7, o que se considera, de forma geral, como sendo um bom resultado.

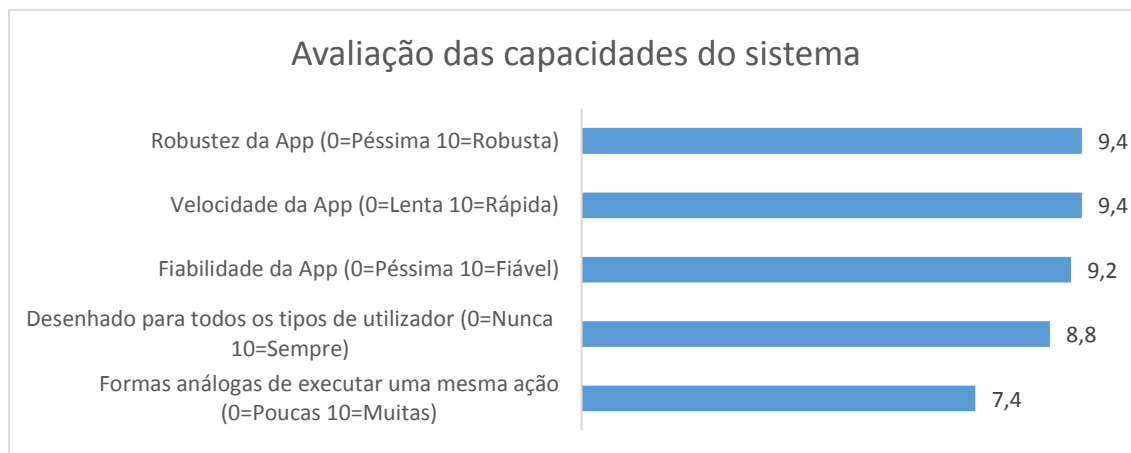


Gráfico 10 – Média das avaliações dadas às capacidades do sistema

## 5.6. Discussão

A utilização da aplicação e os testes de usabilidade realizados por parte dos utilizadores permitiram a recolha de dados importantes através do inquérito presente no Anexo G. Com os resultados obtidos nos inquéritos tornou-se possível tomar conhecimento de certos problemas na aplicação que antes não tinham sido detetados. Por outro lado, também se obteve uma avaliação geral sobre a aplicação em diversos aspetos, ajudando a definir os próximos passos a tomar para se proceder à realização de melhorias na aplicação. Através deste inquérito, os utilizadores verificaram se as funcionalidades implementadas eram realmente importantes e obtiveram-se ainda ideias para novas funcionalidades.

Na análise realizada às respostas obtidas sobre a utilidade das funcionalidades, percebe-se que os utilizadores consideraram quase todas importantes, pois no caso desta aplicação há várias funcionalidades que estão interligadas. Claro que existem algumas funcionalidades que eram dispensáveis, tais como: a apresentação da idade dos utilizadores, a hora e data do último acesso e também a data de registo na aplicação. Contudo, apesar de estas terem sido consideradas mais dispensáveis, não foram referidas pelos utilizadores como funcionalidades que devessem ser retiradas da aplicação.

No que diz respeito às funcionalidades que deveriam ser adicionadas, os utilizadores deram algumas sugestões interessantes, tais como o login através de uma conta de Facebook ou Google.

A reação geral à aplicação foi bastante positiva, da mesma forma que a avaliação do *layout* de visualização, a avaliação da terminologia e sistema de informação, a avaliação da facilidade de aprendizagem da utilização da aplicação e a avaliação das capacidades do sistema também foram muito positivas. Todavia, isto não significa que não haja melhorias a fazer, pois sabe-se que num *software* há sempre algo que pode ser melhorado, corrigindo erros e/ou adicionando funcionalidades que se mostrem realmente necessárias. Se estivermos atentos e verificarmos a evolução das aplicações, normalmente estas começam pelas funcionalidades básicas e nas versões seguintes são corrigidos problemas e adicionadas novas funcionalidades, ou seja, há uma constante melhoria do *software*.

### 5.7. Conclusão

Neste capítulo apresentaram-se todos os testes efetuados pelos utilizadores ao sistema, acompanhados dos resultados e da respetiva análise. Esta foi, sem margem para dúvidas, uma fase essencial para melhorar a qualidade geral do sistema. Saliente-se que, sem se efetuar testes junto dos potenciais utilizadores, é muito complicado melhorar a aplicação, tanto a nível de funcionalidades, como a nível de estabilidade e de usabilidade do sistema.

Após a obtenção dos resultados dos testes e das avaliações recebidas à aplicação, conseguiu-se verificar que a aplicação possuía apenas alguns pequenos erros, mas que não comprometiam o funcionamento do sistema. Contudo, os erros mais relevantes foram corrigidos de forma a melhorar o sistema.

Tendo em conta que esta é uma aplicação que necessita de uma ligação constante à Internet, por vezes, os utilizadores poderão verificar situações em que a informação não é recebida ou em que a aplicação poderá parecer um pouco lenta. É de salientar que o problema de aparente “lentidão” (tempo de processamento dos pedidos efetuados pelo utilizador) não é problema do sistema em si nem do servidor, mas sim da ligação à Internet. Contudo, este facto dependerá muito da qualidade da ligação à Internet que está a ser utilizada e pouco há a fazer em relação ao sistema em si. Note-se que o servidor apresentou boas velocidades de *Download* e *Upload* de ficheiros ao se utilizar o protocolo FTP e também um tempo de resposta baixo ao se realizar testes de *Ping* (latência), daí se afirmar que o problema será da velocidade da ligação de Internet.

Em suma, os resultados obtidos demonstraram que a aplicação está bem desenvolvida e pronta para ser utilizada por qualquer utilizador.

## 6. Conclusões

---

### 6.1. Resumo e Avaliação do Trabalho Efetuado

Com este projeto de mestrado desenvolveu-se uma aplicação que visa facilitar a troca de jogos e de livros entre as pessoas. A aplicação é constituída pela aplicação Android, pela Web API e pela base de dados Microsoft SQL. Note-se que estes dois últimos estão hospedados num servidor.

Este projeto começou a ser desenvolvido depois da realização de um inquérito sobre a ideia desta aplicação (o inquérito está presente no Anexo A). Através do referido inquérito obtiveram-se resultados que indicaram que o desenvolvimento de uma aplicação deste género poderia ter sucesso no que diz respeito à troca de itens entre os utilizadores e na preservação do meio ambiente. Neste sentido, considerou-se ser útil realizar primeiramente um estudo acerca de trabalhos relacionados para verificar o que já existia no mercado, evitando assim estar a criar algo que já existia. Desta forma, constatou-se que, para Android, no momento da pesquisa não se encontraram aplicações deste tipo, pelo menos associadas aos termos pesquisados. Também se realizou um estudo das tecnologias existentes e se selecionou as tecnologias que mais se adequavam a este projeto. Iniciou-se, assim, um *brainstorming* com algumas pessoas que demonstraram interesse nesta ideia e, a partir do *brainstorming*, selecionaram-se as ideias mais relevantes. Seguidamente, foram feitos o levantamento de requisitos e o diagrama de casos de utilização. Também foram realizados diagramas de atividade para esquematizar o funcionamento da aplicação para os casos de uso apresentados.

Após os passos anteriores, ou seja, depois de definidas todas as funcionalidades que a aplicação deveria implementar na sua primeira versão, passou-se à construção do modelo relacional da base de dados e ao planeamento da interface gráfica do utilizador. Para tal, criaram-se cenários de utilização (Ver Anexo E) e protótipos de baixo nível da interface do utilizador e solicitou-se a algumas pessoas que os testassem. Durante os testes, tomaram-se algumas notas, realizaram-se melhorias aos protótipos e, de forma iterativa, foi-se melhorando a interface de utilizador. Após obter um resultado satisfatório, passou-se à fase de implementação do sistema. Importa referir que durante a implementação e de modo informal, foram feitos testes de usabilidade para detetar erros. No final, a aplicação foi testada de forma mais exaustiva para verificar problemas que não tinham sido detetados antes e ainda recolher informações sobre o trabalho realizado.



Note-se que cada funcionalidade implementada exigiu que fossem implementadas as *Stored Procedures* na base de dados, a implementação da funcionalidade na API e, só depois da API estar testada e a realizar as ações pretendidas, é que a funcionalidade era implementada na aplicação Android. Depois, a funcionalidade era testada pelo programador e por outras pessoas de maneira a verificar se a funcionalidade estava bem implementada (sem erros) e se a interface estava intuitiva.

Após todo este processo de desenvolvimento e de melhoria da aplicação, é possível afirmar que a aplicação se encontra totalmente funcional e possui todas as funcionalidades necessárias à realização de propostas, negociações, trocas e outros detalhes mencionados ao longo desta dissertação. Contudo, será alvo de melhorias futuras, tal como qualquer aplicação nos dias de hoje. A aplicação será lançada no mercado Google Play num futuro próximo e as funcionalidades pensadas para a primeira versão foram todas implementadas.

Programar para o OS Android poderá não ser das tarefas mais fáceis. Todavia, com o passar do tempo e com o aprofundar dos conhecimentos, a programação torna-se cada vez mais natural e intuitiva. Apesar das diferenças nas dimensões de ecrã dos dispositivos Android, a organização nativa de um projeto Android facilita bastante a personalização dos *layouts* para dispositivos de várias dimensões e também, em relação a outros aspetos, como por exemplo os vários idiomas, pois o projeto divide os elementos de forma prática e clara para o programador.

O desenvolvimento de uma aplicação nativa possui muitas vantagens, pois o funcionamento é muito mais fluido e natural do que por exemplo uma aplicação *Web*, sendo vista num dispositivo Android. Isto, porque, a aplicação irá apresentar os dados ao utilizador organizados com base no espaço disponível no ecrã do dispositivo, ajustando-se convenientemente. Além deste aspeto, ao contrário de uma página *Web*, a aplicação nativa não necessita carregar a interface do utilizador da Internet, pois esta está armazenada localmente no dispositivo. Uma aplicação, que necessite de uma comunicação frequente com um servidor, apenas necessitará de obter uma quantidade de dados limitada, tornando-se assim, mais eficiente.

Devido à sua predominância no mercado de dispositivos móveis, a aplicação deste projeto está otimizada para o OS Android. O facto de funcionar utilizando uma API poderá se traduzir numa grande vantagem, no que diz respeito à implementação da mesma

aplicação, por exemplo para iOS ou então para a *Web*, pois a API possui grande parte das funcionalidades.

Importa referir que todas as sugestões que possam ser dadas relativamente a melhorias ou novas funcionalidades são bem-vindas, isto porque a aplicação visa ir ao encontro dos interesses e das necessidades dos seus utilizadores e não o contrário.

## 6.2. Trabalho Futuro

Esta é a primeira versão da aplicação e, como é óbvio, existem muitas funcionalidades e melhorias que podem ser implementadas.

Note-se que, apesar de terem sido pensadas muitas outras ideias interessantes para a aplicação, preferiu-se deixá-las para trabalho futuro, isto porque se chegou a uma parte da implementação que se ambicionava desenvolver mais algumas funcionalidades, pois a dada altura já havia um melhor domínio, tanto das ferramentas de desenvolvimento, como dos detalhes da própria programação, o que facilitava o desenvolvimento de outras funcionalidades. Contudo, o tempo é limitado e, como se sabe, nenhuma aplicação que sai no mercado é uma versão final, isto porque nos dias de hoje praticamente todas as aplicações costumam sofrer melhorias ao longo do tempo e é por isso que recebemos sempre atualizações para quase todo o *software* que utilizamos.

De acordo com as sugestões recebidas por vários utilizadores, algumas das funcionalidades e melhorias futuras mais importantes da aplicação serão:

- Melhoria do sistema de adição de novos itens à aplicação por parte do administrador – À data da redação desta dissertação já se iniciou a implementação desta funcionalidade, sob a forma de uma aplicação Windows Forms, utilizando o IDE Visual Studio 2013 for Desktop.
- Permitir o registo e o *login* com conta Google ou Facebook;
- Melhorar a aparência da interface gráfica da aplicação, principalmente para dispositivos de dimensões superiores.
- Lançar a aplicação no mercado Google Play.
- Ao aceder ao perfil de um utilizador poder visualizar os seus itens e os que este deseja.

- Guardar localmente algumas das propostas e respectivas conversas mais recentes, de forma a poderem ser consultadas mesmo sem ligação à Internet.
- Utilização da API do Google Maps para apresentar as distâncias tendo em conta as estradas disponíveis (atualmente a distância está a ser calculada em linha reta).
- Calcular custos de envio dos itens por correio, com base na distância.
- Criação da versão *Web* da aplicação, já que a API permite uma construção mais rápida da mesma.

Estas são as melhorias mais evidentes; contudo, poderá acontecer que com o passar do tempo surjam outras necessidades e, conseqüentemente, seja fundamental implementar outras melhorias.

### 6.3. Considerações Finais

Este projeto teve como principal contribuição dotar os utilizadores de uma ferramenta que possibilita a troca dos seus jogos e de livros de forma simples e prática, priorizando os desejos dos utilizadores e otimizando as trocas por distância, isto é, apresentado os proprietários dos itens ordenados por menor distância em relação ao utilizador de forma a este facilmente escolhê-los. Nesta aplicação criou-se uma interface simples, intuitiva e minimalista, mas ainda assim contendo todas as funcionalidades e opções necessárias e úteis.

É de realçar que esta aplicação além de permitir a tocar de itens que já não são desejados por outros de maior interesse, possibilita também a não poluição do meio ambiente, a economia de recursos naturais e a poupança a nível monetário dos utilizadores, pois não têm de investir diretamente dinheiro para adquirirem um item que desejem.

A aplicação em breve será lançada no mercado Google Play e não terá qualquer custo para o utilizador.

A nível pessoal, este projeto foi um grande desafio por diversas razões que passarei a explicar. Primeiramente, o desenvolvimento de uma aplicação deste género requereu um grande nível de exigência que foi sempre crescendo com a sua implementação. Depois, foi necessário não só utilizar como também conjugar os diversos conhecimentos teóricos e práticos adquiridos ao longo de toda a formação académica (licenciatura e mestrado). Para além destes saberes, foi imperativo aprender muitos outros para conseguir

desenvolver da melhor forma o projeto que tinha em mente. Por fim, mas sem descurar, todo o processo de orientação constante que as minhas orientadoras me forneceram, fez-me refletir e avançar de forma a aplicação se tornar simples e funcional. Importa ainda mencionar que os utilizadores que testaram o projeto foram cruciais na resolução de lacunas e no aprimoramento da aplicação.

Gostava ainda de salientar, de uma forma mais específica, que através do desenvolvimento deste projeto aumentei consideravelmente os meus conhecimentos a nível da programação de aplicações Android, de *Web APIs* (utilizando a linguagem de programação C#), assim como a nível da programação de *Stored Procedures* (linguagem MS SQL) e de funções na própria base de dados. Para mim este projeto foi muito importante para consolidar os conhecimentos aprendidos ao longo do curso, pois abrangeu muitos dos conceitos abordados em diversas unidades curriculares tais como: Sistemas Gestores de Bases de Dados, Desenho e Implementação de *Software*, Segurança em Sistemas de Comunicação, Arquiteturas de *Software*, Engenharia de Requisitos, Interação Humano-Computador, entre outras.

## Referências

---

- [1] “Venda de Livros em Portugal Cai 4,6%,” 03 03 2014. [Online]. Available: <http://www.publico.pt/economia/noticia/venda-de-livros-em-portugal-cai-46-1626902>. [Acedido em 28 outubro 2014].
- [2] R. BORJA-SANTOS, “Portugal é o segundo país da UE onde mais gente acha que a situação está pior,” 12 maio 2014. [Online]. Available: <http://www.publico.pt/sociedade/noticia/portugal-e-o-segundo-pais-da-ue-onde-mais-gente-acha-que-a-situacao-esta-pior-1635583>. [Acedido em 02 novembro 2014].
- [3] “4RS,” [Online]. Available: <http://www.4rs.pt/index.aspx?p=ContactPage>. [Acedido em 02 novembro 2014].
- [4] “Agile Methodology,” [Online]. Available: <http://agilemethodology.org/>. [Acedido em 02 novembro 2014].
- [5] “Google Play,” Google, 2014. [Online]. Available: [https://play.google.com/store/apps?hl=pt\\_PT](https://play.google.com/store/apps?hl=pt_PT). [Acedido em 25 março 2014].
- [6] “Windows Phone Store,” Microsoft, 2014. [Online]. Available: <http://www.windowsphone.com/pt-pt/store>. [Acedido em 26 março 2014].
- [7] “iTunes,” Apple, 2014. [Online]. Available: <https://www.apple.com/pt/itunes/>. [Acedido em 25 março 2014].
- [8] “TrocaJogo,” [Online]. Available: <http://www.trocajogo.com.br/>. [Acedido em 2014 março 27].
- [9] “99 Gamers,” [Online]. Available: <http://99gamers.com/>. [Acedido em 2014 março 27].
- [10] “Trocas Online,” [Online]. Available: <http://www.trocasonline.com/>. [Acedido em 2014 março 28].

- [11] “Venda e Troca de Jogos Usados,” [Online]. Available: <https://www.facebook.com/pages/Venda-e-Troca-de-Jogos-Usados/312348755513457?fref=ts>. [Acedido em 30 março 2014].
- [12] “Banco do Livro Escolar - Troca gratuita de livros escolares,” [Online]. Available: <https://www.facebook.com/Banco.do.livro.escolar>. [Acedido em 30 março 2014].
- [13] “OLX,” [Online]. Available: <http://www.olx.pt/>. [Acedido em 31 março 2014].
- [14] “Coisas,” [Online]. Available: <http://www.coisas.com/>. [Acedido em 31 março 2014].
- [15] M. Weiser, “The computer for the 21st century,” 1988.
- [16] “Gartner Says Worldwide Traditional PC, Tablet, Ultramobile and Mobile Phone Shipments to Grow 4.2 Percent in 2014,” 7 7 2014. [Online]. Available: <http://www.gartner.com/newsroom/id/2791017>. [Acedido em 15 outubro 2014].
- [17] “Android Dashboards,” [Online]. Available: <http://developer.android.com/about/dashboards/index.html>. [Acedido em 17 outubro 2014].
- [18] “Android Developer,” [Online]. Available: <http://developer.android.com/guide/topics/manifest/uses-sdk-element.html>. [Acedido em 17 outubro 2014].
- [19] “Android Developers,” [Online]. Available: <http://developer.android.com/index.html>. [Acedido em 17 maio 2014].
- [20] “Building Your First App,” [Online]. Available: <http://developer.android.com/training/basics/firstapp/index.html>. [Acedido em 28 outubro 2014].
- [21] “Get the Android SDK,” [Online]. Available: <http://developer.android.com/sdk/index.html>. [Acedido em 18 outubro 2014].
- [22] “Introducing JSON,” [Online]. Available: <http://www.json.org/>. [Acedido em 28 outubro 2014].

- [23] “Introduction to Recommender Systems Handbook,” em *Recommender Systems Handbook*, Springer US, 2011, pp. 1-35.
- [24] B. Christiansson, M. Forss, I. Hagen, K. Hansson, J. Jonasson, M. Jonasson, F. Lott, S. Olsson e T. Rosevall, *GoF Design Patterns - with examples using Java and UML2*, Logica Java Architects Training Crew, 2008, p. 3; 24; 50.
- [25] E. Freeman, E. Freeman, B. Bates e K. Sierra, *Head First Design Patterns*, O'Reilly, 2004, pp. 51; 173; 243; 410-416.
- [26] “Activity,” [Online]. Available:  
<http://developer.android.com/reference/android/app/Activity.html>. [Acedido em 01 novembro 2014].
- [27] “Requisitos funcionais e não-funcionais, as duas faces da moeda aplicáveis à engenharia de software,” [Online]. Available:  
<http://www.batebyte.pr.gov.br/modules/conteudo/conteudo.php?conteudo=1718>. [Acedido em 24 outubro 2014].
- [28] “10 Usability Heuristics for User Interface Design,” [Online]. Available:  
<http://www.designprinciplesftw.com/collections/10-usability-heuristics-for-user-interface-design>. [Acedido em 26 outubro 2014].
- [29] “Android Design Principles,” [Online]. Available:  
<http://developer.android.com/design/get-started/principles.html>. [Acedido em 26 outubro 2014].
- [30] “Balsamiq Mockups,” [Online]. Available:  
<http://balsamiq.com/products/mockups/>. [Acedido em 26 outubro 2014].
- [31] “How Many Test Users in a Usability Study?,” [Online]. Available:  
<http://www.nngroup.com/articles/how-many-test-users/>. [Acedido em 27 outubro 2014].
- [32] R. Hanmer, *Pattern-Oriented Software Architecture For Dummies*, John Wiley & Sons, Ltd, 2013.

- [33] “REST,” [Online]. Available: <http://www.caelum.com.br/apostila-vraptor-hibernate/rest/#11-6-refatorando-o-produtocontroller>. [Acedido em 28 outubro 2014].
- [34] “Localizing with Resources,” [Online]. Available: <http://developer.android.com/guide/topics/resources/localization.html>. [Acedido em 1 outubro 2014].
- [35] “Action Bar,” [Online]. Available: <http://developer.android.com/guide/topics/ui/actionbar.html>. [Acedido em 1 outubro 2014].
- [36] “Support Library Features,” [Online]. Available: <https://developer.android.com/tools/support-library/features.html>. [Acedido em 1 outubro 2014].
- [37] “Problemas de início de sessão e palavra-passe,” [Online]. Available: <http://windows.microsoft.com/pt-pt/windows/outlook/forgotten-password>. [Acedido em 5 outubro 2014].
- [38] “A Google Geocoding API,” [Online]. Available: <https://developers.google.com/maps/documentation/geocoding/>. [Acedido em 7 outubro 2014].
- [39] R. W. Sinnott, “Virtues of the Haversine,” em *Sky and Telescope*, vol. 68 (no. 2), 1984, p. 159.
- [40] “The Importance of Customer Feedback,” [Online]. Available: <http://smallbusiness.chron.com/importance-customer-feedback-2089.html>. [Acedido em 10 outubro 2014].
- [41] “Iconography,” [Online]. Available: <http://developer.android.com/design/style/iconography.html>. [Acedido em 1 outubro 2014].



- [42] G. Perlman, "Questionnaire for User Interface Satisfaction," [Online]. Available: <http://hcibib.org/perlman/question.cgi?form=QUIS>. [Acedido em 5 novembro 2014].



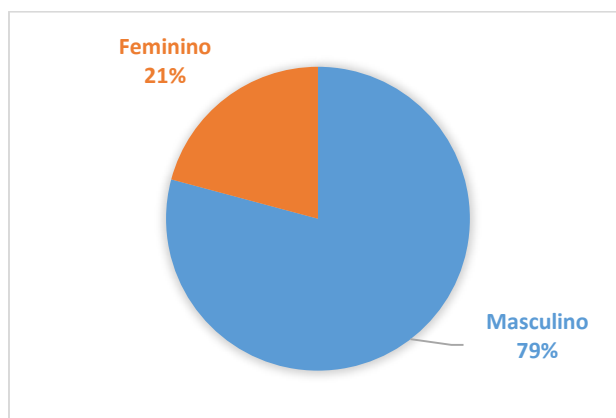
# **Anexos**

## Anexo A – Inquérito inicial para estudo de viabilidade do projeto (com tratamento de dados)

---

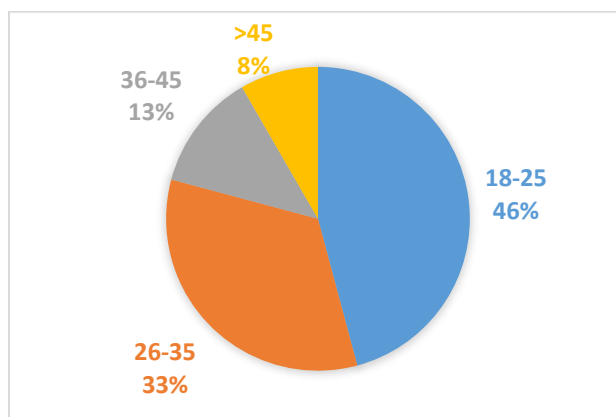
Este inquérito foi divulgado através de *e-mail* e foi respondido por 24 pessoas e tinha como intuito estudar a viabilidade deste projeto, numa fase inicial. Em seguida são apresentadas as perguntas do inquérito seguidas do resultado das respostas. Note-se que o resultado foi apresentado através de gráficos de forma a facilitar a sua análise. Após cada gráfico será realizada a conclusão obtida.

**P1.** Qual o seu género?

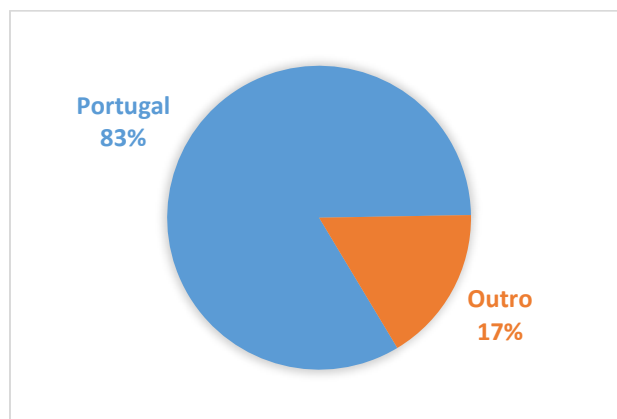


Esta questão serviu para se verificar que os inquiridos eram maioritariamente do sexo masculino (79%). Desta forma sabemos que as respostas irão refletir mais a opinião dos homens do que das mulheres (21%).

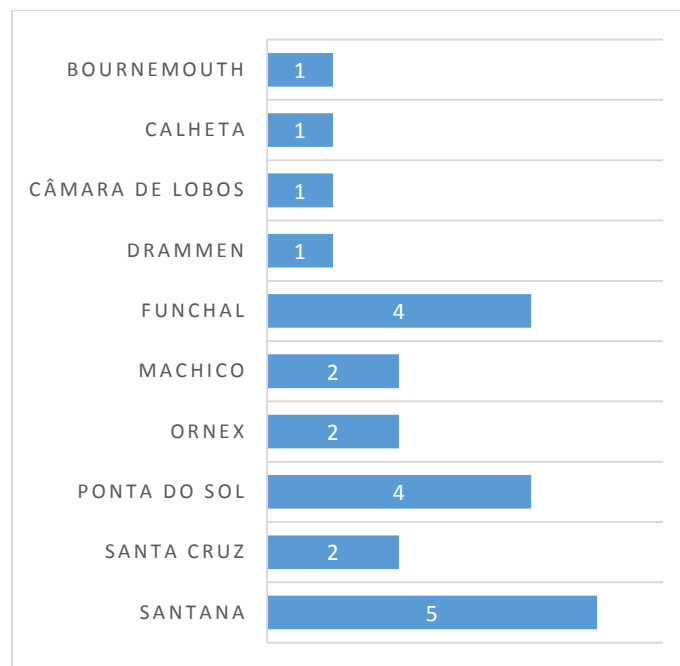
**P2.** Indique a sua faixa etária.



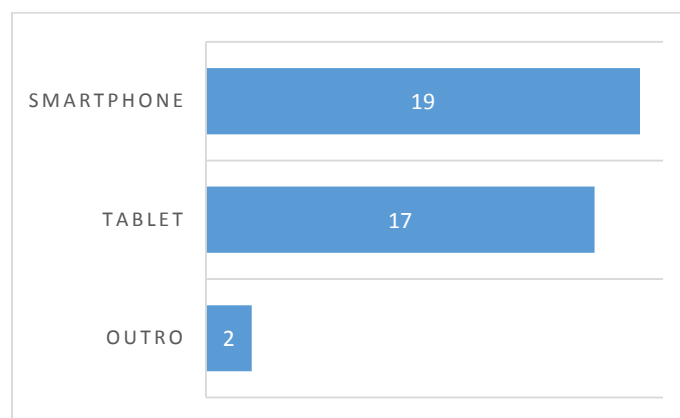
Relativamente à faixa etária dos inquiridos visualiza-se, no gráfico acima, que 79% têm idade compreendida entre os 18 e os 35 anos, os restantes (21%) têm idade superior a 35 anos. Deste modo, sabemos que a opinião predominante será de jovens-adultos.

