

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
COORDENAÇÃO DE LICENCIATURA EM INFORMÁTICA
DESENVOLVIMENTO DE SISTEMAS PARA INTERNET E DISPOSITIVOS MÓVEIS**

MAURO ALBRECHT

**GASTOMETRO – DESENVOLVIMENTO DE UM APLICATIVO ANDROID PARA
AUXILIAR NAS COMPRAS DE SUPERMERCADOS**

MONOGRAFIA DE ESPECIALIZAÇÃO

FRANCISCO BELTRÃO

2014

MAURO ALBRECHT

**GASTOMETRO – DESENVOLVIMENTO DE UM APLICATIVO ANDROID
PARA AUXILIAR NAS COMPRAS DE SUPERMERCADOS**

Monografia de Especialização apresentada a
Coordenação de Licenciatura em Informática, da
Universidade Tecnológica Federal do Paraná
como requisito parcial para obtenção do título de
“Especialista em Desenvolvimento de Sistemas
para Internet e Dispositivos Móveis”.

Orientador: Prof. Robison Cris Brito

FRANCISCO BELTRÃO

2014



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Francisco Beltrão
Diretoria de Pesquisa e Pós-Graduação
Curso de Especialização em Desenvolvimento de
Sistemas para Internet e Dispositivos Móveis

TERMO DE APROVAÇÃO

Dia 02 do mês de outubro de 2014 às: 20:10 horas, na sala COLIN - Anfiteatro do Câmpus Francisco Beltrão, realizou-se a apresentação pública da monografia pelo estudante Mauro Albrecht, intitulada “Desenvolvimento de um aplicativo android para auxiliar nas compras de supermercados.” Finalizada a apresentação e arguição, a Banca Examinadora declarou **aprovada** a monografia do estudante, como requisito parcial para obtenção do título de Especialização em Desenvolvimento e Sistemas para Internet e Dispositivo Móveis.

Professor Robison Cris Brito - UTFPR
(Orientador)

Professor Vinicius Pegorini - UTFPR
(Convidado)

Professor Andrei Carniel- UTFPR
(Convidado)

Professor Dr. Ademir Roberto Freddo - UTFPR
(Coordenação)

A folha de aprovação com as rubricas encontram-se disponíveis na Diretoria de Pesquisa e Pós-Graduação,
UTFPR, Francisco Beltrão

AGRADECIMENTOS

A minha família, que sempre me incentivou e compreendeu que os momentos em que estive ausente eram devido a este trabalho.

Aos amigos Robson Buratti e Soeli dos Santos, que contribuíram com ideias que agregaram muito valor ao trabalho, e agradecer também o tempo que disponibilizaram para conversarmos.

A meu amigo Pedro Alencar Machado, pela ajuda nos diagramas de atividade e de caso de uso, e por incentivar nos momentos difíceis.

Ao colega de trabalho Ladair, que me ajudou muitas vezes com alguns problemas que enfrentei no decorrer do desenvolvimento do código fonte.

E por fim, ao orientador Robison Cris Brito, pela disponibilidade de tempo a me ajudar, por incentivar e motivar para que eu seguisse o desenvolvimento deste trabalho.

RESUMO

Atualmente os *smartphones* se tornaram uma tendência entre as pessoas. Suas funcionalidades vão muito além de apenas fazerem ligações, são aparelhos que permitem enviar e receber dados usando a rede da operadora ou a grande rede mundial de computadores – Internet. Um dos motivos desse sucesso está também na possibilidade de instalar aplicativos que auxiliam as pessoas no dia a dia, desde tarefas simples até tarefas mais complexas, isso tudo com a facilidade de poucos toques na tela do aparelho. Com o objetivo de que as pessoas possam economizar dinheiro e controlar seus gastos de uma maneira mais fácil, o presente trabalho propõe o desenvolvimento de um aplicativo móvel, baseado no sistema operacional Android, que possibilite criar listas para pesquisa de preços, onde o usuário pode selecionar os produtos e os locais para realizar a pesquisa. E ao final do lançamento dos preços para os respectivos locais e produtos, o usuário poderá utilizar uma opção de análise de dados, para ver qual produto tem o preço mais acessível entre vários locais. Recursos como lista de compras e conversor de moedas também estão presentes no aplicativo proposto. Como resultado, foi desenvolvido um aplicativo simples e intuitivo os quais estes são requisitos para aplicativos móveis.

Palavras-Chave: Aplicativo Móvel. Android. Pesquisa. Compras. Celular.

ABSTRACT

Currently smartphones become a trend among the people. Its functionality goes far beyond just making calls, are devices that allow you to send and receive data using the network operated or large global network of computers - Internet. One reason for this success is also the possibility to install applications that assist people in everyday life, from simple tasks to more complex tasks, all this with the ease of a few keystrokes on the handset screen. With the goal that people can save money and control your spending in an easier way, this paper proposes the development of a mobile application based on the Android operating system, which allows to create lists price search, where the user can select products and locations to conduct the research. And at the end of the release prices for their sites and products, users can use an option of data analysis, to see which product has the most affordable price across various locations. Features such as shopping list and currency converter are also present in the proposed application. As a result, we developed a simple and intuitive application which these are requirements for mobile applications.

Keywords: Mobile Application. Android. Search. Shopping. Cellular.

LISTA DE LISTAGENS

Listagem 1 – Exemplo de gravação de um objeto no banco	37
Listagem 2 – Classe mapeada usando ORMLite	39
Listagem 3 – Classe persistência de dados usando genéricos	39
Listagem 4 – Exemplo de manipulação de dados no banco de dados.....	40
Listagem 5 – Envio de SMS	41

LISTA DE FIGURAS

Figura 1 – Diagrama da arquitetura do Android.....	16
Figura 2 – Exemplo do aplicativo Supermarket.	18
Figura 3 – Exemplo da aplicação Boa Lista.....	18
Figura 4 – Esquema de ferramentas utilizadas	19
Figura 5 – Diagrama de casos de uso	24
Figura 6 – Diagrama de atividade para de controle de compra	25
Figura 7 – Diagrama de atividade para criação de uma pesquisa de preço	26
Figura 8 – Diagrama entidade relacionamento.....	27
Figura 9 – Estrutura do código fonte	28
Figura 10 – Samsung Galaxy Y.....	29
Figura 11 – Assistente de criação de layouts	30
Figura 12 – Tela de confirmação de instalação de aplicativos	30
Figura 13 – Tela principal	31
Figura 14 – Compra nacional	32
Figura 15 – Compra internacional	33
Figura 16 – Criação de uma lista de compra	33
Figura 17 – Criação de uma pesquisa de preços	34
Figura 18 – Menu de opções da pesquisa selecionada.....	35
Figura 19 – Lançamento de preços	35
Figura 20 – Tela análise de pesquisa	36
Figura 21 – Envio de SMS.....	37
Figura 22 – Emulador utilizado para testes	43
Figura 23– Testes em um smartphone modelo Samsung Galxy Y.....	44
Figura 24 – Resultado de uma pesquisa de preço	45

LISTA DE SIGLAS

ADT	Android Developer Tools
API	Application Programming Interface
DAO	Data Access Objects
DCU	Diagrama de Caso de Uso
DER	Diagrama Entidade Relacionamento
EPL	Eclipse Public License
IDE	Integrated Development Environment
SDK	Source Development Kit
SQL	Structured Query Language
SMS	Short Message Service
SPC	Serviço de Proteção ao Crédito
UML	Unified Modeling Language
XML	Extensible Markup Language

SUMÁRIO

1.	INTRODUÇÃO	11
1.1	CONSIDERAÇÕES INICIAIS	11
1.2	OBJETIVOS	12
1.2.1	Objetivos Gerais.....	12
1.2.2	Objetivos Específicos	12
1.3	JUSTIFICATIVA	13
1.4	ESTRUTURA DO TRABALHO.....	13
2.	REFERENCIAL TEÓRICO	15
2.1	SMARTPHONES E TABLETS	15
2.2	ANDROID.....	15
2.3	APLICATIVOS MÓVEIS PARA AUXÍLIO NAS COMPRAS.....	17
3.	MATERIAIS E MÉTODOS.....	19
3.1	MATERIAIS.....	19
3.1.1	Android SDK	19
3.1.2	SQLite	20
3.1.3	IDE Eclipse.....	20
3.1.4	Astah Community	21
3.1.5	ORMLite.....	21
3.2	MÉTODOS	22
4.	RESULTADOS	23
4.1	MODELAGEM DO SOFTWARE	23
4.1.1	Diagrama de Caso de Uso	23
4.1.2	Diagrama de Atividade	25
4.1.3	Diagrama Entidade Relacionamento.....	26
4.1.4	Estrutura do Código Fonte	28
4.2	DESENVOLVIMENTO DO APLICATIVO MÓVEL.....	29
4.2.1	Apresentação das funcionalidades	31
4.2.2	Persistência no Banco de Dados	37
4.2.3	Adapters.....	40
4.2.4	Envio de SMS	41
4.2.5	Testes	43
4.2.6	Implantação.....	44
5.	CONCLUSÃO.....	46
5.1	PROBLEMAS ENCONTRADOS	46
5.2	TRABALHOS FUTUROS	47
	REFERÊNCIAS	48

1. INTRODUÇÃO

Este capítulo contém as considerações iniciais do trabalho, as quais apresentam uma visão geral do assunto relacionado ao aplicativo proposto. Os objetivos explicitam as finalidades principais deste trabalho. A justificativa centra-se no tipo de aplicativo desenvolvido e nas tecnologias utilizadas. Ao final do capítulo, é apresentada a estrutura do trabalho, que apresenta como o texto está organizado.

1.1 CONSIDERAÇÕES INICIAIS

Atualmente os *smartphones* se tornaram uma tendência entre as pessoas. Suas funcionalidades vão muito além de apenas fazerem ligações, são aparelhos que permitem enviar e receber dados usando a rede da operado ou a grande rede mundial de computadores – Internet. Assim, é possível enviar fotos, vídeos, mensagens, usá-los para acessar a Internet, ver e-mails, acesso a redes sociais, notícias, entre outras informações.

Um dos motivos desse sucesso está também à possibilidade de instalar aplicativos que auxiliam as pessoas no dia a dia, desde tarefas simples até tarefas mais complexas, isso tudo com a facilidade de poucos toques na tela do aparelho, tornando estes dispositivos computadores portáteis.

Segundo o site Flipit (2014), uma pesquisa recente do SPC (Serviço de Proteção de Crédito) relevou que 34% dos brasileiros acabam gastando mais do que planejavam no supermercado todos os meses. E sugere algumas dicas para economizar, como criar listas de compras. Estas, não só evitam que você esqueça os itens de maior necessidade, como também ajudam a manter a disciplina. Fazer compras para o mês todo pode induzi-lo a exagerar, já que você tende a pensar que, como está comprando para um período prolongado, tem de levar mais do que realmente precisa. Outra dica importante é comprar toda semana, desta maneira você fica por dentro das promoções, e para aproveitar ao máximo estas, consulte os preços em mais de um supermercado.

