

Beatriz da Silveira

Aplicativo para dispositivo móvel para otimizar compras em supermercados

São José - SC

Junho/2017

Beatriz da Silveira

Aplicativo para dispositivo móvel para otimizar compras em supermercados

Monografia apresentada à Coordenação do Curso Superior de Tecnologia em Sistemas de Telecomunicações do Instituto Federal de Santa Catarina para a obtenção do diploma Tecnólogo em Sistemas de Telecomunicações.

Instituto Federal de Santa Catarina – IFSC
Campus São José
Curso Superior de Tecnologia em Sistemas de Telecomunicações

Orientador: Prof. Emerson Ribeiro de Mello, Dr
Coorientador: Prof. Tiago Semprebom, Dr

São José - SC
Junho/2017

Beatriz da Silveira

Aplicativo para dispositivo móvel para otimizar compras em supermercados/
Beatriz da Silveira. – São José - SC, Junho/2017-

67 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Emerson Ribeiro de Mello, Dr

Monografia (Graduação) – Instituto Federal de Santa Catarina – IFSC
Campus São José

Curso Superior de Tecnologia em Sistemas de Telecomunicações, Junho/2017.

1. Android. 2. Lista de compra. I. Emerson Ribeiro de Mello. II. Instituto Federal de Santa Catarina. III. Campus São José. IV. Aplicativo para dispositivo móvel para otimizar compras em supermercados

Beatriz da Silveira

Aplicativo para dispositivo móvel para otimizar compras em supermercados

Monografia apresentada à Coordenação do Curso Superior de Tecnologia em Sistemas de Telecomunicações do Instituto Federal de Santa Catarina para a obtenção do diploma Tecnólogo em Sistemas de Telecomunicações.

Trabalho aprovado. São José - SC, 20 de junho de 2017:

Prof. Emerson Ribeiro de Mello, Dr
Orientador

Prof. Tiago Semprebom, Dr
Coorientador

Prof. Eraldo Silveira e Silva, Dr.
IFSC

Prof. Arliones Stevert Hoeller Junior,
Me.
IFSC

São José - SC
Junho/2017

Agradecimentos

Primeiramente gostaria de agradecer aos meus pais por sempre me guiarem e me ensinarem os valores do estudo, e a prioridade que isso tem ao longo da vida e da carreira profissional.

Meu imenso agradecimento ao meu companheiro e amigo William, por sempre se fazer presente em todos os momentos e nunca ter medido esforços para me auxiliar e guiar sempre que precisei. Sem você nada disso seria possível, muito obrigada por sempre acreditar em mim e me motivar.

Ao professor Mello, meu muito obrigada, por ter me guiado e orientado durante o desenvolvimento dessa monografia com paciência e dedicação. Tenho total admiração e respeito pelo seu trabalho e pelo mestre competente que és. Ao professor Tiago pelo incentivo e apoio nessa jornada mostrando-se prestativo em todos os momentos.

Aos amigos que dividiram comigo as alegrias e dificuldades durante o curso, e todas as comemorações de cada etapa vencida.

Resumo

Tradicionalmente listas de compras são elaboradas utilizando-se papel e caneta. Nesse cenário os produtos são inseridos um a um manualmente e a ordem de inserção na lista é decorrente dos produtos que vamos lembrando. Atualmente devido a popularização dos aplicativos, as pessoas estão migrando das tradicionais listas manuscritas para a sua versão digital. No entanto, a maioria dos aplicativos de compras apresentam-se como uma versão digital da lista em papel. Isto é, trazem algumas facilidades para montar a lista, porém não auxiliam durante as compras, quando o usuário está no mercado utilizando efetivamente a lista de produtos. Desta forma, esse trabalho tem como objetivo reduzir o tempo de confecção das listas de compras e o tempo do usuário dentro dos supermercados, através de um aplicativo móvel para a plataforma Android que irá acompanhar o usuário durante as compras. Esse aplicativo auxiliará o usuário a inserir os produtos no carrinho de compras na melhor ordem possível, de acordo com a organização das seções presentes no supermercado. Isso é possível pois o aplicativo desenvolvido nesse trabalho, denominado Compra Rápida, se utiliza da ordem em que os produtos são removidos da lista de compras passadas, para desta forma prover a organização das seções dos produtos o mais similar possível ao supermercado no qual está sendo realizada a compra.

Palavras-chave: Android, Lista de Compras.

Abstract

The traditional way of creating a shopping list is using a paper and pen, where the products are manually inserted and the order that they are added in the list is derived from the products we are recalling. Currently, due to the popularization of mobile applications, people are migrating from traditional paper lists to a digital version. However, most shopping applications boil down to being a digital version of the paper list, they bring some facilities to create the list, however, they do not help during the shopping, when the user is on market effectively using the shopping list. In this way, this work aims to adapt the shopping lists to minimize the time consumed doing shopping by using a mobile application for the Android platform in which will assist the user during the shopping. This application will help the user to shop the products contained on the shopping list in the best possible order, according to the organization of the sections present in the supermarket. This is possible because the application denominated in this thesis Compra Rápida uses the past purchases from the user, in order to provide a better organization of the shopping list, putting together the sections of the products as similar as possible to the supermarket in which the purchase is being made.

Keywords: Android. Shopping list.

Lista de ilustrações

Figura 1 – Notificações na gaveta de notificação	29
Figura 2 – Aplicação do material design no Gmail	30
Figura 3 – Tipos de listas de compras.	37
Figura 4 – Planta baixa do mercado.	38
Figura 5 – Menu Principal.	39
Figura 6 – Etapas para criar uma lista.	40
Figura 7 – Produtos.	40
Figura 8 – Lista de mercados.	41
Figura 9 – Etapas de uma compra.	42
Figura 10 – Objeto JSON.	42
Figura 11 – Sincronização dos dados com todos os clientes.	44
Figura 12 – Diagrama de casos de uso.	61
Figura 13 – Diagrama Entidade Relacionamento	66

Lista de tabelas

Tabela 1 – Resumo comparitivo entre os aplicativos	35
Tabela 2 – Lista de compra do usuário.	48
Tabela 3 – Ordenação da lista em uma primeira compra.	49
Tabela 4 – Ordem de inserção dos produtos no carrinho pelo usuário.	49
Tabela 5 – Ordenação da lista em uma segunda compra.	50
Tabela 6 – Ordenação da nova lista de compras.	51
Tabela 7 – Ordem de inserção dos produtos no carrinho da nova lista.	51
Tabela 8 – Resultado da ordenação da lista em uma nova compra.	51
Tabela 9 – Caso de Uso: Criar Lista	62
Tabela 10 – Caso de Uso: Listar Lista	62
Tabela 11 – Caso de Uso: Editar Lista	62
Tabela 12 – Caso de Uso: Excluir Lista	62
Tabela 13 – Caso de Uso: Compartilhar Lista	63
Tabela 14 – Caso de Uso: Listar Mercados	63
Tabela 15 – Caso de Uso: Ir as Compras	63
Tabela 16 – Caso de Uso: Consultar Compras	64
Tabela 17 – Caso de Uso: Listar itens e categorias	64

Lista de abreviaturas e siglas

API Application Programming Interface	26
ISM <i>Industrial, Scientific and Medical</i>	24
SMS <i>Short Message Service</i>	19
Wi-Fi <i>Wireless Fidelity</i>	17
GPS <i>Global Positioning System</i>	19
BaaS <i>Backend as a Service</i>	31

Sumário

1	INTRODUÇÃO	19
1.1	Justificativa	20
1.2	Objetivos	20
1.2.1	Objetivo Geral	20
1.2.2	Objetivos Específicos	20
1.3	Organização do texto	21
2	FUNDAMENTAÇÃO TEÓRICA	23
2.1	Tecnologias e serviços de conectividade	23
2.1.1	Telefonia Celular	23
2.1.2	Bluetooth	24
2.1.3	IEEE 802.11 (<i>Wireless Fidelity (Wi-Fi)</i>)	25
2.2	Plataforma Android	26
2.2.1	Permissionamento	26
2.2.2	Geolocalização	27
2.2.3	Notificações na interface do usuário	28
2.2.4	Material design	29
2.3	Sincronização de dados com serviços na nuvem	31
2.4	Trabalhos Relacionados	32
2.4.1	Grocery List - Tomatoes	32
2.4.2	Listonic	33
2.4.3	Out of Milk	33
2.4.4	Bring	33
2.4.5	Resumo comparativo	34
3	COMPRA RÁPIDA	37
3.1	Funcionalidades	38
3.2	Base de dados de produtos e categorias	41
3.3	Algoritmo para ordenação da lista de compras	44
4	CONCLUSÕES	53
4.1	Trabalhos Futuros	54
	REFERÊNCIAS	55

APÊNDICES 57

APÊNDICE A – REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS 59

A.0.1	Requisitos funcionais	59
A.0.2	Análise de requisitos não funcionais	60

APÊNDICE B – CASOS DE USO 61

APÊNDICE C – MODELO ENTIDADE RELACIONAMENTO 65

1 Introdução

O telefone móvel evoluiu muito desde o seu lançamento como produto para o público em geral. Na época de seu surgimento, o grande diferencial da tecnologia era em permitir que o usuário pudesse realizar e receber chamadas mesmo estando fora de sua residência ou local de trabalho. Porém, com o decorrer do tempo novas funções foram adicionadas, como troca de mensagens de texto curtas (*Short Message Service (SMS)*), agenda de contatos, calendário, entre outros. Atualmente, as pessoas carregam em seus bolsos uma versão inteligente desses telefones, também conhecidos como *smartphones*. Antes, funções que eram somente possíveis de serem realizadas em computadores ou em dispositivo designado a uma tarefa específica, como por exemplo tirar fotos, ouvir músicas, geolocalização (*Global Positioning System (GPS)*), jogos e acesso à Internet, podem ser encontrados em um mesmo dispositivo com tudo integrado.

De acordo com ([LUNDEN, 2015](#)) em 2015 mais de 2,6 bilhões de pessoas possuem telefone celular. O mercado de aplicativos para esses dispositivos está rendendo bilhões de dólares todos os anos. De acordo com ([LECHETA, 2013](#)) usuários comuns buscam um celular com um visual elegante, moderno, de fácil navegação e também com uma infinidade de aplicativos e recursos. Tanto as empresas quanto os desenvolvedores buscam uma plataforma moderna e ágil para desenvolver aplicativos. O sistema operacional Android se apresenta como uma plataforma robusta e rica em funcionalidades para agradar tanto os usuários quanto os desenvolvedores, além de ser uma plataforma aberta na qual todos possuem acesso ao seu código fonte e não possui restrições em relação ao computador que está sendo utilizado para a montagem do ambiente de desenvolvimento.

Os aplicativos para dispositivos móveis auxiliam nas tarefas diárias das pessoas, como deslocamento pelas cidades, comunicações com outras pessoas, bloco de anotações, etc. Tarefas antes efetuadas manualmente ou anotadas em papéis como uma lista de compras em um supermercado ou lembretes de tarefas pendentes podem ser automatizadas ou lembradas com dinamicidade e interatividade e assim ajudar a evitar erros ou mesmo esquecimento de alguma tarefa.