**P3. Em que país vive?**

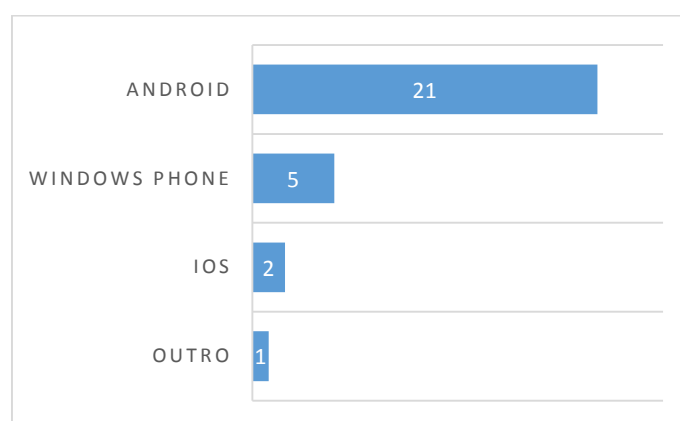
Nesta questão concluímos que a maior parte dos inquiridos vive em Portugal (83%).

**P4. Em que cidade vive?**

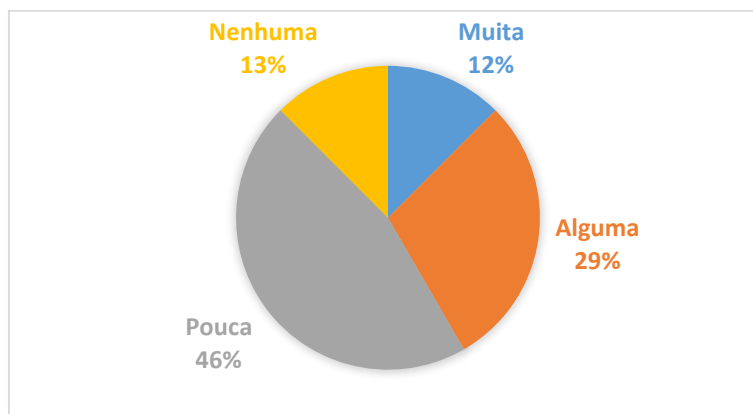
Neste gráfico é possível visualizar em que cidade vivem os inquiridos. Da sua análise conclui-se que grande parte reside na Ilha da Madeira (Portugal), mas também há inquiridos de outros locais, tais como Bournemouth (Inglaterra), Ornex (França) e Drammen (Noruega).

**P5. Atualmente que tipo de equipamentos móveis utiliza?**

Nesta questão apura-se que os utilizadores possuem maioritariamente dois tipos de dispositivos móveis: os *smartphones* (79% - 19 inquiridos) e os *tablets* (71% - 17 inquiridos). Desta forma verificamos que maioria possui os dois tipos de dispositivos.

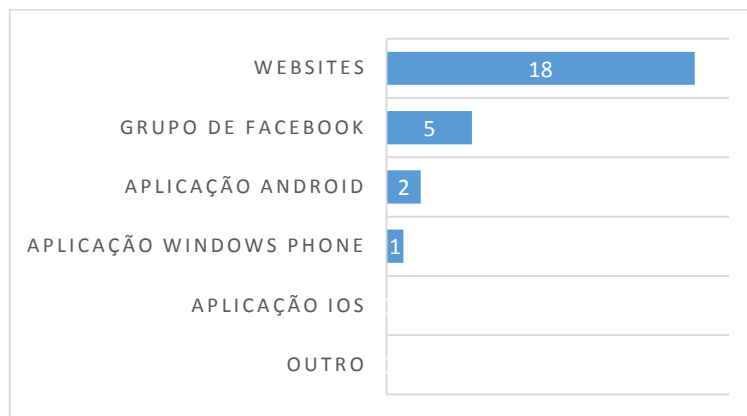
**P6. Qual o sistema operativo que os seus dispositivos possuem?**

Neste ponto constata-se que grande parte dos dispositivos dos utilizadores inquiridos utiliza o sistema operativo Android (72%).

**P7. Qual o seu nível de experiência com trocas e vendas de objetos Online?**

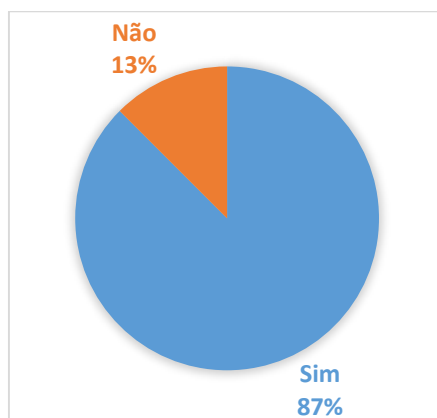
Nesta questão verifica-se que mais de metade (59%) dos inquiridos possui pouca (46%) ou nenhuma (13%) experiência em trocas e vendas de objetos *online*. Contudo ainda há uma parte significativa (41%) de inquiridos que possuem alguma (29%) ou muita experiência (12%).

**P8.** Essa experiência relativa à trocas e/ou vendas online foi adquirida através de:



Nesta questão verifica-se que a maioria da experiência mencionada pelos inquiridos na questão P7 foi adquirida através de *websites* incluindo a rede social Facebook (também *website*). Sendo esta uma questão em que os inquiridos podiam seleccionar várias opções pode-se verificar que alguns deles obtiveram experiência em *websites*, mas também noutros tipos de aplicações.

**P9.** Acha útil a ideia de criar uma aplicação móvel (para *Smartphones*, *Tablets*, etc) que permita aos utilizadores realizarem trocas e/ou vendas de objetos?



Esta foi uma das questões em que as respostas motivaram bastante o desenvolvimento do projeto, isto porque 21 dos inquiridos (87%) acharam útil a criação de uma aplicação de trocas e/ou vendas de objetos para *smartphones* e *tablets*.

**P10.** Se respondeu "Não" à questão anterior, justifique a sua resposta. (3 Respostas)

Resposta 1	<i>“Não sinto tanta confiança em colocar um número de cartão de crédito numa aplicação face a coloca-lo num website devidamente certificado.”</i>
Resposta 2	<i>“Dois dos maiores websites de troca e venda de objetos em Portugal têm versão móvel, e aplicação móvel pelo menos para Android. Falo do OLX e Coisas. Julgo que a criação de uma nova aplicação, com ou sem website, não teria adesão suficiente para justificar o investimento, pois parece-me que as necessidades dos utilizadores (neste caso compradores e vendedores) já estão satisfeitas com estes dois websites.”</i>
Resposta 3	<i>“Porque através de um portal web temos mais comodidade para realizar a troca e também a possibilidade de analisar em paralelo o produto.”</i>

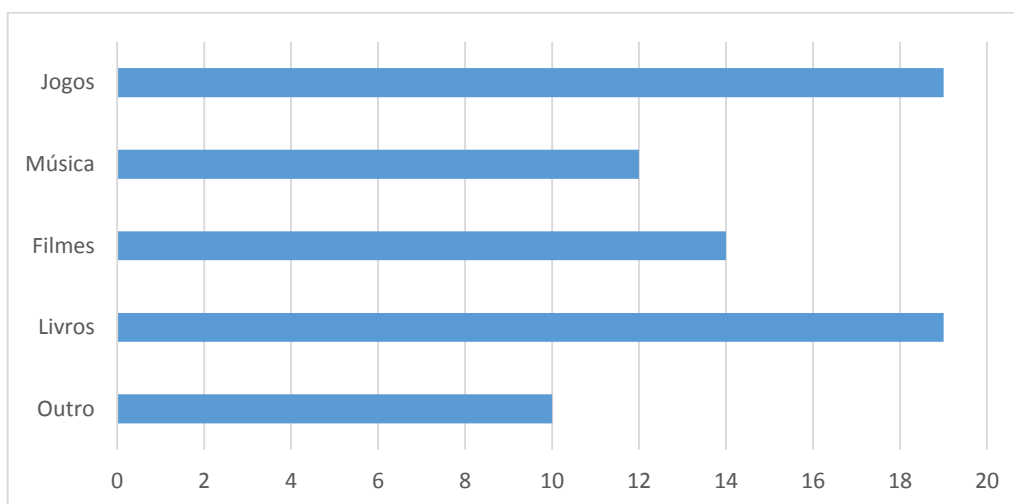
Através da tabela apresentada conseguimos visualizar a justificação dos inquiridos que responderam “não” à questão P9. No entanto apura-se que a resposta 2 é a mais objetiva.

**P11.** A aplicação deveria permitir:

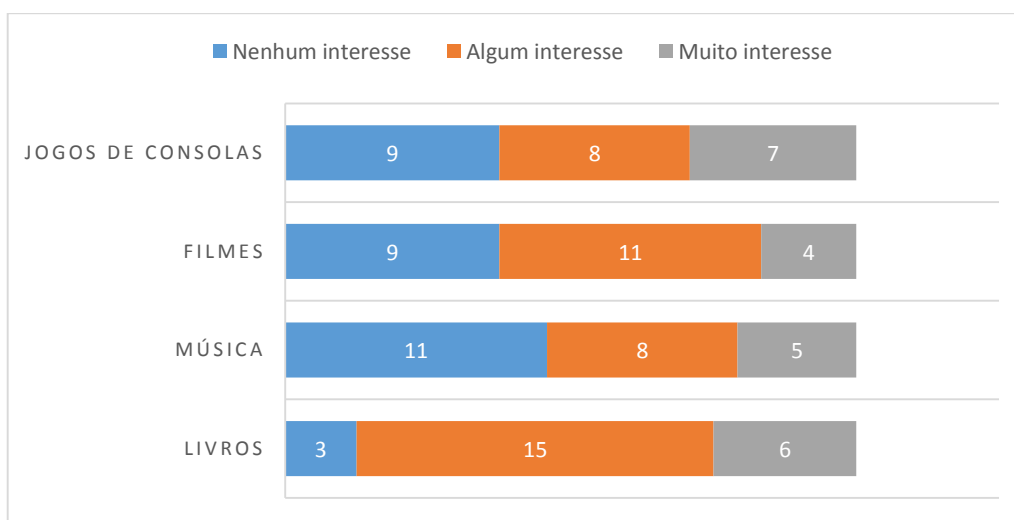


Com esta questão concluímos que a esmagadora maioria dos inquiridos prefere uma aplicação que permita os dois tipos de transações, ou seja, trocas e vendas.

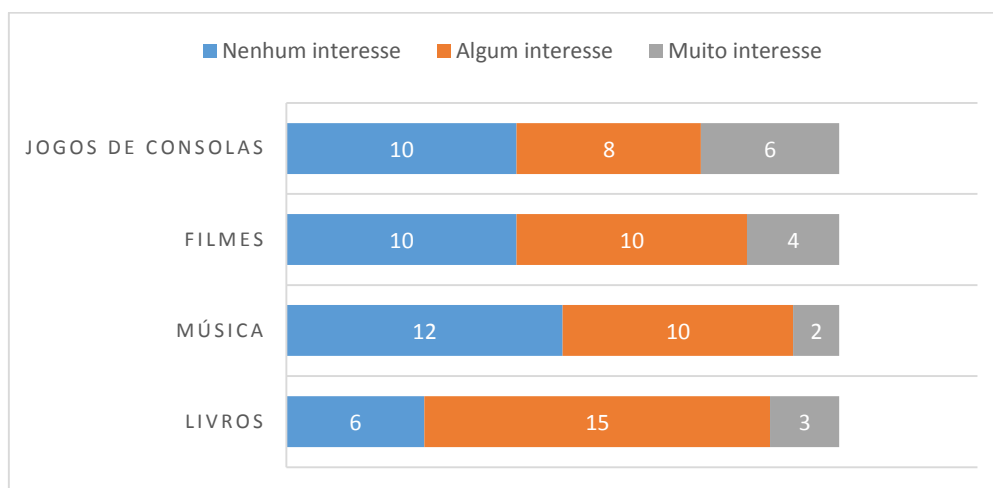


**P12.** Quais as principais categorias de objetos que a aplicação se deveria focalizar?

Os inquiridos, nesta questão, indicaram que as categorias de maior interesse são os jogos e os livros.

**P13.** Nível de Interesse relativo à TROCA de jogos, filmes, música e livros.

Nesta questão os inquiridos indicaram o nível de interesse relativo às trocas dos artigos mencionados e o resultado está patente no gráfico acima. Como se pode ver, a troca de livros foi a que mais interesse teve. Em segundo lugar ficou a troca de jogos com 7 inquiridos muito interessados e 8 com algum interesse, em terceiro lugar ficou os filmes com apenas 4 inquiridos com muito interesse e 11 com algum interesse. Em último tem-se a troca de música.

**P14.** Nível de interesse relativo à VENDA de jogos, filmes, música e livros.

Nesta questão os inquiridos indicaram o nível de interesse relativo às vendas dos artigos mencionados e o resultado é apresentado no gráfico acima. Após a análise conclui-se que a ordem de preferência segue a mesma que para a questão P12, ou seja, os utilizadores acham mais útil a venda de livros, depois de jogos e no final os filmes e a música.

## Anexo B – Diagramas de Casos de Utilização (Versão Completa)

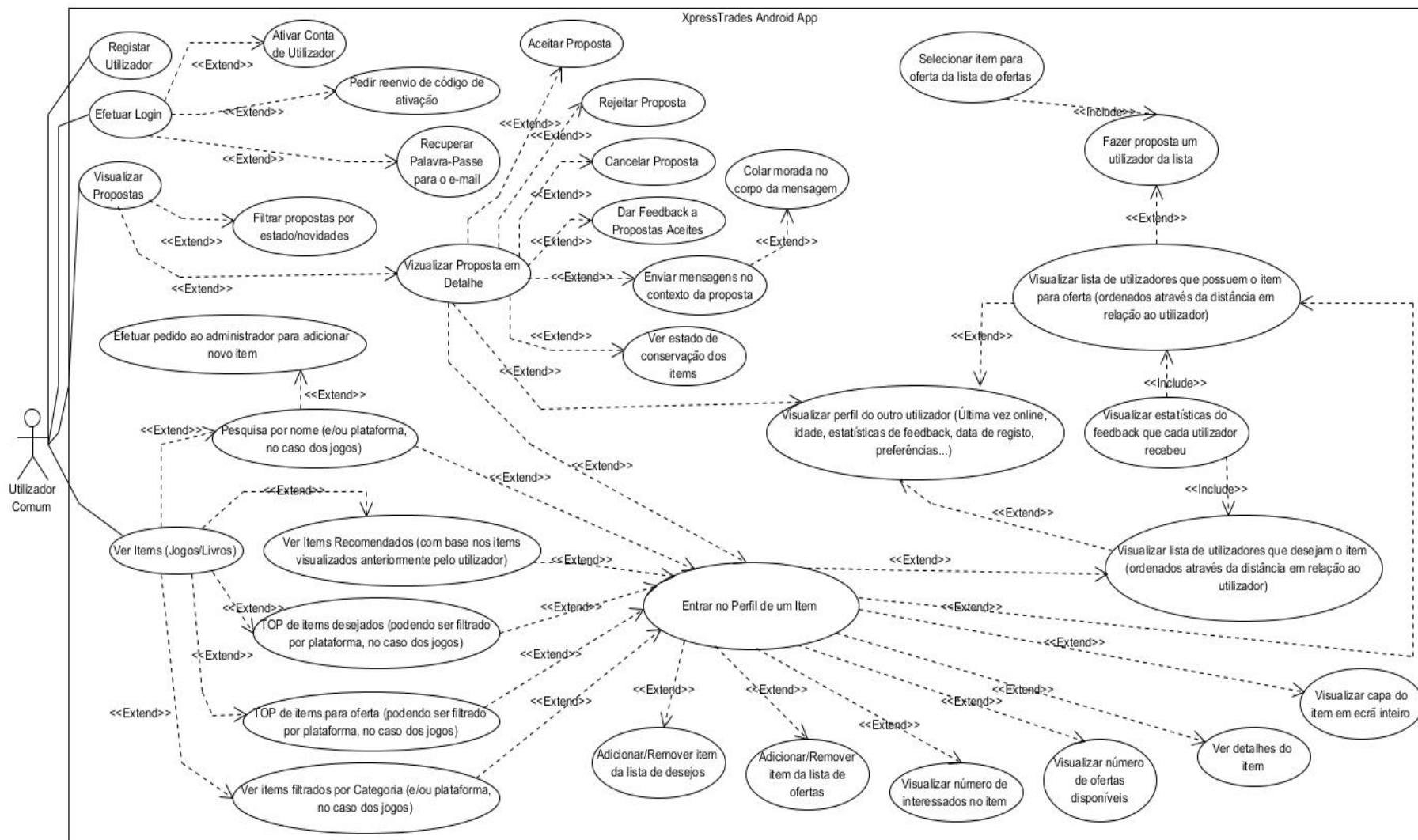


Figura 42 - Diagrama de Casos de Uso (Parte 1)

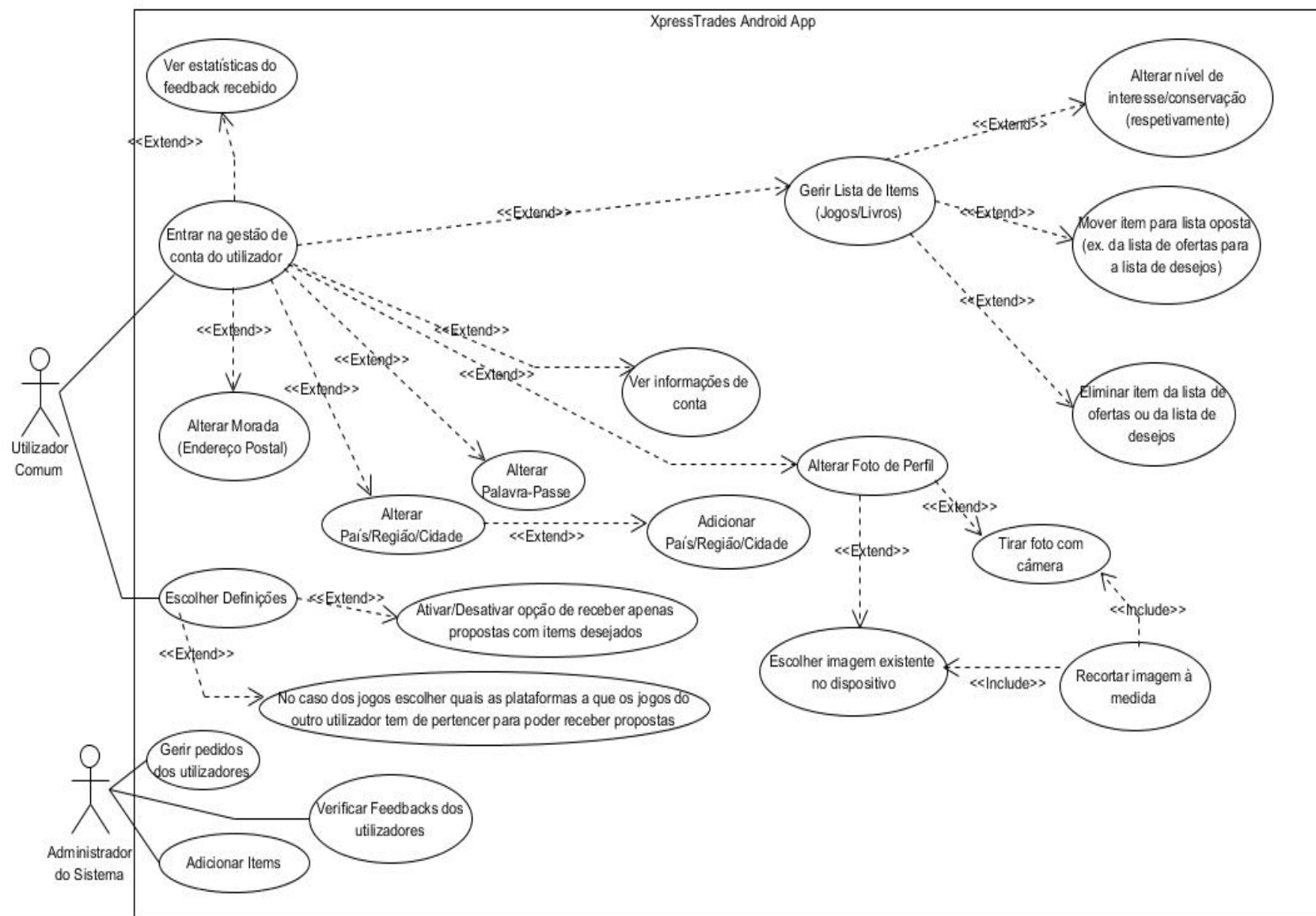


Figura 43 - Diagrama de Casos de Uso (Parte 2)

## Anexo C – Diagramas de Atividade dos Casos de Utilização

Neste anexo, são apresentados os restantes diagramas de atividade, que foram desenhados para os vários casos de utilização.

O caso **ativar opção “Só receber propostas com jogos das plataformas selecionadas”** é muito simples, ou seja, o utilizador deve seleccionar a opção “Definições” a partir do Menu Principal. Neste momento, o sistema apresenta a atividade “Definições” e o utilizador deverá seleccionar as plataformas de jogo às quais os jogos que lhe são oferecidos em propostas têm de pertencer. O utilizador nunca receberá propostas com jogos que não pertençam às plataformas selecionadas. Ao sair da atividade “Definições”, o sistema verifica se o utilizador seleccionou pelo menos uma plataforma. Se o utilizador não seleccionou nenhuma, o sistema apresenta uma mensagem de erro a indicar que necessita de seleccionar pelo menos uma. Caso contrário as suas preferências serão automaticamente guardadas pelo sistema. O diagrama correspondente está na Figura 44.

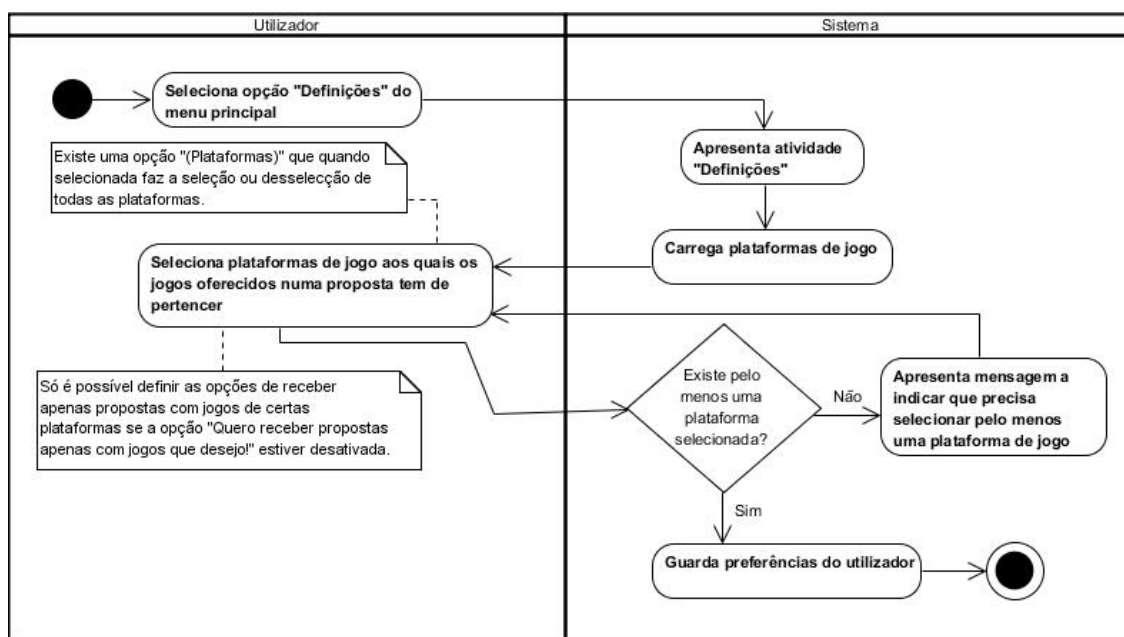


Figura 44 - Diagrama de atividade para o caso "Ativar Opção: Só Receber Propostas com Jogos das Plataformas Selecionadas"

O caso de utilização **ativar opção “Quero receber propostas apenas com livros que desejo”** é análoga à anterior, apenas em vez de o utilizador seleccionar as plataformas tem de seleccionar a opção descrita. O diagrama deste caso está na Figura 45.

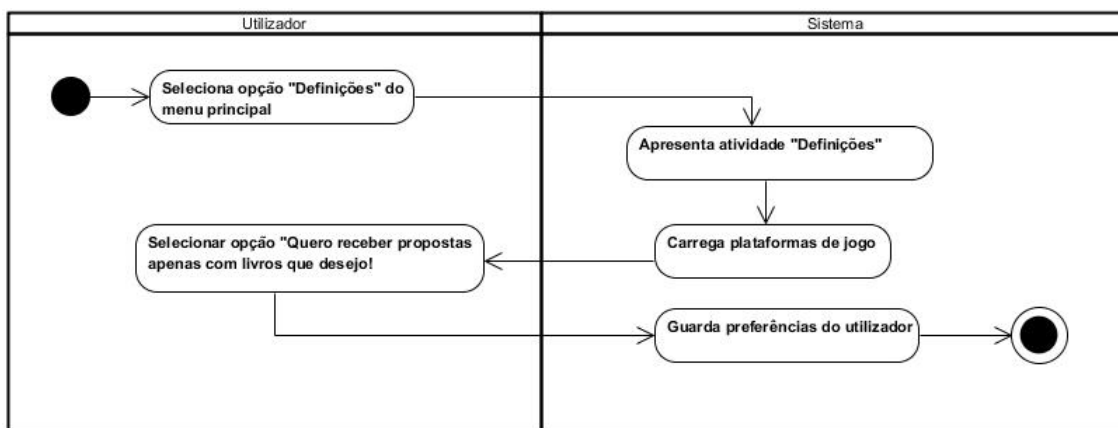


Figura 45 - Diagrama de atividade para o caso "Ativar Opção: Quero Receber Propostas Apenas com Livros que Desejo"

O caso de utilização **ativar opção “Quero receber propostas apenas com jogos que desejo”** é semelhante à anterior, apenas em vez de o utilizador selecionar esta opção para os livros terá de a selecionar para a opção jogos. O diagrama de atividade deste caso está na Figura 46.

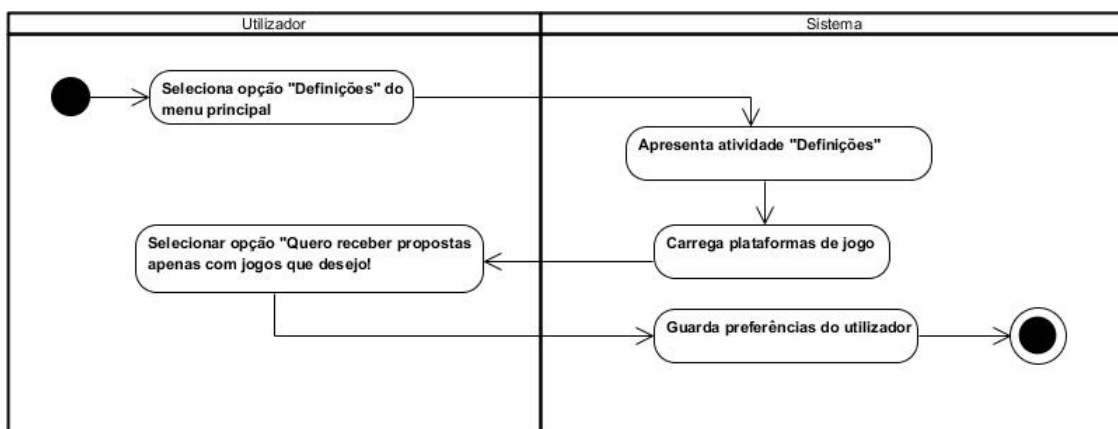


Figura 46 - Diagrama de atividade para o caso "Ativar Opção: Quero Receber Propostas Apenas com Jogos que Desejo"

Para o caso **alterar a foto de perfil** o utilizador deve escolher a opção “A Minha Conta” presente na atividade “Menu Principal”. O sistema apresenta a atividade A Minha Conta. O utilizador deve selecionar a sua fotografia de perfil atual. O sistema apresenta as opções “Tirar Foto Agora” e “Escolher Foto Existente” e o utilizador tem de escolher uma das opções. Para a primeira opção o sistema abre a aplicação da câmara para obter a foto. Já para a segunda opção, o sistema pergunta com que programa o utilizador pretende escolher a foto existente no dispositivo. Após o utilizador selecionar uma foto, o sistema

abre a aplicação de recorte de fotos. O utilizador deve ajustar o tamanho do recorte e a posição do mesmo na foto e validar a ação. O sistema grava a nova imagem e substitui a foto antiga presente na atividade. O diagrama que ilustra este caso de uso está na Figura 47.