Os *smartphones* se tornaram indispensáveis para o cotidiano das pessoas. São usados em qualquer lugar, em casa, no trabalho, restaurantes, trânsito, eventos sociais, cafeterias entre muitos outros lugares. Segundo o Our Mobile Planet (2013), a difusão desses aparelhos atinge 26% da população, e esses proprietários dependem cada vez

mais de seus dispositivos e muitos não saem de casa sem eles. Ainda segundo o Our Mobile Planet (2013), cerca de 82% dos consumidores já pesquisaram um produto ou serviço utilizando o *smartphones* antes da compra, as pesquisas em *smartphones* influenciam as decisões dos compradores. Desta forma, ter um site otimizado para celular é essencial. Além disso, é necessário ter uma estratégia para vários canais (*móBILE* e *web*) a fim de envolver os consumidores nos diversos caminhos até a compra.

Com o propósito de que as pessoas possam economizar dinheiro e controlar seus gastos de uma maneira mais fácil, o presente trabalho propõe o desenvolvimento de um aplicativo para dispositivo móvel, este, baseado no sistema operacional Android, que possibilite o lançamento de itens e valores para o local onde está realizando a compra, dessa forma, o usuário tem o controle sobre o que está comprando e quanto irá pagar, outras funcionalidades presentes no aplicativo são as lista de compra, lista para pesquisa de preços, além de conversão de moedas, para as compras internacionais.

Utilizando o aplicativo proposto, o usuário final tem diversos recursos ao seu alcance em um único aplicativo, que podem auxiliá-lo no dia a dia, isso tudo com a facilidade de poucos toques na tela do aparelho.

1.2 OBJETIVOS

A seguir serão apresentados os objetivos gerais e específicos do trabalho.

1.2.1 Objetivos Gerais

Desenvolver um aplicativo para dispositivo móvel que permita a criação de listas de compras, listas para pesquisas de preços, conversão de moedas e auxilie controle dos gastos em um supermercado.

1.2.2 Objetivos Específicos

Dentre os objetivos específicos do trabalho, destacam-se:

- Desenvolver o aplicativo proposto utilizando a plataforma Android.

- Utilizar o recurso de envio de SMS para envio de algumas informações geradas pelo aplicativo para outras pessoas.
- Disponibilizar o aplicativo para testes e coletar o depoimento dos usuários.

1.3 JUSTIFICATIVA

Atualmente existem algumas aplicações que desempenham a função de lista de compras, entretanto, a maioria destes possuem funcionalidades isoladas. Após pesquisas na loja de aplicativos da Google conhecida como Google Play (2014), nenhum aplicativo que reunisse todos os recursos propostos por este trabalho foi encontrado.

Na Google Play, também foi encontrado alguns aplicativos para comparação de preços, mas para lojas on-line, e no caso do aplicativo proposto a comparação de preços é realizada através de uma pesquisa, onde o usuário vai a campo lançar os valores dos produtos para os seus respectivos locais.

Um aplicativo como o proposto, poderia contribuir muito para seu utilizador, auxiliando no controle de gastos e economia de dinheiro através das pesquisas de preços. Também auxilia nas tarefas mais simples como lista de compras, isso tudo centralizado em um único aplicativo, outra vantagem é a conversão de moedas, no caso de uma compra internacional (nos supermercados da Argentina ou do Paraguai, por exemplo) será possível saber quanto foi gasto em determinado local, em reais.

1.4 ESTRUTURA DO TRABALHO

O trabalho está dividido em 5 capítulos. Destes, o primeiro apresenta a introdução, objetivos e justificativa para a realização deste trabalho.

O Capítulo 2 apresenta uma contextualização teórica sobre os conceitos e características do sistema operacional Android, estes utilizados para o desenvolvimento do aplicativo móvel.

O Capítulo 3 apresenta os materiais e métodos utilizados para a realização deste trabalho.

O Capítulo 4 apresenta os resultados obtidos, que se resume ao aplicativo móvel, e seus diversos recursos.

Finalizando, no capítulo 5 são apresentadas as conclusões, dificuldades encontradas e as sugestões de trabalhos futuros.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta o referencial teórico do presente trabalho. Será apresentado o mercado de dispositivos móveis (*smartphones* e *tablets*), explicado as características do sistema Android e também conceitos básicos de seu funcionamento, e por fim, exemplos de aplicações similares ao software desenvolvido.

2.1 SMARTPHONES E TABLETS

O mercado de *smartphones* e *tablets* está em constante evolução. Surge a cada ano uma grande variedade de novos dispositivos móveis, o que permite aos usuários escolherem qual aparelho é necessário para satisfazer seus desejos e necessidades, assim os usuários começam a substituir o computador por estes novos dispositivos para tarefas diárias, como ler email, acessar rede social, ler *e-books*, etc.

Segundo o grupo de pesquisas Gartner (2014) o número de *smartphones* e *tablets* vendidos com o sistema operacional Android ultrapassará um bilhão este ano, um crescimento de vinte e seis por cento em comparação ao ano de 2013. No geral espera-se que as vendas globais de todos os dispositivos cheguem a 2,48 bilhões de unidades em 2014, um crescimento de 7,6 por cento em relação a 2013.

Ainda segundo Gartner (2014) o mercado de *tablets* no mundo todo deverá ter um crescimento de 47%. Esse aumento ocorre devido à baixa nos preços. Os consumidores continuam comprando *tablets* como um dispositivo adicional que eles levam para toda parte. No mercado de sistemas operacionais para dispositivos o Android continua a ser o mais utilizado tanto em *tablets* quanto em *smartphones*.

2.2 ANDROID

O Google adquiriu a “Android Inc” em julho de 2005. Esta era uma pequena empresa localizada na Califórnia (Estados Unidos), que estava desenvolvendo uma plataforma para dispositivos móveis baseado em Linux, com o principal objetivo de ser uma plataforma flexível e de código aberto. Em 5 de novembro de 2007, o Google anuncia

oficialmente o desenvolvimento do sistema operacional Android, até essa data a Android Inc, manteve em segredo que o Android seria 100% livre.

Apesar de ser desenvolvido inicialmente para *smartphones*, a plataforma Android hoje é usada em diversos aparelhos, como *tablets*, *notebooks* e até relógios. Este sistema operacional está sendo mantido pelo Open Handset Alliance, que é um grupo formado por mais de trinta empresas que se uniram para inovar e acelerar o desenvolvendo de aplicações, proporcionando aos consumidores uma experiência melhor em termos de recursos e custos, pois o Android é a primeira plataforma móvel completa, aberta e livre.

A arquitetura do Android é dividida em camadas: Applications, Application Framework, Libraries e Runtime e Linux Kernel. A Figura 1 mostra todas as camadas, e na sequencia será explicada cada camada com mais detalhes.

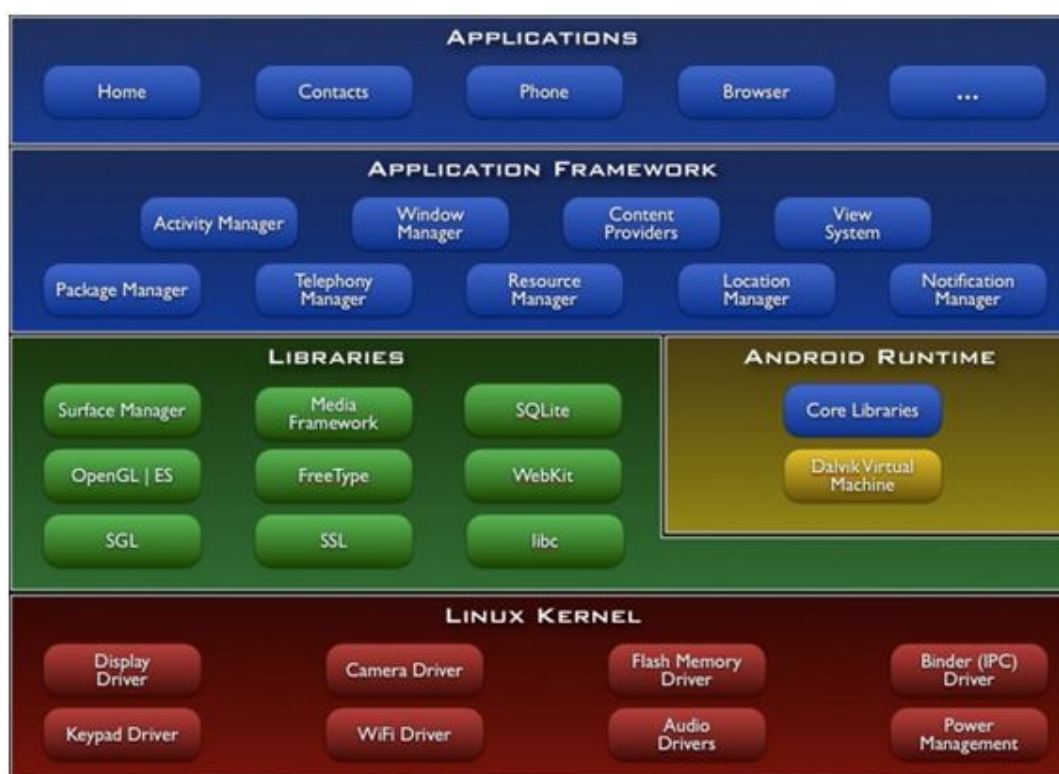


Figura 1 – Diagrama da arquitetura do Android
Fonte: ANDROID DEVELOPER, 2014.

A seguir será apresentada de forma detalhada cada camada do sistema operacional Android, essas informações foram retiradas do Android Developer (2014).

- *Linux Kernel:* é a camada responsável por serviços de mais baixo nível da plataforma, como por exemplo, gerenciamento de memória, gerenciamento de

processos, segurança, rede entre outros. Basicamente esta camada é o núcleo do Android.

- *Libraries*: nesta camada estão as bibliotecas nativas escritas em C e C++, que fazem parte da plataforma, nesta camada encontra-se APIs como o OpenGL ES, que é utilizada para renderização 3D, SQLite que possui recursos de banco de dados, esta camada ainda possui suporte a diversos formatos de áudio e vídeo.
- *Runtime*: é composta pela máquina virtual chamada Dalvik VM, onde as aplicações são executadas, e através de um conjunto de bibliotecas fornecem boa parte das funcionalidades encontradas nas bibliotecas padrão do Java.
- *Application Framework*: esta camada fornece todas as funcionalidades necessárias para a construção de aplicativos, através das bibliotecas nativas. Gerenciadores de serviços de telefonia, notificação e localização são alguns exemplos do que esta camada disponibiliza.
- *Application*: representa as aplicações que executam sobre a plataforma, nela está localizada uma lista de aplicações padrões que incluem cliente de e-mail, programa de SMS, calendário, mapas, navegador entre outros que são desenvolvidos pela comunidade, sendo que todas essas aplicações são escritas na linguagem Java.

2.3 APLICATIVOS MÓVEIS PARA AUXÍLIO NAS COMPRAS

Com relação aos aplicativos que utilizam ou possibilitem criação de listas de compras, bem como o gerenciamento dos gastos realizados durante uma compra, existem alguns aplicativos que são populares entre os usuários da loja de aplicativos da Google – Google Play, onde citam-se o Supermarket e Boa Lista.

O aplicativo Supermarket (2014), criado pela empresa Dukitan Software & Games permite criar listas de compra e controlar o que está comprando e quanto irá pagar. É um aplicativo simples e intuitivo, e a tela principal deste é apresentada na Figura 2.