No caso de uma lista de compras a forma tradicional para cria-lá é usando papel e caneta. Geralmente os produtos são adicionados na lista de maneira corrida, ou seja, um item após o outro, sendo um item por linha. Isso poderia resultar uma lista não otimizada para realizar as compras, ou seja, os itens na lista podem não estar agrupados e ordenados em seções como estariam em um supermercado. Sendo assim, durante a compra no supermercado, em cada seção que a pessoa se encontra ela precisaria percorrer toda a lista a fim de verificar se existe ali algum item daquela seção que ela precise comprar. É

nesse cenário que os aplicativos móveis poderiam ajudar, permitindo ao usuário criar uma lista de compra com os itens agrupados em seções, tornando assim menor o tempo para realizar a compra.

1.1 Justificativa

Atualmente existem diversos aplicativos disponíveis para realizar compras em supermercados para a plataforma Android. A maioria desses aplicativos são uma versão digital da tradicional lista em papel. Isto é, trazem facilidades para montar a lista de compras, com inclusão facilitada de itens e registro de histórico, porém no momento das compras, quando o usuário vai ao mercado utilizar efetivamente essa lista, os produtos estão listados pela ordem de inserção, ou no melhor caso, agrupados por categorias. Ou seja, os aplicativos ainda não possuem uma funcionalidade que facilite o usuário a otimizar o seu tempo durante as compras, auxiliando-o a pegar os produtos na melhor ordem possível, ou seja, os que estão mais próximos uns dos outros.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo principal desse trabalho é desenvolver um aplicativo móvel de lista de compras que minimize o tempo gasto durante as compras em um supermercado, apresentando ao usuário os produtos a serem buscados por ordem de proximidade física. O usuário informará em qual supermercado está fazendo compras e o aplicativo ordenará a lista de produtos dinamicamente conforme a organização das seções e a disposição dos produtos pelas seções do supermercado em questão. Dessa forma, espera-se minimizar o tempo que o usuário despende ao fazer compras em um supermercado.

1.2.2 Objetivos Específicos

- Conceber um algoritmo que permita ordenar os itens de uma lista na fase de compras tendo como base o histórico da ordem que esses itens foram pegos durante em compras anteriores;
- Utilizar as APIs do ambiente de desenvolvimento Android que permitam determinar a posição geográfica de um dispositivo, como o uso da rede celular, redes sem fio Wi-Fi e sinais GPS;
- Desenvolver uma solução de base de dados distribuída para permitir o compartilhamento e atualização das listas de produtos e seções;

- Desenvolver um aplicativo Android que siga as recomendações do Material Design Guide, que indica como as interfaces de usuários deveriam ser construídas;

1.3 Organização do texto

No [Capítulo 1](#) foi apresentada a motivação do trabalho, o objetivo geral bem como os objetivos específicos do mesmo. O restante do documento está organizado da seguinte forma.

No [Capítulo 2](#) é apresentado a fundamentação teórica com os tópicos e conceitos utilizados neste trabalho, como as tecnologias envolvidas para o desenvolvimento do aplicativo proposto. Nesse capítulo também são apresentados os trabalhos relacionados e uma tabela comparativa entre eles com algumas de suas funcionalidades referentes ao objetivo proposto.

No [Capítulo 3](#) é apresentado o Compra Rápida, aplicativo para otimização de compras desenvolvido nesse trabalho. São descritas suas funcionalidades vistas pelo usuário, bem como o algoritmo desenvolvido para organização da apresentação da lista de produtos durante a execução de uma lista de compras.

No [Capítulo 4](#) são apresentadas as conclusões e os trabalhos futuros que podem ser desenvolvidos tendo como base este trabalho. Requisitos funcionais e não funcionais, casos de uso e o modelo entidade relacionamento são apresentados nos apêndices.

2 Fundamentação teórica

Neste capítulo serão apresentados conceitos, tecnologias e funcionalidades do sistema operacional Android, necessários para o desenvolvimento desse trabalho.

2.1 Tecnologias e serviços de conectividade

Nessa seção serão apresentados alguns padrões de comunicação sem fio, suas definições, assim como, suas vantagens e desvantagens para o cenário da aplicação desenvolvida neste trabalho.

2.1.1 Telefonia Celular

A telefonia celular é um sistema de transmissão de dados sem fio, que opera sobre as redes de telefonia celular. Essa tecnologia teve início com a implementação da primeira geração, conhecida como 1G. Essa primeira geração foi desenvolvida somente para comunicação por voz. Ao longo do tempo, o sistema 1G foi substituído pelo 2G, na qual foi projetado para voz e suporte de dados. Posteriormente veio o sistema 3G, que também suporta voz e dados, porém com ênfase crescente em capacidade de dados e com uma velocidade maior ([KUROSE; ROSS, 2012](#)).

De acordo com a ([OPENSIGNAL, 2015](#)), as operadoras de telefonia disponibilizam conexão 3G com uma média de taxa de dados de 1,7 Mbps enquanto o 4G apresenta uma média de 11,7 Mbps. Apesar da discrepância de velocidade a área de cobertura é um fator relevante no momento da contratação do serviço e de seu uso. Segundo pesquisas da ([TELECO, 2015](#)) 76,4% dos municípios são cobertos com tecnologia 3G, o que representa 94,2% da população brasileira.

Para ter acesso a transmissão de dados via telefonia celular o usuário necessita de um aparelho compatível com as tecnologias 3G ou 4G, tendo sua conexão automaticamente estabelecida sem a necessidade de uma configuração extra, como por exemplo senhas ou a seleção de uma rede disponível. A contratação de um plano ou de um pacote de dados é necessária, como também estar dentro de uma área de cobertura de algumas dessas tecnologias, 3G ou 4G.

No que tange a velocidade de transmissão de dados, ela pode sofrer oscilações e variações conforme condições topográficas e/ou outros fatores que interfiram no sinal de transmissão, como condições climáticas, velocidade de movimento e a quantidade de clientes utilizando o serviço na mesma célula de rede¹. Geralmente os pacotes de dados

¹ Área de cobertura de uma estação rádio base

são comercializados com uma franquia mensal, o usuário fica sem o serviço de dados assim que consumir toda sua franquia. As operadoras também possibilitam ao cliente adquirir mais pacotes de dados.

2.1.2 Bluetooth

O padrão 802.15 popularmente conhecido como Bluetooth é uma tecnologia de rede pessoal sem fio, possuindo curto alcance para transmissão de dados entre dispositivos, com baixa taxa de transmissão e consumo energético reduzido. É possível interconectar aparelhos eletrônicos sem fio, tais como, telefones celulares, notebooks, fones de ouvido, *wearables*, equipamentos médicos, impressoras, entre outros. Esse protocolo utiliza a faixa *Industrial, Scientific and Medical* ([ISM](#)) e opera na frequência de 2,45 GHz, a qual é uma frequência de rádio aberta, aceita em praticamente em qualquer lugar do mundo ([ALECRIM, 2013](#)).

Em uma conexão Bluetooth, em que dois ou mais dispositivos se comunicam é criado uma rede pessoal, denominada *piconet*. Nesse tipo de rede, o dispositivo que iniciou a conexão assume o papel de mestre, enquanto os demais dispositivos são denominados escravos. É função do dispositivo que iniciou a conexão a tarefa de regular a transmissão dos dados que irá trafegar pela rede, além, de regular o sincronismo entre os dispositivos ([ALECRIM, 2013](#)).

Em uma rede *piconet*, é possível ter até oito dispositivos conectados simultaneamente, no qual, um será o mestre e os sete restantes serão os escravos. É possível elevar o número de dispositivos conectados entre eles, a partir da sobreposição de *piconetes*. Isso é possível fazendo com que uma *piconet* se comunique com outra *piconet* que esteja dentro do limite de alcance, esse processo é denominado de *scatternet*. Dentro de uma *scatternet* um escravo poderá fazer parte de mais de uma *piconet*, enquanto o mestre é responsável por sua própria *piconet*, não interagindo com outras redes ao seu redor ([ALECRIM, 2013](#)).

Os dispositivos que se conectam em uma rede Bluetooth são classificados de acordo com a potência e o alcance, em três classes: Classe 1 os dispositivos possuem potência máxima de 100 mW, e um alcance máximo de aproximadamente 100 metros; Classe 2 possui potência máxima de 2.5 mW e com um alcance máximo de aproximadamente 10 metros; Classe 3 permite aos seus dispositivos uma potência máxima de transmissão de até 1 mW enquanto seu alcance fica em torno de 1 metro.

Dispositivos de classes diferentes podem se comunicar e transmitir dados entre eles, porém a distância máxima entre eles deverá ser a distância do dispositivo que possui o menor alcance. Em um cenário onde ocorre somente a comunicação entre dispositivos móveis não teríamos esse problema, pois a maioria dos celulares, segundo ([ALECRIM, 2017](#)) trabalha com alcance de até 10 metros, ou seja, classe 2. No entanto, caso a comunicação

seja entre um celular e um fone de ouvido ou qualquer outro dispositivo com Bluetooth Classe 3 eles irão se comunicar se a distância entre eles for inferior a 1 metro. Apesar de ser uma distância curta, já é o suficiente para um fone de ouvido se conectar a um celular, por exemplo ([ALECRIM, 2013](#)).

A velocidade de transmissão de dados entre dispositivos utilizando a tecnologia Bluetooth é relativamente baixa: até a versão 1.2, a taxa pode alcançar, no máximo, 1 Mbps. Na versão 2.0, esse valor chega a alcançar 3 Mbps. Embora essas taxas sejam baixas, são suficientes para uma conexão satisfatória entre a maioria dos dispositivos. O Bluetooth está em constante evolução, a versão 3.0 é capaz de atingir taxas de até 24 Mbps enquanto a versão 4.0 possui a mesma taxa de transmissão, porém com maior eficiência energética.

O Bluetooth 4.0, ou também conhecido como BLE, é uma tecnologia de rede compatível com as versões anteriores ao Bluetooth 3.0 porém com menor consumo de energia e o mesmo alcance. Esta tecnologia foi desenvolvida para aplicações que apenas precisam enviar poucas informações periodicamente, como transferência de dados de sincronização entre usuários ou um sistema de localização interno, através de beacons² que podem auxiliar na localização do cliente ou de seções de produtos do supermercado.

2.1.3 IEEE 802.11 ([Wi-Fi](#))

Presente em grande parte dos ambientes, as redes locais sem fio (WLAN) são uma das mais importantes tecnologias de acesso à Internet. O padrão de rede mais utilizado é o IEEE 802.11 popularmente conhecido como [Wi-Fi](#) ([KUROSE; ROSS, 2012](#)).

O [Wi-Fi](#) pode operar nas faixas de 2,4GHz e 5GHz, permitindo diferentes taxas de transmissão de dados, podendo chegar à 1.3Gbps e com um alcance na transmissão dos dados entre dispositivos eletrônicos de até 200 metros, dependendo da tecnologia do aparelho e do padrão que ele possui suporte ([PLAZA, 2014](#)).

A simplicidade e a versatilidade do [Wi-Fi](#) são tão grandes que se tornou viável o uso dessa tecnologia em diversos locais, tais como: supermercados, restaurantes, shoppings, residências, entre outros, necessitando apenas que o usuário tenha um dispositivo compatível com a tecnologia [Wi-Fi](#). Embora haja estudos que comprovem o benefício de possuir uma rede Wi-Fi, ainda não há estudos que mostrem a parcela de estabelecimentos comerciais que disponibilizam redes Wi-Fi para seus clientes. De acordo com o ([DEXI, 2016](#)) 79% dos lojistas que implementaram Wi-Fi como cortesia para seus clientes concluíram que é uma solução eficiente para os negócios, pois perceberam que obtiveram uma retenção maior de seus clientes dentro da loja e a percepção de um valor maior agregado no produto oferecido.