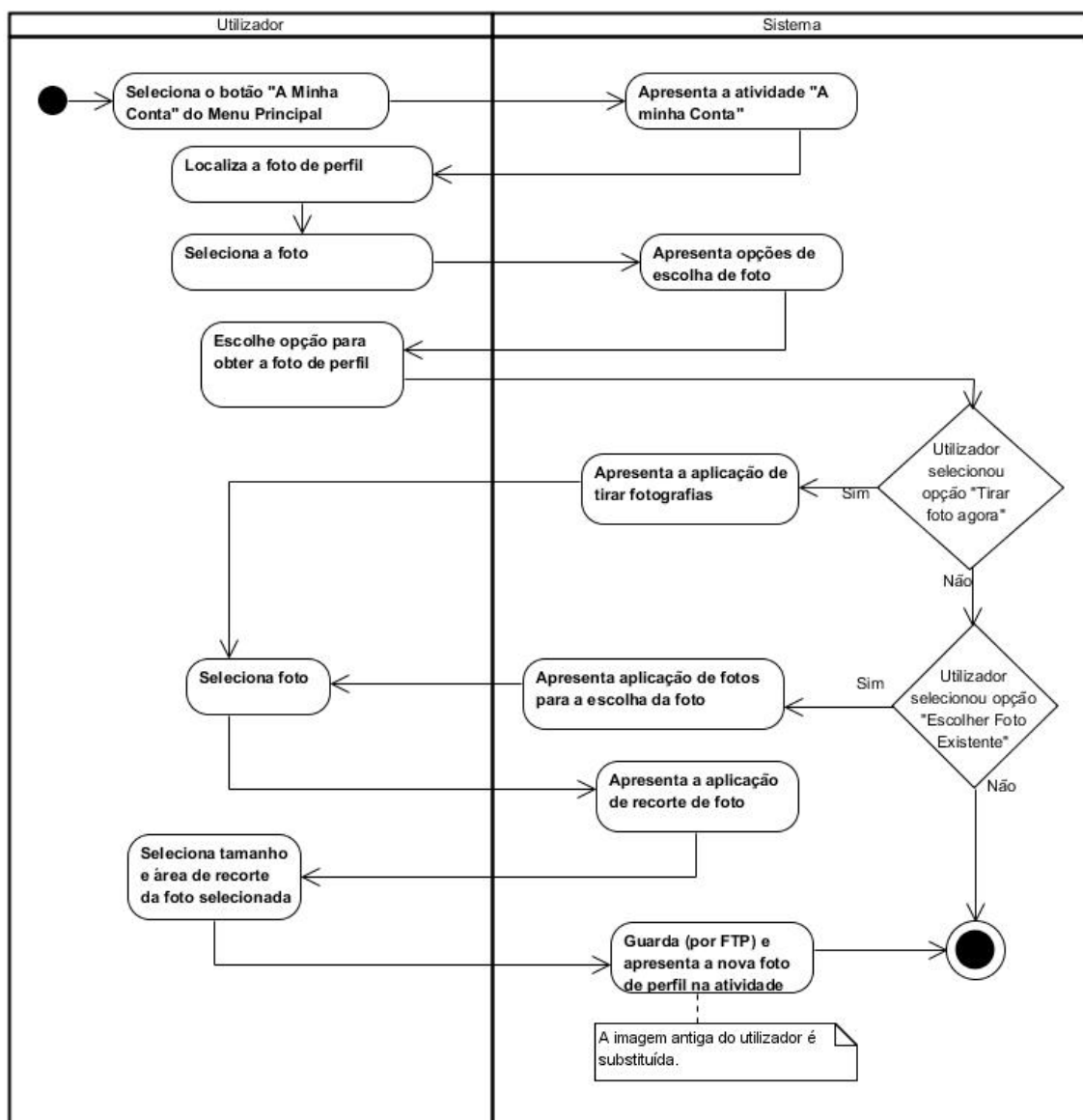


Figura 47 - Diagrama de atividade para o caso "Alterar a Foto de Perfil"

Para o caso de uso **alterar palavra-passe** o utilizador tem de escolher a opção "A Minha Conta" a partir da atividade "Menu Principal". O sistema irá apresentar a atividade "A Minha Conta". O utilizador deve localizar os formulários de alteração de palavra-passe, começando pelo formulário "Palavra-passe atual". O sistema irá apresentar o teclado. O utilizador deve escrever a sua palavra-passe atual e selecionar o formulário seguinte. O

sistema passará para o formulário que corresponde à nova palavra-passe. O utilizador deve escrever a sua nova palavra-passe e selecionar o formulário seguinte. O sistema seleciona o formulário que corresponde à reescrita da nova palavra-passe. O utilizador deverá reescrever a nova palavra-passe e selecionar a opção “Guardar Alterações”. O sistema verifica se a palavra-passe atual é a correta, se a nova palavra-passe tem pelo menos oito caracteres e se a palavra-passe nova corresponde nos dois formulários destinados a esta. Caso as três verificações anteriores sejam verdadeiras o sistema altera a palavra-passe do utilizador e notifica-o do sucesso da alteração. Este caso de uso está descrito no diagrama presente na Figura 48.

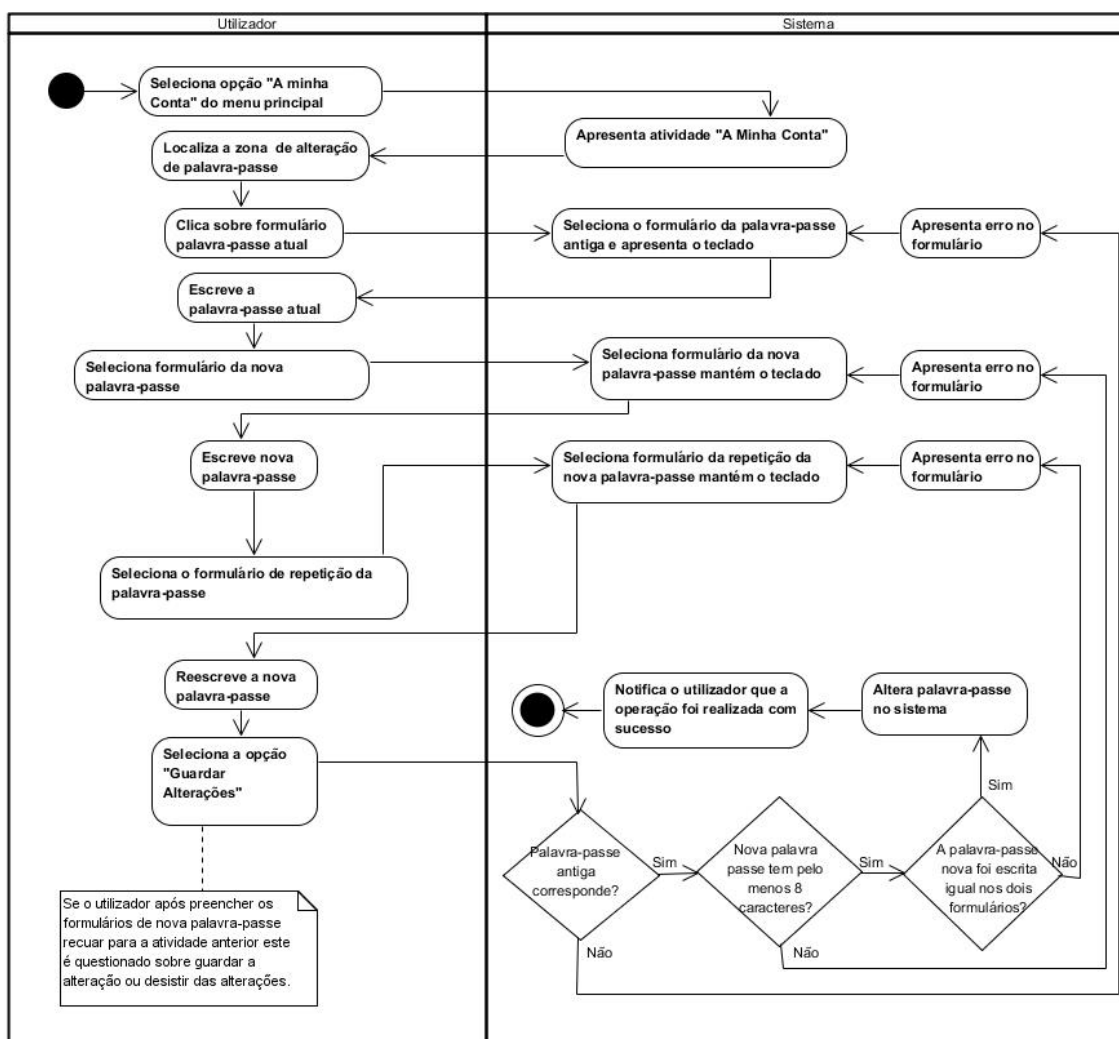


Figura 48 - Diagrama de atividade para o caso "Alterar a Palavra-Passe"

Para o caso **alterar morada** (endereço postal) o utilizador deve aceder à atividade “A Minha Conta”, tal como nos casos anteriores. Procurar o formulário que possui o nome da rua e selecionar a opção “Alterar”. Nesse momento a aplicação apresenta a morada



(caso alguma tenha sido introduzida). O utilizador deverá proceder à alteração pretendida e submeter ao sistema. O sistema atualizará a morada na atividade, ficando visível ao utilizador. Após este passo o utilizador deverá escolher a opção “Guardar Alterações”. O sistema procederá a atualização da morada do utilizador e apresentará uma mensagem de sucesso. Nenhuma verificação é efetuada à morada introduzida, pois não é um campo obrigatório. O diagrama de atividade deste caso está representado na Figura 49.

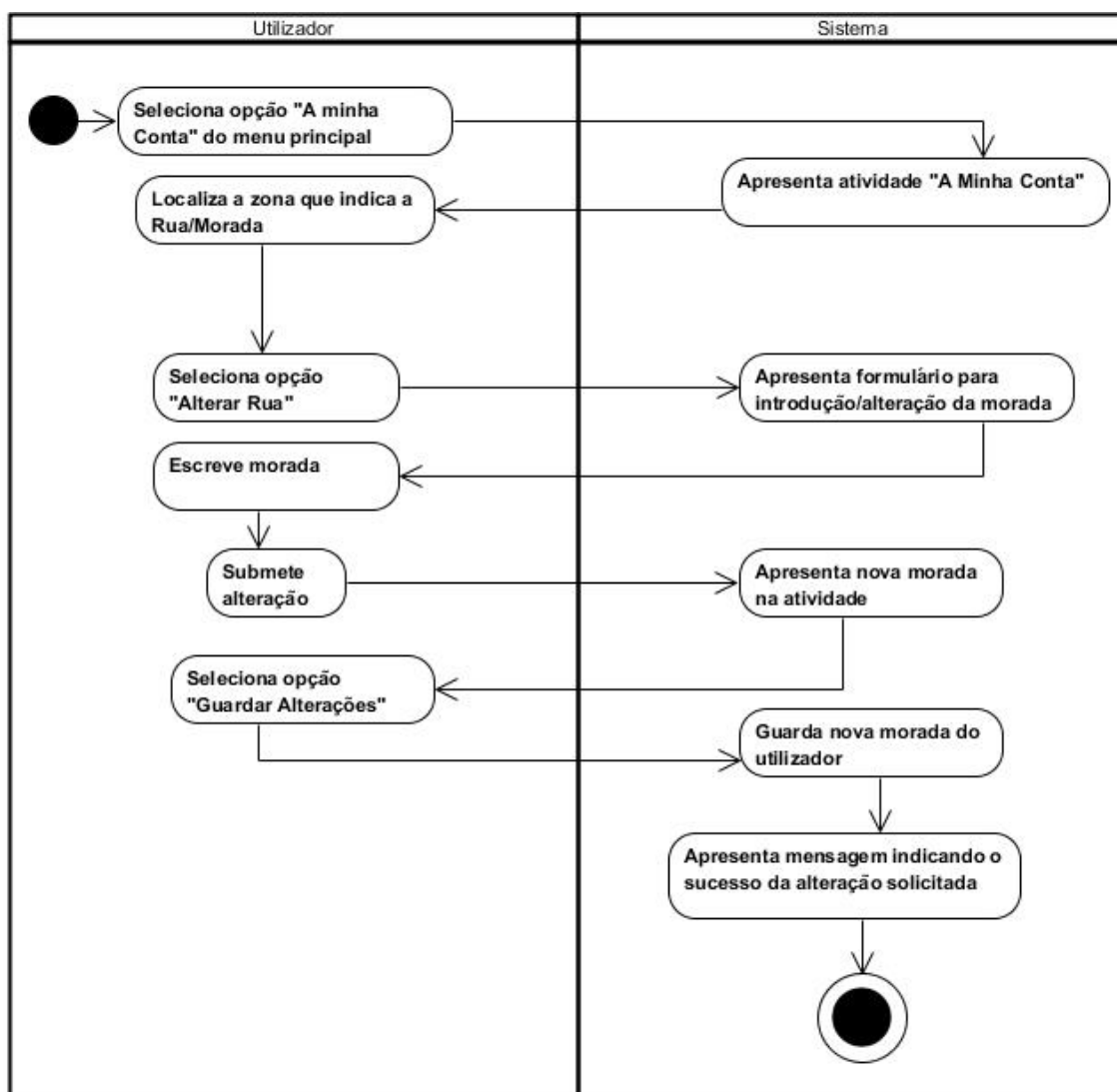


Figura 49 - Diagrama de atividade para o caso "Alterar morada"

No caso **alterar cidade**, o utilizador deve aceder à atividade “A Minha Conta”. No momento em que atividade “A Minha Conta” é apresentada, o sistema também carrega a lista de Países. O utilizador deverá procurar pela zona onde são apresentados o País, Região e Cidade. Neste caso o utilizador se pretender uma cidade de um outro país, terá de começar por selecionar o País. Nesse momento, o sistema apresentará a lista de países.

O utilizador pode escolher um país existente ou adicionar, caso ainda não exista. Após a escolha do país o sistema apresenta o país escolhido e carrega a lista de regiões desse país (caso existam). No momento seguinte o utilizador deverá seleccionar a região. O sistema apresenta a respetiva lista, o utilizador tem de escolher ou adicionar a região pretendida. Após a seleção da região, o sistema apresentará a região seleccionada e iniciará o carregamento das cidades dessa região. De forma análoga, o utilizador escolherá ou adicionará uma cidade. Em seguida deverá seleccionar a opção “Guardar Alterações”. O sistema procederá à alteração da cidade do utilizador, mas antes se o utilizador adicionou um novo país, região e/ou cidade este irá adicioná-los ao sistema. Este caso está descrito no diagrama da Figura 50.

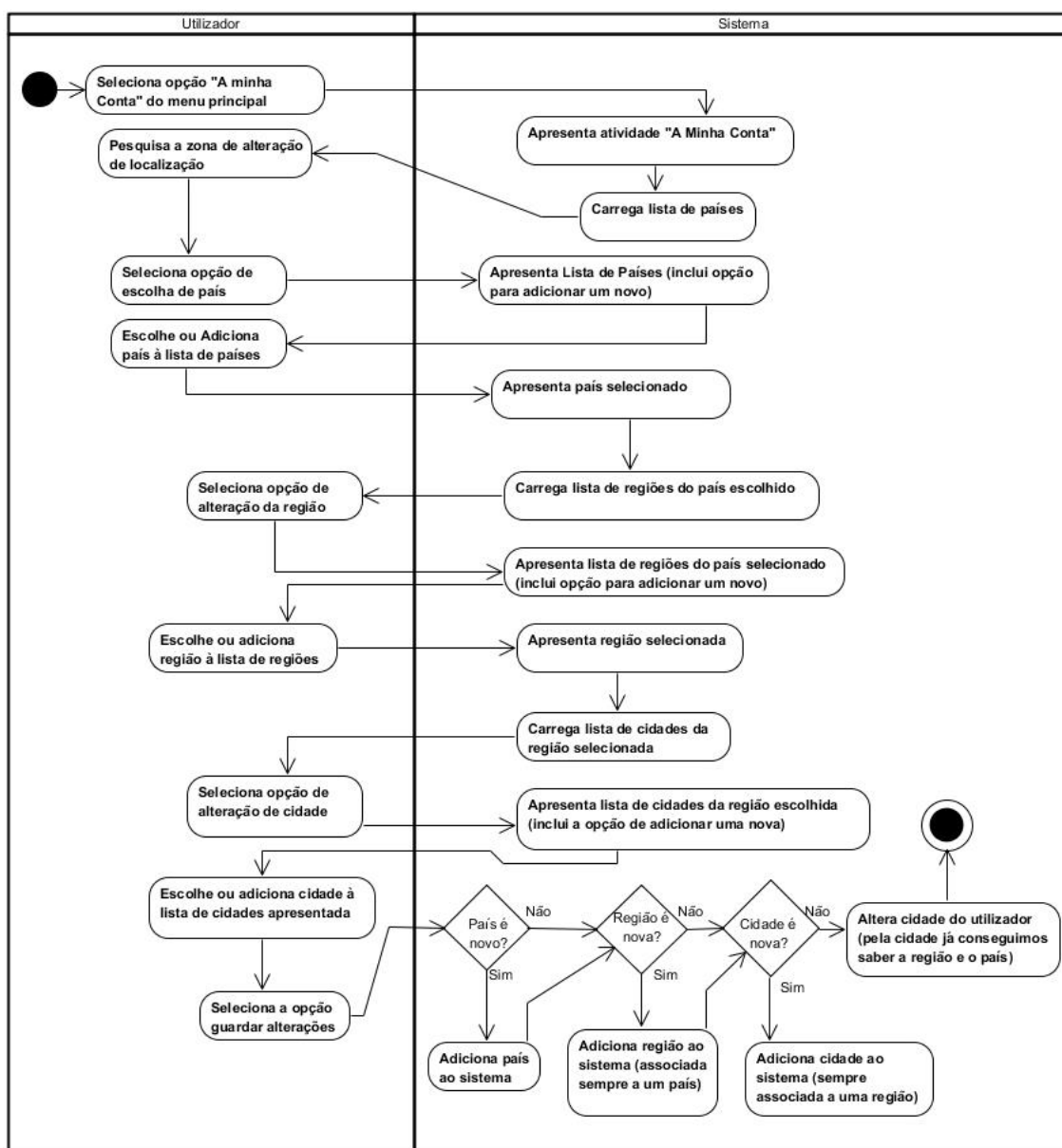


Figura 50 - Diagrama de atividade para o caso "Alterar cidade"

Para **pesquisar um item pelo nome**, o utilizador seleciona a opção “Jogos” ou “Livros” da atividade “Menu Principal”, conforme a categoria de itens a que pretende aceder. Após o utilizador escolher a opção, o sistema apresenta a respetiva atividade. No caso dos Jogos estes têm a particularidade de poderem ser filtrados por plataforma de jogo. O utilizador deve selecionar a opção de pesquisa. O sistema seleciona a zona destinada à Pesquisa (caso ainda não esteja selecionada), também seleciona a zona de escrita das palavras a utilizar na pesquisa e apresenta o teclado. O utilizador tem de escrever o nome ou a sigla do item que quer pesquisar e dar ordem à aplicação para iniciar a pesquisa. O sistema pesquisa o item pelo nome e pela categoria selecionada. No caso dos jogos, se o utilizador tiver uma plataforma específica selecionada a pesquisa também será filtrada pela mesma, apresentando os resultados apenas para essa plataforma. Se o sistema não encontrar itens correspondentes aos termos introduzidos pelo utilizador, este apresentará uma mensagem a dizer que não encontrou resultados, caso contrário apresentará a lista dos itens encontrados. Este caso é representado esquematicamente no diagrama da Figura 51.

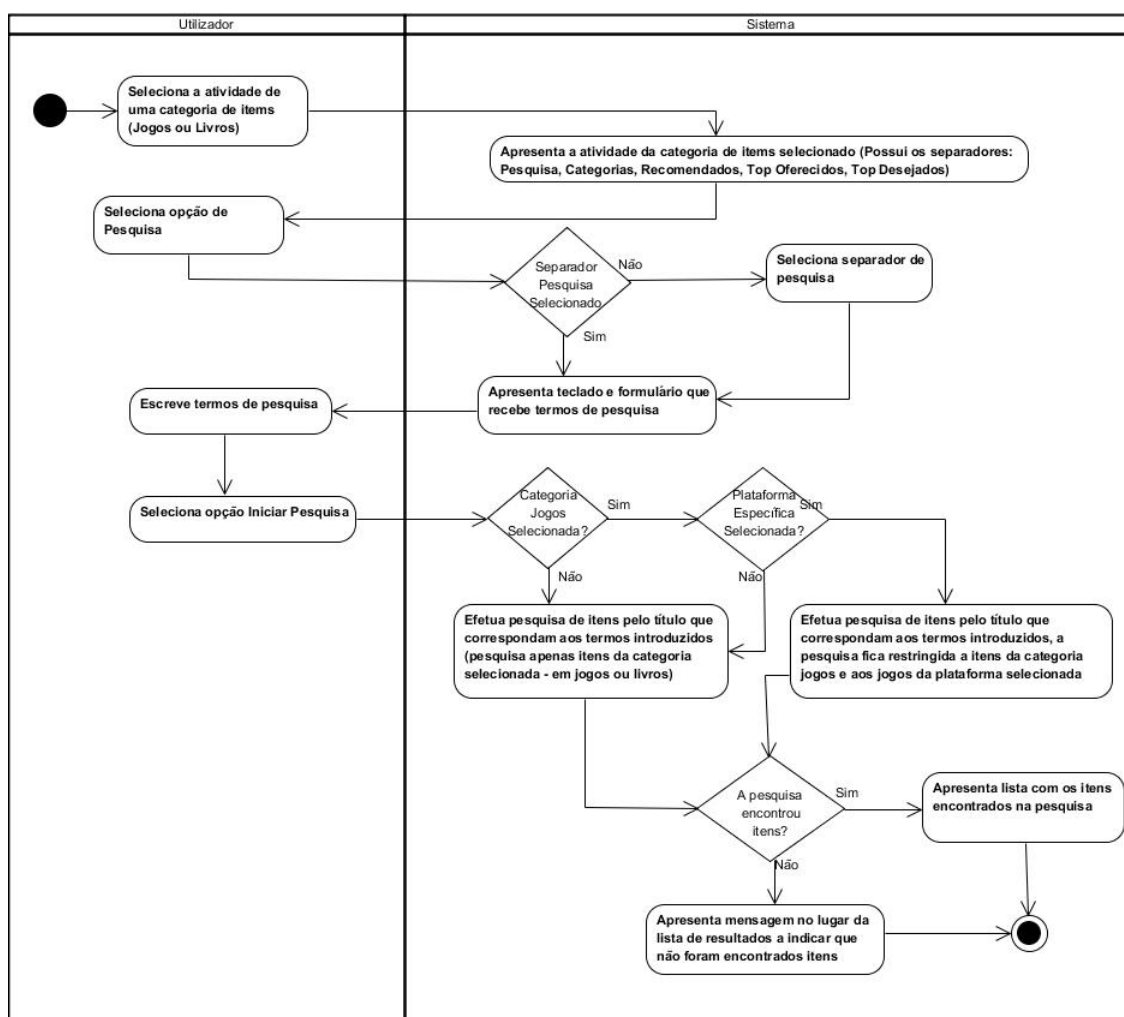


Figura 51 - Diagrama de atividade para o caso "Pesquisar um Item pelo Nome"

O caso **pedir ao administrador para adicionar novo item** faz sentido na sequência do caso anterior. Quando, numa pesquisa, o utilizador não encontrar o item que pretende deve solicitar a sua adição ao administrador. Para tal, o utilizador estando no separador de pesquisa do caso anterior, deverá seleccionar a opção “Pedir para adicionar novo item”. O sistema apresentará a atividade de pedido de adição de novo item. O sistema carregará para a categoria de item seleccionada as listas de títulos, géneros, publicadores/editores. Quando o utilizador estiver a preencher os formulários, as listas anteriores serão utilizadas para apresentar sugestões para completar o formulário. O utilizador deve preencher o formulário do pedido indicando o título do item, o género, a plataforma, a versão, o publicador e a data de lançamento. No final deve solicitar o envio do pedido. O sistema enviará o pedido e notificará o utilizador da entrega do pedido ao administrador. Este está representado no diagrama da Figura 52.

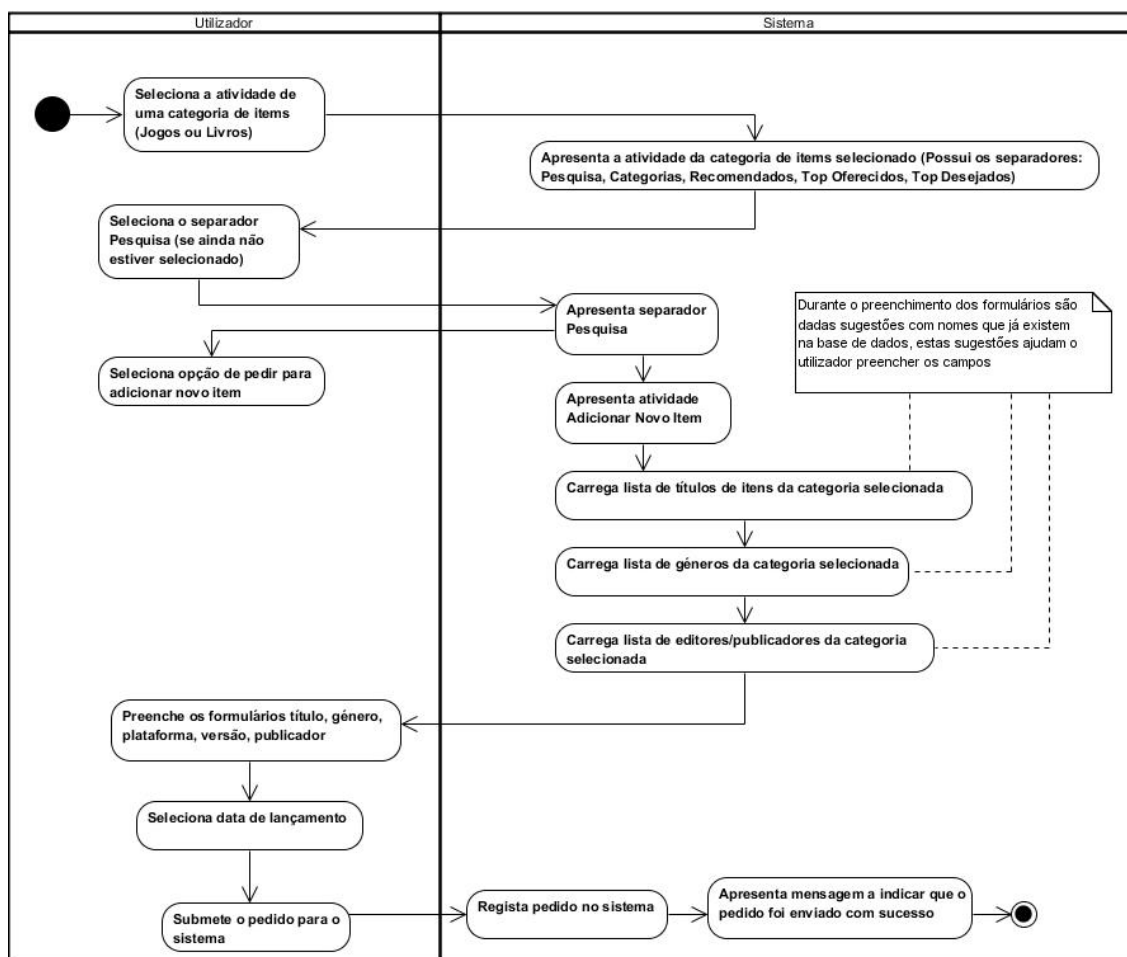


Figura 52 - Diagrama de atividade para o caso "Pedir ao Administrador para Adicionar Novo Item"

Para **efetuar uma pesquisa por género**, o utilizador tem de escolher a categoria Jogos ou Livros, análogo ao caso anterior. Após a escolha, o sistema apresenta a respetiva

atividade e o utilizador deve seleccionar o separador “Géneros”. Depois, o sistema apresenta uma lista dos géneros da categoria seleccionada e o utilizador deverá seleccionar o género que pretende. Após esta escolha, o sistema apresentará a atividade que lista itens de uma determinada categoria. Esta ao iniciar pesquisará e apresentará todos os itens que se incluam no género seleccionado pelo utilizador. É importante referir, que tal como no caso anterior, se a categoria escolhida for a dos Jogos, esta pesquisa ficará filtrada pela plataforma de jogo seleccionada. Este caso é representado no diagrama da Figura 53.