Figura 2 – Exemplo do aplicativo Supermarket.
Fonte: SUPERMARKET, 2014.

O aplicativo o BoaLista, desenvolvido pela Dot Legend, permite criar listas de compras de modo fácil e intuitivo e descobrir onde realizar a compra pelo melhor preço, para isso o aplicativo consulta uma base de dados on-line, que é alimentada conforme os usuários utilizam o aplicativo.

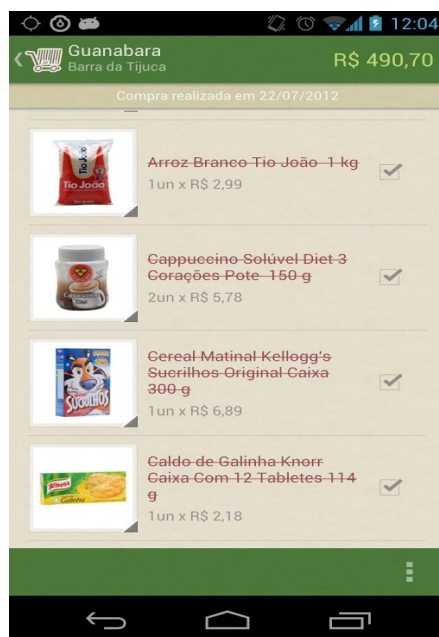


Figura 3 – Exemplo da aplicação Boa Lista.
Fonte: DOT LEGEND, 2014.

3. MATERIAIS E MÉTODOS

Este capítulo apresenta as ferramentas utilizadas para a elaboração do aplicativo para pesquisa de preços e controle de compra. Também serão apresentadas as metodologias utilizadas para atingir o resultado final.

3.1 MATERIAIS

Sabendo das necessidades e dos objetivos deste trabalho, foi possível escolher as ferramentas e tecnologias disponíveis no mercado para o desenvolvimento deste projeto. Na Figura 4 são apresentadas as ferramentas utilizadas para criação do aplicativo.



Figura 4 – Esquema de ferramentas utilizadas
Fonte: Autoria própria

3.1.1 Android SDK

Segundo Android SDK (2014), o kit de desenvolvimento Android padrão (Android SDK) fornece as bibliotecas e as ferramentas necessárias para desenvolver, testar e depurar aplicativos Android. Mesmo quem não é desenvolvedor pode testar o Android SDK, porque é fácil de configurar e executar. A principal linguagem de programação para

desenvolvimento de aplicativos para esta plataforma é a linguagem Java, sendo esta muito utilizada no mercado e nas instituições de ensino.

O SDK pode ser obtido Android SDK (2014) e distribuído sob a licença Apache 2.0. A versão utilizada neste trabalho foi a 22.2.1.

3.1.2 SQLite

O SQLite (2014) é uma biblioteca compacta escrita em linguagem C que implementa um banco de dados SQL embutido, desta forma ele é diferente da maioria dos bancos de dados SQL. O SQLite não tem um processo de servidor separado, na verdade ele lê e escreve arquivos diretamente em disco comuns. É muito utilizado no Android por não depender de instalação de servidores e não possui necessidade de configurações. É utilizado por grandes corporações como Mozilla e Adobe.

O SQLite oferece quase todas as funções de um banco de dados "convencional", como por exemplo, views, múltiplas tabelas, triggers, índices. Também é possível usá-lo em todas as plataformas sem nenhum problema, como em aplicações móveis, web e desktop.

Neste trabalho foi utilizada a versão 3.6.22, esta varia conforme a versão do SDK do Android. SQLite (2014) pode ser baixado e utilizado gratuitamente.

3.1.3 IDE Eclipse

IDE Eclipse (2014) é uma IDE gratuita, criada com a linguagem Java, assim, é multiplataforma. Permite aos desenvolvedores criar aplicações em várias linguagens como C++, PHP, Python e Java. É disponibilizada sob a licença EPL (Eclipse Public License) que é uma licença para software livre, utilizando esta é permitido usar, modificar, copiar e distribuir versões modificadas do software. Possibilita a integração com diversos plugins que acrescentam novas funcionalidades ao Eclipse.

A IDE Eclipse não traz suporte nativo ao desenvolvimento de aplicativos Android, por este motivo é necessária a instalação do Plugin ADT (2014), que fornece um ambiente de desenvolvimento de nível profissional para a construção de aplicativos Android. O plugin ADT amplia os recursos do Eclipse para que o programador possa rapidamente criar novos

projetos para Android, criar interfaces do aplicativo, depurar seus aplicativos e até mesmo exporta-los, a fim de distribuir a sua aplicação.

A versão do Eclipse utilizada no trabalho é a “4.2.1”, e pode ser baixada pelo site IDE Eclipse (2014). Já versão do plugin ADT é a “22.2.1” e pode ser baixado pelo site Plugin ADT (2014)

3.1.4 Astah Community

O Astah é uma ferramenta de diagramação UML, com ela é possível à criação de diagramas de caso de uso, diagramas de classe, além de outros diagramas, tais como diagrama de fluxo de dados, diagrama de entidade-relacionamento e outras funcionalidades úteis na fase de especificação e projeto de um sistema.

Segundo o Astah Community (2014), este ambiente permite criar diagramas rapidamente possuindo inúmeros recursos. Apesar de o Astah possuir versões pagas, o Astah Community é de uso gratuito.

Neste projeto, foi utilizada esta ferramenta para a criação do diagrama de caso de uso, diagrama de atividade e o diagrama de classe, a versão utilizada foi a “Astah Community 6.8.0”, que pode ser baixada Astah Community (2014).

3.1.5 ORMLite

ORMLite (2014) fornece algumas funcionalidades simples e leves para persistência de objetos Java em bancos de dados relacionais, evitando a complexidade. É fácil de usar, sendo que esta ferramenta gera automaticamente as tabelas no banco de dados, é flexível para construir consultas simples e complexas, assim como utiliza anotações Java para mapeamento dos objetos.

A utilização deste *framework* de persistência simplifica a criação da camada de persistência de dados do aplicativo proposto, em poucas linhas de código é possível armazenar no banco de dados um objeto, com as informações desejadas.

A versão utilizada neste projeto foi a “4.47”. Esta ferramenta está disponível para download em ORMLite (2014).

3.2 MÉTODOS

O desenvolvimento do aplicativo para controle de gastos e pesquisas de preço foi dividido em várias etapas, seguindo o modelo sequencial linear de Pressman (2006). Essas etapas são:

- a) **Análise:** O caso de uso descreve o cenário que mostra as funcionalidades do aplicativo, para o qual será proposto uma solução. Os casos de uso serão criados utilizando a ferramenta Astah Community, para produzir o diagrama UML.
- b) **Projeto:** Nesta fase serão criados alguns diagramas de atividade, que demonstram o fluxo básico do aplicativo. Também será criado um diagrama de classe utilizando a linguagem UML.
- c) **Desenvolvimento:** O desenvolvimento será realizado utilizando as tecnologias listadas na seção 3.1 - Materiais.
- d) **Teste:** Testes são necessários para garantir o funcionamento esperado do aplicativo criado. Será utilizado o emulador fornecido pelo SDK do Android juntamente com um aparelho real Samsung Galaxy Y.
- e) **Implantação:** Nesta fase será feita a entrega do aplicativo a o um estabelecimento comercial da cidade de Dois Vizinhos-PR para aprovar o funcionamento do mesmo.

4. RESULTADOS

Este capítulo apresenta uma visão geral da análise realizada para desenvolvimento do aplicativo, esta não será apresentada de forma detalhada, por não ter sido feito um levantamento de requisitos com usuários e sim conforme necessidades pessoais. Desta forma será apresentado o diagrama entidade relacionamento, diagrama de caso de uso, diagrama de atividade e a estrutura do código fonte desenvolvido.

Também será apresentado o aplicativo desenvolvido, focando nas telas e algumas de suas funcionalidades. E para finalizar, serão apresentados detalhes técnicos do desenvolvimento do aplicativo móvel.

4.1 MODELAGEM DO SOFTWARE

Nesta seção, são apresentados artefatos gerados que auxiliaram no desenvolvimento do aplicativo, são eles, diagramas de casos de uso e diagrama entidade relacionamento. Não é apresentado o diagrama de classe, devido à complexidade do projeto, sendo assim será apresentada a estrutura do código fonte gerado.

4.1.1 Diagrama de Caso de Uso

Na visão de Bezerra (2007, p. 54), “Um caso de uso representa um relato de uso de certa funcionalidade do sistema em questão, sem revelar a estrutura e o comportamento internos desse sistema”.

O diagrama de casos de uso (DCU) corresponde a uma visão externa do sistema e representa graficamente os atores, casos de uso e relacionamentos entre esses elementos. O diagrama de caso de uso tem o objetivo de ilustrar em um nível alto de abstração quais elementos externos interagem com que funcionalidades do sistema. (BEZERRA, 2007, p. 57)

Neste tópico é apresentado o diagrama de caso de uso do aplicativo proposto conforme Figura 5.

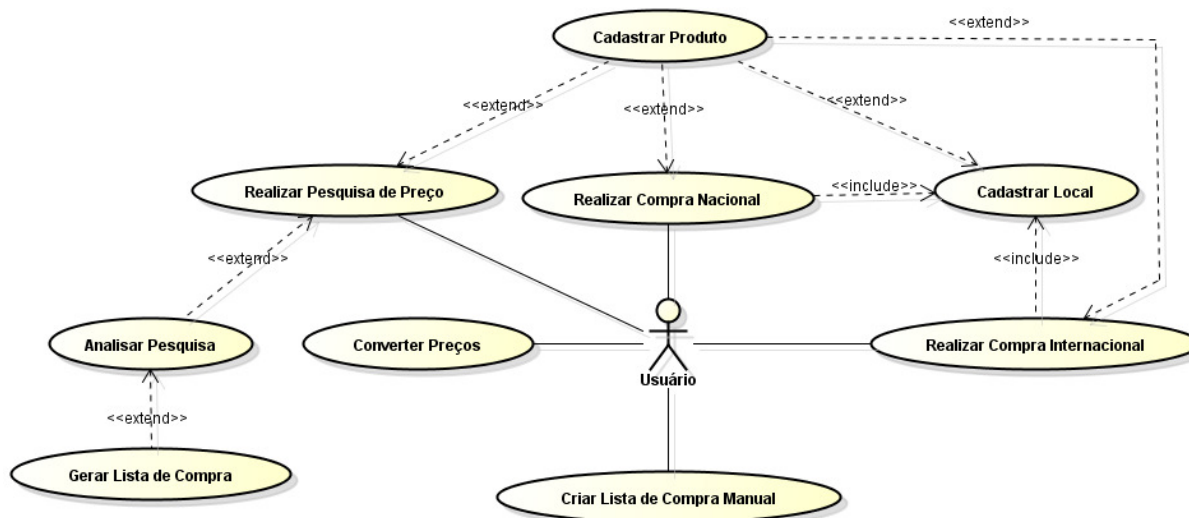


Figura 5 – Diagrama de casos de uso
Fonte: Autoria própria

O caso de uso apresenta as principais funcionalidades do aplicativo, a funcionalidade “Realizar Pesquisa de Preço”, compreende toda a parte de criação e manutenção de uma pesquisa de preço, esta funcionalidade permite o usuário selecionar os produtos e locais desejados para criar uma pesquisa, após a criação desta é possível realizar o lançamento dos preços para os respectivos locais, após o lançamento é possível analisar os dados, e a partir de uma análise é possível gerar uma lista de compra.