Para que o dispositivo do usuário possa se associar em uma rede Wi-Fi de um

² <https://endeavor.org.br/beacon/>

estabelecimento, este precisa passar pelo processo de autenticação, que pode ser feito de várias formas como: autenticação por senha, *Captive Portal* ou login social. Tratando-se de estabelecimentos de grande porte, é necessário incluir múltiplos pontos de acesso para permitir que o usuário desloque-se livremente sem a perda da conexão ou autenticação constante por parte do usuário.

2.2 Plataforma Android

Atualmente, de acordo com a pesquisa realizada pela IDC (2015) a plataforma móvel mais utilizada no mundo é o Android³. A plataforma Android desfruta hoje de um papel de destaque no mercado, tanto pela quantidade significativa de dispositivos produzidos como também por oferecer uma *Application Programming Interface (API)* rica, disponibilizando fácil acesso a vários recursos de hardware, tais como Wi-Fi e GPS, além de boas ferramentas para o desenvolvedor (MONTEIRO, 2013).

2.2.1 Permissionamento

O sistema de permissão nos dispositivos visa proporcionar segurança aos usuário limitando o acesso aos dados privados. No momento da instalação de um aplicativo o usuário é informado sobre quais recursos e dados do seu dispositivo móvel a aplicação está requisitando acesso. Se o usuário não concordar com os recursos e dados solicitados, este pode então interromper o processo de instalação do aplicativo.

No Android as permissões são classificadas em duas categorias: normais ou perigosas. As permissões normais não representam risco para a privacidade do usuário ou a operações do sistema e são concedidas automaticamente pelo sistema, por exemplo quando o aplicativo faz uso do Wi-Fi. Enquanto para as permissões classificadas como perigosas, é necessário a permissão explícita do usuário para a execução do recurso solicitado, se enquadram como permissões perigosas⁴ a utilização da localização do usuário, acesso ao espaço de armazenamento de dados entre outros.

Anteriormente à versão 6.0 do Android o sistema de permissão era muito abrangente, possibilitando que desenvolvedores mal intencionados inseriram códigos maliciosos camuflados dentro da aplicação ou abrindo brechas para que terceiros possam extrair dados privados dos usuários a partir dos recursos liberados. A partir da versão 6 do Android esse problema foi solucionado e agora a solicitação de permissões é feita durante a execução do aplicativo e não mais somente na instalação.

³ O sistema operacional Android possui 82,8% dos usuários de dispositivos móveis, entre abril e junho de 2015

⁴ <https://developer.android.com/guide/topics/permissions/requesting.html>

O sistema operacional Android, por padrão, não permite que nenhum aplicativo execute operações que possam impactar negativamente em outros aplicativos, no sistema operacional, ou no usuário; isso inclui realizar a leitura ou escrita de dados privados do usuário, em arquivos de outros aplicativos, realizar acesso a rede entre outros. Caso algum aplicativo necessite de algum recurso que é provido por outro, é necessário declarar as permissões que ele necessita no arquivo de *manifest*⁵ presente no projeto do aplicativo. Desta forma, o sistema Android se encarregará de informar ao usuário sobre a necessidade do acesso a dados externos e solicitará ao usuário permissão para a execução.

Existem diferentes formas para o pedido de aprovação de acesso aos recursos serem requisitados, na qual dependem da versão do sistema que o aplicativo irá executar. Aplicativos instalados em dispositivos que estejam executando o Android 5.1 ou uma versão anterior, para as permissões classificadas como perigosas listadas no *manifest* o usuário deverá conceder a permissão ao instalar o aplicativo; caso contrário, o sistema não permitirá instalá-lo. Caso, o aplicativo seja instalado em um dispositivo que esteja rodando o Android 6.0 ou uma versão posterior, o aplicativo irá solicitar aprovação para, cada, permissão perigosa listada no *manifest*. O usuário pode conceder ou negar cada permissão e o aplicativo deverá ser programado para lidar com casos onde nem todos os recursos solicitados foram liberados pelo usuário.

2.2.2 Geolocalização

Uma das vantagens dos aplicativos, que utilizam o recurso de *Global Positioning System (GPS)*, é que se tem a possibilidade de obter a localização do usuário. Dependendo das funcionalidades do aplicativo isso pode ser algo essencial. Por exemplo, o aplicativo que informa os postos de combustíveis com os melhores preços levando em consideração a sua localização, apresentando os postos mais próximos. Para desenvolver um aplicativo para Android que tenha reconhecimento de localização, podemos utilizar [GPS](#) e os sinais de 3G e [Wi-Fi](#).

O GPS apresenta a tecnologia mais precisa, entretanto, funciona apenas em locais abertos, com elevado consumo energético (bateria), além de haver demora no envio das informações ao usuário. As redes móveis, utilizam os sinais de 3G e de [Wi-Fi](#) para determinar a localização. Esse tipo de tecnologia funciona tanto em ambientes fechados quanto em ambientes abertos, tem resposta rápida e não consomem tanta bateria quanto o GPS. Além disso, o aplicativo tem a possibilidade de obter a localização do usuário, com maior precisão, utilizando ambas tecnologias discutidas acima ([ANDROID](#), b).

O desenvolvedor pode se deparar com alguns desafios ao tentar obter a localização do usuário, utilizando tanto as redes móveis quanto o [GPS](#). Essas informações podem

⁵ O arquivo de manifesto apresenta informações essenciais sobre o aplicativo ao sistema Android, necessárias para o sistema antes que ele possa executar o código do aplicativo.

conter erros e serem imprecisas, independente do local de leitura. Abaixo estão listados, alguns dos erros, listados pelo ([ANDROID, b](#)), que podem ocorrer ao tentar obter a localização do usuário:

- Muitas fontes de localização: GPS, Cell-ID, e Wi-Fi podem fornecer uma pista da localização do usuário. Determinar qual dessas tecnologias utilizar é uma estratégia para obter precisão, velocidade e economia de bateria;
- Deslocamento do usuário: o desenvolvedor deverá levar em conta as mudanças de localização do usuário, e deverá estimar de quanto em quanto tempo o aplicativo deverá obter a localização do cliente;
- Variação de precisão: as localizações obtidas de várias fontes não são consistentes na precisão. A localização obtida de uma fonte pode ser mais precisa que a de outras.

O [Google Places API](#) é um banco de dados, que de acordo com ([PLACE..., 2016](#)) tem em sua base de dados mais de 100 milhões de locais, que incluem empresas, pontos de interesse e localizações geográficas que são atualizados com frequência por meio de listagens confirmadas pelos proprietários e contribuições moderadas por usuários. Com essa [API](#) é possível saber os estabelecimentos próximos e filtrá-los de acordo com alguns parâmetros de entrada. Entre os parâmetros disponíveis estão: *location*, na qual obtém os estabelecimentos próximos através da latitude e longitude informadas; *rankby*, que traz uma resposta por ordem dos estabelecimentos mais relevantes; *type*, que retorna os estabelecimentos através do tipo informado, podendo ser cafeteria, banco, hospital, igreja, museu, supermercados entre outros.

Os parâmetros listados acima podem ser utilizados individualmente ou combinados, ou seja, é possível inserir vários parâmetros em uma mesma consulta e ela retornará a combinação de todos eles.

2.2.3 Notificações na interface do usuário

As notificações tem por objetivo alertar o usuário sobre eventos importantes assim que ocorrem e sobre registros que aconteceram e que para o cliente no momento do ocorrido o evento passou despercebido, por exemplo uma notificação de uma chamada perdida que por algum motivo o usuário não pode dar atenção. Para o usuário as notificações irão aparecer no dispositivo como um ícone na área de notificação, e geralmente são compostas por mensagens curtas. Para visualizar os detalhes da notificação será necessário que o usuário abra a gaveta de notificações, conforme ilustrado na [Figura 1](#).

De acordo com ([ANDROID, c](#)) as notificações são compostas basicamente por Ícone: simboliza o aplicativo que gerou a notificação, pode também indicar o tipo de notificação,

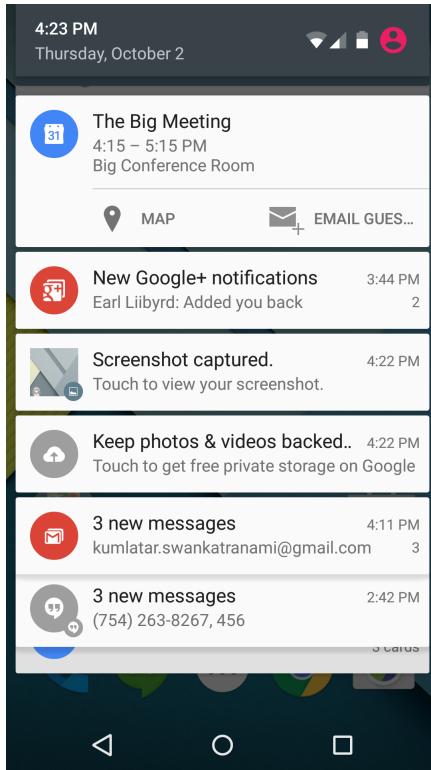


Figura 1 – Notificações na gaveta de notificação. Fonte: ([ANDROID, c](#))

caso o mesmo aplicativo gere mais de uma notificação; Título informa a atividade e um texto adicional enquanto a Data nos informa sobre a data e a hora da notificação

2.2.4 Material design

O Material design é um conjunto de boas práticas que possui como propósito inicial unificar o visual do Android com os aplicativos do Chrome OS e com os serviços web do Google, além de melhorar a experiência do usuário com a aplicação ([DUARTE, 2015](#)).

Atualmente, os profissionais da área de desenvolvimento de aplicativos procuram saber mais sobre o contexto de utilização de seu aplicativo pelo seu público alvo. Pois a experiência do usuário está cada dia mais relevante na construção e no desenvolvimento de um aplicativo. Para isso é necessário extrair informações sobre como o usuário irá utilizá-lo, auxiliando assim o desenvolvimento de um produto com uma melhor usabilidade e eficiência de uso pelo usuário, engajando-o e fazendo com que ele compartilhe suas experiências positivas com outros usuários ([ANDROID, a](#)).

Ao utilizar um aplicativo o usuário deseja uma consistência em sua experiência com o celular, ele deseja utilizar o aplicativo sem a necessidade de aprender um novo fluxo de navegação ou no momento de localizar opções de configurações em locais escondidos, ou até mesmo realizar uma ação básica como adicionar um novo item. Cognitivamente o usuário tende a ter uma experiência melhor com o produto se ele estiver no padrão,

de maneira simples e intuitiva (MATERIAL..., 2017). É simples de entender ao fazer uma analogia com um motorista brasileiro que irá dirigir um carro na Inglaterra. Nos primeiros dias ele irá se sentir desconfortável e provavelmente fará "algo errado". Da mesma forma um usuário que interage com uma aplicação na qual não respeita a linguagem de seu ambiente fazendo com que o usuário fique desorientado, pois está acostumado com elementos visuais da plataforma.