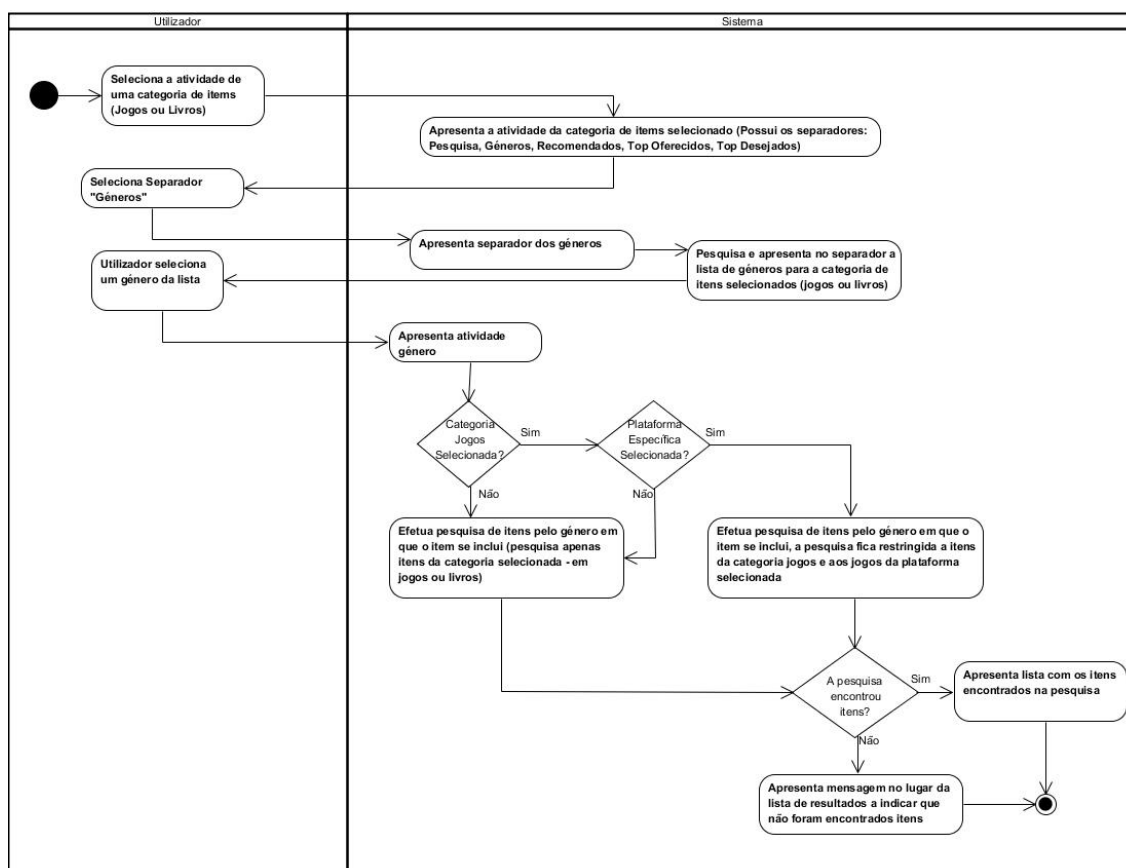


Figura 53 - Diagrama de atividade para o caso "Efetuar Pesquisa por Género"

Para **visualizar a lista de itens recomendados** o utilizador deve escolher uma categoria, Jogos ou Livros. Após esta escolha, o sistema apresenta a respetiva atividade. O utilizador deverá seleccionar o separador Recomendados. O sistema acederá aos cinco géneros preferidos do utilizador e pesquisará dois itens aleatórios de cada um desses géneros, perfazendo um total de dez itens recomendados. No caso da categoria Jogos, mais uma vez, a pesquisa será restringida a jogos da plataforma seleccionada. Caso o utilizador ainda não possua géneros preferidos suficientes para perfazer o top de cinco géneros preferidos, o sistema irá automaticamente buscar itens aleatórios (de qualquer género) para perfazer

os dez itens recomendados. Contudo os primeiros elementos respeitarão os géneros que já estiverem como preferidos do utilizador. Este caso de uso é apresentado no diagrama da Figura 54.

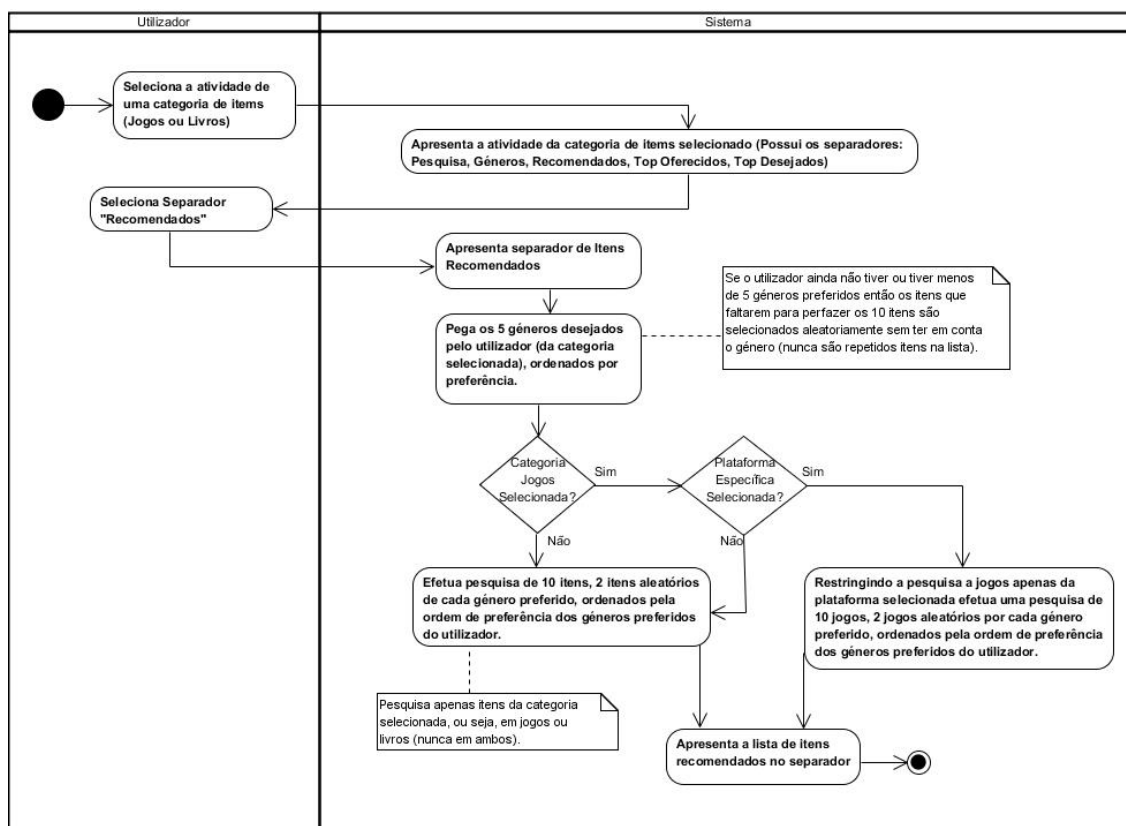


Figura 54 - Diagrama de atividade para o caso "Visualizar Lista de Itens Recomendados"

Para o caso **visualizar Top Desejados**, o utilizador escolhe a opção “Jogos” ou “Livros” a partir da atividade “Menu Principal”. O sistema apresenta a respetiva atividade e o utilizador deve selecionar o separador “Top Desejados”. Neste momento, o sistema pesquisará e apresentará, numa lista os trinta itens com mais utilizadores interessados, ordenados de forma decrescente relativamente ao número de interessados. No caso da categoria selecionada ser a dos Jogos, este Top será restringido à plataforma selecionada no momento (caso esteja alguma selecionada). Este caso está descrito no diagrama da Figura 55.

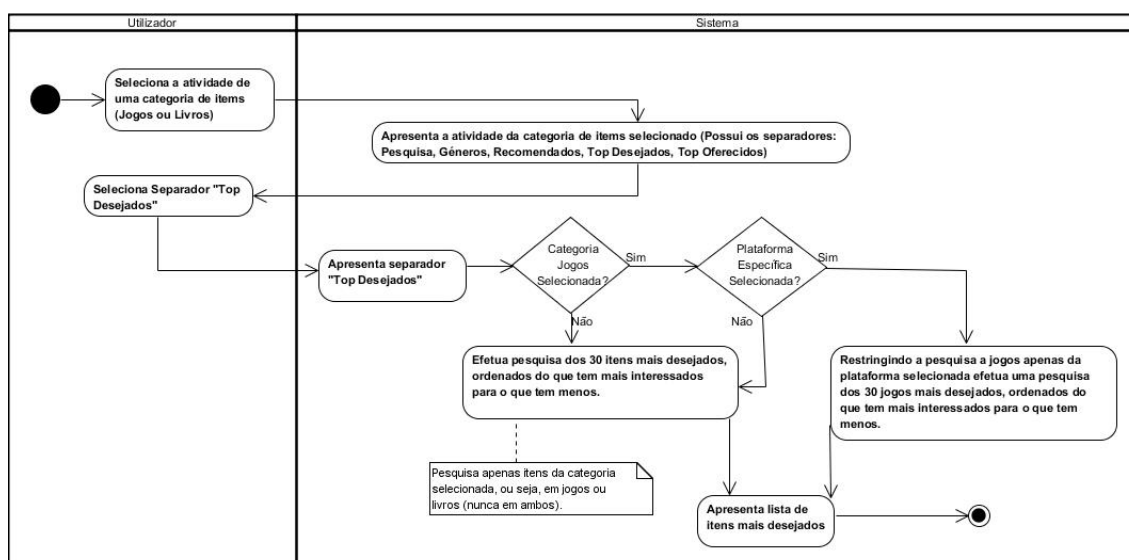


Figura 55 - Diagrama de atividade para o caso "Visualizar Top Desejados"

O caso **visualizar Top Oferecidos** é muito semelhante ao anterior, mas em vez de ter em conta o número de utilizadores interessados, tem em conta o número de utilizadores proprietários. Este está representado no diagrama da Figura 56.

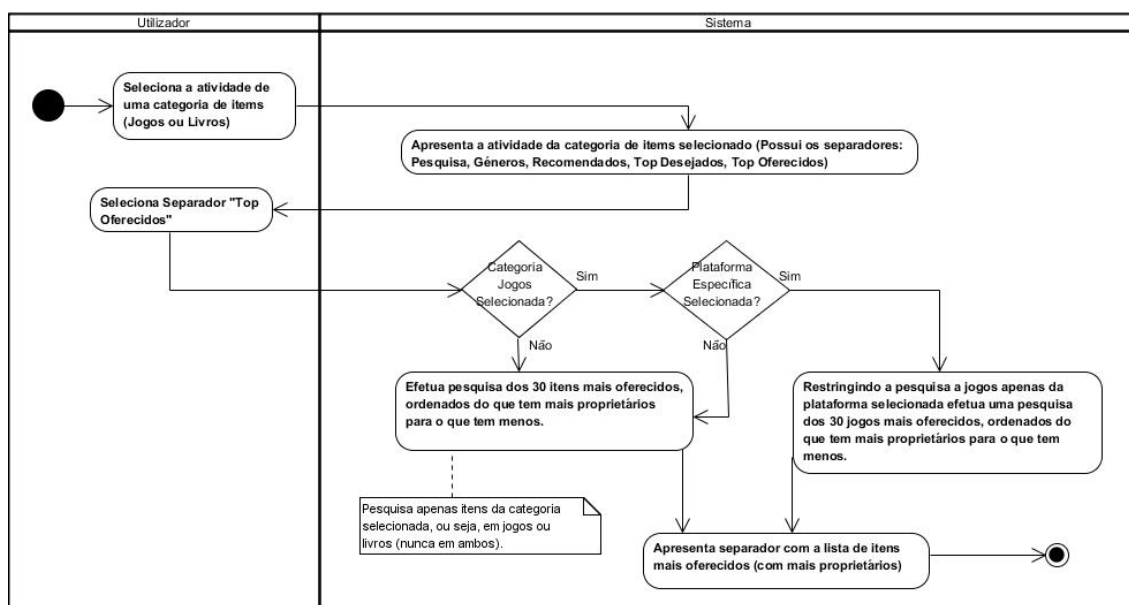


Figura 56 - Diagrama de atividade para o caso "Visualizar Top Oferecidos"

Após os casos que explicam as diversas formas de visualizar diversas listas de itens, está na altura de apresentar o caso **adicionar um item à lista de desejos**. Para tal, o utilizador deve selecionar um item de uma das listas mencionadas nos casos anteriores. Existem duas formas de adicionar um item à lista de desejos. A primeira consiste em pressionar durante cerca de 2 segundos o item (jogo ou livro) desejado, o sistema apresentará as

opções para o item selecionado, o utilizador deverá escolher a opção “Adicionar item à lista de desejos”, o sistema irá apresentar uma lista com os vários níveis de desejo em relação ao item e o utilizador escolhe o nível que considera mais realista. Após o utilizador escolher o nível de desejo, o sistema irá verificar se o utilizador não possui o item em nenhuma das suas listas. Caso não o possua, o item é adicionado e é apresentada uma mensagem de sucesso. Caso contrário, é apresentado o motivo pelo qual o item não pode ser adicionado à lista de desejos. A segunda opção consiste em o utilizador selecionar um item. O sistema apresentará a atividade “Detalhes do Item”, constituído por três separadores: “Perfil”, “Proprietários” e “Interessados”. O sistema irá verificar se o utilizador já possuiu o item em alguma das suas listas (de desejos ou ofertas). Se não possuir, no separador “Perfil” serão apresentados as opções “Eu tenho!” e “Eu desejo!”. Se existir em alguma das listas é apresentada a respetiva opção de remover o item da lista. Neste caso, o utilizador deve selecionar “Eu desejo”. O sistema apresentará a lista com os vários níveis de desejo possíveis. O utilizador deverá escolher o nível de desejo dessa lista. Posto isto, o sistema adiciona o item à lista de desejos, atualiza o número de interessados pelo item e atualiza as opções para a situação atual, tudo isto no separador “Perfil”. Este caso está explícito no diagrama da Figura 57.

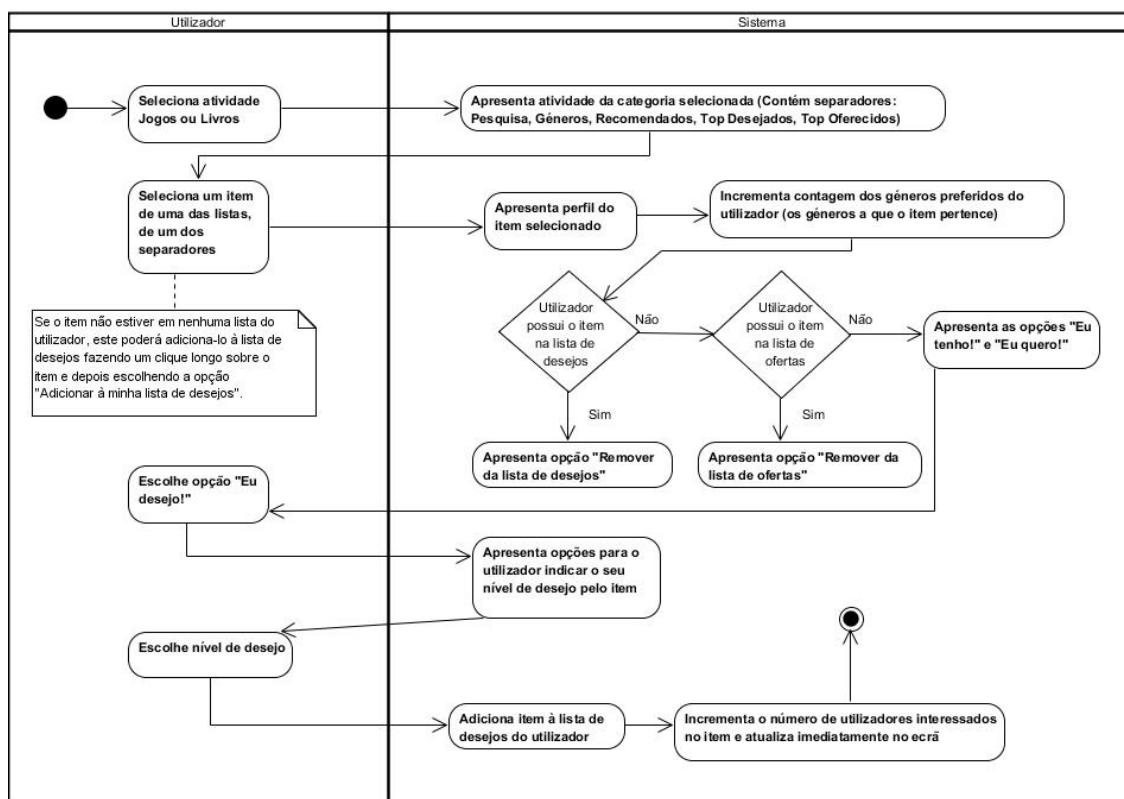


Figura 57 - Diagrama de atividade para o caso "Adicionar item à lista de desejos"



O caso **adicionar um item à lista de ofertas** segue a mesma lógica do caso anterior. As diferenças estão em que no separador “Perfil” (na atividade “Detalhes do Item”) o utilizador deverá seleccionar a opção “Eu tenho!”. Após isto, o sistema apresenta uma lista com os vários níveis de conservação em que o item se pode encontrar. O utilizador escolhe o estado de conservação, de acordo com o estado real de conservação do item que possui. Seguidamente, o sistema adiciona o item à lista de ofertas do utilizador. Caso o utilizador prefira, de forma análoga ao caso anterior, poderá adicionar o item simplesmente pressionando este durante cerca de dois segundos, de uma das listas de itens mencionadas noutros casos de utilização anteriores e quando o sistema apresentar as opções para o item seleccionado deverá escolher a opção “Adicionar à lista de ofertas”. Após seleccionar esta opção, será apresentada a lista para a escolha do estado de conservação do item, análogo à outra forma de adicionar o item à lista de ofertas. Este é representado no diagrama da Figura 58.

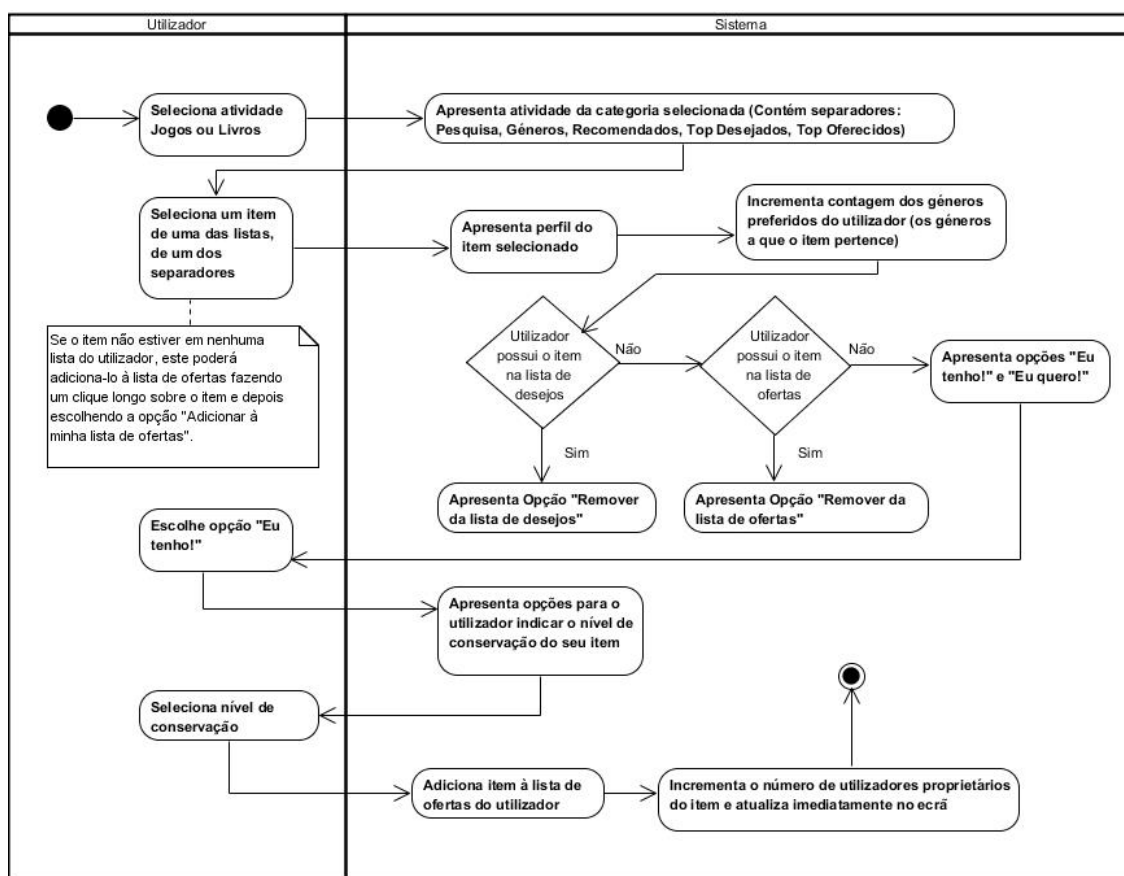


Figura 58 - Diagrama de atividade para o caso "Adicionar item à lista de ofertas"

Outro caso de utilização é o **remover item da lista de desejos**. Para este existem duas possibilidades. A primeira possibilidade é representada no Diagrama Figura 59.

Basicamente é feita através da seleção da opção “Remover item da lista de desejos”, quando o utilizador se encontra a visualizar o perfil do item que pretende remover da sua lista de desejos. A segunda possibilidade consiste em o utilizador seleccionar a opção “A Minha Conta” na atividade “Menu Principal”. O sistema apresenta a atividade “A Minha Conta”. Se o utilizador pretender gerir as listas de desejos e de ofertas dos livros deverá seleccionar a opção “Os Meus Livros”. Se pretender gerir as listas relativas aos Jogos deverá seleccionar a opção “Os Meus Jogos”. Após a escolha de uma destas opções, o sistema apresentará a respetiva atividade de gestão das listas, sendo que esta é constituída por dois separadores: “Eu Tenho” e “Eu Desejo”. O primeiro apresenta a lista de ofertas e o segundo a lista de desejos. Para este caso o utilizador deve seleccionar o separador “Eu Desejo”. O sistema carregará a lista dos itens que o utilizador deseja, respeitando a categoria seleccionada. O utilizador pressiona durante cerca de dois segundos o item que pretende alterar. O sistema apresenta uma lista de opções para o item seleccionado. O utilizador escolhe a opção “Remover o Item”. Após isto, o sistema remove o item da lista de desejos do utilizador. Esta segunda possibilidade é apresentada no diagrama da Figura 60.

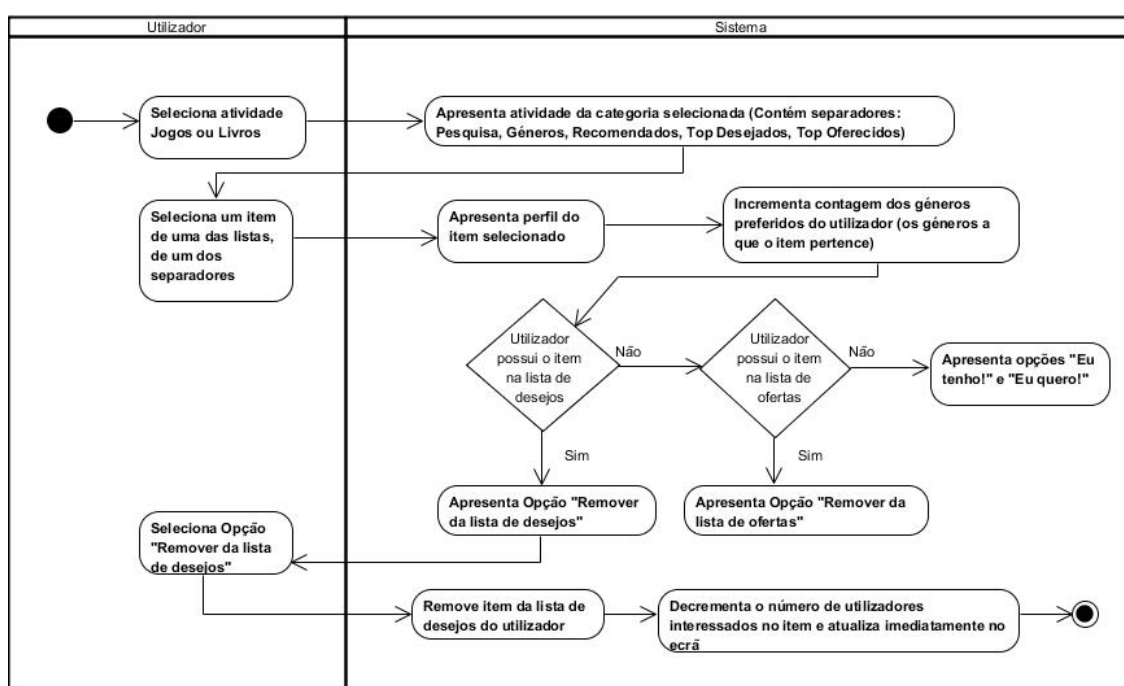


Figura 59 - Diagrama de atividade para o caso "Remover item da lista de desejos" (Modo 1)

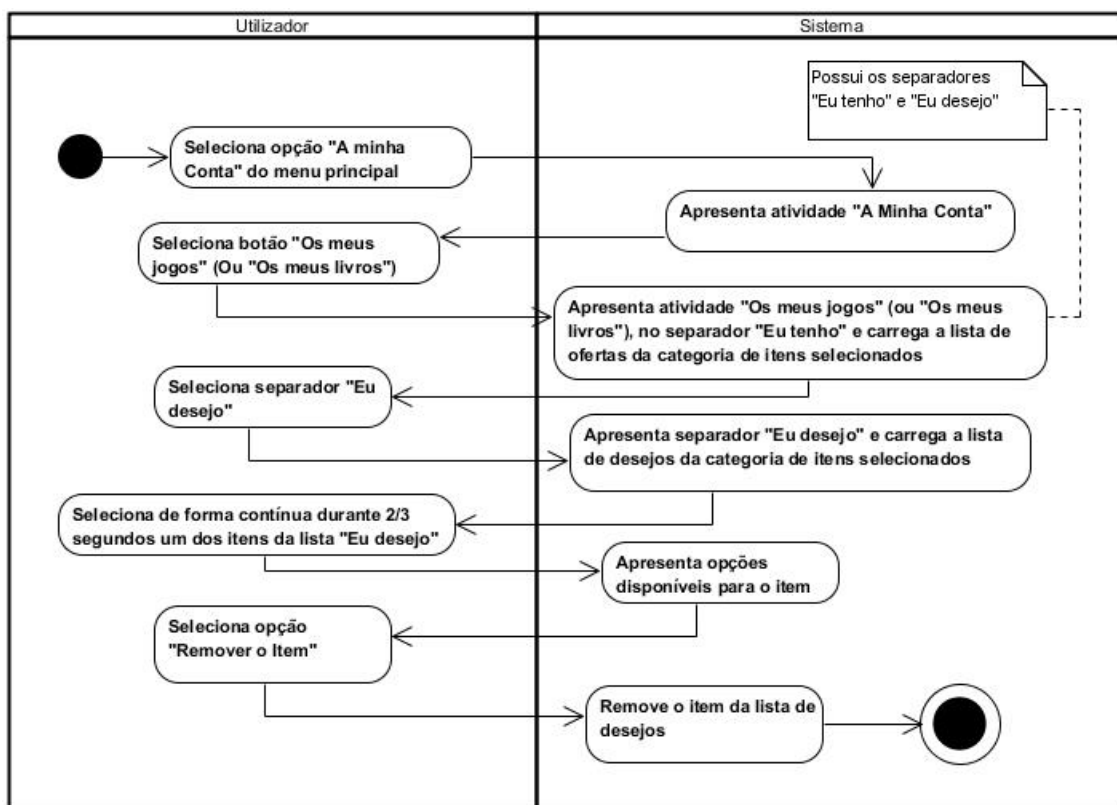


Figura 60 - Diagrama de atividade para o caso "Remover item da lista de desejos" (Modo 2)

Análogo ao caso anterior é o caso de **remover item da lista de ofertas**. Para este existem novamente duas possibilidades. A primeira é representada no diagrama da Figura 61. É feita através da seleção da opção “Remover item da lista de ofertas”, quando o utilizador se encontra a visualizar o perfil do item que pretende remover da sua lista de ofertas. Relativamente à segunda possibilidade, é análoga à segunda possibilidade no caso mencionado anteriormente, mas o utilizador em vez de pressionar durante cerca de dois segundos um item do separador “Eu Desejo”, deverá fazê-lo num item da lista presente no separador “Eu Tenho”. O resto do procedimento é igual, exceto o facto de que o item é removido da lista de ofertas. Esta possibilidade é apresentada no diagrama da Figura 62.