A funcionalidade “Analisar Pesquisa” é utilizada para verificar o resultado de uma pesquisa, nela é mostrado os produtos e o valor total dos itens para cada local pesquisado. Após o usuário verificar qual é o melhor local para realizar a compra ele pode utilizar a opção “Gerar Lista de Compra”, nessa opção é gerado uma lista de compra com os itens e preços lançados para o local selecionado.

As funcionalidades “Realizar Compra Nacional” e “Realizar Compra Internacional” tem o mesmo funcionamento, permitindo controlar o que está sendo comprado e quanto será pago no final da compra. O que diferencia uma da outra é que na opção “Realizar Compra Internacional” é possível informar a moeda (Dólar, Peso, Euro) e o valor cotado para esta no dia, com esses dados informados no aplicativo o valor dos itens serão convertidos para o Real.

Já na opção “Criar Lista de Compra Manual” é possível criar uma lista de compra informando item a item, e por fim a opção “Converter Preços” é utilizada para converter um valor para determinada moeda, deverá ser informado o valor do item e o valor da cotado para a moeda, e a conversão será feita em reais.

4.1.2 Diagrama de Atividade

Segundo Pressman (2007 p.737) “O diagrama de atividade mostra o comportamento de um sistema ou parte um sistema através do fluxo de controle entre ações que o sistema executa”.

Através do diagrama de atividade é possível observar como uma atividade depende uma da outra. Na Figura 6, é apresentado o diagrama de atividade para entendimento do processo de compras no aplicativo.

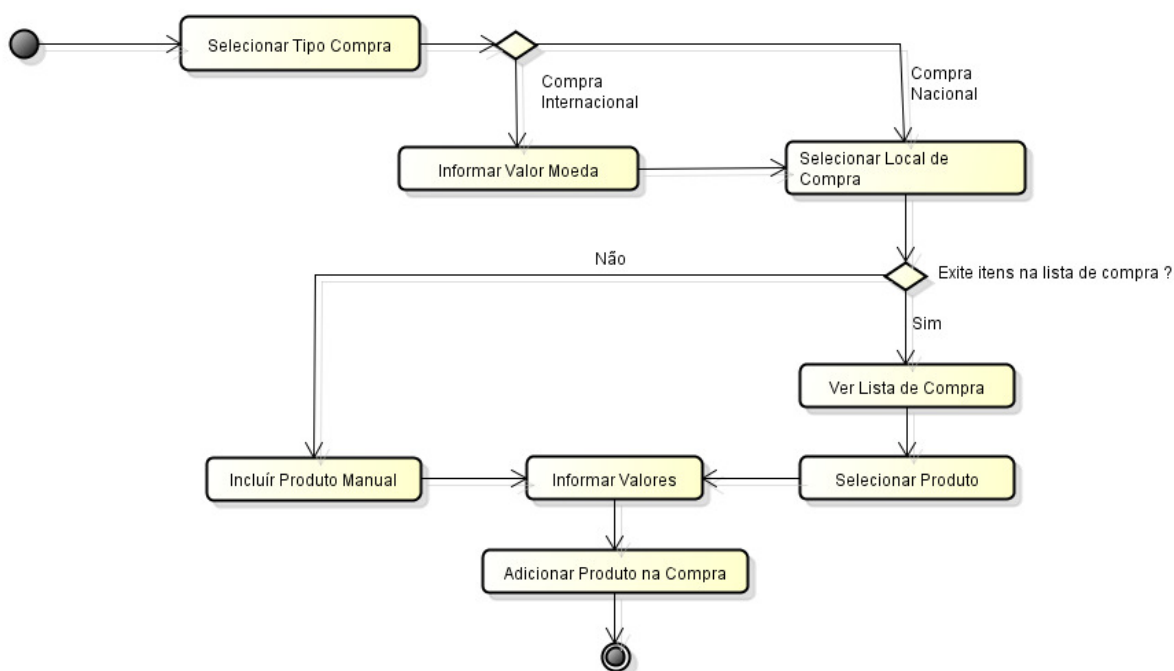


Figura 6 – Diagrama de atividade para de controle de compra
Fonte: Autoria própria

Na Figura 7 é apresentado o diagrama que representa o processo para a criação de uma pesquisa de preços.

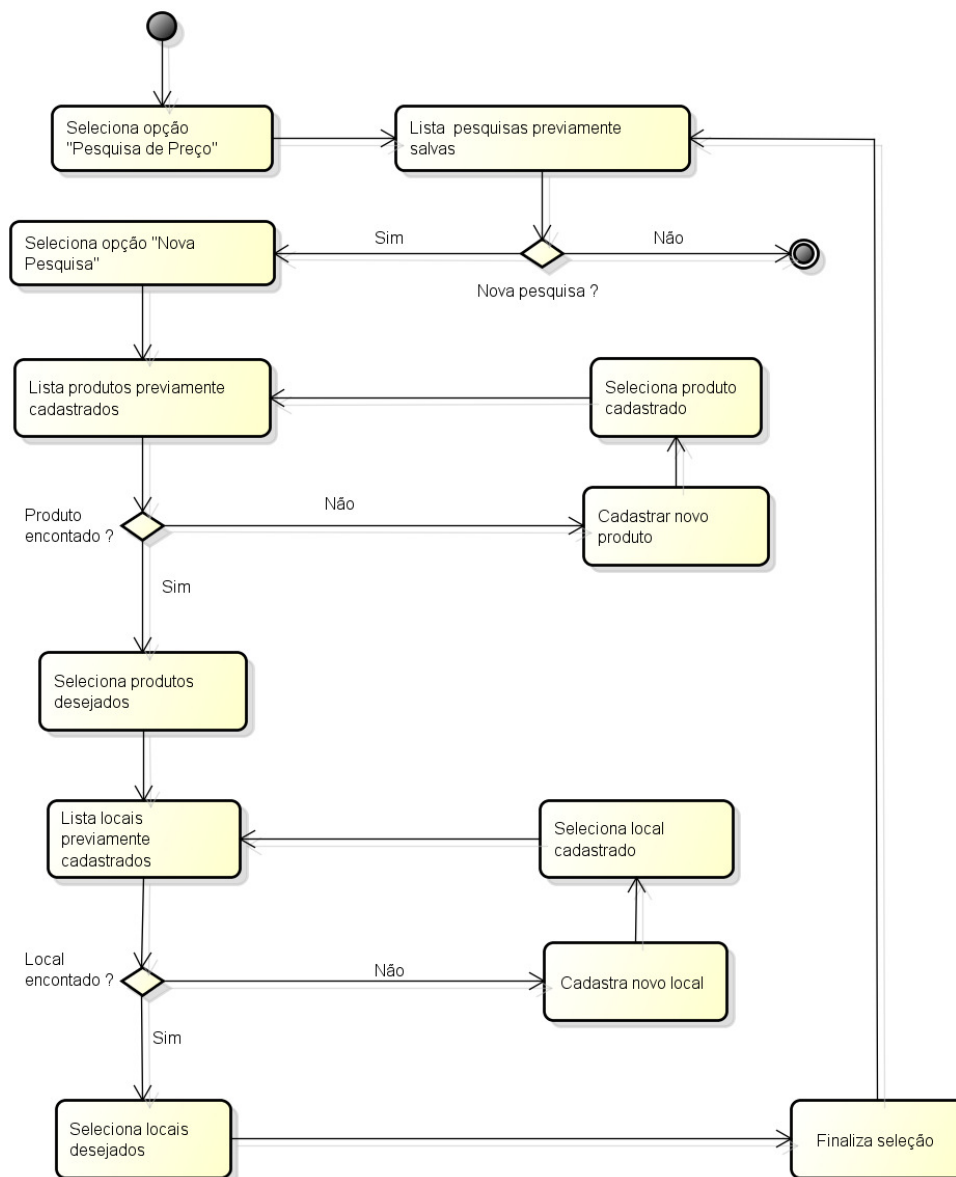


Figura 7 – Diagrama de atividade para criação de uma pesquisa de preço
Fonte: Autoria própria

4.1.3 Diagrama Entidade Relacionamento

De acordo com Rezende (2002, p.201) “O principal propósito do DER é representar os objetos de dados e suas relações, sendo que cada entidade termina representada pelo menos uma tabela de dado”.

O DER representa toda a estrutura lógica do banco de dados graficamente. A sua simplicidade e a clareza gráfica podem explicar, em grande parte, a ampla disseminação do uso do modelo E-R.

Na Figura 8 será apresentado o diagrama de entidade relacionamento do aplicativo proposto.

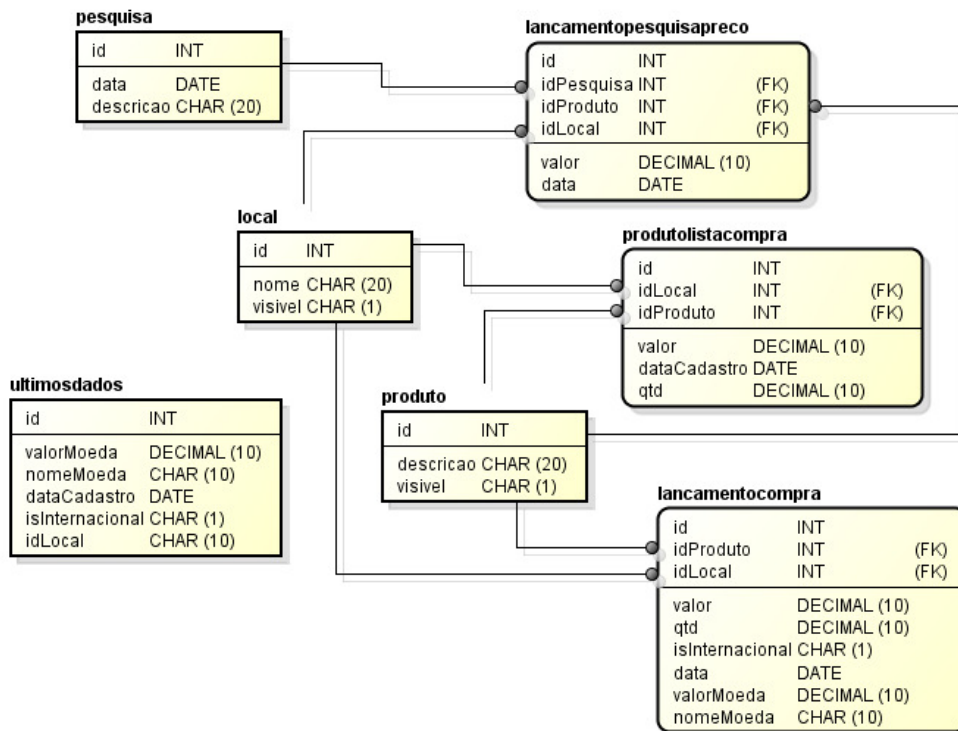


Figura 8 – Diagrama entidade relacionamento
Fonte: Autoria própria

No diagrama pode-se observar a tabela “produtolistacompra”, esta armazena os itens de uma lista de compra, já a tabela “lancamentocompra” armazena os itens que o usuário esta comprando em determinado local, desta forma é possível saber quanto ele já gastou nas compras, outra tabela importante do diagrama é a tabela “lancamentopesquisapreco”, que é utilizada em uma pesquisa de preço, esta tabela armazena os produtos com seus respectivos preços e locais.