Desta forma, o material design vem ao encontro dessas necessidades levantadas, resultando em uma aplicação com um fluxo constante tanto na navegação quanto na utilização. Desta forma seguir o material design, não é somente uma forma de deixar o aplicativo visualmente mais agradável, é trazer toda uma experiência ao usuário para deixá-lo confortável na utilização do aplicativo. Além disso, segundo Cordeiro (2015) o Google não realizou somente um esforço para deixar todos os Androids parecidos, o Google chegou ao Material Design por meio de muita experimentação e observando como os usuários interagiam com os elementos de seus telefones inteligentes. Com base nessas experiências, buscou criar uma linguagem de design que fosse o mais simples e acessível possível, podemos ver em prática a utilização do material design na Figura 2, onde o aplicativo de leitura de e-mails do Google passou a utilizar um novo visual seguindo as recomendações do *Material Design*

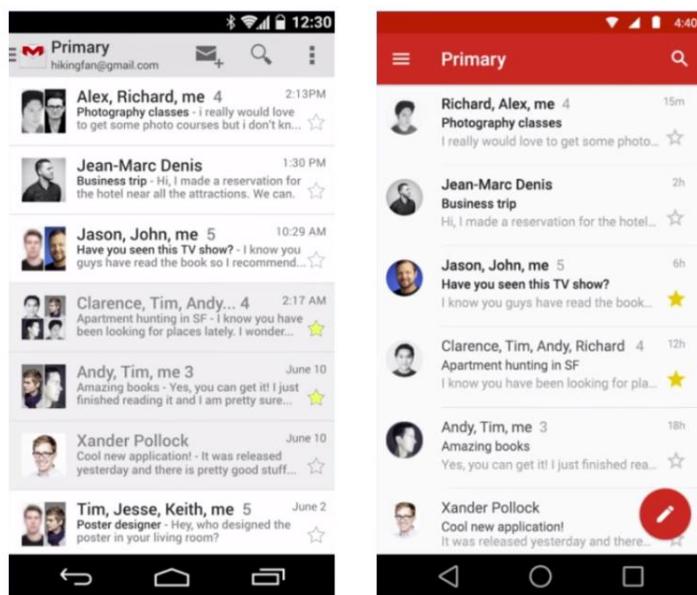


Figura 2 – Aplicação do material design no Gmail. Fonte: (WELCOME..., 2014)

Os elementos disponíveis não foram projetados pensando apenas em uma aparência e um visual elegante, outros elementos como o conforto foram levados em consideração, por exemplo a tipografia, o material design faz com que o conteúdo possa ser lido confortavelmente em qualquer dispositivo, independente do tamanho; as animações tornam as funcionalidades mais intuitivas, não sendo criadas apenas para impressionar; o conteúdo se

ajusta automaticamente ao tamanho da tela de forma a aproveitá-la de maneira inteligente.

Para auxiliar a aplicação dessas boas práticas em uma aplicação real, o Google disponibiliza uma biblioteca denominada *Design Support Library*⁶ que auxilia no desenvolvimento de aplicações que sigam as especificações propostas.

2.3 Sincronização de dados com serviços na nuvem

Atualmente diversos serviços estão disponíveis aos desenvolvedores para auxiliar no processo de desenvolvimento das aplicações. Entre esses serviços, um específico vem evoluindo no ramo de desenvolvimento *mobile*, o *Backend as a Service* (*BaaS*).

O *BaaS* é um modelo de serviço de computação em nuvem que serve como um intermediário, que fornece aos desenvolvedores maneiras de conectar seus aplicativos Web e aplicações móveis a serviços disponíveis em servidores remotos através das interfaces de programação de aplicativos (APIs) e kits de desenvolvedores de software (SDKs), abstraindo assim a infraestrutura do lado do servidor. Desta forma os desenvolvedores se concentram mais na experiência do usuário com a aplicação ao invés de lidar com a infraestrutura e codificação do *backend*.

Entre as vantagens de se utilizar o *BaaS* está a economia de dinheiro e menor tempo de desenvolvimento, pois com esse serviço é possível utilizar infraestrutura com servidores remotos, diminuindo o consumo com manutenção com servidores, além de dispensar horas de um desenvolvedor backend. Outra vantagem de se utilizar o *BaaS* é de possuir ao seu dispor um servidor que possa escalar seu processamento e armazenamento de dados conforme a demanda dos usuários da aplicação, além de possuir os mais atuais protocolos de segurança em suas APIs, possibilitando assim maior segurança nos dados dos usuários sem a necessidade de ter um desenvolvedor para implementar esse nível de segurança. Entre os principais recursos fornecidos pelo *BaaS* estão, banco de dados em tempo real, armazenamento de dados, integração com redes sociais possibilitando autenticação, postagem e compartilhamento de conteúdo com mais facilidade, notificações, testes automatizados, relatório de erros, entre outros.

Entre as desvantagens há o controle sobre o código fonte e o acesso ao backend ser mais restrito. Outro fator negativo e que o desenvolvedor deve ser cauteloso e avaliar de forma criteriosa é ao contratar os fornecedores do serviço. É uma decisão de longo prazo e deve ser dada preferência a fornecedores que permitam a migração de dados caso haja necessidade ou caso a empresa pare de fornecer suporte ao serviço.

⁶ <https://developer.android.com/training/material/design-library.html>

2.4 Trabalhos Relacionados

Os aplicativos disponíveis para a plataforma Android podem ser encontrados na *Google Play Store*, dentre os diversos aplicativos disponíveis há uma vasta gama destinados a lista de compras. Dentro desse grupo, foram selecionados os aplicativos que possuem um número expressivo de *downloads*, além de uma avaliação positiva com uma média maior do que quatro pontos, dentro de uma escala de zero à cinco. Outros aspectos levados em consideração foram a presença das funcionalidades semelhantes a proposta desse projeto, na qual servirão como base para a elaboração do aplicativo proposto. Entre as funcionalidades analisadas estão: execução da compra; presença de lista colaborativa; histórico de compras; indicação do mercado; categorização dos produtos por seções.

Na funcionalidade de execução da compra é considerado que o aplicativo irá tornar uma lista de compras estática em algo dinâmico, permitindo que o usuário interaja com a lista durante as compras, incluindo ou excluindo itens, como também alterando o status de alguns itens e marcando-os como pegas ou faltantes no estabelecimento. A funcionalidade da lista colaborativa acontece durante a execução de uma compra, onde é compartilhado uma lista de compras com outros usuários, e é esperado que os usuários possam interagir entre eles em tempo real. Essas interações incluem, inserção ou exclusão dos itens na lista de compras, como também alteração do status dos produtos, sincronizando essas modificações com todos os membros que compartilham dessa lista.

Dentre os aplicativos disponíveis foram selecionados os seguintes: *Grocery List - Tomatoes*, *Listonict*, *Out of Milk* e *Bring*.

2.4.1 Grocery List - Tomatoes

O *Grocery List*⁷ é um aplicativo disponível para Android e iOS, possuindo ainda uma versão web. Ele possui uma versão paga e outra gratuita. Em sua versão gratuita é possível criar somente uma lista de compras e os itens são adicionados através de seus nomes. Em seu modo de execução de compras, o aplicativo utiliza-se da mesma lista estática, possuindo seus itens separados por seções, com um acréscimo de algumas funcionalidades que realizam a inclusão, e exclusão dos itens da lista. É possível visualizar o histórico de compras, como também inserir o nome do supermercado no qual a compra foi realizada. Por fim o aplicativo apresenta compartilhamento da lista, sendo possível enviar a lista a outros usuários através do endereço eletrônico, importando assim essa lista no aplicativo de outros usuários.

⁷ <https://play.google.com/store/apps/details?id=com.meucarrinho>

2.4.2 Listonic

*Listonic*⁸, é um aplicativo disponível para Android, iOS e Windows Phone. Esse aplicativo tem como proposta auxiliar os usuários a organizarem suas listas de compras de forma inteligente e prática. Esse aplicativo apresenta um modo de execução de compras na mesma interface da lista de produtos, sendo possível marcá-los como pegas ou não. Na mesma lista de produtos ainda é possível adicionar anotações aos produtos, podendo inserir informações como, preço, quantidade e disponibilidade na loja. Todos os produtos presentes na lista possuem uma categorização, sendo possível realizar o ordenamento dessa lista por agrupamento de seções ou por ordem alfabética. O Listonic ainda apresenta o compartilhamento da lista de compras, permitindo a sincronização dos itens em tempo real, podendo marcar o item como pego ou alterando o status de um produto com outros usuários. Por fim, o Listonic apresenta em sua tela principal as listas de produtos já criadas pelo usuário, que é o histórico das compras.

2.4.3 Out of Milk

*Out of Milk*⁹, é um aplicativo tanto para o sistema operacional Android quanto para iOS. Esse aplicativo apresenta como proposta auxiliar os usuários a manter o controle de suas tarefas e de suas listas de compras. Esse aplicativo realiza a criação da lista de produtos e possibilita a execução da compra na mesma tela, apresentando no rodapé da tela um gerenciador de compras, onde está disponível a quantidade de produtos presentes na lista e a quantidade de itens já pegas. Ainda é possível alterar os status dos produtos bem como listar os produtos por categorias. O compartilhamento da lista de produtos é possível através da utilização do endereço de e-mail, permitindo a sincronização dos itens inseridos no carrinho ou não, como também seu status. O aplicativo apresenta em seu menu lateral uma listagem de todas as listas já criadas, possibilitando assim uma visualização de suas listas já utilizadas em compras.

2.4.4 Bring

*Bring*¹⁰, é um aplicativo disponível para Android e iOS. Esse aplicativo traz como proposta auxiliar os usuários a criarem suas listas de compras de forma rápida e elegante. O modo de execução das compras opera em conjunto com a criação da lista de compras, não havendo assim uma separação lógica entre a criação da lista de compras e a ação da compra. Seus itens são expostos através de imagens, não possuindo a possibilidade da alteração dos status dos produtos. Seus produtos não são ordenados por categoria, sendo todos expostos lado a lado dentro da lista. Esse aplicativo apresenta compartilhamento da

⁸ <<http://www.listonic.com/content/en>>

⁹ <<https://www.outofmilk.com/>>

¹⁰ <<http://www.getbring.com/>>

lista de compra, permitindo sincronização em tempo real dos itens com a possibilidade de envio de notificações aos outros usuários. No menu lateral estão disponíveis as listas já criadas pelo usuário, não possuindo o conceito de finalização de uma lista para geração de um histórico de compras.

2.4.5 Resumo comparativo

Com base nos aplicativos analisados, observa-se que a funcionalidade de lista colaborativa está presente em três dos quatro aplicativos analisados. Dentre os aplicativos que possuem a colaboração da lista de compras, o aplicativo Bring se destaca entre eles, trazendo uma sincronização rápida e sem interação com o usuário, permitindo ainda o envio de notificações quando há alterações na lista. Os outros aplicativos, Listonic e Out of Milk, possuem a sincronização de forma lenta, sendo necessário ainda algumas vezes realizar uma intervenção manual para ocorrer a atualização. Por fim, o aplicativo Grocery list realiza somente o compartilhamento dos itens presentes na lista de compras, não realizando efetivamente a sincronização dos status dos produtos.

No contexto da funcionalidade de execução das compras, nenhum aplicativo apresenta uma clara informação se a lista de produtos está pronta para execução de uma compra, sendo possível marcar itens como pegos ao mesmo tempo em que inserimos novos na lista. O aplicativo Out of Milk apresenta o melhor de conceito de execução de compras entre os quatro aplicativos analisados, ele introduz um painel de gerenciamento de compras no rodapé da lista, informando a quantidade de itens disponíveis na lista e a quantidade já inseridas no carrinho de compras.