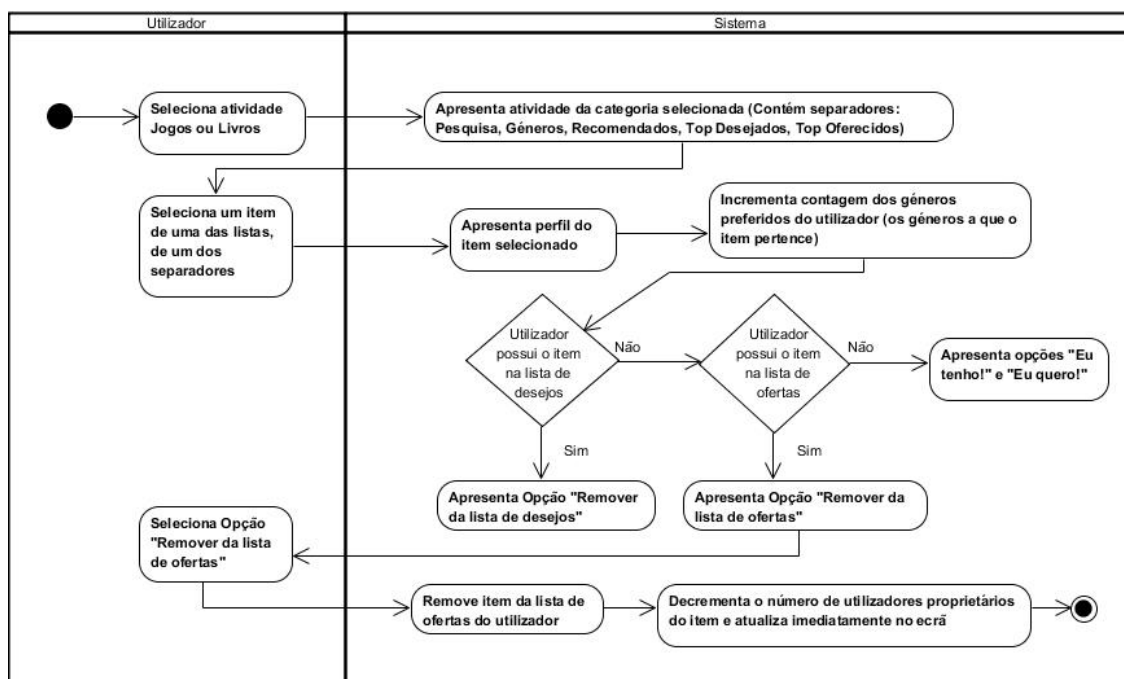


Figura 61 - Diagrama de atividade para o caso "Remover item da lista de ofertas" (Modo 1)

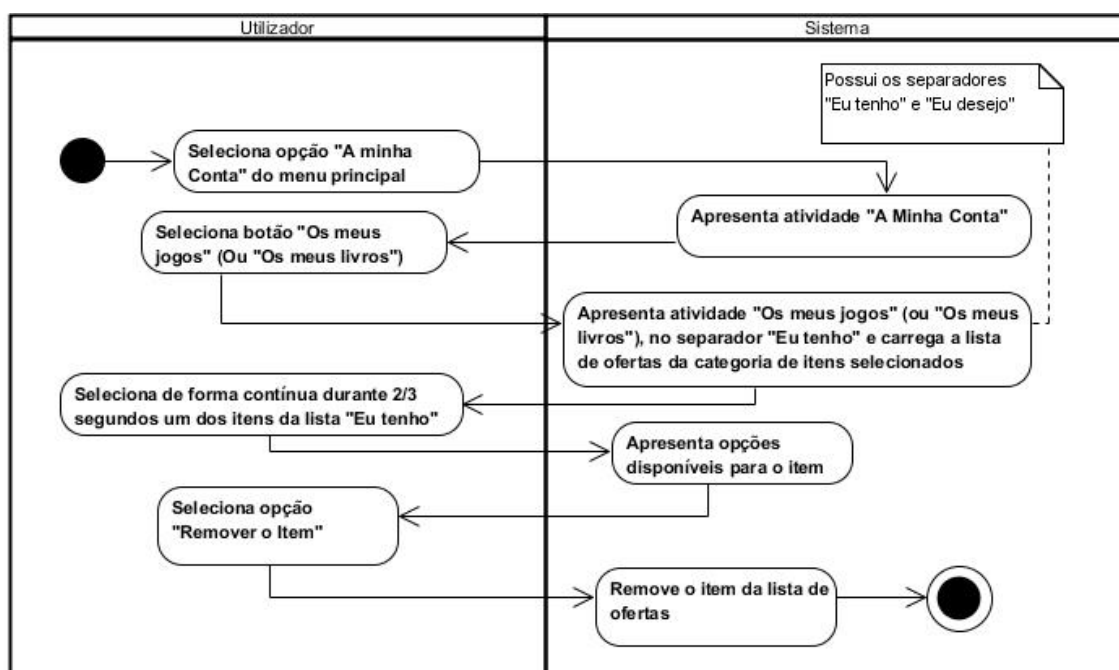


Figura 62 - Diagrama de atividade para o caso "Remover item da lista de ofertas" (Modo 2)

Aproveitando a contextualização dos dois casos anteriores (na segunda possibilidade explanada para realizar a mesma ação) temos o caso **editar estado de conservação de um item**. Supondo que o utilizador já se encontra na atividade de gestão de listas, no separador “Eu Tenho” este terá de pressionar durante cerca de dois segundos o item que

deseja alterar. O sistema irá apresentar as opções disponíveis para o item. O utilizador deve seleccionar a opção “Alterar estado de conservação”. Posto isto, o sistema apresenta uma lista de opções para o estado de conservação. O utilizador escolhe uma das opções, o sistema atualiza o estado de conservação. Este caso é representado no diagrama da Figura 63.

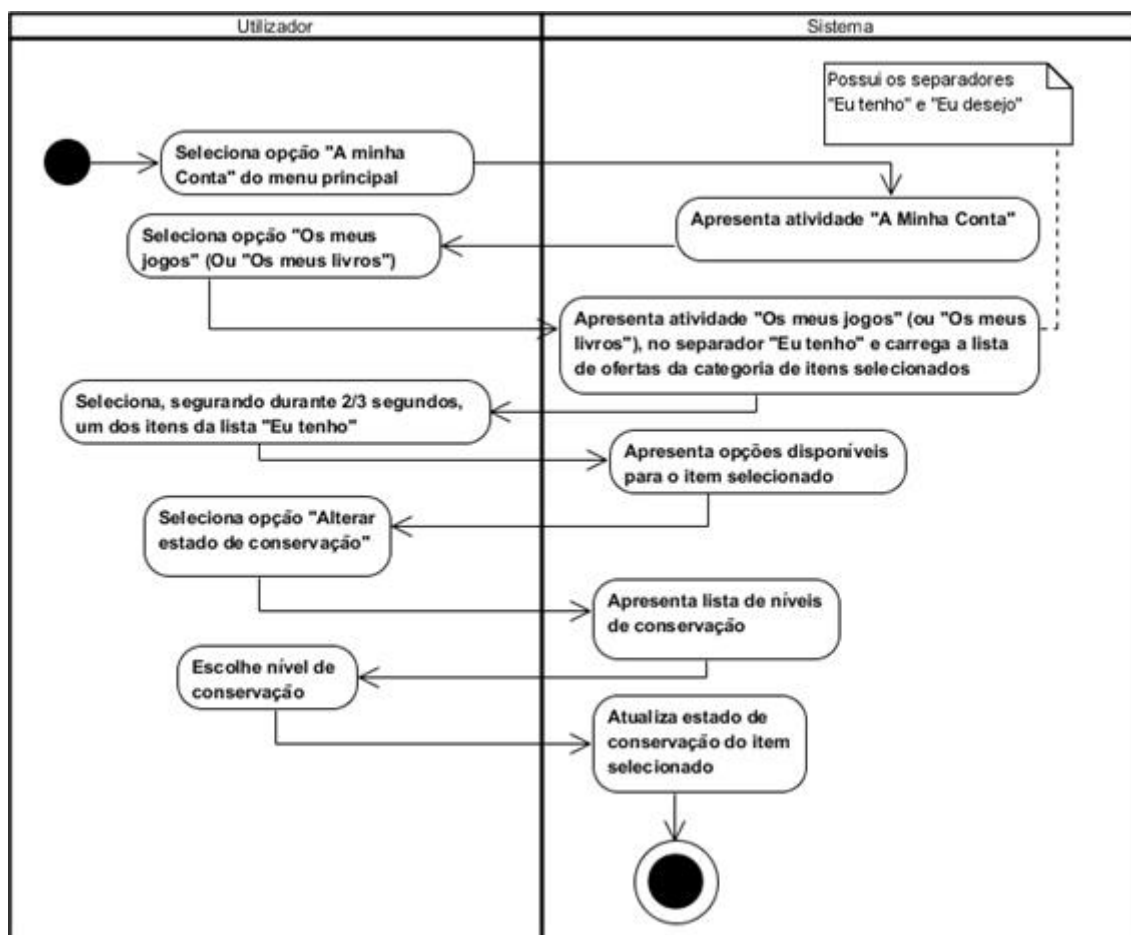


Figura 63 - Diagrama de atividade para o caso "Editar Estado de Conservação de um Item"

Para o caso editar **nível de desejo em relação a um item** o procedimento é análogo ao caso anterior, mas em vez do utilizador estar no separador “Eu Tenho” deverá seleccionar um item do separador “Eu Desejo”. Este caso é apresentado no diagrama da Figura 64.

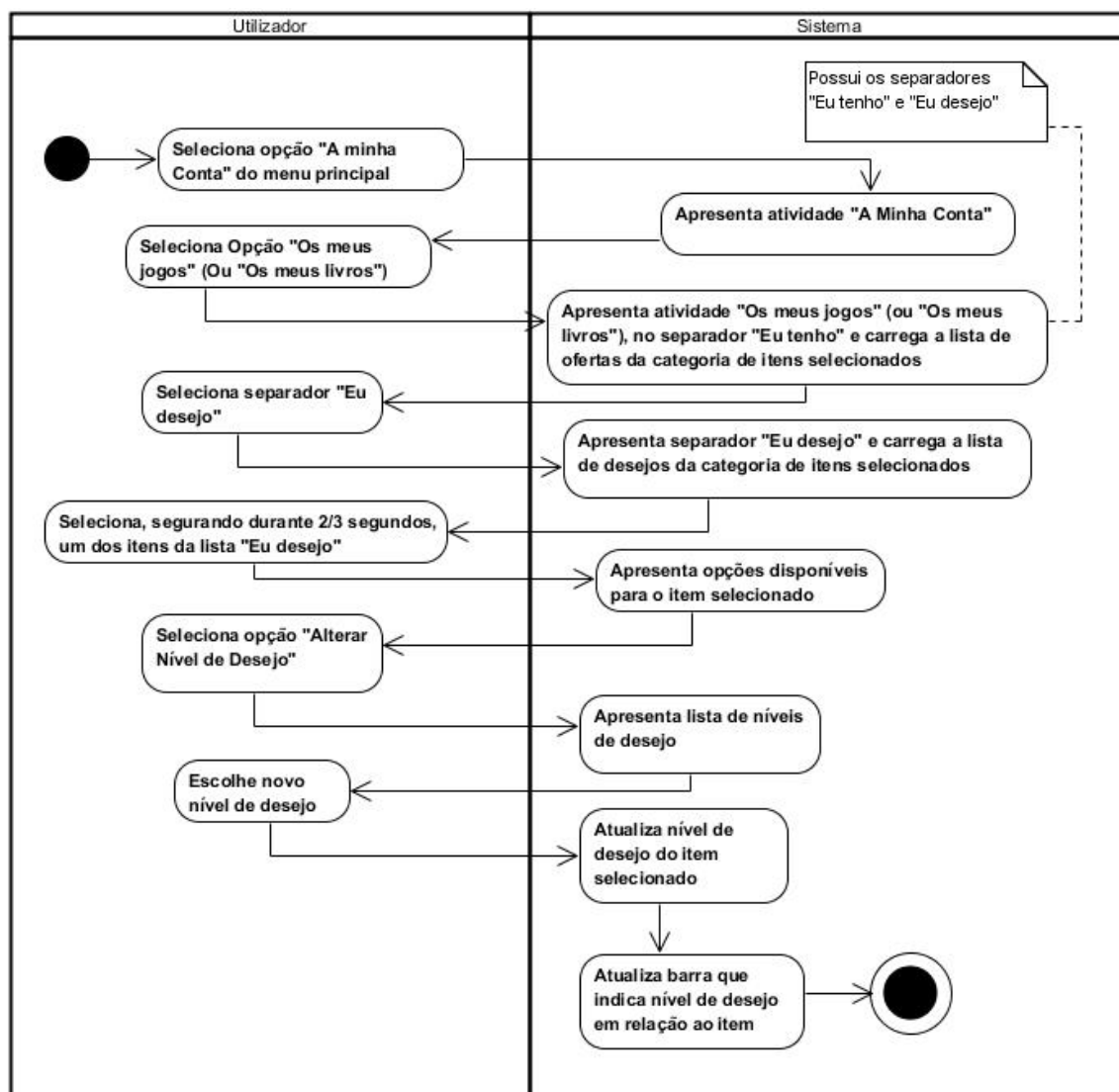


Figura 64 - Diagrama de atividade para o caso "Editar Nível de Desejo em Relação a um Item"

O caso de utilização **mover item da lista de ofertas para a lista de desejos** consiste em o utilizador pressionar durante cerca de dois segundos um item da lista presente no separador “Eu Tenho”, na atividade “As Minhas Listas”. O sistema apresenta a lista de opções para o item. O utilizador escolhe a opção “Mover para a lista de desejos”. O sistema apresenta uma lista com as opções dos níveis de desejo em relação ao item e o utilizador escolhe o nível de desejo que sente em relação ao item. O sistema remove o item da lista de ofertas e adiciona-o à lista de desejos. Este caso é descrito através do diagrama da Figura 65.

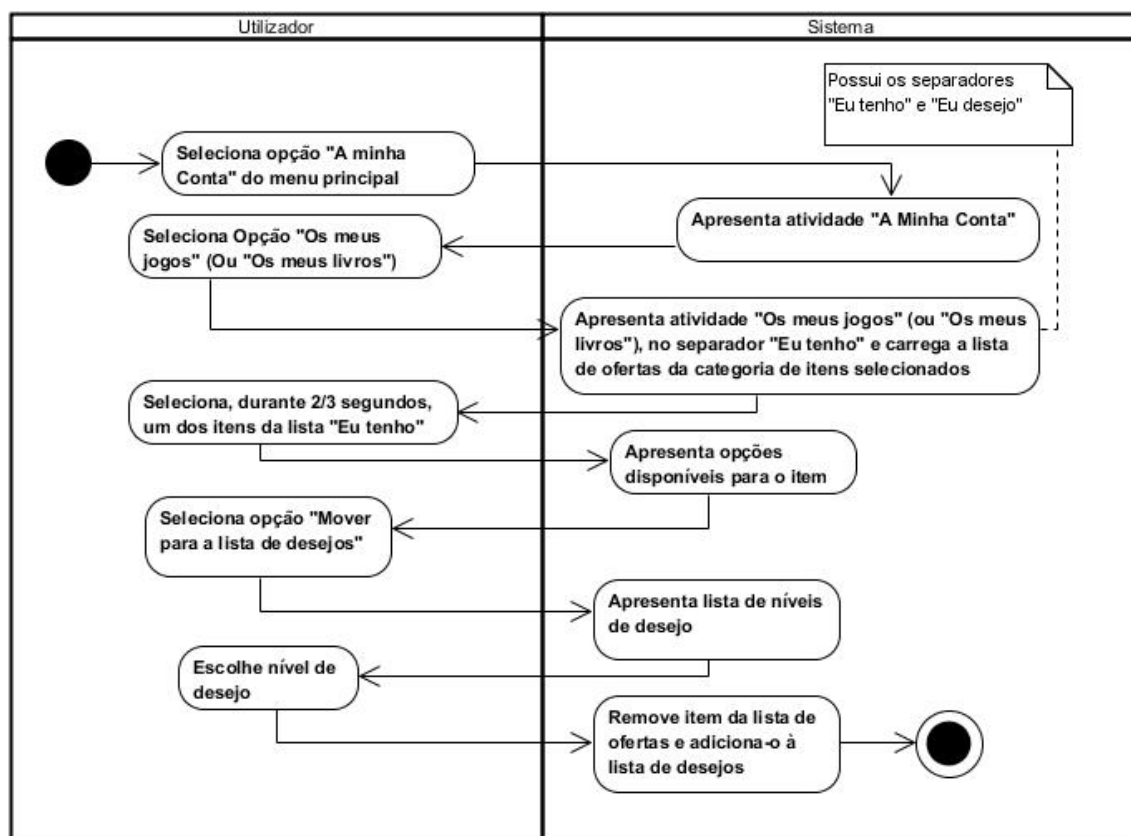


Figura 65 - Diagrama de atividade para o caso "Mover Item da Lista de Ofertas para a Lista de Desejos"

O caso **mover item da lista de desejos para a lista de ofertas** é análogo ao caso exposto anteriormente. A diferença está em que o utilizador deverá pressionar durante cerca de dois segundos um item da lista presente no separador "Eu Desejo" e, em vez do nível de desejo, o utilizador terá de escolher o estado de conservação do item. Este caso está representado no diagrama da Figura 66.

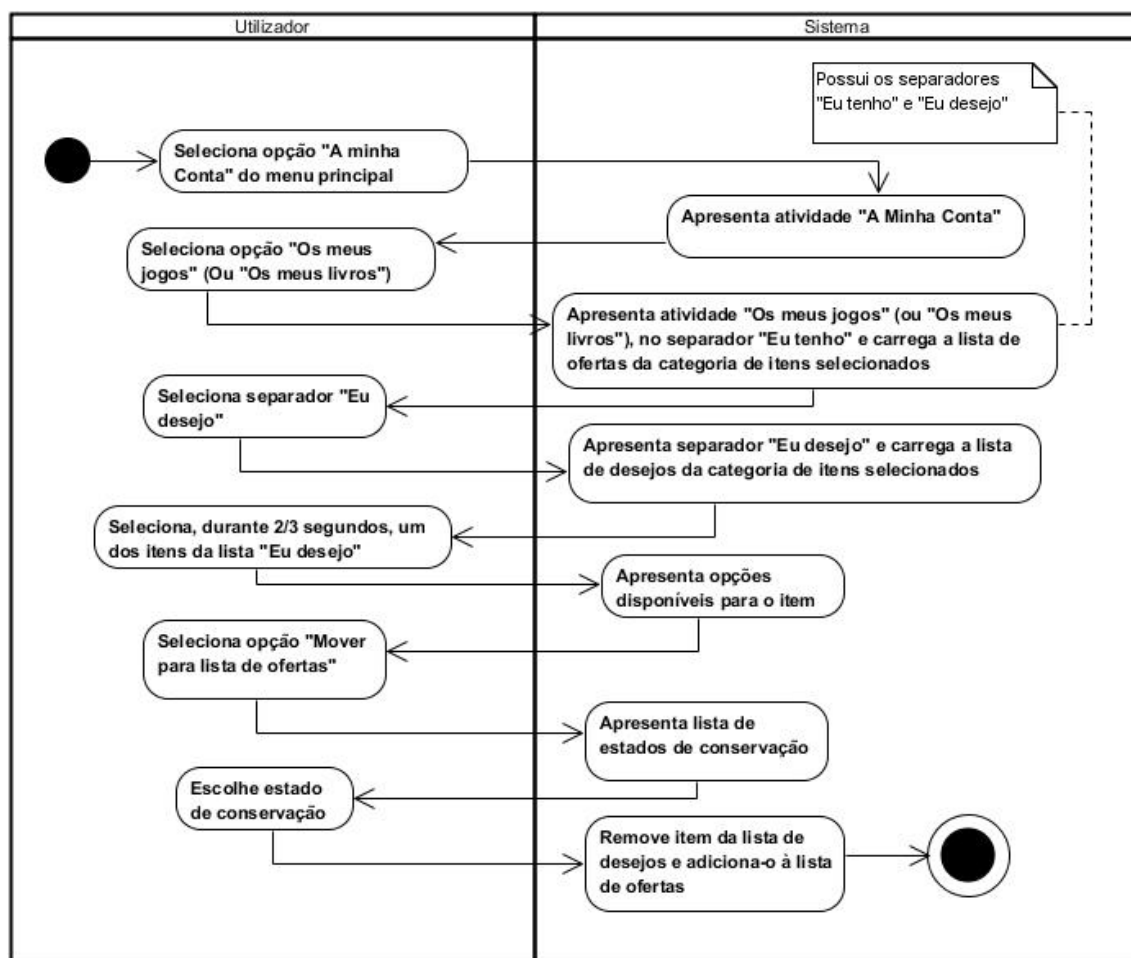


Figura 66 - Diagrama de atividade para o caso "Mover Item da Lista de Desejos para a Lista de Ofertas"

Para o caso de uso **visualizar lista de interessados em um item** o utilizador deve selecionar um item. O sistema apresentará a atividade “Detalhes do Item”. O utilizador deve selecionar o separador “Interessados”. O sistema pesquisa e apresenta uma lista de todos os utilizadores que desejam o item e ordena-os por distância em relação ao utilizador. É utilizada uma função com a fórmula de Haversine para calcular essas distâncias. Este caso está desenhado no diagrama da Figura 67.



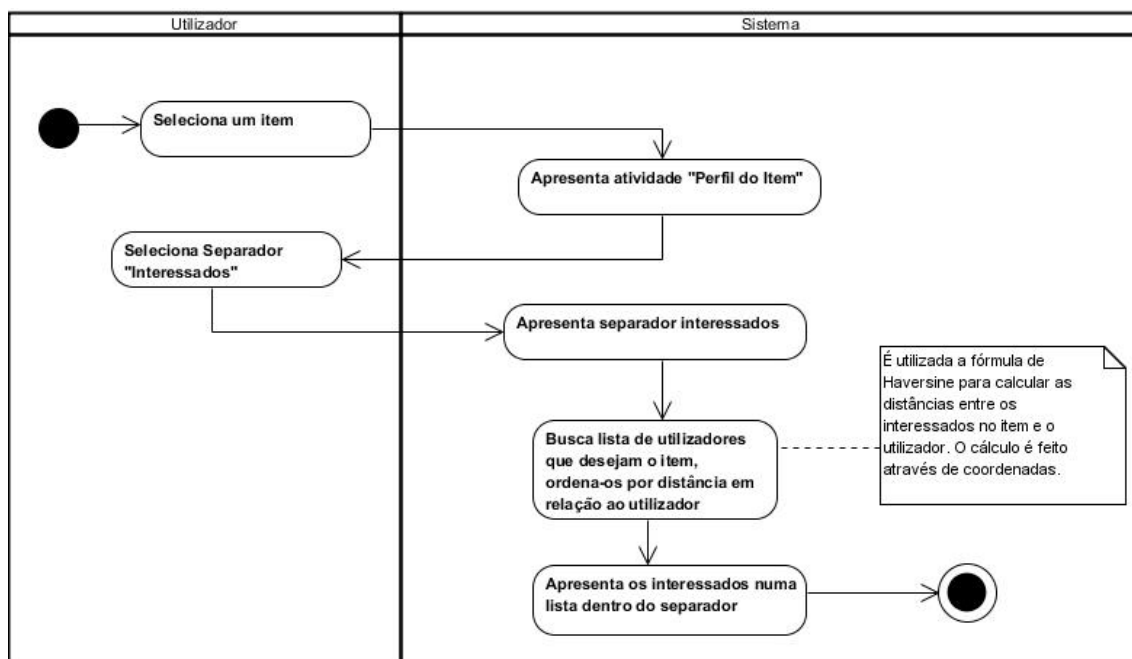


Figura 67 - Diagrama de atividade para o caso "Visualizar Lista de Interessados em um Item"

O caso **visualizar lista de proprietários** é análogo ao caso precedente, porém o separador selecionado pelo utilizador é o "Proprietários". Neste o sistema busca e apresenta a lista de todos os utilizadores que possuem o item. O diagrama elucidativo está apresentado na Figura 68.

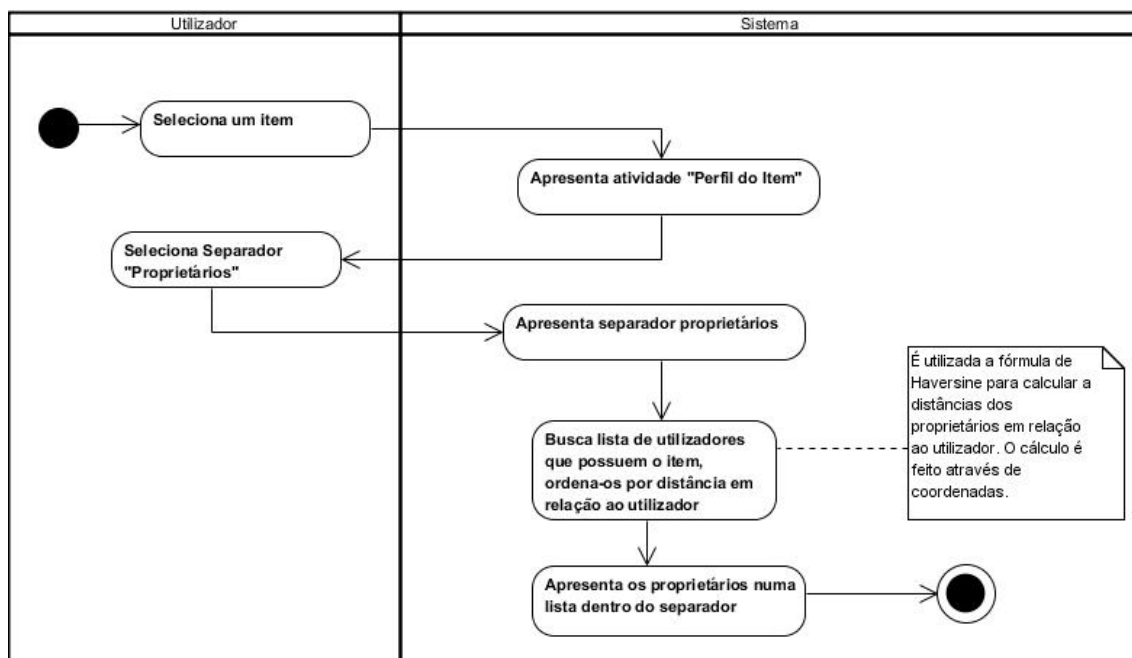


Figura 68 - Diagrama de atividade para o caso "Visualizar Lista de Proprietários de um Item"

Para o caso **aceder ao perfil público de um utilizador**, o utilizador tem de seleccionar a foto de um dos utilizadores da lista de proprietários ou da lista de interessados. O diagrama da Figura 69 ilustra este caso.