A tabela “local” armazena os locais de compras e de pesquisa de preços, já a tabela “produto” grava os produtos utilizados no aplicativo. As demais tabelas são utilizadas para controle do aplicativo.

4.1.4 Estrutura do Código Fonte

Esta seção apresenta, no lugar do diagrama de classe, a estrutura de armazenamento de classes no aplicativo. O aplicativo possui 56 classes, o que demonstra a complexidade do mesmo.

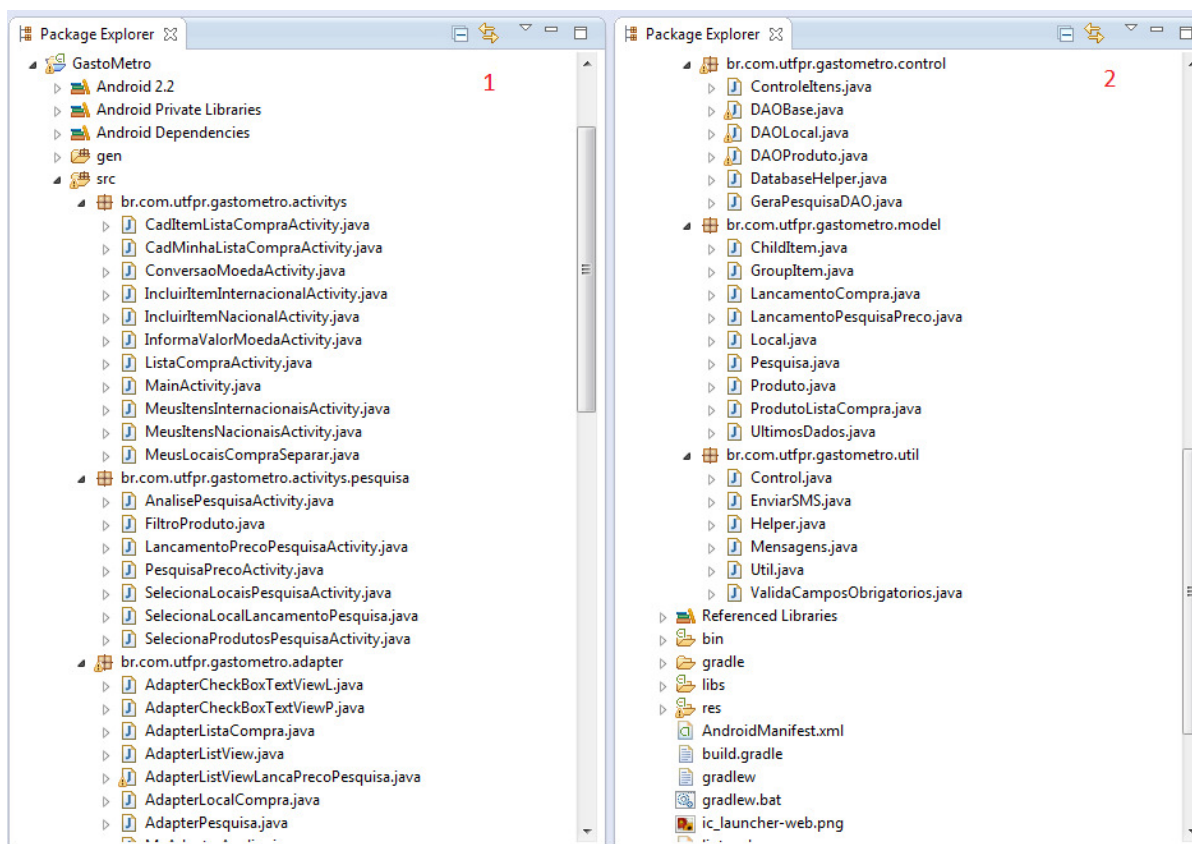


Figura 9 – Estrutura do código fonte
Fonte: Autoria própria

A estrutura do código fonte foi dividida em seis pacotes, para facilitar o entendimento do código, desta forma cada pacote possui tarefas específicas.

Os pacotes “br.com.utfpr.gastometro.activities” e “br.com.utfpr.gastometro.activities.pesquisa” contêm as classes para criação das telas do aplicativo. As classes responsáveis por montar as listagens no aplicativo ficam no pacote “br.com.utfpr.gastometro.adapter”.

No pacote “br.com.utfpr.gastometro.control”, estão localizadas as classes responsáveis pela conexão com o banco de dados, já no pacote “br.com.utfpr.gastometro.model”, possui as classes de modelo, que são classes mapeadas

e utilizadas para gravação dos dados no banco, cada classe deste pacote representa uma tabela no banco de dados.

Por fim, o pacote “br.com.utfpr.gastometro.util” contém as classes que são utilizadas em todo o projeto para reaproveitamento de código, possui classes para mensagens, validação de valores, tratamento de data e hora, entre outras funcionalidades.

4.2 DESENVOLVIMENTO DO APLICATIVO MÓVEL

Para desenvolvimento do aplicativo móvel, foi utilizado a versão 2.2 do Android, sendo esta chamada de “Froyo”. A escolha desta versão foi feita, pois à versão do aparelho móvel utilizado para testes possuía esta versão do Android instalada. Outra justificativa para esta versão é porque os *smartphones* comercializados hoje com o sistema Android dificilmente possuem uma versão inferior a esta instalada. O *smartphone* utilizado foi o Samsung Galaxy y, apresentado na Figura 10.



Figura 10 – Samsung Galaxy Y
Fonte: RICARDO ELETRO, 2014

Para a criação das interfaces visuais de cada tela do aplicativo, foi utilizado o assistente visual disponibilizado pelo próprio plugin ADT, instalado no Eclipse, este assistente é baseado em XML. É muito simples sua utilização, basta arrastar os componentes desejados para a área disponível e organizá-los de acordo com a

necessidade, desta maneira o código XML é gerado automaticamente. A Figura 11 exemplifica a criação de uma das interfaces do aplicativo.



Figura 11 – Assistente de criação de layouts
Fonte: Autoria própria

Todos os aplicativos desenvolvidos em Android, possuem obrigatoriamente o arquivo AndroidManifest.xml, onde é definido as permissões de acesso a recursos do dispositivo móvel. No aplicativo desenvolvido é possível enviar o resultado de uma pesquisa de preço para outros celulares, através do envio de SMS. Assim, para a utilização do recurso de envio de SMS é necessário adicionar neste arquivo uma permissão de uso desse recurso. Desta maneira, quando o aplicativo for instalado será informado ao usuário quais recursos serão utilizados.

Esse método segue a ideia de transparência do Android, evitando que aplicações mal intencionadas utilizem recursos do aparelho sem o conhecimento do proprietário, sendo exemplificada na Figura 12.

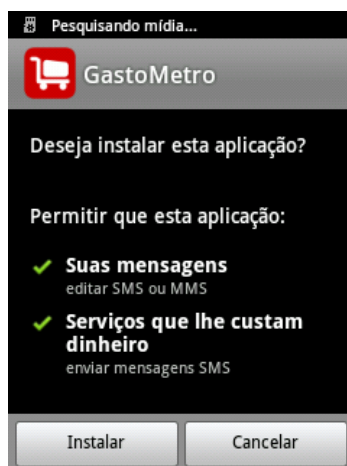


Figura 12 – Tela de confirmação de instalação de aplicativos
Fonte: Autoria própria

4.2.1 Apresentação das funcionalidades

Ao iniciar o aplicativo através do dispositivo móvel, a tela principal é apresentada, nela encontram-se as funcionalidades do aplicativo, distribuídas em cinco opções de menu, sendo elas: “Compra Nacional”, “Compra Internacional”, “Lista de Compras”, “Pesquisa de Preço” e “Conversão de Moedas”, conforme ilustrado na Figura 13. As funcionalidades citadas serão abordadas com maiores detalhes no transcorrer deste capítulo.



Figura 13 – Tela principal
Fonte: Autoria própria

Ao acessar através da tela principal o item “Compra Nacional”, será apresentada a tela para compra, onde o usuário deve definir o nome do local onde está comprando, e em seguida pode incluir os itens que está comprando, desta forma ele saberá o quanto já gastou até o momento, ver Figura 14.



Figura 14 – Compra nacional
Fonte: Autoria própria

Ao acessar o item “Compra Internacional”, será apresentada a tela para informar o valor e tipo da moeda, que será utilizada para conversão de valores, após isto, será aberta a tela onde o usuário deve definir o nome do local onde está comprando, e em seguida poderá incluir os itens que está comprando, desta forma ele saberá o quanto já gastou até o momento em reais conforme Figura 15.

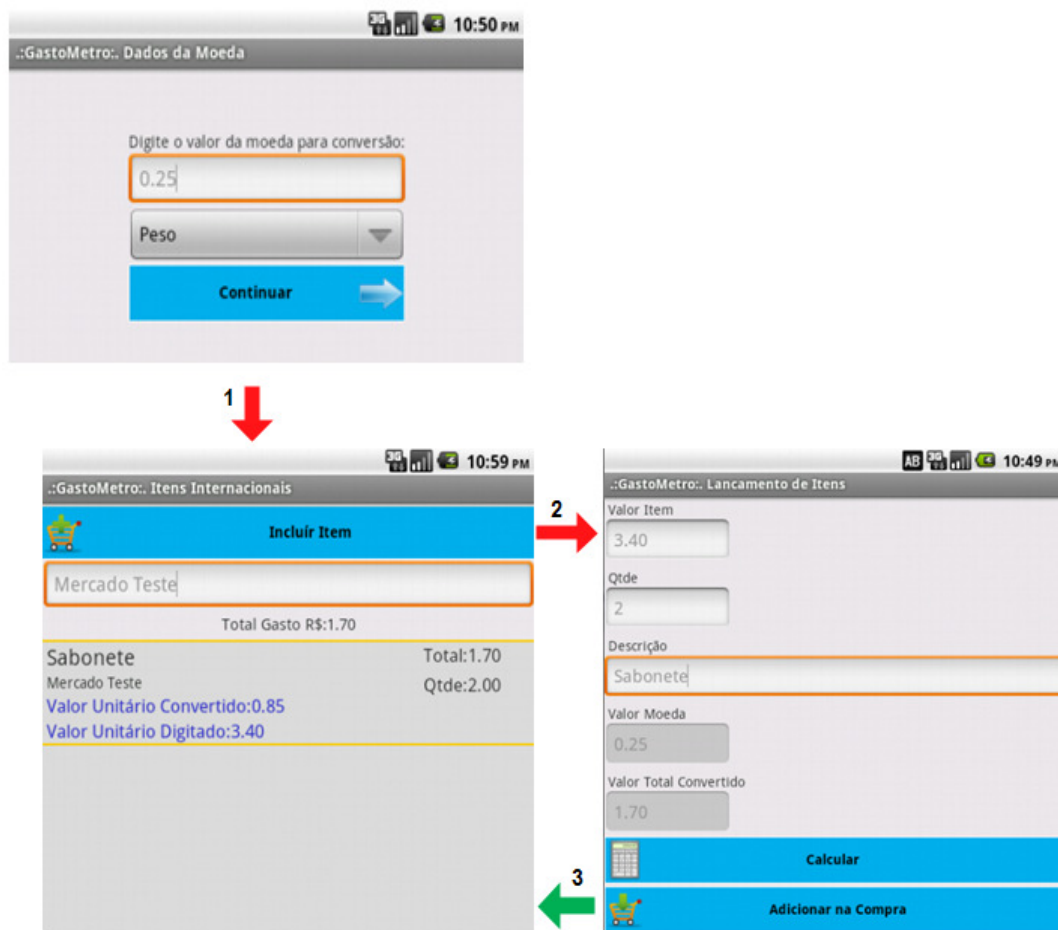


Figura 15 – Compra internacional
Fonte: Autoria própria

A opção “Lista de Compra”, quando acessado, apresenta ao usuário a tela para criação de uma lista de compra, inserindo os itens que deseja comprar de forma fácil e simples. Ver figura 16.