Dentro os aplicativos analisados, somente o Grocery List possui um histórico de compras bem definido, permitindo ao finalizar a lista incluir algumas informações como o nome do mercado. Os outros aplicativos analisados apresentavam apenas uma listagem das listas já criadas, podendo modificá-las a qualquer momento, incluindo assim novos itens e iniciando uma nova compra. Nenhum dos aplicativos testados apresentou alguma forma de ordenação de acordo com o mercado em que o cliente está realizando suas compras. Foi visto que é possível agrupar os itens de uma lista por seções ou ordená-los por ordem alfabética, mas nada a otimizar o tempo. Além disso, outro aspecto que não foi encontrado em nenhum dos aplicativos foi a indicação dos mercados próximos, na qual o cliente possa inserir em sua lista o local que deseja realizar suas compras ou até mesmo para que possa simular uma ordenação de sua lista de compras de acordo com o mercado correspondente. A [Tabela 1](#) permite comparar os aplicativos apresentados nessa seção, contando com o trabalho proposto denominado Compra Rápida, através de suas características mais relevantes para o desenvolvimento deste trabalho.

	Listas Colaborativas	Execução das Compras	Histórico das Compras
Grocery list	indisponível	atrelado a lista	possui
Out of Milk	tempo real	atrelado a lista	possui
Bring	tempo real	atrelado a lista	indisponível
Listonic	tempo real	atrelado a lista	possui
Compra Rápida	indisponível	lista modelo	possui
	Categorização de itens	Mercados próximos	
Grocery List	indisponível	indisponível	
Out of milk	por seções	indisponível	
Bring	indisponível	indisponível	
Listonic	por seções	indisponível	
Compra Rápida	por seções	Google places API	

Tabela 1 – Resumo comparitivo entre os aplicativos.

3 Compra Rápida

Existem disponíveis para a plataforma Android diversos aplicativos destinados à criação de listas de compras para supermercados, que buscam substituir as tradicionais listas feitas à mão. Esses aplicativos buscam auxiliar o usuário a confeccionar a sua lista de compras, deixando-o preparado para a execução das compras. No entanto, durante as compras, o usuário não possui auxílio do aplicativo, pois apesar de algumas funcionalidades existirem para ordenamento dos produtos elas não são destinadas para otimizar as compras no supermercado. Ou seja, apesar das tecnologias terem evoluído, os aplicativos atuais continuam copiando as tradicionais listas em papéis, onde a criação da lista é função principal, conforme mostra a [Figura 3](#).



Figura 3 – Tipos de listas de compras.

Ao iniciar uma compra, tipicamente os itens presentes na lista estão ordenados de maneira aleatória, seguindo a ordem que foi ditada pela memória do usuário no momento da confecção da lista, como mostra a [Figura 3a](#). Isso acaba dificultando a compra dos produtos, pois esses itens não estão seguindo nenhuma ordem existente no mercado em questão, fazendo com que o usuário verifique a todo momento, se ainda há mais algum item na lista que pertence à seção atual que ele se encontra.

Esse processo de verificar a lista de compras integralmente a cada nova seção do supermercado, a fim de verificar se nenhum dos itens listados foi esquecido, faz com que o cliente despenda mais tempo durante suas compras. Há aplicativos que trazem a opção de agrupar os itens por categoria, como na [Figura 3b](#), mas visto que os mercados não seguem um padrão de organização das seções, essa opção não evita que a cada seção do mercado o usuário tenha que percorrer a lista de compras novamente.

Tendo esses problemas atuais como motivação, o Compra Rápida traz como principal objetivo auxiliar o processo de compras do usuário economizando tempo, através do ordenamento da lista de compra de acordo com o mercado onde o cliente se encontra. Auxiliando desta forma o usuário, indicando uma ordem a ser seguida dentro dos mercados de acordo com os itens presentes em sua lista de compras, a fim de minimizar o tempo despendido durante as compras.

Na [Figura 4](#) é ilustrada a ordenação das seções de um supermercado hipotético, onde o caminho indicado pela linha preta ilustra o percurso que um usuário normalmente percorre durante as compras. Em compras que o usuário utiliza uma lista comum, onde os produtos da lista não estão agrupados, é possível que um usuário passe despercebido por um dos itens, pois está abaixo na lista, forçando o usuário a retornar a uma seção já visitada para buscar esse produto. O aplicativo Compra Rápida visa auxiliar os usuários a percorrerem de modo contínuo as seções do mercado, evitando retornar a alguma seção já visitada através do agrupamento dos itens pelas seções, e ordenando essas seções de acordo com o mercado em questão.

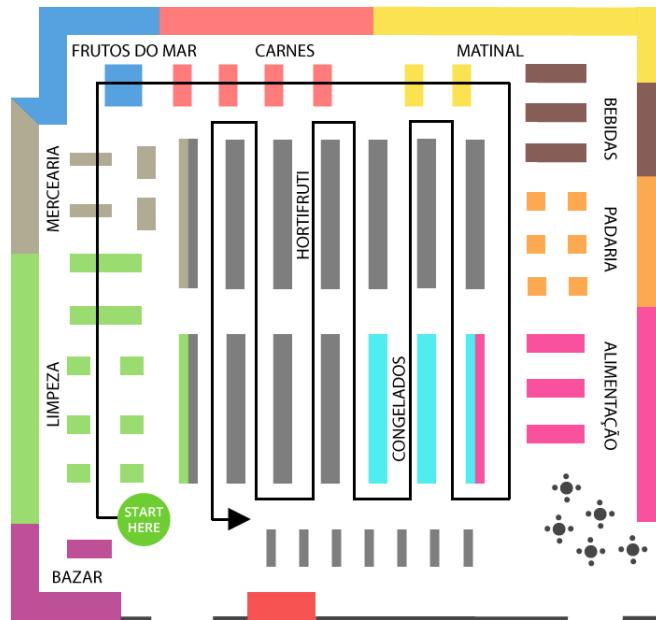


Figura 4 – Planta baixa do mercado.
(DETTO, 2015)

Nas próximas seções são apresentadas as funcionalidades, as ferramentas usadas para concepção da solução e por fim o algoritmo usado para fazer a ordenação da lista de compras durante sua execução.

3.1 Funcionalidades

O aplicativo Compra Rápida possui seis funcionalidades disponíveis aos usuários na tela principal da aplicação. Cada funcionalidade é acessível através de botões, conforme

ilustrado na [Figura 5](#). Nessa figura é possível observar as seguintes funcionalidades: ir às compras, listar listas, listar mercados, listar produtos, listar histórico de compras e criar uma lista, representado por um botão rosa, com um símbolo de "+" centralizado.

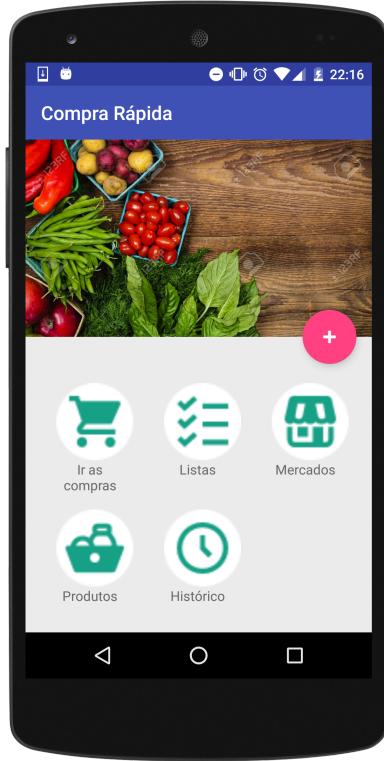
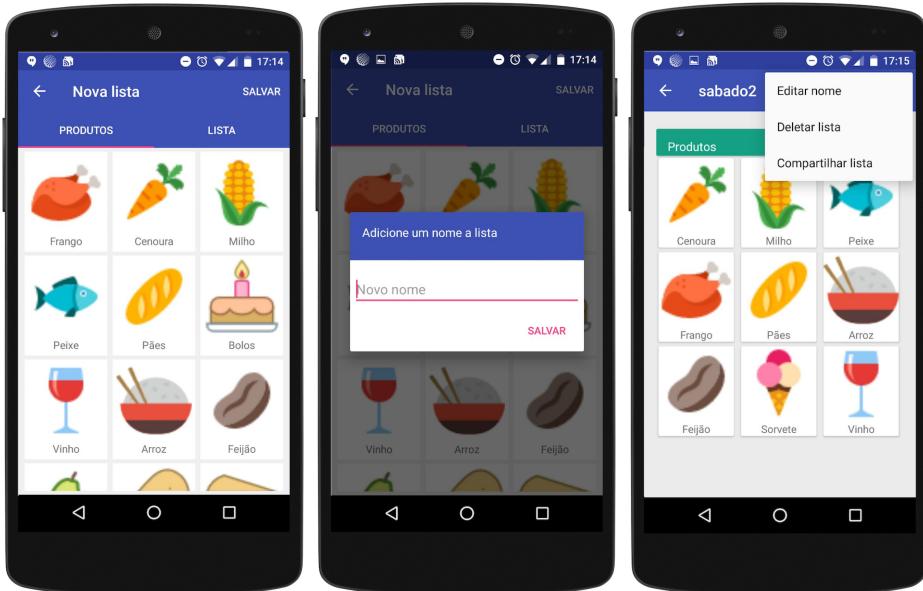


Figura 5 – Menu Principal.

Através da funcionalidade de criar listas ([Figura 6](#)), o usuário inserirá itens na lista de compras, conforme apresentado na [Figura 6a](#) para serem posteriormente utilizados em compras futuras, sendo referenciados pelo nome da lista, que é um campo obrigatório de preenchimento ([Figura 6b](#)). Ainda na tela de criação da lista, é possível visualizar os produtos já inseridos e os produtos que ainda podem ser adicionados na lista, para facilitar a visualização foi decidido separar os produtos já inclusos e os produtos que podem ser adicionados em abas diferentes.

A funcionalidade listas, permite visualizar todas as listas já criadas e disponíveis para a execução de uma compra, como também selecionar uma delas para realizar a edição do nome ou a sua exclusão conforme [Figura 6c](#). Todas as listas disponíveis podem ser compartilhadas através das aplicações que o usuário possui em seu celular, como, por exemplo, mensageiros instantâneos, serviços de e-mail ou mídias sociais, enviando dessa forma uma listagem dos itens presentes em sua compra.

A aplicação ainda possibilita a visualização dos itens disponíveis em sua base de dados, como também as categorias a qual cada item pertence. Isso ocorre através da funcionalidade de listagem dos produtos, como ilustra a [Figura 7](#), listando os itens por ordem alfabética e agrupados por seções.



(a) Criar lista.

(b) Salvar lista.

(c) Nova lista.

Figura 6 – Etapas para criar uma lista.