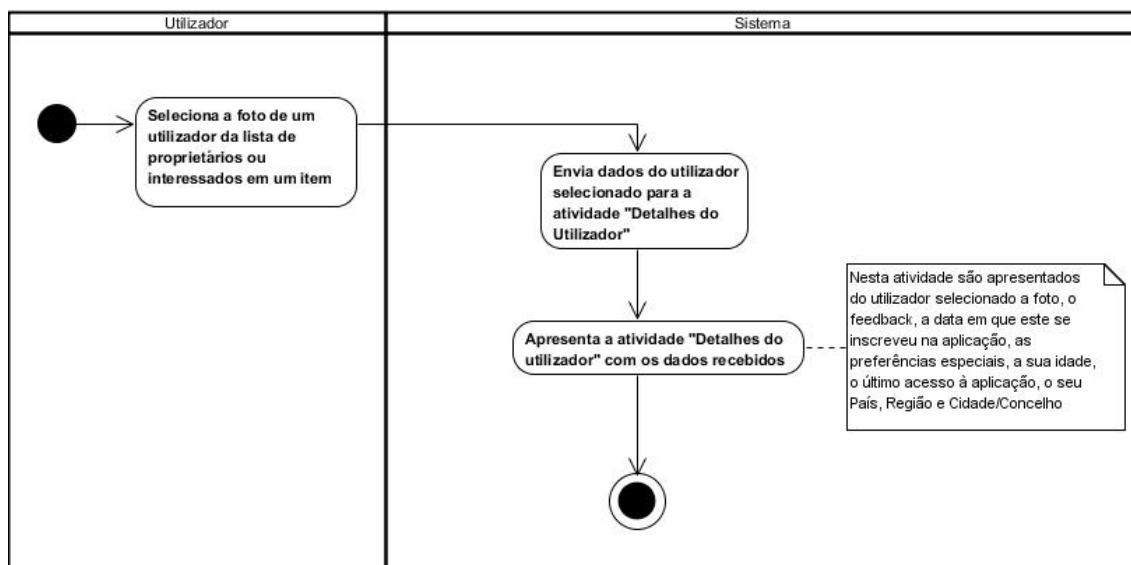


Figura 69 - Diagrama de atividade para o caso "Aceder ao Perfil Público de um Utilizador"

O caso **enviar proposta** é um dos mais importantes. Para tal, o utilizador tem de se encontrar no separador "Proprietários" da atividade "Detalhes do Item". O utilizador tem de seleccionar a opção "Enviar Proposta" de um dos utilizadores da lista de proprietários. O sistema abre a atividade "Enviar Proposta" e carrega os itens possíveis para oferta por parte do utilizador. Os itens possíveis têm em conta as preferências do outro utilizador (o que irá receber a proposta). Note-se que qualquer item que já esteja numa proposta que se encontre no estado "Em Negociação" não aparece como item possível para oferecer em troca. Caso seja uma proposta de troca de livros, o sistema verifica se o outro utilizador só deseja receber propostas de trocas de itens da sua lista de desejos. Caso esta preferência se verifique, o sistema selecciona os itens possíveis para oferta que se encontrarem ao "mesmo tempo" na lista de ofertas do utilizador que pretende efetuar a proposta e também na lista de desejos do utilizador que irá receber a proposta. Se não houver correspondências, o utilizador não pode fazer a proposta e no lugar da lista de itens para oferecer o sistema irá apresentar uma mensagem a explicar os motivos de não poder fazer a proposta. Caso seja uma proposta de troca de jogos o sistema verifica se o outro utilizador só quer receber propostas com jogos da sua lista de desejos. Se esta preferência for verificada, o procedimento é igual ao descrito para os livros, mas neste caso para itens da categoria jogos. Além disto, quando se trata de jogos, o outro utilizador poderá apenas

aceitar propostas que contenham jogos de determinadas plataformas. Se for o caso, os itens da lista de ofertas do utilizador que pretende efetuar a proposta serão filtrados pelas plataformas selecionadas pelo outro utilizador. Se não existirem itens para oferta, o sistema apresenta uma mensagem ao utilizador no lugar da lista de ofertas possíveis. Posto isto, e assumindo que o utilizador possui itens para oferecer na proposta, o utilizador terá de selecionar a opção “Selecionar o meu item”. O sistema apresentará a lista de itens possíveis para oferecer na proposta, o utilizador escolhe o item e o sistema passa a apresentar o item para oferta e o outro item. Depois o utilizador seleciona a opção “Enviar Proposta” e o sistema guarda-a e apresenta uma mensagem a indicar o sucesso do envio da proposta. Este caso está representado no diagrama da Figura 70.

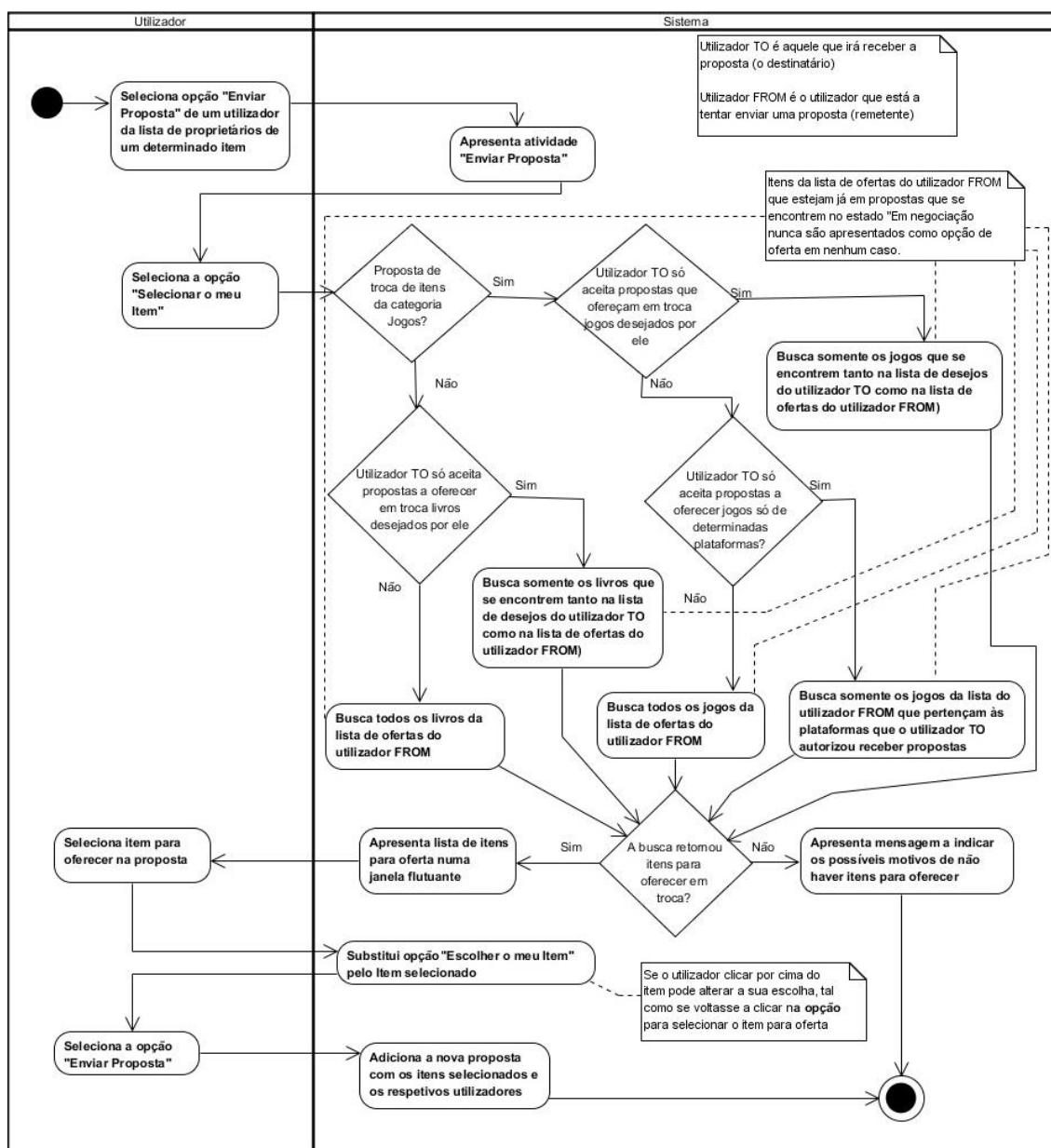


Figura 70 - Diagrama de atividade para o caso "Enviar Proposta"

Após a explanação do caso anterior, faz sentido apresentar o caso **aceitar proposta recebida**. Neste sentido, o utilizador tem de seleccionar a opção “Propostas” a partir da atividade “Menu Principal”. O sistema apresenta a atividade “Propostas” no separador “Recebidas” (o outro separador designa-se “Enviadas”). O utilizador deve solicitar a lista de filtros. O sistema apresentará a lista dos filtros possíveis para a proposta. O utilizador escolhe o filtro “Em negociação” e o sistema apresenta a lista de propostas recebidas (se não existirem, o sistema apresenta uma mensagem a informar). O utilizador acede a uma proposta seleccionando-a na lista. O sistema apresenta a atividade “Detalhes da Proposta”, como se trata de uma proposta recebida e em negociação, o sistema mostra as opções aceitar e rejeitar. Neste caso, o utilizador deverá seleccionar a opção “Aceitar”. O sistema apresenta uma janela com um diálogo de confirmação e o utilizador confirma-a. O sistema altera o estado da proposta para o estado “Aceite”. Este caso é representado pelo diagrama da Figura 71.

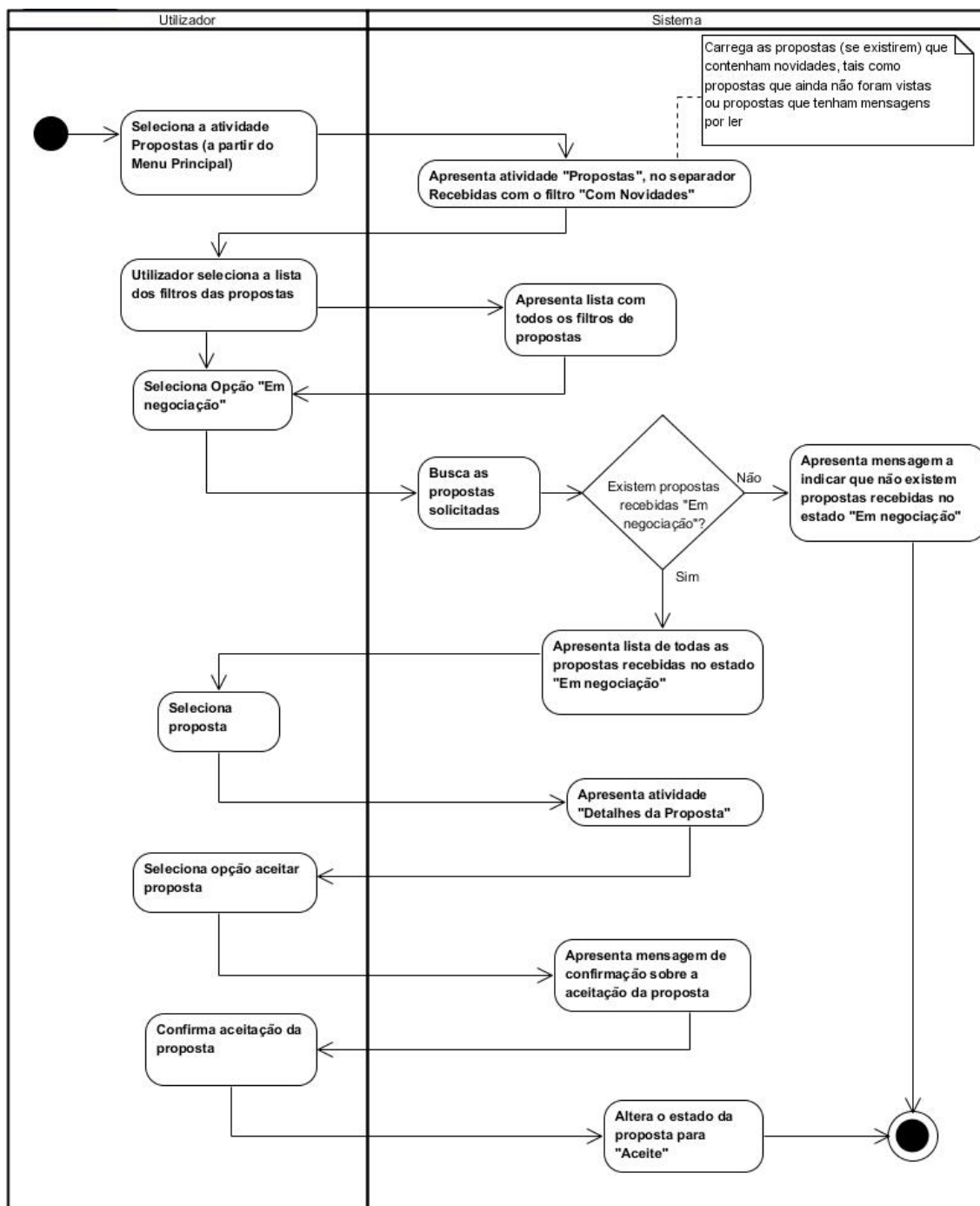


Figura 71 - Diagrama de atividade para o caso "Aceitar Proposta Recebida"

O caso de utilização **rejeitar proposta recebida** é semelhante ao anterior. Porém, o utilizador em vez de selecionar a opção "Aceitar" deverá selecionar a opção "Rejeitar". Neste caso, em vez de a proposta passar para o estado "Aceite" passará para o estado "Rejeitada". Este caso é apresentado no diagrama da Figura 72.

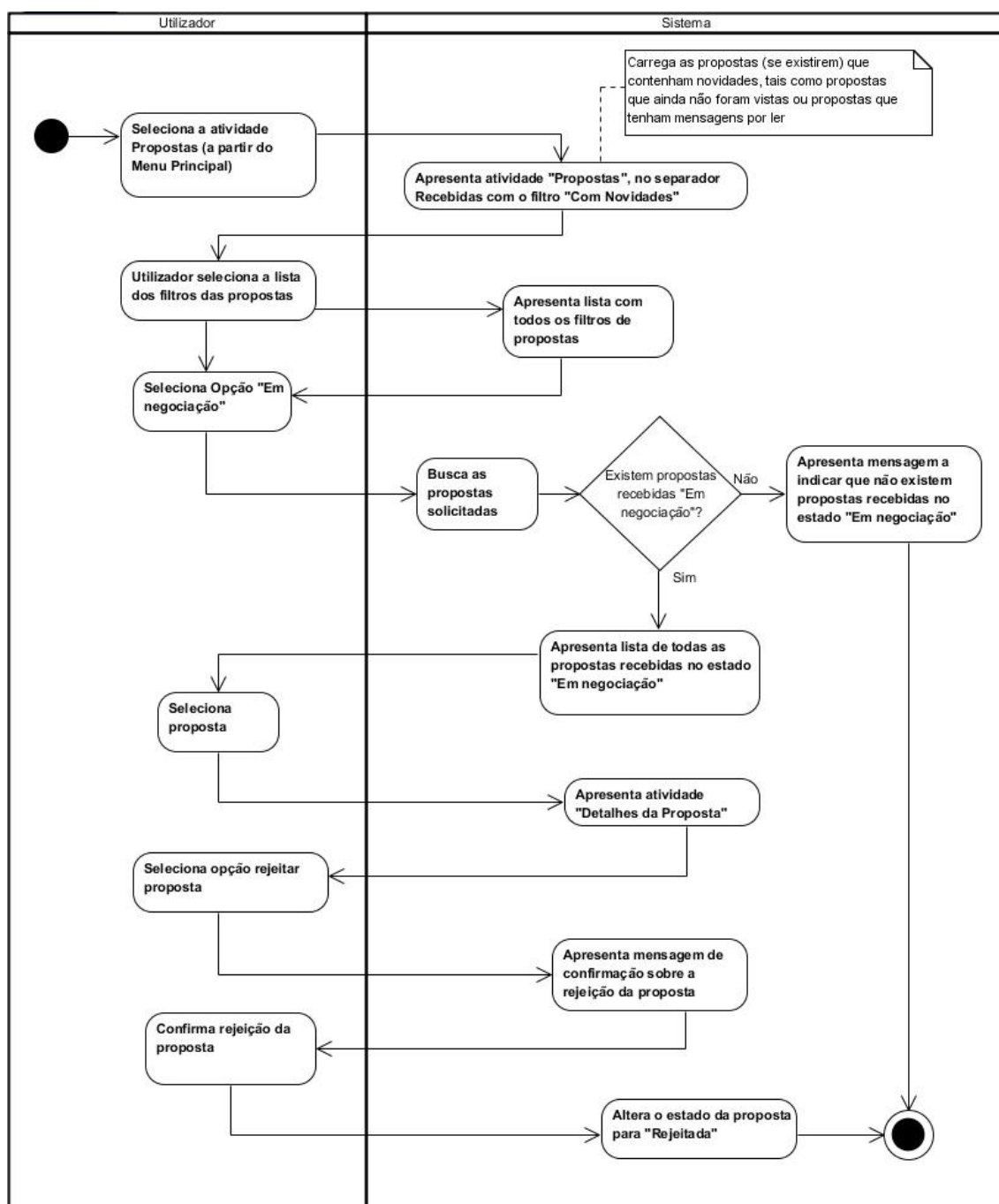


Figura 72 - Diagrama de atividade para o caso "Rejeitar Proposta Recebida"

Aproveitando a sequência, outro caso é **cancelar proposta enviada**. A opção de cancelar só pode ser efetuada pelo utilizador que enviou a proposta. Sendo assim, o utilizador para realizar este ato tem de se encontrar a visualizar as propostas enviadas que estão no estado "Em negociação". Neste ponto, seleciona uma proposta e o sistema apresenta a atividade "Detalhes da Proposta". O utilizador tem de escolher a opção "Cancelar Proposta". O sistema abre um diálogo de confirmação e o utilizador confirma que quer cancelá-la. O sistema altera o estado da proposta para o estado "Cancelada" (se o outro utilizador estiver

a ver a proposta irá ser notificado que a proposta foi cancelada) e termina a atividade “Detalhes da Proposta”. Este caso está especificado no diagrama da Figura 73.

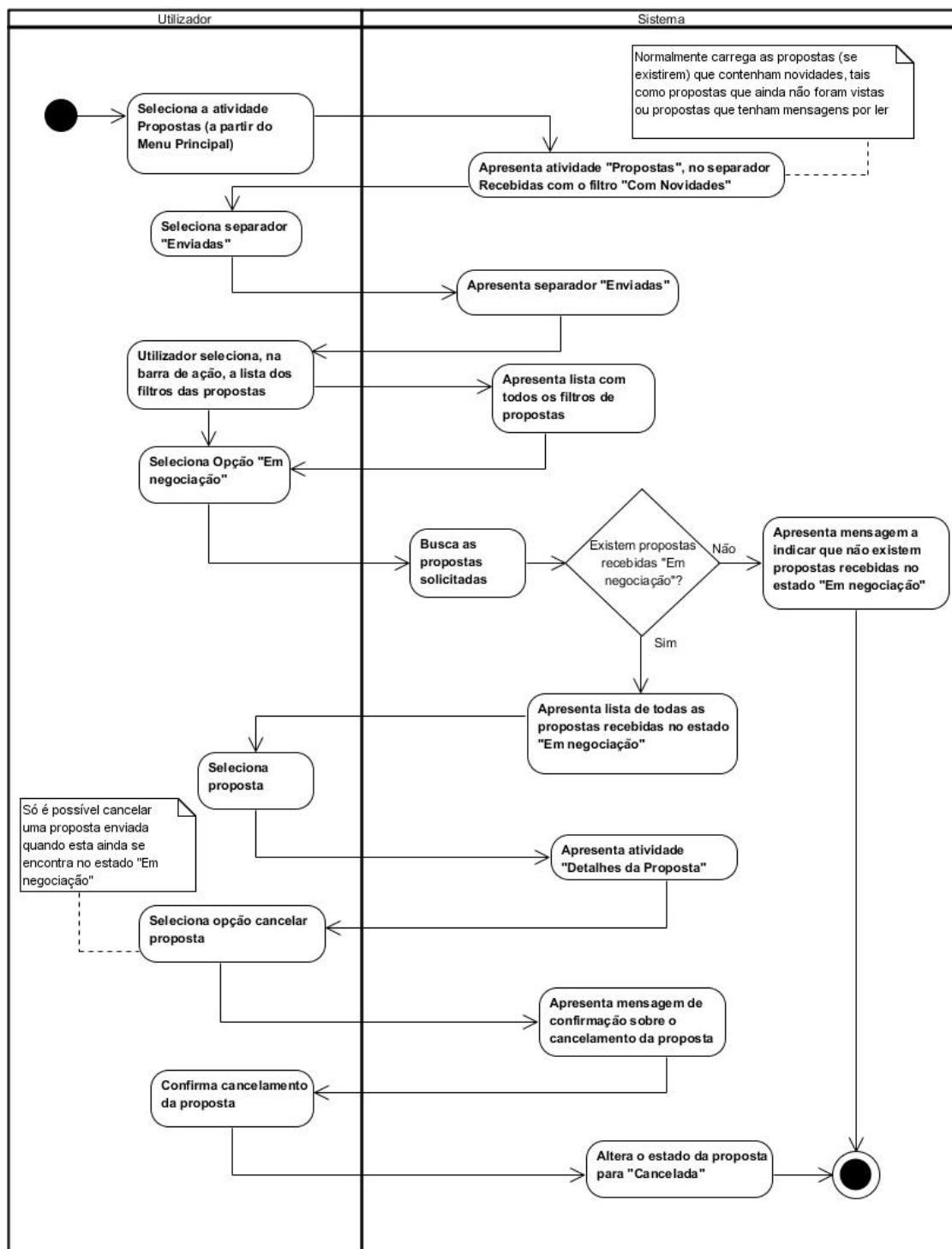


Figura 73 - Diagrama de atividade para o caso "Cancelar Proposta Enviada"

O caso **dar feedback** (avaliar) consiste em o utilizador escolher a opção “Propostas” a partir da atividade “Menu Principal”. O sistema abre a atividade “Propostas”, no separador “Recebidas” e com o filtro “Com novidades” selecionado. O utilizador solicita

a lista de filtros, o sistema apresenta a lista de filtros e o utilizador escolhe o filtro “Esperando *Feedback*”. O sistema carrega todas as propostas que aguardam *feedback* (todas têm de estar no estado aceite). O utilizador seleciona uma proposta (enviada ou recebida), o sistema abre a atividade “Detalhes da Proposta” e apresenta a opção “Dar *Feedback*”. O utilizador seleciona essa opção e o sistema apresenta a atividade “Dar *Feedback*”. O utilizador preenche a avaliação geral (positiva, neutra ou negativa), escreve um comentário sobre como correu a troca com o outro utilizador, avalia numa escala de 0,5 a 5 sobre três pontos distintos em relação ao outro utilizador: nível de educação (respeito para com os outros), honestidade em relação ao estado de conservação do item oferecido e assiduidade/pontualidade. O utilizador após preencher estes dados tem de submetê-los para o sistema. O sistema abre um diálogo de confirmação, o utilizador confirma e o sistema grava o *feedback* fechando a atividade. Ao voltar à atividade “Detalhes da Proposta” a opção “Dar *Feedback*” é substituída por uma mensagem a indicar que o *feedback* já foi dado. Este é apresentado no diagrama da Figura 74.



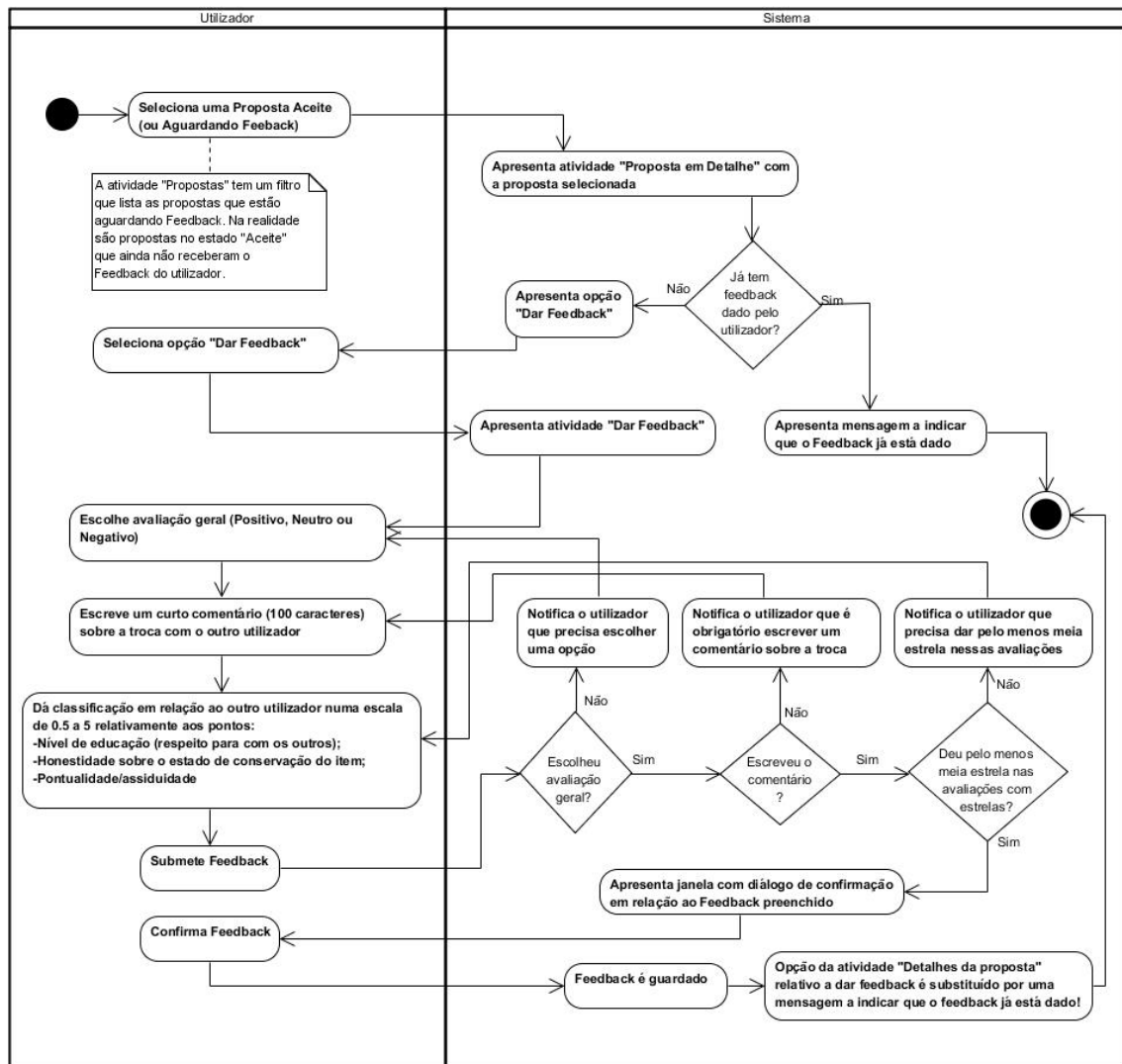


Figura 74 - Diagrama de atividade para o caso "Dar Feedback (Avaliar)"

O caso **enviar mensagens numa proposta** consiste em o utilizador escrever uma mensagem utilizando o formulário destinado a esse propósito, quando este se encontra na atividade "Detalhes da Proposta". Nesta atividade o sistema disponibiliza ao utilizador uma opção que permite colocar a sua morada no corpo da mensagem. Após o utilizador escrever a mensagem, apenas tem de selecionar a opção "Enviar". O sistema guarda a mensagem e notifica o utilizador de que esta foi enviada com sucesso. Este caso é apresentado no diagrama da Figura 75.