Figura 16 – Criação de uma lista de compra
Fonte: Autoria própria

Ao tocar o menu “Pesquisa de Preço”, o usuário pode criar uma lista para pesquisa de preços, onde ele define os produtos e locais que deseja pesquisar os preços, este processo é mostrado na Figura 17.

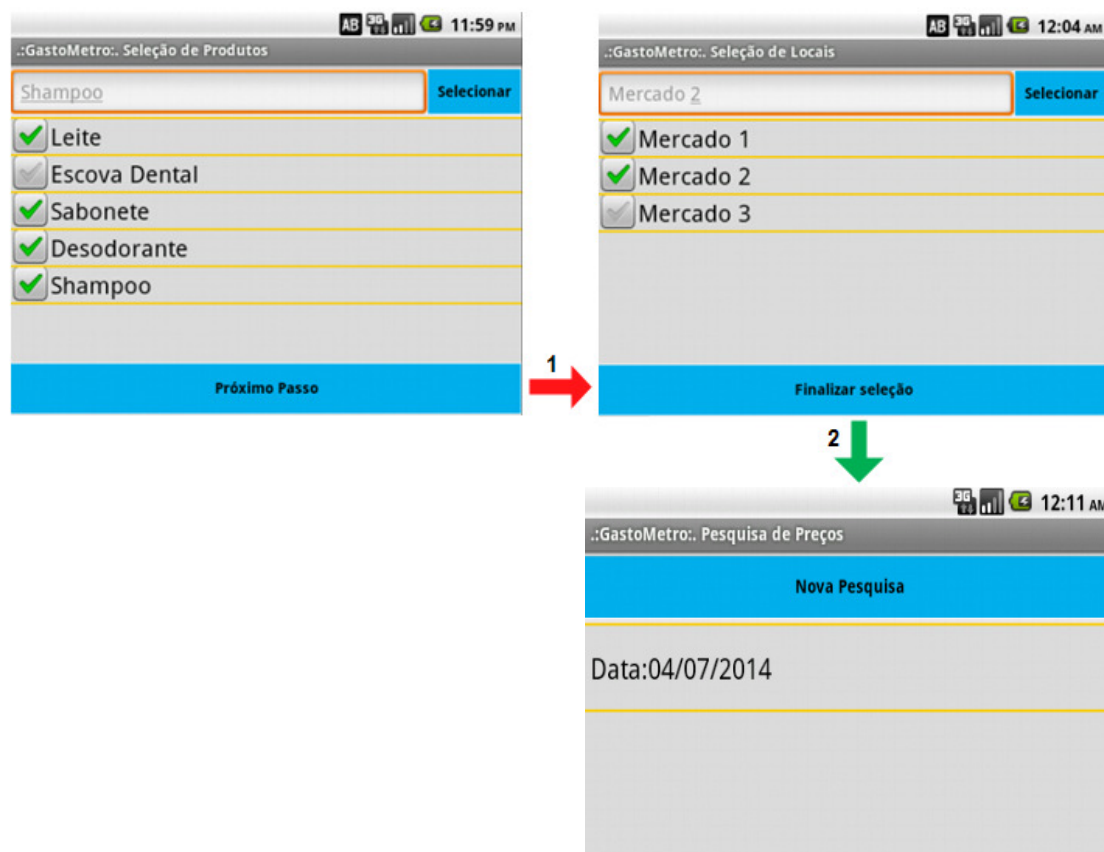


Figura 17 – Criação de uma pesquisa de preços
Fonte: Autoria própria

Ao tocar e segurar na pesquisa criada, esta representada pela data na Figura 17, será apresentada as opções disponíveis, que são: Lançar Preços; Editar Lista; Análise Pesquisa; Enviar por SMS, Excluir; Cancelar - Figura 18, essas opções serão detalhadas logo mais no decorrer do texto.



Figura 18 – Menu de opções da pesquisa selecionada
Fonte: Autoria própria

A opção “Lançar Preços”, é utilizada para selecionar um local entre vários, para lançamento dos preços, após a seleção do local, é apresentada a tela com os produtos para ser informado os seus respectivos preços, a seguir na Figura 19 é exemplificado o processo.



Figura 19 – Lançamento de preços
Fonte: Autoria própria

A opção de “Editar Lista” segue a mesma ideia do processo de criação da lista, podendo o usuário incluir ou tirar itens e locais.

Uma das opções mais importantes do aplicativo é “Analisar Pesquisa”, esta possibilita ao usuário a análise dos preços lançados. Existem duas formas de analisar os preços, a primeira é Produto/Local, onde mostra todos os produtos e seus respectivos

locais de lançamento, já a opção Produto/Local e Melhor preço, mostra os produtos e locais em que cada produto tem o melhor preço. Para ver as opções de análise é preciso escolher o menu do *smartphone*, conforme Figura 20.

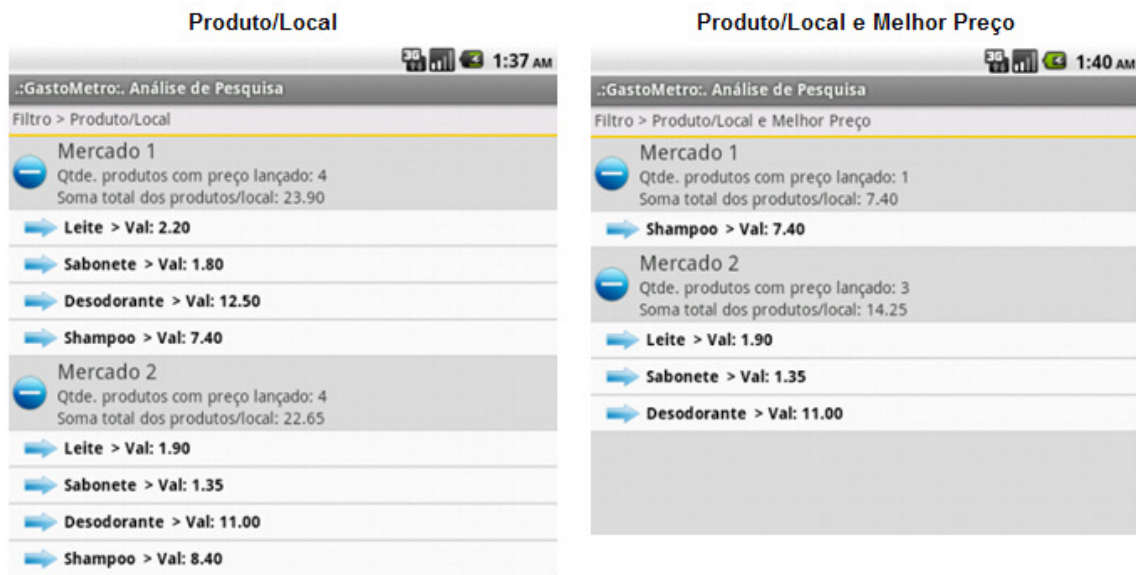


Figura 20 – Tela análise de pesquisa
Fonte: Autoria própria

Acessando a opção “Enviar por SMS”, o aplicativo converte em texto os dados lançados na pesquisa, concatenando Produto, Local e Valor. Dessa forma é possível enviar uma pesquisa para quem o usuário desejar - Figura 21.

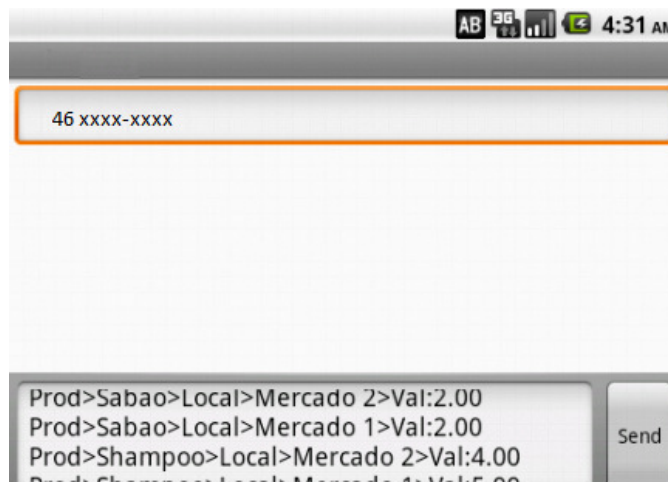


Figura 21 – Envio de SMS
Fonte: Autoria própria

Já a opção “Excluir”, é utilizada para exclusão de uma pesquisa de preço. E por fim o item “Cancelar” fecha o menu de opções.

A seguir serão apresentadas as soluções criadas, juntamente com seus respectivos códigos, que tornaram o aplicativo móvel funcional.

4.2.2 Persistência no Banco de Dados

Para o armazenamento dos dados inseridos no aplicativo, foi utilizado o banco de dados nativo do Android, este chamado SQLite, juntamente com o *framework* objeto-relacional ORMLite que é um *framework* de persistência que simplifica a forma como as informações são inseridas no banco de dados do aplicativo – Listagem 1.