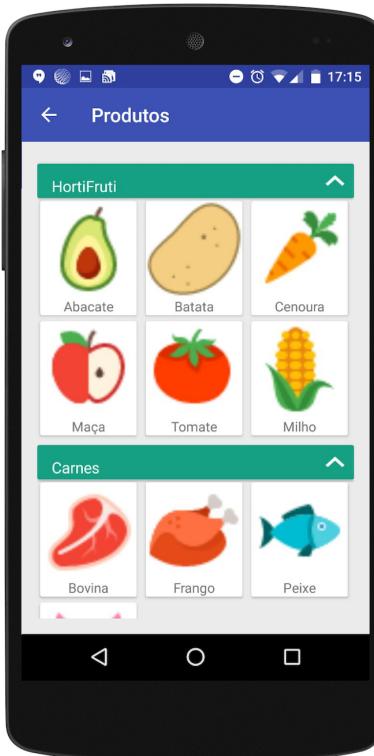


Figura 7 – Produtos.

A funcionalidade de listagem dos mercados ocorre através da utilização do aparelho GPS do usuário e de uma conexão com a Internet, nessa listagem estão inclusos os mercados dentro de um raio de 5km metros de proximidade com o aparelho celular do usuário e estão ordenados pelo grau de avaliação dos usuários dados ao Google [Figura 8](#).

Por fim, o aplicativo ainda possui a funcionalidade de ir às compras, onde o usuário

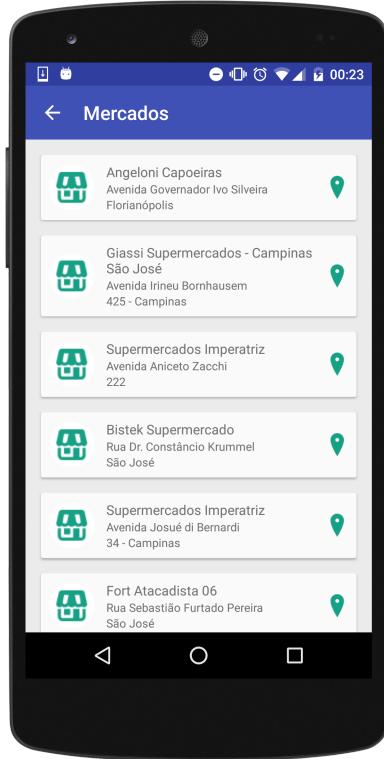


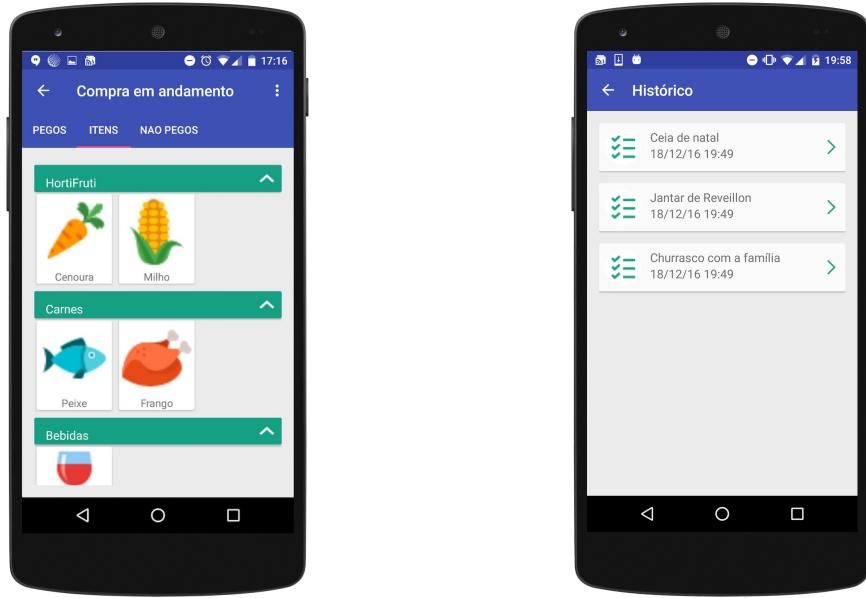
Figura 8 – Lista de mercados.

informa qual lista irá utilizar para realizar as compras, e o mercado na qual a compra será realizada, a fim de otimizar as seções dos produtos presentes na lista para o mercado especificado pelo usuário. Diferente de uma lista de produtos, uma lista de compras permite com que o usuário altere o estado de um produto ao longo das compras, podendo indicar a inserção de um produto no carrinho de compras, ou informar a indisponibilidade desse produto, conforme ilustra [Figura 9a](#). Ao finalizar a compra, o usuário finaliza a lista de compra, inserindo-a no histórico de compras, porém não alterando a lista de produtos [Figura 9b](#).

3.2 Base de dados de produtos e categorias

Uma necessidade comum em todos os aplicativos de compras é a utilização de uma base de dados para o armazenamento dos produtos que estão disponíveis no aplicativo. Essa base de dados pode conter textos descritivos dos produtos ou imagens ilustrativas dos mesmos, assim como a categorização de cada um dos produtos. Uma estratégia que precisa ser definida pelo desenvolvedor da aplicação é a forma de armazenamento dessa base de dados, tendo a possibilidade de armazená-la localmente ou requisitar os dados através de serviços externos.

Em uma base de dados estática, o usuário não necessita de conexão com a Internet, pois todos os dados estão presentes localmente na aplicação. No entanto, caso ocorra



(a) Compra em andamento.

(b) Histórico de Compras.

Figura 9 – Etapas de uma compra.

```
{
  "categories": [
    {
      "id": 1,
      "name": "Hortifruti"
    },
    {
      "id": 2,
      "name": "Carnes"
    },
    {
      "id": 3,
      "name": "Lacticínios"
    }
  ],

  "products": [
    {
      "category": {
        "id": 1,
        "name": "Hortifruti"
      },
      "id": 1,
      "name": "Abacate"
    },
    {
      "category": {
        "id": 1,
        "name": "Hortifruti"
      },
      "id": 2,
      "name": "Batata"
    }
  ]
}
```

Figura 10 – Objeto JSON.

alguma atualização na base de dados, como a inclusão de novos produtos, ou alteração de algum produto já existente, o usuário irá necessitar atualizar sua aplicação para possuir esses novos dados. Enquanto, em uma solução que utiliza uma base de dados remota, todos os usuários estarão com sua base de produtos atualizadas de acordo com a última atualização disponível. Essa solução remota proporciona uma dinamização da aplicação, pois não requer que o usuário atualize manualmente o aplicativo. Entretanto, traz como desvantagem a indisponibilidade de acesso aos dados caso ocorra uma falha com a conexão de rede.

Desta forma, o aplicativo Compra Rápida segue uma estratégia de armazenamento de dados híbrido, onde os dados serão requisitados a um serviço externo, garantindo assim a atualização dos produtos, enquanto que ao mesmo tempo esses dados serão persistidos no banco de dados local da aplicação, garantindo assim o acesso aos dados mesmo sem conexão. Como solução para o serviço de armazenamento de dados externo, a aplicação utiliza o serviço de banco de dados em tempo real provido pelo Firebase, o acesso a essas informações é através de um serviço *REST*, que transmite os dados no formato *JSON*. Enquanto a persistência de dados, ocorre através da utilização do banco de dados do *SQLite*, que recebe os dados e armazena-os na tabela de produtos e categorias disponíveis na aplicação. A estrutura utilizada localmente está armazenada em um banco de dados relacional, sendo assim a estrutura utilizada são tabelas na qual cada uma representa uma entidade específica. As únicas tabelas que sofrem alterações a cada atualização que o aplicativo recebe do Firebase são as tabelas de produtos e categorias, todas as outras são tabelas com informações específicas do cliente que não precisam ser sincronizadas. Toda a estrutura do banco de dados relacional está descrito no [Apêndice C](#).

Como mencionado anteriormente, a estrutura dos dados armazenados no Firebase encontra-se no formato *JSON*, conforme ilustrado pela [Figura 10](#). As entidades armazenadas nessa estrutura são somente os produtos e categorias, pois para essa primeira versão do Compra Rápida os únicos dados que têm a necessidade de estarem sincronizados com todos os clientes são os produtos e categorias, conforme mostrado na [Figura 11](#). Desta forma a cada nova atualização realizada pelo administrador da base de dados, como ilustrado na etapa 1 da [Figura 11](#), ocorrerá a sincronização dos dados presentes nas aplicações conforme ilustrado na etapa 2 da [Figura 11](#) através do envio de um pacote que possui o formato *JSON*.

Essas atualizações ocorrem utilizando o conceito de *publish* e *subscribe*, de tal forma que as aplicações recebem somente as modificações nas entidades de seu interesse. Nesta primeira versão, as aplicações estão somente recebendo atualizações dos produtos e categorias, como mencionado anteriormente. Para o recebimento das atualizações é necessário conexão com a Internet, caso a aplicação não esteja conectada à Internet no momento em que o servidor altera esses dados, o recebimento ocorrerá posteriormente sem

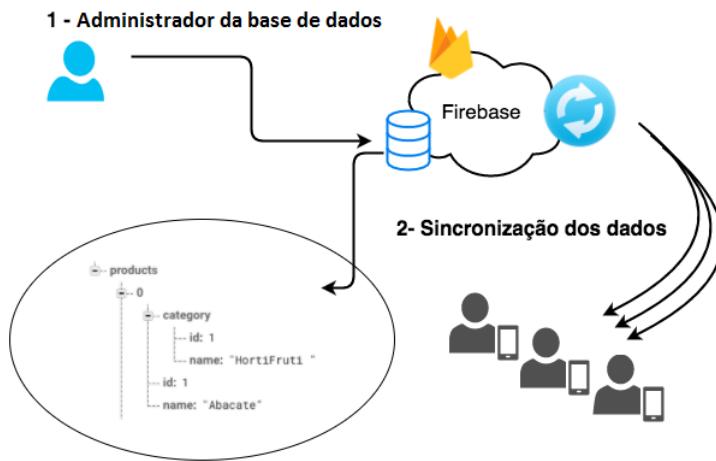


Figura 11 – Sincronização dos dados com todos os clientes.

a perda de nenhum dado.

3.3 Algoritmo para ordenação da lista de compras

O fluxo normal de uma compra ocorre a partir da criação de uma lista de produtos, através da opção criar lista, como mencionado no início deste capítulo. Na tela de criação da lista de produtos, o usuário terá acesso a todos os produtos cadastrados na base de dados da aplicação. O usuário irá escolher os produtos que ele deseja inserir na lista e após o término, deve-se salvar a lista de produtos. Uma vez salva essa lista, ela poderá ser acessada novamente na opção de visualização das lista, onde é possível visualizar os produtos já inseridos, ou alterar o nome da lista.

Após o usuário já possuir disponível uma lista de produtos, ele poderá iniciar uma compra. Para o início da compra é necessário informar a lista de produtos que será utilizada, e o nome do mercado na qual a compra será efetuada. Essas informações são essenciais para o funcionamento do algoritmo elaborado para esse trabalho, pois ele irá cruzar as informações das seções presentes na lista, com as compras já realizadas no mercado selecionado pelo usuário. De acordo com o histórico de compras, será possível reorganizar a lista de produtos, reaproximando as seções para se aproximar o máximo possível com o mercado selecionado, essa aproximação é dependente do número de compras já realizadas nesse mercado, desta forma quanto mais compras, com mais itens de seções diferentes possuir a compra, a reorganização dos produtos será mais precisa em uma próxima ida ao mercado.

Caso o usuário esteja realizando uma compra pela primeira vez em um determinado mercado, a aplicação irá ordenar as seções pela ordem ditada pela consulta ao banco de dados local. Porém se uma determinada seção nunca esteve presente em suas compras passadas, no momento que o usuário inserir essa seção ela irá para ao final da lista, pois

não há dados suficientes para se tirar alguma conclusão a respeito do posicionamento da seção na lista.