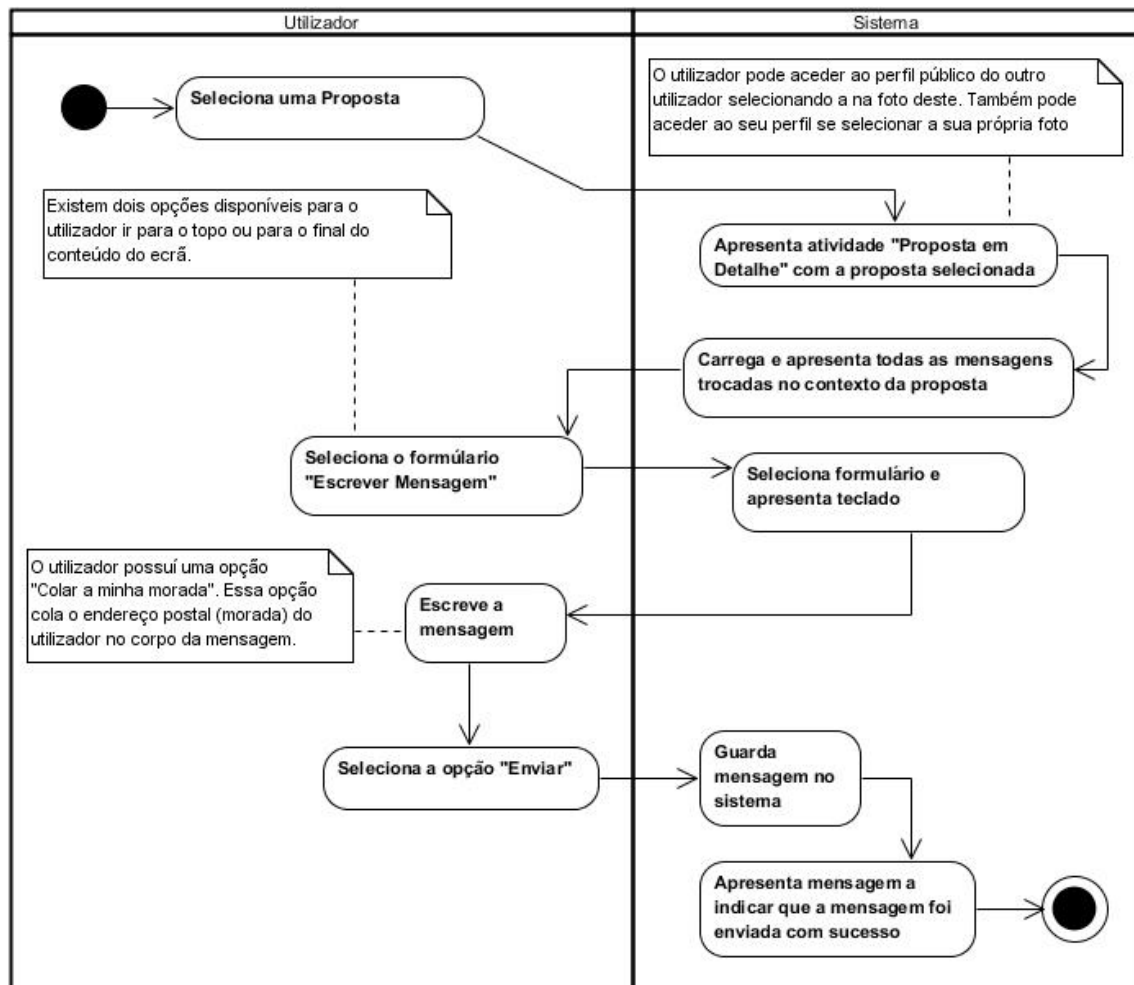


Figura 75 - Diagrama de atividade para o caso "Enviar Mensagem numa Proposta"

O caso **consultar pedidos de adição de itens** consiste em o administrador aceder ao *software* de gestão da base de dados, introduzir as credenciais de acesso e depois aceder aos registos da tabela "requests". Este caso é descrito no diagrama da Figura 76.

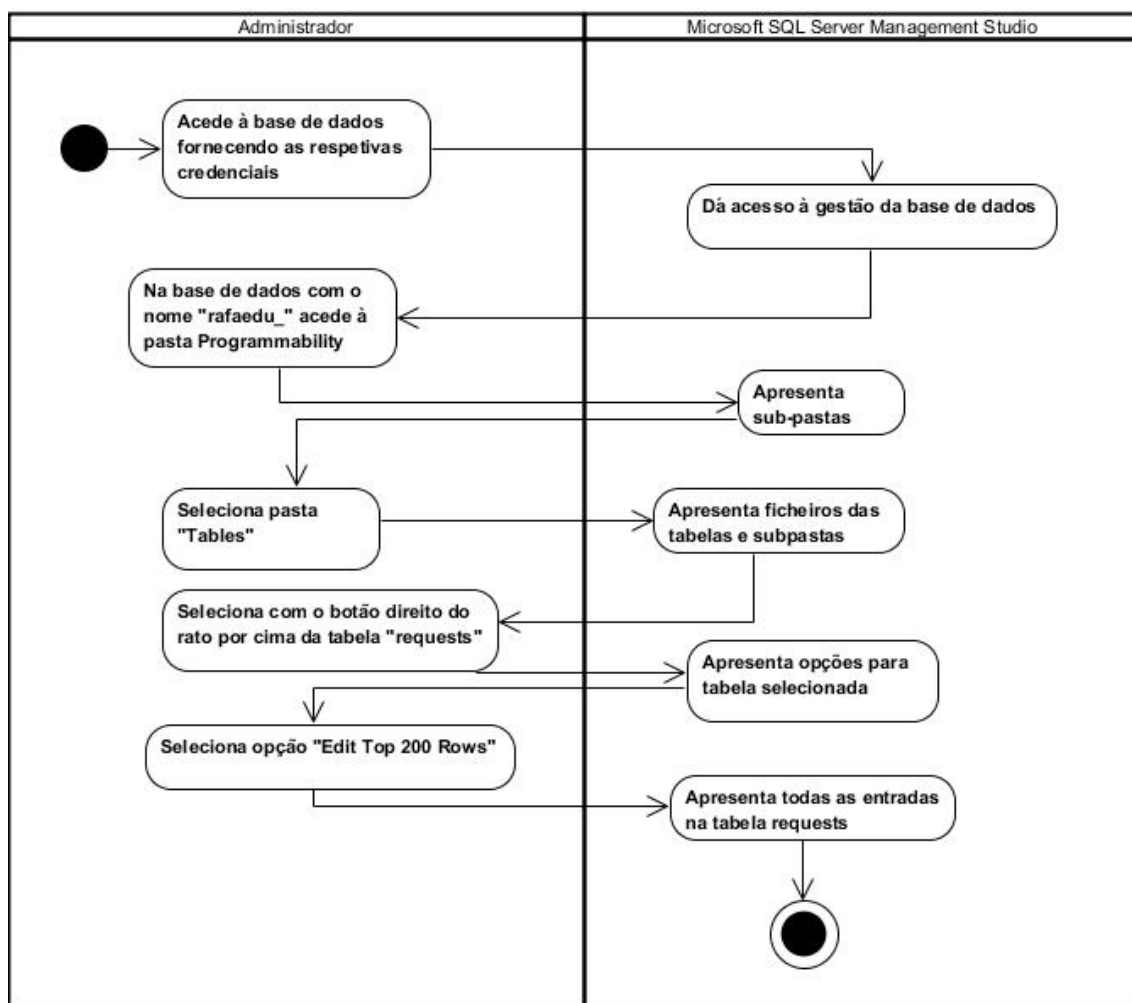


Figura 76 - Diagrama de atividade para o caso "Consultar Pedidos de Adição de Itens" (Administrador)

O caso **adicionar novo item** consiste em o administrador aceder à base de dados utilizando o respetivo *software*. Neste caso o administrador tem de pesquisar pela *Stored Procedure* com o nome “uspAddItem” e executá-la. Para executá-la tem de preencher vários campos relativos ao item tais como: título, género, publicador ou editor, categoria do item, etc. A *Stored Procedure* encarrega-se de verificar se os dados introduzidos já existem nas respetivas tabelas e se não existirem adiciona a respetiva informação. O item em si consiste na combinação de vários atributos de outras tabelas. Para melhor se entender o funcionamento desta podemos visualizar o diagrama da Figura 77.

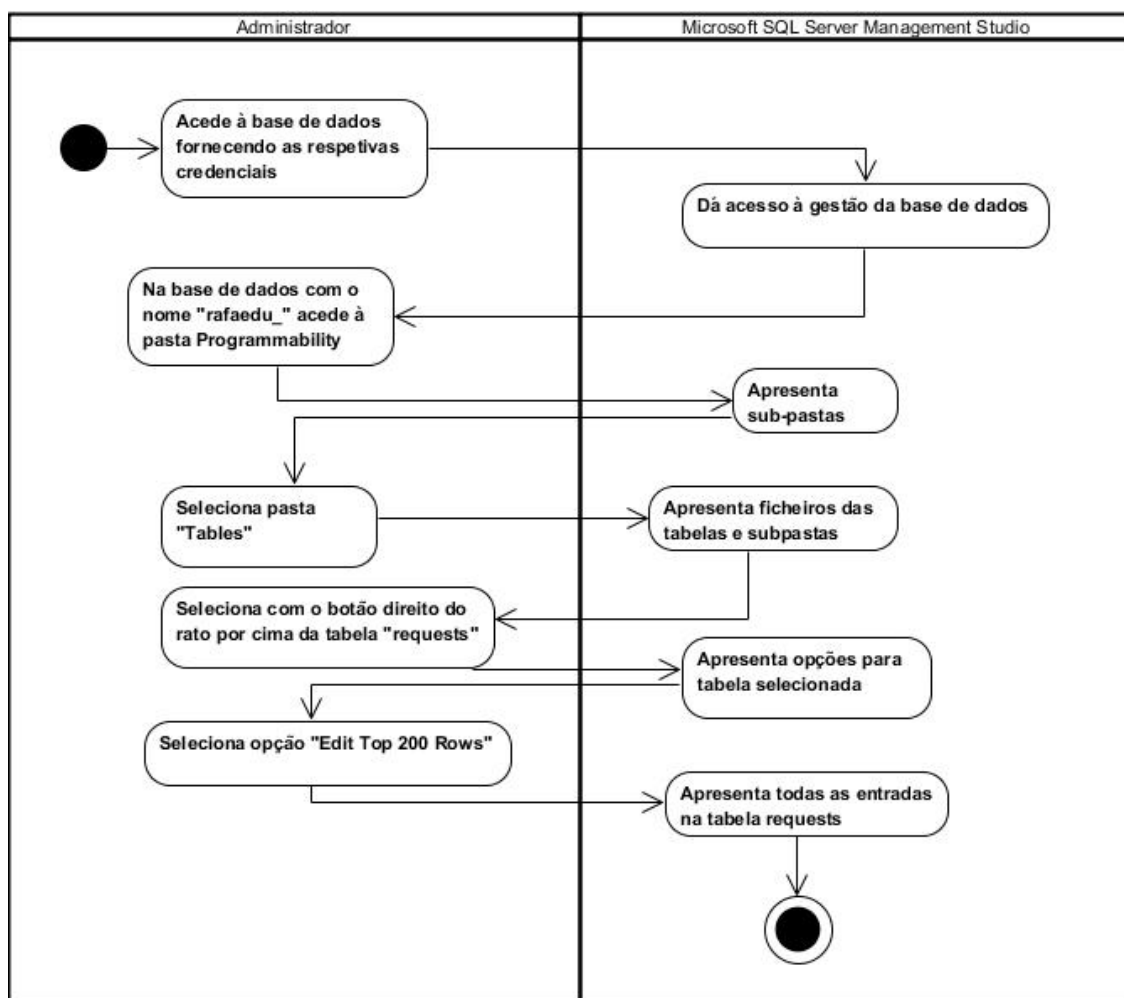


Figura 77 - Diagrama de atividade para o caso "Adicionar Novo Item" (Administrador)

Na Figura 78 está representado o caso de **verificação de novas propostas e mensagens**, que é um serviço. Este ocorre sem a interação do utilizador. Periodicamente, o serviço executa o procedimento descrito no diagrama referido, ou seja, este verifica se existem novas mensagens ou propostas que ainda não tenham sido notificadas. Se existirem, o sistema apresenta uma notificação ao utilizador. O utilizador ao selecioná-la é encaminhado para a atividade "Propostas" com o filtro "Com Novidades" selecionado.

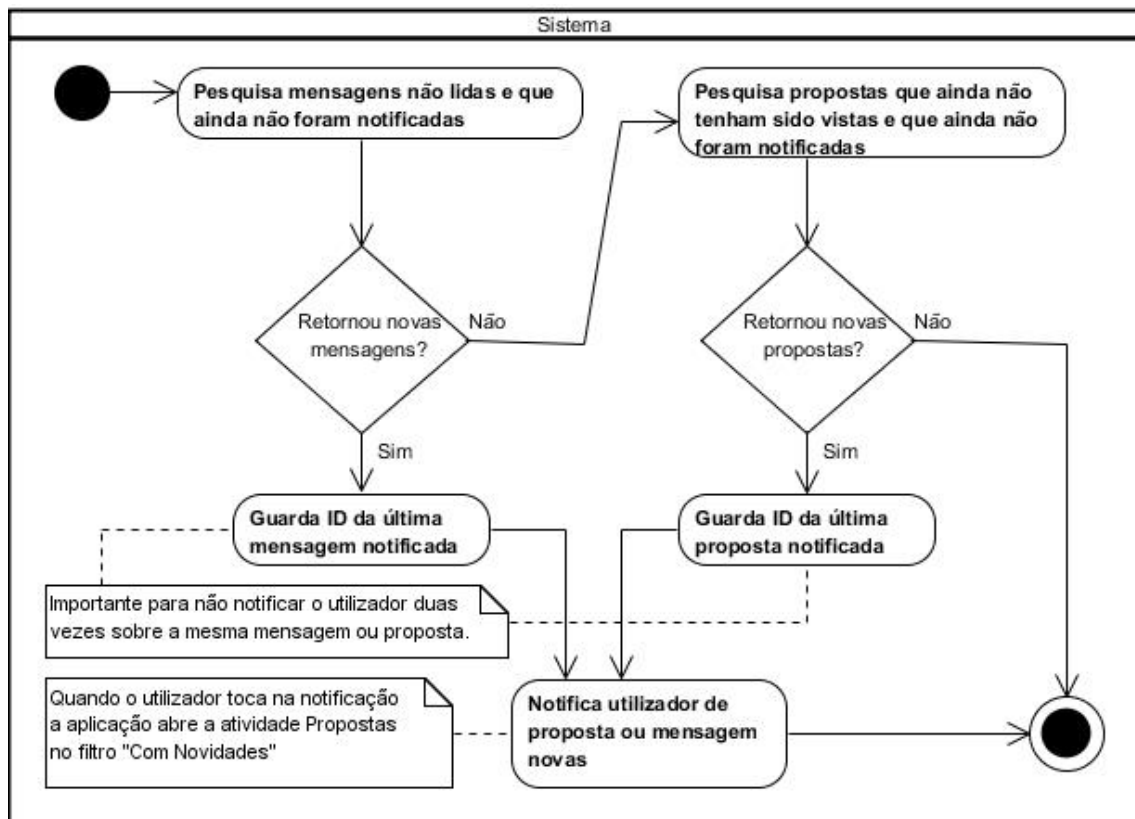
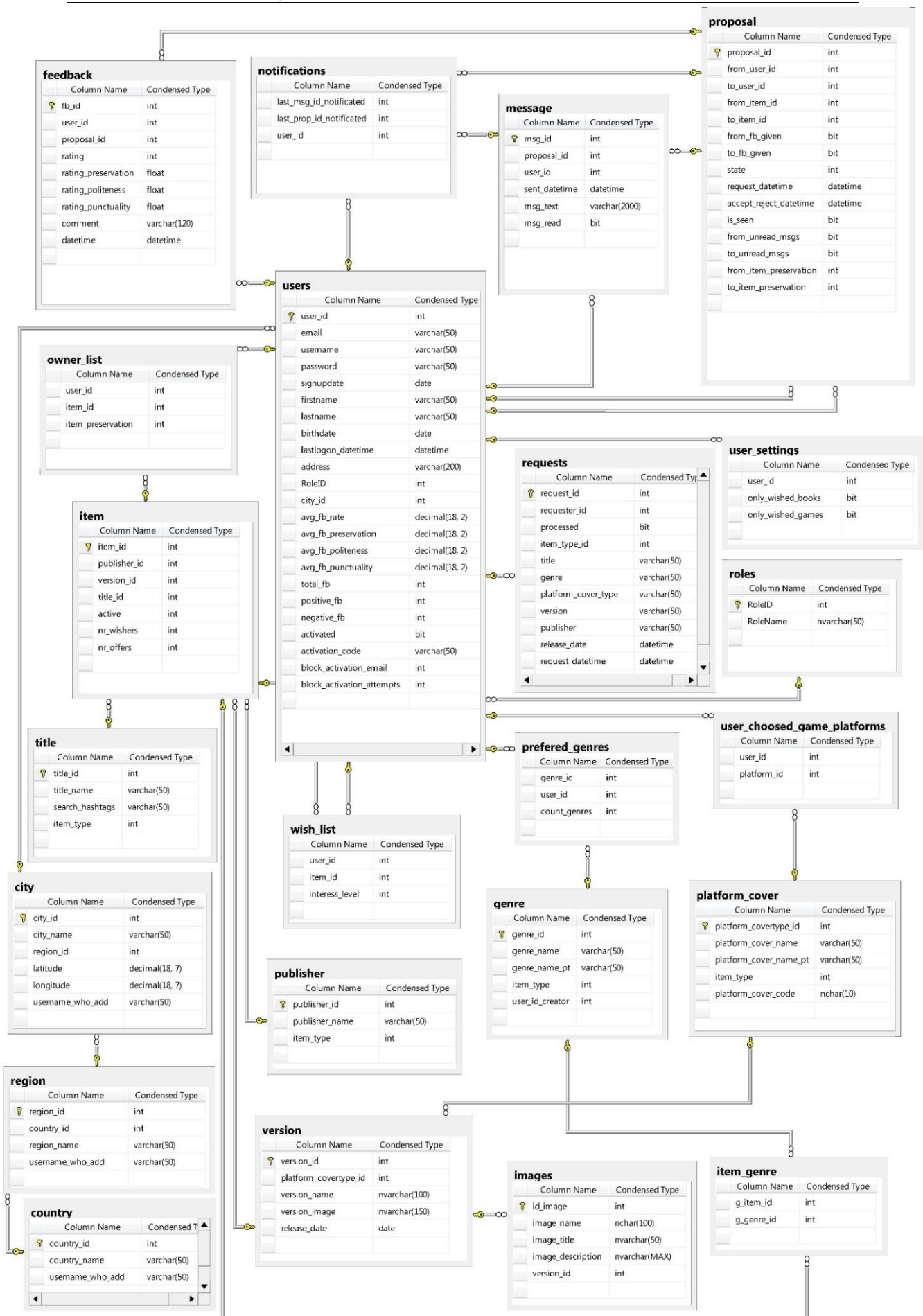


Figura 78 - Diagrama de atividade para o caso "Verificação de Novas Propostas e Mensagens" (Serviço)

## Anexo D – Explicação das entidades e atributos da base de dados



Observando a figura acima, é possível visualizar as entidades e os seus respetivos atributos (incluindo os seus tipos). De modo a explicar de forma pormenorizada cada entidade e cada atributo dessa entidade tentou-se seguir uma sequência lógica na apresentação das entidades e na explanação fornecida em seguida. Note-se que os tipos dos atributos não são referidos por serem explícitos na figura.

- Entidade *country* - A entidade *country* armazena nomes de países. Possui os seguintes atributos:
  - *country\_id* – Este é um identificador único (ID) que é atribuído automaticamente sempre que se adiciona um novo país à base de dados.
  - *country\_name* – Nome do país (na língua do mesmo).
  - *username\_who\_add* – Armazena o nome de utilizador do utilizador que adicionou o país

Exemplo 1:

<i>country_id</i>	<i>country_name</i>	<i>username_who_add</i>
1	Portugal	user1

- Entidade *region* – Esta entidade guarda o nome de regiões. Note-se que a região tem de pertencer a um país e o país ao qual ela pertence tem de ser adicionado antes desta (se não existir ainda).
  - *region\_id* – ID único atribuído quando é adicionada uma nova região para identifica-la inequivocamente.
  - *country\_id* – ID do país ao qual a região pertence. Saliente-se que tem de pertencer sempre a um país.
  - *region\_name* – Nome da região.
  - *username\_who\_add* – Armazena o nome de utilizador do utilizador que adicionou a região.

Exemplo 2:

region_id	country_id	region_name	username_who_add
2	1	Ilha da Madeira	user1

Neste exemplo é possível verificar que a região pertence ao país que possui o `country_id = 1`, neste caso, tendo em conta o exemplo 1 podemos concluir que a região “Ilha da Madeira” pertence ao País “Portugal”.

- Entidade `city` – Esta entidade como se pode deduzir armazena nomes de cidades e a sua localização em coordenadas GPS. Uma cidade tem de pertencer a uma região já existente.
  - `city_id` – ID único atribuído quando é adicionada uma nova cidade, é utilizado para identifica-la inequivocamente.
  - `city_name` – Nome da cidade.
  - `region_id` – ID da região ao qual a cidade pertence.
  - `latitude` – Coordenada GPS referente à latitude onde se localiza a cidade.
  - `longitude` – Coordenada GPS referente à longitude onde se localiza a cidade.
  - `username_who_add` – Armazena o nome de utilizador do utilizador que adicionou a cidade.
- Entidade `roles` – Esta armazena os tipos de utilizador que podem existir na aplicação. Cada utilizador irá ter níveis de privilégios diferentes.
  - `RoleID` - Análogo ao primeiro atributo das entidades anteriores.
  - `RoleName` – Nome dado ao tipo de utilizador (por exemplo: Administrador ou Utilizador Padrão).



- Entidade users – Guarda as informações dos utilizadores da aplicação.
  - user\_id – ID único atribuído quando é adicionado um novo utilizador, é utilizado para identifica-lo inequivocamente.
  - email – Endereço de correio eletrónico do utilizador.
  - username – Nome de utilizador.
  - password – Palavra-passe do utilizador.
  - signupdate – Data em que o utilizador se registou na aplicação.
  - firstname – Nome (verdadeiro) do utilizador.
  - lastname – Apelido (verdadeiro) do utilizador.
  - birthdate – Data de nascimento do utilizador.
  - lastlogon\_datetime – Data e hora do último acesso do utilizador à aplicação.
  - address – Armazena o endereço postal do utilizador.
  - RoleID – Guarda o ID do “papel” ou tipo do utilizador na aplicação (ver explicação da entidade role)
  - city\_id – Guarda o ID da cidade ao qual o utilizador pertence.
  - avg\_fb\_rate – Média da avaliação (*feedback*) global recebido pelo utilizador.
  - avg\_fb\_preservation – Média da avaliação (*feedback*) recebida referente à honestidade do utilizador em relação ao estado de conservação dos itens que ofereceu em troca.
  - avg\_fb\_politeness – Média da avaliação (*feedback*) recebida dos outros utilizadores com quem o utilizador realizou trocas em relação à sua educação.

- avg\_fb\_punctuality – Média da avaliação (*feedback*) relativo à sua pontualidade e/ou assiduidade nos compromissos que possa ter assumido na realização de uma troca com os outros utilizadores.
- total\_fb – Número total de avaliações (*feedbacks*) que o utilizador recebeu.
- positive\_fb – Número de avaliações (*feedbacks*) positivas.
- negative\_fb – Número de avaliações (*feedbacks*) negativos.
- activated – Indica estado de ativação da conta do utilizador.
- activation\_code – Contém código de ativação da conta de utilizador.
- block\_activation\_email – Contém o número de *e-mails* enviados para o *e-mail* do utilizador com o código de ativação.
- block\_activation\_attempts – Contém o número de vezes que o utilizador tentou ativar a sua conta.
- Entidade user\_settings – Armazena preferências do utilizador.
  - user\_id – ID do utilizador ao qual se referem as preferências.
  - only\_wished\_books – Preferência se o utilizador só aceita receber propostas com livros da sua lista de desejos.
  - only\_wished\_games – Preferência se o utilizador só aceita receber propostas com jogos da sua lista de desejos.
- Entidade user\_choosed\_game\_platforms – Guarda as plataformas de jogo preferidas pelo utilizador e tem como intuito não permitir que os outros utilizadores façam propostas com jogos de plataformas que não sejam desejadas. Permite várias entradas de um mesmo utilizador (para o caso de haver mais do que uma plataforma preferida). Se o utilizador não tiver plataformas preferidas registadas nesta entidade/tabela é assumido que este aceita propostas de jogos de todas as plataformas.
  - user\_id – Identifica o utilizador pelo seu ID

- platform\_id – Identifica a plataforma de jogo preferida pelo utilizador. Os jogos que não pertençam às plataformas preferidas pelo utilizador não podem ser oferecidos pelos outros utilizadores numa proposta.
- Entidade publisher – Neste momento esta guarda publicadores de jogos e editoras de livros.
  - publisher\_id – Identificador único do publicador (para poder ser referenciado).
  - publisher\_name – Nome do publicador ou editor (de acordo com a categoria do item).
  - item\_type – identifica se o publicador/editor lança itens da categoria jogos ou da categoria livros.
- Entidade platform\_cover – Esta contém as plataformas de jogo, no caso da categoria jogos ou o tipo de capas de livros, no caso da categoria livros. Desta forma utilizou-se apenas uma tabela para dois aspetos diferentes.
  - platform\_covertime\_id – Identificador da plataforma de jogo (no caso de itens da categoria jogos) ou do tipo de capa (no caso de itens da categoria livros)
  - platform\_cover\_name – Contém o nome da plataforma de jogo em inglês (Exemplo: Playstation 4) ou no caso dos livros, o nome do tipo de capa do livro (Exemplo: Soft Cover, Hard Cover).
  - platform\_cover\_name\_pt – No caso de itens pertencentes à categoria livros, este contém o nome do tipo de capa do livro em português. (Exemplo: capa mole, capa rígida).
  - item\_type – Identifica o tipo de categoria (jogos ou livros) a que pertence essa plataforma ou capa, conforme o caso.
  - platform\_cover\_code – Este possui siglas para referenciar mais facilmente as plataformas sem ser necessário lembrar o ID da mesma e sem ser

necessário escrever os nomes por extenso para identifica-las. Por exemplo, é utilizada a sigla PS4 para referenciar a plataforma de jogo Playstation 4.

- Entidade version – Esta possui informações específicas de uma versão de um item.
  - version\_id – Identificador único de uma entrada na entidade/tabela versão.
  - platform\_covertime\_id – Identifica a plataforma de jogo ou tipo de capa do livro.
  - version\_name – Nome ou número da versão (se existir).
  - version\_image – Contém URL com a localização da imagem da versão.
  - release\_date – Data de lançamento desta versão.
- Entidade title – Contém os títulos dos itens, quer sejam jogos ou livros.
  - title\_id – Identificador do título.
  - title\_name – Título do item (jogo ou livro).
  - search\_hashtags – Nomes ou siglas pelo qual o título é conhecido para otimizar as pesquisas por nome (Por exemplo a *sigla* GT6 refere-se ao título Gran Turismo 6).
  - item\_type – Indica em que categoria de itens esse título é utilizado (se é em jogos ou em livros).
- Entidade item – Esta reúne a informação de várias entidades de modo a formar um item. Note-se que não pode haver itens iguais. É considerado um item diferente aquele que não corresponder em um ou mais destes três atributos: publisher\_id, version\_id ou title\_id em relação aos que já existem.
  - item\_id – Identificador único do item.
  - publisher\_id – Identificador do publicador deste item (ver entidade publisher).
  - version\_id – Identificador da versão (ver entidade version).

- title\_id – Identificador do título do item (ver entidade title).
- active – Utilizado para indicar se o item deve ser mostrado aos utilizadores.
- nr\_wishers – Contém o número de interessados no item.
- nr\_offers – Contém o número de utilizadores que possuem o item.
- Entidade genre – Esta entidade possui os géneros dos itens.
  - genre\_id – Identificador do género.
  - genre\_name – Nome do género em inglês.
  - genre\_name\_pt – Nome do género em português.
  - item\_type – Indica em que categoria de itens esse género é utilizado (se é em jogos ou em livros).
  - user\_id\_creator – Indica o ID do utilizador que adicionou o género.
- Entidade item\_genre – Entidade que interliga os itens aos seus géneros. Tenha-se em conta que cada item pode ter mais de um género, daí a necessidade desta.
  - g\_item\_id – Identifica o ID do item a que o atributo seguinte se refere.
  - g\_genre\_id – Identifica um dos géneros do item. Para atribuir os vários géneros a um item são necessárias várias entradas nesta tabela relativas a esse item, todavia essas entradas deixam de ser necessárias se o item possuir apenas um género.
- Entidade preferred\_genres – Esta é utilizada para armazenar o número de vezes que o utilizador entrou no perfil de itens de determinado género. É útil para o funcionamento do sistema de recomendação.
  - genre\_id – Identifica o género (ver entidade genre).
  - user\_id – Identifica o utilizador.