Listagem 1 – Exemplo de gravação de um objeto no banco

```

1. public class InformaValorMoedaActivity extends Activity {
2.     private UltimosDados ud = null;
3.     private Button btnContinuarCompraInternacional;
4.
5.     @Override
6.     protected void onCreate(Bundle savedInstanceState) {
7.         super.onCreate(savedInstanceState);
8.         setContentView(R.layout.informa_valor_moeda_internacional);
9.         this.getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STATE_HIDDEN);
10.
11.         setValorMoedaDia();
12.         carregaDados();
13.         btnContinuarCompraInternacional =
14.             (Button) findViewById(R.id.buttonContinuarCompraInternacional);
15.         btnContinuarCompraInternacional
16.             .setOnClickListener(new OnClickListener() {
17.                 @Override
18.                 public void onClick(View arg0) {
19.                     EditText ed_ValorMoeda = (EditText) findViewById(R.id.txt_valor_moeda);
20.                     if (ed_ValorMoeda.getText().toString().trim()

```

```

21.         .equals("")) {
22.             ed_ValorMoeda.setError("Digite o valor da moeda");
23.             return;
24.         }
25.         if (Double.parseDouble(ed_ValorMoeda.getText()
26.             .toString().trim()) <= 0) {
27.             ed_ValorMoeda.setError("Digite o valor positivo!");
28.             return;
29.         }
30.         // alteraValorMoedaDia(Double.valueOf(ed_ValorMoeda.getText().
31.             toString().trim()));
32.
33.         Spinner spinerMoeda = (Spinner) findViewById(R.id.spinner_tipo_moeda);
34.
35.         try {
36.
37.             if (ud == null) {
38.                 ud = new UltimosDados();
39.             }
40.             ud.setNomeMoeda(spinerMoeda.getSelectedItem().toString());
41.             ud.setValorMoeda(Double.valueOf(ed_ValorMoeda.getText().toString().trim()
42.                 .replace(",", ".")));
43.             ud.setIsInternacional("T");
44.             salvaDados();
45.             Intent intent = new Intent(InformaValorMoedaActivity.this,
46.                 MeusItensInternacionaisActivity.class);
47.             startActivity(intent);
48.             finish();
49.         } catch (Exception e) {
50.             e.printStackTrace();
51.         }
52.     }
53. });
54. }
55. private void carregaDados() {
56.     ud = new DAOBase<UltimosDados>(this, UltimosDados.class).getUltimosDados("T");
57. }
58. public void voltar(){
59.     Intent intent = new Intent(this, MainActivity.class);
60.     finish();
61.     startActivity(intent);
62. }
63.
64. public void salvaDados() {
65.     try {
66.         new DAOBase<UltimosDados>(this, UltimosDados.class).Create(ud);
67.     } catch (Exception e) {
68.         // TODO Auto-generated catch block
69.         e.printStackTrace();
70.     }
71. }
72.
73. public void setValorMoedaDia() {
74.     ud = new DAOBase<UltimosDados>(this, UltimosDados.class).getUltimosDados("T");
75.     EditText ed_ValorMoeda = (EditText) findViewById(R.id.txt_valor_moeda);
76.     if (ud != null) {
77.         ed_ValorMoeda.setText(Util.formatarValor(ud.getValorMoeda()).replace(",", "."));
78.     }
79. }

```

Na linha 66 da Listagem 1, é mostrado como é feito a gravação de um objeto usando o ORMLite, nesse caso é passado um objeto do tipo UltimosDados, e a partir deste o ORMLite sabe em qual banco gravar, tabela, campo deve gravar os dados que são passados.

Na Listagem 2, é mostrado como é feito o mapeamento dos objetos para que possam ser gravados na base, nesse mapeamento é definido qual tabela será gravada, quais campos do objeto serão persistidos e por fim qual a chave primaria.

Listagem 2 – Classe mapeada usando ORMLite

```

1.  @DatabaseTable
2.  public class UltimosDados {
3.
4.      @DatabaseField(generatedId = true)
5.      private Integer id;
6.      @DatabaseField
7.      private double valorMoeda;
8.      @DatabaseField
9.      private String nomeMoeda;
10.     @DatabaseField
11.     private Date dataCadastro = Util.removeTime(
12.         new Date(Calendar.getInstance().getTimeInMillis()));
13.     @DatabaseField
14.     private String isInternacional = "F";
15.     @DatabaseField
16.     private Integer idLocal;
17.
18.     private Local local;
19.
20.     public UltimosDados() {
21.
22.     }

```

O `@DatabaseTable`(linha 1 da Listagem 2) define que o objeto `UltimosDados` poderá ser persistido e que o nome da tabela do banco será `ultimosdados`, já o `@DatabaseField` define quais campos do objeto será persistido, e qual o tipo de dado de cada campo.

E por fim, na Listagem 3, é apresentado a classe que responsável pela gravação dos dados. A classe `DAOBase` utiliza um recurso chamado generics (linha 1 e 2), o qual não é necessário definir o tipo de dado que a classe vai trabalhar, dessa forma diz-se que ela trabalhará com um tipo genérico, que no caso utilizado foi definido como `E`, conforme observado na Listagem 1, linha 66. Pode-se observar que no momento da criação de um objeto da classe `DAOBase`, foi definido qual tipo de dado ela iria trabalhar.

Listagem 3 – Classe persistência de dados usando generics

```

1.  public class DAOBase<E> extends DatabaseHelper<E> {
2.
3.      protected Dao<E, Integer> dao;
4.
5.      private Class<E> type;
6.      private Context context;
7.
8.      public DAOBase(Context context, Class<E> type) {
9.          super(context);
10.         this.type = type;
11.         this.context = context;
12.         SetDAO();
13.     }
14.

```

```

15.     protected void SetDAO() {
16.         try {
17.             if (dao == null) {
18.                 dao = DaoManager.createDao(connectionSource, type);
19.             }
20.         } catch (Exception ex) {
21.             ex.printStackTrace();
22.         }
23.     }
24.
25.     public void Create(E obj) throws Exception {
26.         try {
27.             dao.createOrUpdate(obj);
28.         } catch (Exception ex) {
29.             throw ex;
30.         }
31.     }

```

Utilizando este recurso consegue-se trabalhar com uma única classe para persistir todos os objetos necessários no aplicativo todo.

4.2.3 Adapters

A exibição de elementos em uma lista é muito comum em aplicações móveis, o usuário vê uma lista de itens e pode percorrê-los de forma fácil e intuitiva, mesmo nos dispositivos com telas menores. No aplicativo desenvolvido foram utilizadas diversas listas, para apresentar itens, pesquisas, lançamentos, resultados.

Para realizar essa apresentação de dados em listas é utilizado um recurso do Android chamado Adapter, que é uma ponte entre a visão e os dados apresentados nessa visão. Com ele é possível manipular a apresentação dos dados em um ListView.

Utilizando o método `getView()` é possível manipular a apresentação do ListView linha a linha. Existem alguns Adapters que são do próprio SDK, basta atribuir os dados para o adapter e atribuí-lo a um componente, e os dados serão apresentados em uma lista padrão, isso é muito útil para listas simples, quando é preciso criar layouts complexos com muitos campos, é preciso criar Adapters personalizados, os quais devem herdar de `BaseAdapter`, na Listagem 4 é mostrado um Adapter personalizado que foi utilizado no projeto.

Listagem 4 – Exemplo de manipulação de dados no banco de dados

```

1.     public class AdapterListaCompra extends BaseAdapter {
2.         private LayoutInflater mInflater;
3.         private List<ProdutoListaCompra> itens;
4.         public AdapterListaCompra(Context context, List<ProdutoListaCompra> itens) {
5.             this.itens = itens;
6.             mInflater = LayoutInflater.from(context);
7.         }
8.         public int getCount() {
9.             return itens.size();

```



```

10.     }
11.     public ProdutoListaCompra getItem(int position) {
12.         return itens.get(position);
13.     }
14.     public long getItemId(int position) {
15.         return position;
16.     }
17.     public View getView(int position, View view, ViewGroup parent) {
18.         // Pega o item de acordo com a posição.
19.         ProdutoListaCompra item = itens.get(position);
20.         view = inflater.inflate(R.layout.item_lista_compra, null);
21.         ((TextView) view.findViewById(R.id.list_view_txt_nome_produto)).
22.             setText(item.getProduto().getDescricao());
23.         if(item.getIdLocal() != null){
24.             ((TextView) view.findViewById(R.id.tvQtd)).setVisibility(TextView.GONE);
25.             ((TextView) view.findViewById(R.id.list_view_txt_qtd)).setVisibility(TextView.GONE);
26.             ((TextView) view.findViewById(R.id.tvLocalListaCompra)
27.                 .setVisibility(TextView.VISIBLE);
28.             ((TextView) view.findViewById(R.id.tvLocalListaCompra)).
29.                 setText(item.getLocal().getNome());
30.         }else{
31.             ((TextView) view.findViewById(R.id.list_view_txt_qtd)).
32.                 setText(Util.formatarValor(item.getQtd()));
33.             ((TextView) view.findViewById(R.id.tvLocalListaCompra)).setVisibility(TextView.GONE);
34.         }
35.         return view;
36.     }
37. }

```

O adapter é alimentado com uma lista de itens linha 5, e para cada item dessa lista é disparado o método `getView()` linha 17, e neste é recuperado os dados do item que serão inseridos em uma linha da listagem linha 21 a linha 33.

4.2.4 Envio de SMS

Para envio de SMS foi utilizado o Intent, que é um recurso da SDK do Android, este pode ser personalizado para o envio de mensagens, conforme Listagem 5.

Listagem 5 – Envio de SMS

```

1.     public class EnviarSMS extends Activity {
2.         private int idPesquisa;
3.         private DAOBase<LancamentoPesquisaPreco> daoPesq;
4.
5.         @Override
6.         protected void onCreate(Bundle savedInstanceState) {
7.             super.onCreate(savedInstanceState);
8.             Bundle param = getIntent().getExtras();
9.             if (param != null) {
10.                 idPesquisa = param.getInt("idPesquisa");
11.             }
12.             daoPesq = new DAOBase<LancamentoPesquisaPreco>(this, LancamentoPesquisaPreco.class);
13.             enviaSMS();
14.         }
15.
16.         public String getBodySMS() {
17.             try {
18.                 StringBuilder msg = new StringBuilder();
19.                 ArrayList<LancamentoPesquisaPreco> pesquisa = (ArrayList<LancamentoPesquisaPreco>)
20.                     daoPesq
21.                         .getProdutosAnalise("idPesquisa", idPesquisa + "", "");
22.                 msg.append("Pesquisa de preço enviada pelo app GastoMetro...\r\n");
23.                 for (LancamentoPesquisaPreco pesq : pesquisa) {

```

```

23.         msg.append("\r\nProd>" + pesq.getProduto().getDescricao());
24.         msg.append(">Local>" + pesq.getLocal().getNome());
25.         msg.append(">Val:" + Util.formatarValor(pesq.getValor()));
26.     }
27.     return msg.toString();
28. } catch (Exception e) {
29.     // TODO Auto-generated catch block
30.     e.printStackTrace();
31. }
32. return null;
33. }
34.
35. private void enviaSMS() {
36.     String body;
37.     body = getBodySMS();
38.     try {
39.         Intent smsIntent = new Intent(Intent.ACTION_VIEW);
40.         smsIntent.putExtra("sms_body", body);
41.         smsIntent.setType("vnd.android-dir/mms-sms");
42.         startActivity(smsIntent);
43.         this.finish();
44.     } catch (Exception e) {
45.         Toast.makeText(getApplicationContext(),
46.             "Não foi possível enviar o SMS", Toast.LENGTH_LONG).show();
47.         e.printStackTrace();
48.     }
49. }
50. }

```

O Intent é utilizado para realizar uma ação, geralmente são criados a partir de ações do usuário e representam a intenção de realizar algo, por exemplo, iniciar a reprodução de uma música, iniciar o aplicativo de correio eletrônico, abrir uma nova tela na aplicação, entre outras inúmeras ações.

É um recurso muito utilizado no Android, através dele é possível que as aplicações colaborem entre si. Através de Intents é possível iniciar novas activities, como fazer uma busca e selecionar um contato do telefone, abrir a aplicação de mapas com coordenada de localização GPS, abrir uma pagina web, etc.