Nota-se que o algoritmo não é influenciado por compras realizadas em mercados distintos, cada mercado possui seu histórico e suas formas de análise para reorganização das seções presentes no supermercado. Além disso, nessa primeira versão do aplicativo, as compras efetuadas pelos usuários não contribuirão para otimizar a compra de outros usuários que também estejam se utilizando do mesmo aplicativo.

O sistema de ordenação proposto nesse trabalho, possui três estados de atuação, que são: início das compras, inserção dos produtos e finalização das compras. Os estados são acionados de acordo com a interação do usuário com a aplicação. O estado início das compras ocorre logo após o usuário iniciar a execução de uma compra, ordenando os produtos da lista. O estado de inserção dos produtos ocorre durante a compra, extraíndo informações sobre a ordem de inserção dos produtos no carrinho de compra, e por fim, o estado finalização das compras ocorre quando o usuário seleciona a opção de finalizar a compra, inicializando desta forma os cálculos da distância entre as seções presentes na lista.

No estado início das compras o usuário indica qual lista, criada previamente, será executada e em qual supermercado. Nesse momento é feita a ordenação da lista de produtos de acordo com o mercado em que se encontra. Para isso ocorrer, o algoritmo irá verificar previamente se o mercado em questão já possui compras passadas, caso positivo, é feito a ordenação conforme descrito no Algoritmo 1. O ordenamento das seções presentes na lista de produtos ocorre através das distâncias entre as seções, conforme demonstrado na linha 3 do Algoritmo 1. Nessa linha, é verificado qual seção foi selecionado pelo usuário. Essa seção por sua vez, será a referência para início do ordenamento da lista. As seções seguintes serão ordenadas de acordo com a distância em relação a seção referência, sendo

que a seção referência será a seção com menor distância na iteração anterior.

```

Entrada: lista_id, seção_referência, mercado_id
Saída: Lista ordenada de seções de acordo com o mercado_id

1 seções ← seções presentes na lista_id;
2 seções_tmp ← seções;
3 seções_ordenadas ← seção_referência
4 enquanto i < tamanho do conjunto seções faz
5   seções_tmp ← remova seção_referência do conjunto seções_tmp;
6   proxima_seção ← seção presente no conjunto seções_tmp que possui o
     menor valor no campo distância em relação a seção_referência;
7   seções_ordenadas ← seções_ordenadas + proxima_seção ;
8   seção_referência ← proxima_seção;
9 fim
10 Retorna seções_ordenadas.

```

Algoritmo 1: Ordenação das seções presentes na lista.

O segundo estado do algoritmo ocorre enquanto o usuário realiza as compras, quando o usuário marca um produto como pego, é indicado ao aplicativo a ordem de inserção desse produto no carrinho de compras. Fazendo com que desta forma, seja possível ter uma ideia inicial da ordem em que as seções estão sendo visitadas pelo usuário. Por exemplo, caso um usuário pegue um produto da seção *Mercearia* e em seguida pegue um produto da seção *Limpeza*, a seção *Mercearia* é classificada como a primeira seção visitada, enquanto, a seção *Limpeza* é considerada a segunda seção visitada. Essas informações sobre a ordem de visitação das seções são utilizadas pela última etapa do ordenamento das seções, onde são realizados os cálculos das distâncias entre as seções.

A rotina de extração da ordem das seções do mercado, ocorre no terceiro e último estado do algoritmo, quando o usuário finaliza as compras. Nesse estado, representado pelo Algoritmo 2, são utilizadas todas as informações extraídas durante as compras anteriores, guardadas como a média das distâncias entre as seções em compras passadas, como descrito na linha 9 do Algoritmo 2. Utilizando-se da distância entre as sessões da compra atual, extraída no segundo estado do algoritmo, com a média das compras anteriores é calculado

um novo valor para a distância entre duas sessões, que será atualizado no banco de dados.

```

Entrada: mercado_id, lista_seções_visitadas

1 secao_visitada ← 0;
2 peso_acumulado ← 0,4;
3 peso_atual ← 0,6;
4 enquanto secao_visitada < tamanho da lista_seções_visitadas faça
5   seção_referência ← lista_seções_visitadas[secao_visitada];
6   proxima_secao ← secao_visitada + 1;
7   enquanto (proxima_secao < tamanho da lista_seções_visitadas) faça
8     seção_relação ← lista_seções_visitadas[proxima_secao];
9     distancia_entre_seções ← (proxima_secao - secao_visitada);
10    total_de_compras ← Número de compras anteriores entre
11      seção_referência e seção_relação;
12    soma_distancias_anteriores ← Soma das distâncias anteriores entre
13      seção_referência e seção_relação;
14    media_nova ←
15       $\frac{\text{soma_distancias_anteriores} * \text{peso_acumulado} + \text{distancia_entre_seções} * \text{peso_atual}}{(\text{total_de_compras} * \text{peso_acumulado}) + \text{peso_atual}}$ 
16    proxima_secao++;
17  fim
18  secao_visitada++;
19 fim

```

Algoritmo 2: Rotina de extração da ordem das seções do mercado.

No Algoritmo 2, o processamento da distância entre a categoria referência com todas as outras categorias presentes na compra será uma iteração pequena e rápida. Em um primeiro momento essa combinação par a par sem repetição apresenta ser custosa, pois aparenta gerar muitas combinações. Porém um mercado possui entre 10 a 20 seções o que representa um valor de 45 a 190 multiplicações, algo que um celular pode lidar facilmente hoje em dia sem haver nenhum atraso no processamento.

Exemplificando melhor esse algoritmo, serão empregados os dados de uma compra fictícia, onde será utilizado a lista de compras apresentada na [Tabela 2](#). A ordem presente nessa tabela está de acordo com a ordem de inserção dos produtos na lista pelo usuário.

Ao iniciar uma compra, o usuário deverá selecionar a lista que deseja utilizar, sendo nesse exemplo a lista da [Tabela 2](#), e logo em seguida deverá informar a seção que deseja iniciar suas compras, e o mercado. Ao selecionar o mercado o algoritmo de ordenação atuará sobre a lista selecionada junto com a seção escolhida como inicial, nesse exemplo o mercado em questão não possui nenhum histórico de compras, logo o algoritmo de ordenação não atuará nesse primeiro momento pois não há dados suficientes. A ordenação dessa lista será dependente da ordem da consulta ao banco, sendo a ordenação final possível

de visualizar na [Tabela 3](#).

Ordem	Seções	Produtos
1	Hortifruti	Cenoura Milho Abacate
2	Carne	Peixe Frango
3	Padaria	Bolo Pão
4	Alimento	Arroz Feijão
5	Congelado	Picolé Gelo
6	Bebida	Vinho Água

Tabela 2 – Lista de compra do usuário.

Dando prosseguimento às compras, a [Tabela 4](#) apresenta a ordem na qual os itens da lista foram inseridos no carrinho pelo usuário durante suas compras no mercado. Ao finalizar a compra as informações do ordenamento das seções serão utilizadas pelo algoritmo, para abastecer o histórico e melhorar a ordenação das seções em compras futuras no mesmo estabelecimento.

Os cálculos para ordenação dos produtos na lista irão somente considerar as seções visitadas pelo usuário durante as compras. O algoritmo iniciará seus cálculos pela primeira seção visitada e analisando sua distância¹ entre todas as outras seções. Analisando a [Tabela 4](#) é notado que a seção *Alimento* foi a primeira seção a ser visitada, logo essa seção será a primeira a possuir seus cálculos de distância efetuados, com as seguintes iterações:

Alimento-Bebida,
Alimento-Congelado,
Alimento-Hortifruti,
Alimento-Padaria,
Alimento-Carne

Segundo esse mesmo exemplo, a segunda seção visitada foi a seção Bebidas, conforme é possível verificar na [Tabela 4](#). Para essa seção seus cálculos de distâncias serão efetuados utilizando as seguintes combinações:

Bebida-Congelado,

¹ Neste trabalho o termo distância representa um valor sem unidade de medida, que quanto menor o seu valor significa que as seções estão mais próximas entre si.

Ordem	Seções	Produtos
1	Alimento	Arroz Feijão
2	Hortifruti	Cenoura Milho Abacate
3	Carne	Peixe Frango
4	Bebida	Vinho Água
5	Padaria	Bolo Pão
6	Congelado	Picolé Gelo

Tabela 3 – Ordenação da lista em uma primeira compra.

Ordem	Seções	Produtos
1	Alimento	Arroz Feijão
2	Bebida	Vinho Água
3	Congelado	Picole Gelo
4	Hortifruti	Cenoura Milho Abacate
5	Padaria	Bolo Pão
6	Carne	Peixe Frango

Tabela 4 – Ordem de inserção dos produtos no carrinho pelo usuário.

Bebida-Hortifruti,
Bebida-Padaria,
Bebida-Carne

O conjunto *Bebida-Alimento* foi omitido nessa interação, porque essa distância já foi realizada no passo anterior, quando foram realizados os cálculos das distâncias relacionadas a seção Alimentos.

Dentre essas combinações, serão calculadas as distâncias entre todas as seções, e posteriormente seus valores serão salvos no banco de dados para futuras consultas. Para não ser necessário manter um histórico de todos os valores entre as distâncias das seções em compras anteriores, ocasionando assim um número maior de requisições ao banco de dados, foi utilizada a média do relacionamento entre as seções em conjunto com o número de compras já realizadas entre essas duas seções para calcular uma nova média atualizada desse relacionamento, seguindo a seguinte fórmula:

$$\text{MediaTotalNova}[\text{Seção1} - \text{Seção2}] = \frac{\text{soma_distancias_anteriores} * \text{peso_acumulado} + \text{distancia_entre_seções} * \text{peso_atual}}{(\text{total_de_compras} * \text{peso_acumulado}) + \text{peso_atual}}$$

Onde *soma_distancias_anteriores* contém o somatório das distâncias das compras anteriores dessas duas seções, enquanto *total_de_compras* é o número de compras já realizadas entre essas duas seções. Por fim, *peso_acumulado* e *peso_atual* são pesos utilizados na média ponderada, 0,4 para o *peso_acumulado* e 0,6 para o *peso_atual*. Esses pesos foram definidos empiricamente e não foram realizados experimentos exaustivos para determinar sua melhor distribuição.

Ordem	Seções	Produtos
1	Alimento	Arroz Feijão
2	Bebida	Vinho Água
3	Congelado	Picole Gelo
4	Hortifruti	Cenoura Milho Abacate
5	Padaria	Bolo Pão
6	Carne	Peixe Frango

Tabela 5 – Ordenação da lista em uma segunda compra.

Os cálculos das distâncias para essa primeira compra, exemplificada na [Tabela 4](#) será simples, pois a média armazenada é zero e o número de compras também é zero. Na [Tabela 5](#) podemos verificar como fica a ordenação dessa mesma lista em uma próxima compra no mesmo mercado.

Para exemplificar o comportamento do algoritmo ao conter em sua lista seções que nunca participaram de uma compra, será utilizada a [Tabela 6](#) que exemplifica uma nova lista de compras diferente da primeira compra, porém sendo executada ainda no mesmo estabelecimento. Nota-se que as seções Hortifruti, Carne, e Padaria foram retiradas da lista de compras e as seções Congelados e Enlatados foram adicionadas a essa nova lista. As seções adicionadas nunca participaram de uma compra, logo não possuem um valor de distância em relação a qualquer outra seção, e por esse motivo serão inseridas ao final da lista.