- count\_genres – Guarda o número de vezes que o utilizador visitou itens do género especificado.
- Entidade wish\_list – Armazena as listas de desejos dos utilizadores.
  - user\_id – Identifica o utilizador.
  - item\_id – Identifica o item.
  - interest\_level – Indica o grau de interesse do utilizador pelo item.
- Entidade owner\_list – Armazena as listas de ofertas dos utilizadores.
  - user\_id – Identifica o utilizador.
  - item\_id – Identifica o item.
  - item\_preservation – Indica o estado de conservação do item do utilizador.
- Entidade proposal – Esta é a entidade responsável por armazenar as propostas efetuadas pelos utilizadores.
  - proposal\_id – Identifica a proposta.
  - from\_user\_id – ID do utilizador que fez a proposta.
  - to\_user\_id – ID do utilizador que recebeu a proposta.
  - from\_item\_id – ID do item que o utilizador que fez a proposta ofereceu nesta.
  - to\_item\_id – ID do item do utilizador que recebeu a proposta.
  - from\_fb\_given – Guarda o valor que indica se o *feedback* já foi dado pelo utilizador que enviou a proposta.
  - to\_fb\_given – Guarda o valor que indica se o *feedback* já foi dado pelo utilizador que recebeu a proposta.
  - state – Armazena o estado da mensagem: Aceite, Rejeitada, Cancelada, Em negociação.

- request\_datetime – Guarda a data e a hora que a proposta foi realizada.
- accept\_reject\_datetime – Regista a data em que a proposta foi aceite ou rejeitada.
- is\_seen – Armazena o valor que indica se a proposta já foi vista pelo utilizador que a recebeu.
- from\_unread\_msgs – Guarda o valor que indica se há novas mensagens enviadas pelo utilizador que enviou a proposta.
- to\_unread\_msgs – Guarda o valor que indica se há novas mensagens enviadas pelo utilizador que recebeu a proposta.
- from\_item\_preservation – Guarda o estado de conservação do item do utilizador que fez a proposta, no momento em que este fez a proposta.
- to\_item\_preservation – Guarda o estado de conservação do item do utilizador que recebeu a proposta, no momento em que recebeu a proposta.
- Entidade message – Destinada ao armazenamento de todas as mensagens trocadas no contexto das propostas.
  - msg\_id – Identifica as mensagens unicamente.
  - proposal\_id – Indica a que proposta esta mensagem está associada.
  - user\_id – Indica o id do utilizador que enviou a mensagem.
  - sent\_datetime – Regista a data e a hora do envio da mensagem.
  - msg\_text – Armazena a mensagem propriamente dita, o texto.
  - msg\_read – Valor destinado a indicar se a proposta já foi lida pelo outro utilizador da proposta.
- Entidade notifications – Auxilia o serviço de notificações da aplicação. Guarda para cada utilizador a última mensagem e a proposta que foi notificada a este.

- last\_msg\_id\_notificated – ID da última mensagem que o utilizador recebeu notificação.
- last\_prop\_id\_notificated – ID da última proposta que o utilizador recebeu notificação.
- user\_id – ID do utilizador a que se referem os atributos anteriores.
- Entidade *feedback* – Esta armazena os *feedbacks* dados pelos utilizadores.
  - fb\_id – Identifica o *feedback* inequivocamente.
  - user\_id – ID do utilizador que está a receber o *feedback*.
  - proposal\_id – ID da proposta a que o *feedback* está relacionado.
  - rating – Armazena a avaliação geral do *feedback*: Positivo, Neutro ou Negativo.
  - rating\_preservations – Guarda o valor da avaliação dada relativa à honestidade da descrição do estado de conservação do item.
  - rating\_politeness – Regista o valor da avaliação relativa ao nível de educação.
  - rating\_punctuality – Armazena o valor da avaliação da pontualidade e da assiduidade.
- Entidade requests – Esta armazena os pedidos de adição de itens feitos ao administrador.
  - request\_id – ID do pedido.
  - requester\_id – ID do utilizador que realizou o pedido.
  - processed – Indica se o pedido já foi atendido pelo administrador.
  - item\_type\_id – Indica o tipo de item que é suposto adicionar (jogo ou livro).



- title – Título do item.
  - genre – Género principal do item.
  - platform\_cover\_type – Tipo de capa (livros) ou plataforma (jogos).
  - version – Nome da versão.
  - publisher – Nome do editor (livros) ou publicador (jogos).
  - release\_date – Data de lançamento do item.
  - request\_datetime – Data e hora em que o pedido foi realizado.
- 
- Entidade images – Esta foi projetada para o futuro. Armazenará endereços de *screenshots* dos itens.
    - image\_id – Identificador da imagem.
    - image\_name – Nome/caminho para a imagem.
    - image\_title – Título da imagem.
    - image\_description – Descrição da imagem em texto.
    - version\_id – ID da versão do item a que a imagem pertence.

## **Anexo E – Cenários de utilização para teste de protótipos**

---

Neste anexo pode-se visualizar os cenários de utilização que os utilizadores de teste utilizaram para testar os protótipos. Note-se que se utilizou os mesmos cenários para os testes da interface de utilizador (*mockups*) e para os testes de usabilidade da aplicação final.

**Cenário 1:** Tu encontras-te a visualizar o Menu Principal da aplicação e pretendes navegar pelos jogos e realizar algumas ações que te vais ir lembrando no momento.

- Pretendes aceder à categoria Jogos.
- Tu possuis a consola Playstation 4 e queres filtrar a categoria jogos para jogos apenas da tua plataforma.
- Entretanto tu decides que queres pesquisar o jogo Need For Speed Rivals para a tua consola.
- Tu queres ver o perfil do jogo que pesquisaste.
- Agora queres ver a imagem da capa do jogo no ecrã inteiro.
- Decides que desejas assinalar este como um jogo que desejas obter.
- Entretanto pensas melhor e decides ver quem são os utilizadores que possuem o jogo que desejas.
- Decides que queres enviar uma proposta ao utilizador que se encontra mais próximo de ti.

**Cenário 2:** Tu encontras-te no Menu Principal da aplicação.

- Pretendes visualizar os livros mais oferecidos no momento.
- Entretanto verificas que por acaso possuis um desses livros e já não o queres, então decides adicioná-lo na tua lista de ofertas.

- Passado pouco tempo, ao tirares o livro da estante, este cai e rasga metade de uma página, contudo não afeta a sua leitura, pois colaste-a. Como és uma pessoa honesta, vais querer aceder à tua lista de ofertas e alterar o estado de conservação do teu livro para o novo estado de conservação.
- Entretanto apercebeste que na tua lista de ofertas possuis um outro livro que já não queres oferecer em troca porque ofereceste-o ao teu primo.
- Já agora, decides ver a tua lista de desejos dos livros e alteras o nível de desejo de um livro.

**Cenário 3:** Tu encontras-te no menu principal e entretanto recebes uma notificação de uma proposta nova. Vais negociá-la da melhor forma.

- Tu és notificado que recebeste uma proposta nova.
- Acedes aos detalhes dessa proposta.
- Trocas algumas mensagens com o outro utilizador.
- Chegas a um entendimento com ele e decides que vais aceitar a proposta de troca. Aceitas a proposta sem pensar duas vezes.
- Entretanto o utilizador vive ainda a alguma distância de ti e decidem que vão enviar os jogos por correio.
- Não necessitas de escrever a tua morada. Apenas escolhes a opção que cola a tua morada no corpo da mensagem que queres enviar. Já agora, não te esqueças de mencionar o teu nome completo antes de colar a morada (endereço postal).

**Cenário 4:** Estavas em casa, distraído a organizar a tua coleção de jogos. Entretanto vais à janela e reparas que os CTT acabaram de parar em frente à tua casa com uma encomenda do tamanho de um jogo. Aproximas-te, vês que a encomenda vem em teu nome e és solicitado a assinar o aviso de receção. Abres a encomenda e verificas que o jogo está um

pouco riscado, mas funciona. O utilizador que te enviou o jogo dizia que o jogo estava como novo.

- Ficaste um pouco desiludido e queres dar o *Feedback* a essa proposta.
- Acedes à respetiva proposta.
- Ainda assim dás *feedback* positivo e comentas que o nível de conservação do jogo não correspondeu ao descrito. Na avaliação da honestidade em relação ao estado de conservação do item vais dar apenas duas estrelas para indicar a tua insatisfação em relação a esse ponto.

**Cenário 5:** Fizeste uma troca em que enviaste o teu item por correio para o outro utilizador e vice-versa. Contudo a tua encomenda nunca chegou e foi retornada ao outro utilizador.

- Vais verificar se a tua morada (na aplicação) está correta e reparas que há um engano.
- Tendo em conta que usaste a opção de colar a morada no corpo da mensagem vais à tua conta colocar a tua morada correta!

## Anexo F – Pedidos da Aplicação Android e Ações Executadas na API

Tabela 5 - Tabela com as ações da API e respectivas Stored Procedures da base de dados:

Ação	Ação solicitada pela Aplicação Android	Ação API	Parâmetros recebidos pela API	Resultado API (JSON)	Stored Procedure executada na Base de Dados	Parâmetros da Stored Procedure
1	Lista de todos os usernames e e-mails registados para indicar ao utilizador se já existem	getAllUsernamesAndEmails	-	Listas de todos os usernames e e-mails registados na DB	uspGetAllUsernamesAndEmails	-
2	Obter lista de países	getCountries	-	Lista de Países	uspManageZones	@ActionID=1
3	Obter lista de regiões de um país	getRegions	country_id	Lista de regiões para o país fornecido	uspManageZones	@ActionID=1 @CountryID
4	Obter lista de cidades de uma região	getCities	region_id	Lista de cidades para a região fornecida	uspManageZones	@ActionID=1 @RegionID
5	Registo de Utilizador	signUp	username, password, email, firstname, lastname, birthdate, address, roleId, countryname, regionname, cityname	Número que indica sucesso ou insucesso (indicando respetivo erro)	uspUsersManagement	@ActionID=3, @User_Name, @User_Password, @User_Email, @User_FirstName, @User_LastName, @User_BirthDate, @User_Address, @User_RoleID, @User_CountryName,

						@User_RegionName, @User_CityName
6	Efetuar Login	getLogin	username password (enviado codificado* e no header)	Alguns Dados do Utilizador (se ele existir e os dados estiverem corretos)	uspUsersManagement	@ActionID=1 @User_Name @User_Password
7	Reenviar mensagem para e-mail com código de ativação	Activation	email	Retorna se foi possível enviar o e-mail através de um número	uspAccountActivation	@ActionID=1, @Email
8	Ativar conta de utilizador	Activation	email code	Retorna valor a indicar se foi ativado com sucesso ou não	uspAccountActivation	@ActionID=2, @Email, @Code
9	Recuperar Password (Envia mensagem para e- mail com a mesma)	recoverPassword	email	Retorna resultado da operação através de um valor numérico	uspRecoverPassword	@Email
10	Alterar dados de perfil	changeUserPerfil	username, current_pw, newAddress, newPassword, countryName, regionName, cityName	Retorna resultado da operação através de um valor numérico	uspChangeAccountData	@User_Name, @User_Password, @User_Address, @User_NewPassword, @User_CountryName, @User_RegionName, @User_CityName
11	Guardar Definições do Utilizador	saveUserSettings	userId, owb, owg, ps	Retorna resultado da operação através de um valor numérico	uspManageSettings	@ActionID = 2, @UserId, @OnlyWishedBooks,

						@OnlyWishedGames, @PlatformsSelected
12	Dados públicos de um utilizador	getUserPublicInfo	UserId	Retorna dados públicos do utilizador do Id recebido	uspUsersManagement	@ActionID=5, @UserId
13	Dados públicos de utilizador e atualização da data e hora do último acesso	getUserPublicInfo UpdateLogon	UserID	Retorna dados públicos do utilizador do Id recebido	uspUsersManagement	@ActionID=6, @UserId
14	Contador de géneros vizualizados	countGenres	username, password, itemId	Retorna resultado da operação através de um valor numérico	uspUserStatistics	@ActionID=1, @Username, @Password, @ItemId
15	Obter plataformas de jogo na língua do sistema	GetPlatformCover	ItemType, Language	Lista de Plataformas de Jogo	uspGetPlatforms	@ItemType, @Language
16	Obter lista de todos os jogos de uma plataforma	GetItemsByPlatform	PlatformCode ItemType	Lista de todos os jogos de uma determinada plataforma	uspGetItemsByPlatform	@PlatformCode @ItemType
17	Pesquisar itens por nome, categoria	GetItemsByName	ItemName ItemType	Devolve lista de itens solicitados	uspGetItemsByName	@ActionID = 1 @ItemName @ItemType
18	Pesquisar jogos por nome e plataforma	GetItemsByName	ItemName ItemType PlatformCode	Devolve lista de itens solicitados	uspGetItemsByName	@ActionID = 2 @ItemName @ItemType @PlatformCode

19	Lista de todos os títulos, géneros, publicadores, plataformas registados filtrados por categoria	GetItemNamesByType	itemType	Devolve todos os títulos de uma categoria	uspGetItemNamesByType	@ItemType
20	Pedido ao administrador para adicionar novo item	sendItemAddRequest	uid, type, title, genre, platform_cover_type, v, pub, rel_date	Retorna resultado da operação através de um valor numérico	uspSendRequest	@RequesterId, @ItemTypeId, @Title, @Genre, @PlatformCoverType, @Version, @Publisher, @ReleaseDate
21	Obter lista de géneros de uma categoria específica (Jogos ou Livros)	GetGenre	itemTypeId	Retorna lista de géneros de uma categoria de itens recebida no parâmetro itemTypeId	uspGetCategoryByType	@ItemType
22	Pegar lista de jogos pelo género e plataforma de jogo	GetItemsByGenre	genreId, itemType, platformCode	Retorna jogos pelo género e plataforma	uspGetItemsByGenre	@ActionID=2, @GenreID, @ItemType, @PlatformCode
23	Pegar lista de itens pelo seu género e categoria	GetItemsByGenre	genreId, itemType	Retorna itens pela categoria recebida e pelo género recebido	uspGetItemsByGenre	@ActionID=1, @GenreID, @ItemType
24	Lista de jogos recomendados filtrados por uma plataforma de jogo específica	getRecommendedItemsByPlatform	username, password, platformCode	Retorna lista de jogos recomendados para o utilizador	uspUserStatistics	@ActionID=4, @Username, @Password, @PlatformCode



25	Lista de itens recomendados para o utilizador por categoria (jogos/livros)	getRecommendedItems	username, password, itemType	Retorna lista de itens recomendados da categoria escolhida	uspUserStatistics	@ActionID=3, @Username, @Password, @ItemType
26	Pegar lista de itens mais desejados filtrados por categoria	GetTopWished	itemType	Retorna lista de itens mais desejados da categoria escolhida	uspGetTops	@ActionID=1, @ItemType
27	Pegar lista de itens mais desejados filtrados por categoria	GetTopWished	itemType, platformCode	Retorna lista de jogos mais desejados filtrados por plataforma de jogo	uspGetTops	@ActionID=3, @ItemType, @PlatformCode
28	Pegar lista de itens mais oferecidos filtrados por categoria	GetTopOffers	itemType	Retorna lista de itens mais oferecidos da categoria escolhida	uspGetTops	@ActionID=2, @ItemType
29	Pegar lista de jogos mais oferecidos filtrados por categoria e plataforma	GetTopOffers	itemType, platformCode	Retorna lista de jogos mais oferecidos filtrados por plataforma de jogo	uspGetTops	@ActionID=4, @ItemType, @PlatformCode
30	Pegar item pelo seu ID	getItem	ItemId	Devolve item identificado pelo ItemId	uspGetItem	@ActionID=1, @ItemId
31	Adicionar item à lista de ofertas	OfferslistAddItem	ItemId, UserId, PreservationLevel	Devolve resultado da operação (sucesso/erro)	uspOfferslistAddItem	@User_Id, @Item_Id, @Item_Preservation
32	Editar item da lista de ofertas	OfferslistEditItem	UserId, ItemId, PreservationLevel	Devolve resultado da operação (sucesso/erro)	uspOfferslistEditItem	@User_Id, @Item_Id, @Item_Preservation
33	Remover item da lista de ofertas	OfferslistRemoveItem	UserId, ItemId	Devolve resultado da operação (sucesso/erro)	uspOfferslistRemoveItem	@User_Id, @Item_Id

34	Apresenta lista de possíveis itens para oferecer em troca (tendo em conta as preferências do outro utilizador)	getUserOwnerList Proposal	FromUserId, ToUserId, ItemType, WishedItemId	Devolve lista de possíveis ofertas para uma proposta	uspOfferslistGetItemsBy Type	@ActionID=3, @User1_Id, @User2_Id, @ItemType, @Item_Id
35	Pegar dados de um item que pertence a um determinado utilizador	getUserOwnerList ByType	UserId, ItemType, ItemId	Devolve item de um determinado utilizador (com detalhes de conservação ou nível de desejo)	uspOfferslistGetItemsBy Type	@ActionID=1, @User1_Id, @ItemType, @Item_Id
36	Pegar lista de utilizadores que são proprietários de um item	getOwnersOfItem	username, password, ItemId	Devolve lista de utilizadores que possuem determinado item	uspGetItemOwners	@Item_Id, @Username, @Password
37	Pegar lista de ofertas de determinado utilizador filtrado por categoria	getUserOwnerList ByType	UserId, ItemType	Devolve lista de itens que pertencem à lista de ofertas do utilizador com determinado UserId	uspOfferslistGetItemsBy Type	@ActionID=2, @User1_Id, @ItemType
38	Adicionar item à lista de desejos	InsertWishedItem	ItemId, UserId, InterestLevel	Devolve resultado da operação (sucesso/erro)	uspWishlistAddItem	@User_Id, @Item_Id, @InterestLevel
39	Editar Item da Lista de Desejos	WishlistEditItem	UserId, ItemId, InterestLevel	Devolve resultado da operação (sucesso/erro)	uspWishlistEditItem	@User_Id, @Item_Id, @InterestLevel
40	Remover item da lista de desejos	WishlistRemoveItem	UserId, ItemId	Devolve resultado da operação (sucesso/erro)	uspWishlistRemoveItem	@User_Id, @Item_Id

41	Pegar lista utilizadores que desejam um determinado item	getWishersOfItem	username, password, ItemId	Devolve lista de utilizadores que desejam determinado item	uspGetItemWishers	@Item_Id, @Username, @Password
42	Pegar um item da lista de desejos de um determinado utilizador (permite ver nível de desejo)	getUserWishList ByType	UserId, ItemType, ItemId	Devolve um item da lista de desejos de um utilizador	uspWishlistGetItemsByType	@ActionID=3, @User_Id, @ItemType, @Item_Id
43	Pegar Lista de Desejos do utilizador filtrado por categoria	getUserWishList ByType	UserId, ItemType	Devolve lista de itens que pertencem à lista de desejos de um utilizador	uspWishlistGetItemsByType	@ActionID=2, @User_Id, @ItemType
44	Enviar proposta	sendProposal	FromUserId, ToUserId, FromItemId, ToItemId	Devolve resultado da operação (sucesso/erro)	uspSendProposal	@FromUserId, @ToUserId, @FromItemId, @ToItemId
45	Ver propostas	getProposals	UserId, State, ActionID (1 para propostas enviadas e 2 para propostas recebidas)	Devolve lista de propostas de acordo com estado e Id de utilizador e ação (enviadas/recebidas)	uspGetProposals	@ActionID, @UserId, @State
46	Pegar mensagens de uma proposta	getProposalMessages	id, username	Devolve mensagens de uma proposta	uspGetProposalMessages	@ProposalId, @Username
47	Enviar mensagem	SendMessage	text, pid, uid	Devolve resultado da operação (sucesso/erro)	uspSendMessage	@Text, @ProposalId, @UserId
48	Alterar estado de uma proposta	changeProposalState	proposalId, newState	Devolve resultado da operação (sucesso/erro)	uspChangeProposalState	@ProposalId, @NewState

49	Estado atual de uma proposta	getProposalState	proposalId	Devolve número correspondente ao estado atual de uma proposta	uspGetProposalState	@ProposalId
50	Enviar Feedback	sendFeedback	actionid, uid, pid, r, pre, pol, pun, comment	Devolve resultado da operação (sucesso/erro)	uspSendFeedback	@user_id, @proposal_id, @rating, @rating_preservation, @rating_politeness, @rating_punctuality, @comment, @action_id
51	Verificar se há notificações	checkForNotifications	username, password	Devolve resultado da operação (indicando se há novidades não notificadas ainda)	uspCheckNotification	@Username, @Password

## Anexo G – Inquérito de avaliação aos testes de usabilidade

---

O inquérito que se apresenta neste anexo foi usado pelos utilizadores que testaram a aplicação. Este inquérito teve também como finalidade recolher opiniões e de os utilizadores avaliarem a aplicação quanto à usabilidade. É importante mencionar que o presente inquérito foi criado com base no modelo de inquérito disponível online designado de “Questionnaire for User Interaction Satisfaction” (QUIS) de Gary Perlman [42].”

### Inquérito de Usabilidade XpressTrades (Protótipo Android App V1)

Este inquérito destina-se a avaliar o protótipo da aplicação XpressTrades desenvolvida para o Sistema Operativo Android. A ideia deste é apurar de forma quantificada o quanto os utilizadores estão satisfeitos com a aplicação.

Se ainda não transferiste a aplicação podes transferi-la clicando no link seguinte:

[goo.gl/qzKQFq](https://goo.gl/qzKQFq)

Antes de responderes a este inquérito por favor tenta realizar na App as ações descritas nos cenários de utilização apresentadas neste documento: [goo.gl/xxscDf](https://goo.gl/xxscDf)

**\*Obrigatório**

### Dados Pessoais

O seu nome não será divulgado.

Qual é o seu nome? \*

Primeiro e Último

Qual é a sua idade? \*

Qual o seu género? \*

☐ Masculino

☐ Feminino

## Experiência de Utilização

Perguntas sobre a experiência da utilização da aplicação XpressTrades.

### Opiniões sobre a usabilidade da Aplicação.

Escreve os problemas (defeitos, erros, etc) que tenhas encontrado na App!

Quando esta aplicação for lançada para o mercado Google Play achas que vais utilizar esta aplicação no teu dia-a-dia? \*

Quando a App for lançada toda a gente com Android poderá transferi-la e utiliza-la.

- ☐ Sim
- ☐ Não

Quais as funcionalidades que mais utilizaste na aplicação? \*

- ☐ Pesquisa de jogos ou livros por nome
- ☐ Pesquisa de jogos ou livros por categoria
- ☐ Visualizar itens recomendados
- ☐ Visualizar lista de Top Oferecidos
- ☐ Visualizar lista de Top Desejados
- ☐ Gerir lista de desejos e lista de ofertas (de jogos ou livros)
- ☐ Enviar propostas
- ☐ Filtrar jogos por plataforma
- ☐ Visualizar perfil de um item
- ☐ Visualizar lista de proprietários
- ☐ Visualizar lista de interessados
- ☐ Ver perfil dos outros utilizadores
- ☐ Visualizar perfil de outro utilizador
- ☐ Gerir Propostas (Aceitar/Rejeitar/Cancelar)
- ☐ Dar feedback (A propostas aceites)
- ☐ Outra:

Em que dispositivo te é mais prático utilizar a aplicação? \*

- ☐ Tablet
- ☐ Smartphone

### Que funcionalidades achas que deveriam ser adicionadas à aplicação?

### Que funcionalidades achas que deveriam ser retiradas da aplicação?

Se achas que se devem manter todas as funcionalidades não seleções nenhuma opção.

- ☐ Definições para filtrar as propostas que queres receber
- ☐ Receber notificações quando uma nova proposta ou mensagem é recebida
- ☐ Possibilidade de colar a tua morada numa mensagem para o outro utilizador envolvido na proposta
- ☐ Foto de perfil
- ☐ Visualizar lista de interessados em um item
- ☐ Visualizar lista de proprietários de um item
- ☐ Ver o nível de desejo dos interessados em um item
- ☐ Ver o nível de conservação de um item de um determinado proprietário
- ☐ Ver resumo do feedback de cada utilizador
- ☐ Ver número de interessados e número de proprietários de um item
- ☐ Visualizar capa do item em ecrã inteiro
- ☐ Outra:

### Classifica da Utilidade das Funcionalidades Atuais da Aplicação \*

	0 (Inútil)	1	2	3	4	5	6	7	8	9	10 (Muito Útil)
Pesquisar itens por nome	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Pesquisar itens por género	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Recomendação de itens com base no histórico de itens visualizados	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Apresentação dos itens mais desejados	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Apresentação dos itens mais oferecidos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Possuir e gerir listas de ofertas e de desejos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Visualizar número de interessados e proprietários	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ver resumo do feedback dos outros utilizadores	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sistema de feedback	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Enviar Propostas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Gerir Propostas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Enviar mensagens no contexto das propostas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Colar morada do utilizador no corpo de uma mensagem	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Visualizar cidade em que os outros utilizadores se encontram	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Ordenar utilizadores (proprietários e interessados) por distância em relação a ti	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Filtrar jogos pela plataforma	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Poder escolher que tipo de propostas queres receber (não permitir que os outros utilizadores façam propostas que não te interessam)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Alterar a tua imagem de perfil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Alterar a tua palavra-passe	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Alterar a tua cidade	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>



Alterar a tua morada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Visualizar o feedback que tens recebido sem especificar quem avaliou	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Apresentar o número de feedbacks que recebeste	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mostrar o perfil público dos outros utilizadores	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mostrar a última vez que um utilizador esteve online na App	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Apresentar idade dos outros utilizadores	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Apresentar data em que os outros utilizadores se tornaram membros da aplicação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

## Satisfação com Interface de Utilizador

Perguntas sobre a aplicação e a respetiva interface do utilizador.

### Reação Geral à Aplicação

Perguntas genéricas sobre a tua experiência com a aplicação.

	0	1	2	3	4	5	6	7	8	9	10
0=Terrível 10=Excelente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
0=Difícil 10=Fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
0=Frustrante 10=Satisfatória	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
0=Inadequada 10=Adequada	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
0=Aborrecida 10=Estimulante	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
0=Rígida 10=Flexível	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Layout de visualização

Relativo a aparência da interface de utilizador

	0	1	2	3	4	5	6	7	8	9	10
Leitura dos caracteres (0=Difícil 10=Fácil)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Animações nas tarefas (0=Pioram 10=Melhoram)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Negritos e Cores nos Layouts (0=Pioram 10=Melhoram)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Organização da Informação (0=Confusa 10=Objetiva)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Disposição dos Layouts (0=Confusa 10=Objetiva)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Terminologia e sistema de informação

Relativo à linguagem utilizada na aplicação.

	0	1	2	3	4	5	6	7	8	9	10
Termos utilizados no sistema (0=Inconsistentes 10=Consistentes)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Terminologia relacionada com a tarefa (0=Nunca relacionada 10=Sempre relacionada)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Posição das mensagens no ecrã (0=Inconsistente 10=Consistente)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Instruções para a introdução de dados (0=Confuso 10=Clara e Objectiva)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Apresentação do progresso durante uma tarefa (0=Nunca 10=Sempre)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mensagens de Erro (0=Dificultam 10=Auxiliam)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Aprendizagem

Esta parte avalia a facilidade de se aprender a utilizar a aplicação.

	0	1	2	3	4	5	6	7	8	9	10
Aprender a utilizar a aplicação (0=Difícil 10=Fácil)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Explorar as funcionalidades (0=Difícil 10=Fácil)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Lembrar dos nomes e comandos (0=Difícil 10=Fácil)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Mensagens de ajuda (0=Pioram 10=Ajudam)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

### Capacidades do Sistema

Esta parte serve para avaliar as capacidades do sistema.

	0	1	2	3	4	5	6	7	8	9	10
Velocidade da App (0=Lenta 10=Rápida)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Fiabilidade da App (0=Péssima 10=Fiável)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Robustez da App (0=Péssima 10=Robusta)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Formas análogas de executar uma mesma ação (0=Poucas 10=Muitas)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Desenhado para todos os tipos de utilizador (0=Nunca 10=Sempre)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>