No aplicativo desenvolvido foi utilizando o Intent para abrir a tela de envio de SMS linha 39, foram definidos alguns parâmetros para o Intent, em seguida é passado para o método startActivity() e a tela de envio é aberta. Em poucas palavras o Intent possui as informações necessárias para ativar um componente da aplicação.

O Intent também foi muito utilizado para a navegação entre as telas do aplicativo, onde era definido alguns parâmetros para o Intent e a tela desejada era aberta, esse recurso é muito útil e simples de usar.

4.2.5 Testes

Não é possível garantir o funcionamento de um produto sem passar pela fase de testes, esta fase é muito importante no processo de criação de um produto, nela é identificado inconsistências nos dados gerados pelo produto.

No aplicativo desenvolvido foi utilizado para testes o próprio emulador fornecido pelo SDK do Android, juntamente com um *smartphone* Samsung Galaxy Y, a cada recurso novo criado para o aplicativo, era gerado uma nova versão e esta era testada a fim de identificar possíveis erros. Na Figura 22 é apresentado o emulador utilizado para testar o aplicativo.

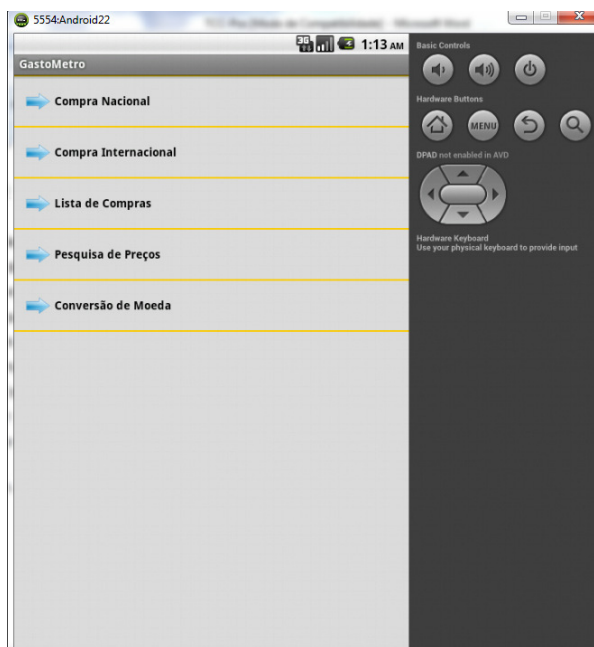


Figura 22 – Emulador utilizado para testes
Fonte: Autoria própria

Na Figura 23 é apresentado o *smartphone* que também foi utilizado para testes. O uso deste *smartphone* ajudou a verificar o funcionamento do envio de SMS utilizado enviar uma pesquisa de preço para outros *smartphones*. Além disso, também foi possível identificar outros erros que o aplicativo tinha como, por exemplo, não fechar as telas, erros em tempo de execução, erros de armazenamento das informações, entre outros. Desta forma com a realização de testes foi possível corrigir estas situações tornando o aplicativo mais confiável.



Figura 23– Testes em um smartphone modelo Samsung Galxy Y
Fonte: Autoria própria

4.2.6 Implantação

Nesta etapa foi disponibilizada uma versão do aplicativo para um supermercado da cidade de Dois Vizinhos-PR. O estabelecimento utilizou a pesquisa de preço durante um mês. Através do aplicativo foi criada uma pesquisa de preços, onde foram definidos quais locais e produtos fariam parte da pesquisa, logo após o lançamento dos preços foi possível analisar quais produtos possuem melhor preço por local.

Na Figura 24 é possível observar a diferença total de preços entre os dois estabelecimentos pesquisados, ambos da cidade de Dois Vizinhos, neste caso foram pesquisados 30 produtos, e foi possível através do aplicativo verificar que se a compra fosse realizada no Mercado 2 teria uma economia de R\$ 61.79.

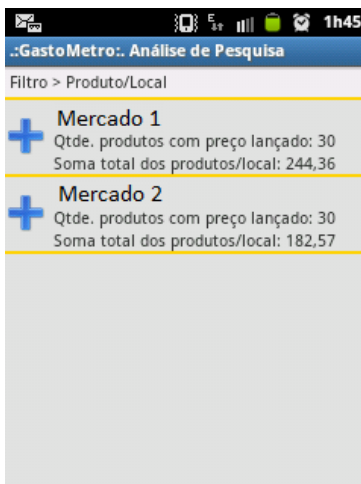


Figura 24 – Resultado de uma pesquisa de preço
Fonte: Autoria própria

Esta funcionalidade é muito útil para estabelecimentos comerciais, permitindo analisar os seus preços em relação aos seus concorrentes, podendo assim ajustá-los e manter seus preços competitivos.

A seguir é apresentado o depoimento do responsável pelo estabelecimento comercial da cidade de Dois Vizinhos-PR que utilizou o aplicativo.

“A ideia do aplicativo foi muito boa. Estar antenado com os preços dos concorrentes é vital para manter o empreendimento no mercado, no entanto há uma dificuldade em estar adentrando o estabelecimento concorrente e realizar esta pesquisa. Com o aplicativo desenvolvido para celular, facilitou muito isso, visto que a pesquisa pode ser feita de forma praticamente imperceptível.

Falando do aplicativo em si, é bem transparente, a interface é autoexplicativa e tem boa usabilidade, sendo possível o uso tanto para pesquisa com o concorrente, quanto para mandar para os clientes com a sugestão de lista de compras. Temos algumas ideias para desenvolver novos recursos, as quais estamos passando a medida que vão aparecendo, o objetivo é que o aplicativo seja usado semanalmente, auxiliando muito na criação de promoções e estratégias para obtenção e ampliação da carteira de clientes.”

5. CONCLUSÃO

O aplicativo desenvolvido tem a finalidade de controlar as compras e quanto irá pagar, além de permitir a criação de pesquisas de preços, listas de compras e converter moedas no caso de compras internacionais.

Neste trabalho, inicialmente apresentou-se toda a documentação necessária para criação e compreensão do aplicativo proposto, entre essa documentação encontra-se o diagrama de caso de uso essencial para entender as funcionalidades do aplicativo e o diagrama de banco de dados o qual mostra as tabelas do bando de dados necessárias para o projeto.

Com esta documentação percebe-se a importância de trabalhar com engenharia de software, envolvendo os processos de engenharia de requisitos e projeto. Estas duas fases foram fundamentais, pois ofereceram o embasamento necessário para o bom desenvolvimento do projeto e auxiliou para construir corretamente o que foi proposto.

Com a documentação definida, o processo de desenvolvimento se tornou mais simples, utilizando as ferramentas citadas na seção 3.1, bem com a metodologia definida na seção 3.2, foi possível atingir os objetivos iniciais: o desenvolvimento e os testes do aplicativo.

5.1 PROBLEMAS ENCONTRADOS

No decorrer do desenvolvimento do trabalho foram encontradas algumas dificuldades com relação à ferramenta de desenvolvimento Eclipse, utilizada para desenvolvimento do código fonte. Em alguns momentos, ao importar alguns projetos para analisar o código fonte, a ferramenta limpava todos os arquivos do projeto. O erro foi desconhecido e atrapalhou o desenvolvimento do projeto.

Outro problema era a lentidão para depurar o código fonte, quando não se consegue identificar o que está causando determinado problema, é utilizado o recurso de debug que é a depuração linha a linha da aplicação, e ao realizar esta operação, ocorria um erro de Timeout, obrigando a começar a depuração novamente desde o início, esse caso foi resolvido no fim do projeto desmarcando algumas configurações na própria IDE Eclipse.

5.2 TRABALHOS FUTUROS

Para trabalhos futuros sugere-se a possibilidade de comercialização deste aplicativo. Já houve pessoas e empresas interessadas, pois a aplicação ajuda muito na pesquisa de preços de concorrentes de forma discreta.

Também poderá ser reformulado a parte de usabilidade do aplicativo utilizando o recurso de ActionBar, que é uma barra de navegação superior, que auxilia na navegação do usuário na aplicação.

Outro recurso, é a criação de uma opção de sugestão de compra, com base nas compras realizadas o aplicativo iria sugerir os produtos que o usuário teria que comprar para não faltar em sua casa.

REFERÊNCIAS

ANDROID DEVELOPER. Disponível em <http://developer.android.com/index.html>. Acesso em: 28 jul. 2014.

ANDROID SDK. Disponível em <http://developer.android.com/sdk/index.html>. Acesso em: 10 jul. 2014.

ASTAH COMMUNITY. Disponível em <http://www.astah.net>. Acesso em: 12 jul. 2014.

BEZERRA, Eduardo. **Princípios de análise e Projeto de Sistemas com UML**. 2ª Edição. Rio de Janeiro: Campus, 2007.

DOT LEGEND. Disponível em https://play.google.com/store/apps/details?id=net.dotlegend.boalista&hl=pt_BR. Acesso em: 10 out. 2014.

FLIPIT. **Como economizar no supermercado**. Disponível em <http://www.flipit.com/br/guiaeconomizar/economize-no-supermercado>. Acesso em 21 jun. 2014.

GARTNER. **Gartner Says Worldwide Traditional PC, Tablet, Ultramobile and Mobile Phone Shipments On Pace to Grow 7.6 Percent in 2014**. Disponível em <https://gartner.com/newsroom/id/2645115>. Acesso em 29 jul. 2014.

GOOGLE PLAY. Disponível em https://play.google.com/store?hl=pt_BR. Acesso em: 19 set. 2014.

IDE ECLIPSE. Disponível em <https://www.eclipse.org/downloads>. Acesso em 19 jul. 2014

OUR MOBILE PLANET. Disponível em <https://www.services.google.com/fh/files/misc/omp-2013-br-local.pdf>. Acesso em 22 jul. 2014.

ORMLITE. **Lightweight Object Relational Mapping (ORM) Java Package** - Disponível em <http://www.ormlite.com>. Acesso em 10 jul. 2014.

PLUGIN ADT. Disponível em <http://developer.android.com/tools/sdk/eclipse-adt.html>. Acesso em 19 jul. 2014

PLAY STORE. Disponível em <https://play.google.com/store/apps>. Acesso em 19 jun. 2014.

PRESSMAN, Roger S. **Engenharia de Software**. 6ª Edição. São Paulo: McGraw-Hill do Brasil, 2006.

PRESSMAN, Roger S. **Engenharia de Software**. 7ª Edição. São Paulo: McGraw-Hill do Brasil, 2007.

REZENDE, Denis Alcides. **Engenharia de Software e Sistemas de Informação**. 2ª Edição. Rio de Janeiro: Brasport, 2002.

RICARDO ELETRO. Disponível em <http://www.ricardoeletro.com.br/Produto/Samsung-Galaxy-Y-S5360-Prata-Celular-Smartphone-Desbloqueado-Claro-Android-23-Camera-2MP-WiFi-3G-MP3-Player-Bluetooth-e-Fone-de-Ouvido/44-491-496-151792>. Acesso em 16 jul. 2014. Acesso em: 30 ago et. 2014.

SQLITE. Disponível em <http://www.sqlite.org/about.html>. Acesso em 16 jul. 2014.

SUPERMARKET. **Supermercado: vamos às compras**. Disponível em https://play.google.com/store/apps/details?id=com.dukitan.android.supermarket&hl=pt_BR. Acesso em: 19 set. 2014.