Ao término da compra, os algoritmos de cálculo de distância serão executados novamente. Para as seções Alimento e Bebida, suas médias serão atualizadas, enquanto que para as novas seções suas distâncias serão criadas. Na [Tabela 7](#), é possível verificar a ordem na qual os produtos foram pegos por um usuário no momento da compra, enquanto na [Tabela 8](#) é possível verificar a ordenação dessa lista em uma nova compra no mesmo estabelecimento.

Ordem	Seções	Produtos
1	Alimento	Arroz Feijão
2	Bebida	Vinho Água
3	Congelado	Picole Gelo
4	Laticínio	Queijo Iogurte
5	Enlatado	Pepino Ervilha

Tabela 6 – Ordenação da nova lista de compras.

Ordem	Seções	Produtos
1	Alimento	Arroz Feijão
2	Bebida	Vinho Água
3	Laticínio	Queijo Iogurte
4	Congelado	Picolé Gelo
5	Enlatado	Pepino Ervilha

Tabela 7 – Ordem de inserção dos produtos no carrinho da nova lista.

Ordem	Seções	Produtos
1	Alimento	Arroz Feijão
2	Bebida	Vinho Água
3	Laticínio	Queijo Iogurte
4	Congelado	Picolé Gelo
5	Enlatado	Pepino Ervilha

Tabela 8 – Resultado da ordenação da lista em uma nova compra.

4 Conclusões

Este trabalho teve como objetivo o desenvolvimento de um aplicativo para a plataforma Android para ser utilizado por usuários que desejam otimizar o andamento de suas compras em mercados. A otimização é realizada por meio da coleta dos dados das compras do usuário, a fim de que em compras futuras em um mesmo mercado, a lista de compras esteja organizada de forma mais similar ao mercado.

De um modo geral o aplicativo Compra rápida supriu as necessidades de reorganizar as listas de produtos de acordo com o mercado que o cliente se encontra. Os produtos são agrupados em categorias e sua reorganização é calculada através dos dados coletados em compras passadas realizadas pelo usuário.

O algoritmo desenvolvido nesse trabalho fez uso do banco de dados do dispositivo móvel, para realizar as interações necessárias com os dados disponíveis para melhor ordenar a lista de compras do usuário. A cada nova compra que o usuário realiza em um mesmo mercado a lista irá agrupar os produtos por seções e essas consequentemente ordenadas o mais similar possível ao mercado. Foi possível também obter de forma satisfatória a localização do usuário através do aparelho celular do mesmo.

A localização do mercado onde o cliente se encontra para realizar as compras é obtida através do serviço de geolocalização do Google conhecido como *Google Places*, onde através dessa ferramenta é possível obter o nome do estabelecimento passando como parâmetro a posição geográfica do usuário. Enquanto a base de dados de produtos é atualizada através de um serviço distribuído.

A base de produtos do usuário é mantida em um serviço de banco de dados distribuído com assinatura *publish - subscribe*, provida pela plataforma Firebase. Esse serviço permite com que os dados cadastrados possam ser sincronizados imediatamente após alguma alteração na base remota, permitindo assim, a consistência dos dados entre todos os usuários, sem a necessidade de alterações no código-fonte do projeto e, consequentemente, o envio dessas atualizações para a loja de aplicativos do Google.

O objetivo geral desse trabalho de organizar dinamicamente a lista de produtos do usuário de acordo com o mercado em que ele se encontra foi alcançado utilizando o aplicativo Compra Rápida. Embora atualmente têm-se uma base de dados limitada, os testes realizados com os produtos disponíveis mostraram que as seções são ordenadas de maneira próximas a cada iteração, satisfazendo o objetivo principal deste trabalho, reduzir o tempo do usuário durante suas compras, em diferentes locais sem a necessidade da adição de uma infraestrutura extra nos supermercados para realização da ordenação. Por fim, a implementação da funcionalidade de lista colaborativa foi iniciada e parcial-

mente desenvolvida, fazendo uso da conexão de dados Wi-Fi ou de redes móveis para o compartilhamento das informações, no entanto sua implementação ficou incipiente, sendo uma opção para os trabalhos futuros.

4.1 Trabalhos Futuros

Para continuidade desse projeto, sugere-se como trabalhos futuros a aplicação de *beacons* com *Bluetooth Low Energy*(BLE), para captar a posição do usuário dentro do supermercado, e assim detectar em quais corredores ele está passando, podendo utilizar esses dados para reorganizar a lista de produtos durante as compras no mercado. No entanto, além do desenvolvimento do algoritmo de reorganização dinâmica, será necessário avaliar se o custo para a implantação desses dispositivos de *beacons* são comercialmente viáveis.

Uma outra sugestão é a inclusão de uma funcionalidade de sincronização das listas com outros usuários em modo colaborativo, através de um login único utilizando uma plataforma conhecida pelo usuário, como Facebook ou Gmail. A sincronização permite que os usuários verifiquem quais produtos estão sendo inseridos no carrinho ao longo das compras, atualizando assim a lista de compras de todos os outros convidados envolvidos nessa compra. Evitando desta forma que as pessoas comprem itens duplicados ou se esqueçam de um produto que foi adicionado somente após o início das compras,

Por fim, uma funcionalidade que poderá ser adicionada ao aplicativo é a inserção facilitada de itens através de métodos como reconhecimento de voz, leitura de códigos de barra ou aproximação com RFID. Uma outra opção válida é a integração com assistentes pessoais disponíveis nos sistemas operacionais tais como Google Now, Cortana ou Alexa fazendo a interação através desses assistentes inserirem itens à lista de produtos já existente ou a criação de uma lista nova.

Referências

- ALECRIM, E. *Tecnologia Bluetooth: o que é e como funciona?* 2013. <<http://www.infowester.com/bluetooth.php>>. (Visited on 12/05/2015). Citado 2 vezes nas páginas 24 e 25.
- ALECRIM, E. *Tecnologia Bluetooth: o que é e como funciona?* 2017. <<https://www.infowester.com/bluetooth.php>>. (Visited on 05/04/2017). Citado na página 24.
- ANDROID. *Introduction - Material design - Google design guidelines.* <<http://www.google.com.br/design/spec/material-design/introduction.html#introduction-principles>>. (Visited on 12/07/2015). Citado na página 29.
- ANDROID. *Location Strategies / Android Developers.* <<http://developer.android.com/intl/pt-br/guide/topics/location/strategies.html>>. (Visited on 12/07/2015). Citado 2 vezes nas páginas 27 e 28.
- ANDROID. *Notificações / Android Developers.* <<http://developer.android.com/intl/pt-br/guide/topics/ui/notifiers/notifications.html>>. (Visited on 12/07/2015). Citado 2 vezes nas páginas 28 e 29.
- CORDEIRO, F. *Introdução ao Material Design.* 2015. Disponível em: <<http://www.androidpro.com.br/android-material-design-introducao>>. Citado na página 30.
- DETTO, A. *Grocery Store Layout.* 2015. Disponível em: <<http://talkingcents.consumercredit.com/wp-content/uploads/2015/07/Grocery-Store-Layout.png>>. Citado na página 38.
- DEXI. *Wifi na loja.* 2016. <<https://www.dexi.com.br/exp-wifi-na-loja>>. (Visited on 05/04/2017). Citado na página 25.
- DUARTE, M. *Material Now.* 2015. Google I/O 2014. Disponível em: <<https://www.google.com/events/io>>. Citado na página 29.
- IDC. *IDC: Smartphone OS Market Share 2015, 2014, 2013, and 2012.* 2015. <<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>>. (Visited on 12/02/2015). Citado na página 26.
- KUROSE, J. F.; ROSS, K. W. *Computer Networking: A Top-Down Approach (6th Edition).* [S.l.]: Pearson, 2012. ISBN 0132856204. Citado 2 vezes nas páginas 23 e 25.
- LECHETA, R. R. *Google Android-3ª Edição: Aprenda a criar aplicações para dispositivos móveis com o Android SDK.* [S.l.]: Novatec Editora, 2013. Citado na página 19.
- LUNDEN, I. *Smartphone Users Globally By 2020, Overtaking Basic Fixed Phone Subscriptions.* 2015. <<https://techcrunch.com/2015/06/02/6-1b-smartphone-users-globally-by-2020-overtaking-basic-fixed-phone-subscriptions/>>. (Visited on 20/03/2017). Citado na página 19.
- MATERIAL design - Introduction. 2017. <<https://material.io/guidelines/material-design/introduction.html>>. (Visited on 05/15/2017). Citado na página 30.

MONTEIRO, J. B. *Google Android. Crie Aplicações Para Celulares (Em Portuguese do Brasil)*. [S.I.]: CASA DO CODIGO, 2013. ISBN 8566250028. Citado na página 26.

OPENSIGNAL. *State of Mobile Networks: Brazil (Aug 2015) - OpenSignal*. 2015. <<http://opensignal.com/reports/2015/08/brazil/state-of-the-mobile-network>>. (Visited on 12/06/2015). Citado na página 23.

PLACE Search. 2016. <<https://developers.google.com/places/web-service/search?hl=pt-br>>. (Visited on 05/04/2017). Citado na página 28.

PLAZA, W. R. *Entendendo o Wi-Fi 802.11ac*. 2014. <<http://www.hardware.com.br/artigos/entendendo-wifi-802.11-ac/>>. (Visited on 02/16/2016). Citado na página 25.

TELECO. *teleco.com.br*. 2015. <http://www.teleco.com.br/3g_cobertura.asp>. (Visited on 12/06/2015). Citado na página 23.

WELCOME to the Y Axis - Google Material Design. 2014. <<https://www.linkedin.com/pulse/20140722183716-30240682-welcome-to-the-y-axis-google-material-design>>. (Visited on 05/04/2017). Citado na página 30.

Apêndices

APÊNDICE A – Requisitos funcionais e não funcionais

Através da análise das necessidade do aplicativo, foram definidos requisitos funcionais e não funcionais. Os requisitos funcionais serão aqueles que definem como funciona o algoritmo e a interação do usuário com o aplicativo, enquanto os requisitos não funcionais irão definir o desempenho, confiabilidade e usabilidade do aplicativo.

A.0.1 Requisitos funcionais

A seguir são listados os requisitos funcionais da aplicação:

- **RF1:** O sistema deve permitir que o usuário possa criar listas de produtos.
- **RF2:** Ao criar uma lista de produtos o sistema deverá solicitar um nome a lista.
- **RF3:** O sistema deve permitir que o usuário possa inserir itens a uma lista de produtos
- **RF4:** O sistema irá organizar os itens adicionados à lista de produtos em suas seções correspondentes.
- **RF5:** O sistema deve permitir que o usuário tenha acesso às listas de produtos já criadas.
- **RF6:** O sistema deve permitir que o usuário possa renomear suas listas de produtos já criadas.
- **RF7:** O sistema deve permitir que o usuário possa excluir suas listas de produtos já criadas.
- **RF8:** O sistema deve permitir que o usuário selecione uma das listas de produtos para iniciar uma compra.
- **RF9:** O sistema deverá solicitar ao usuário que inclua o mercado onde irá realizar as compras.
- **RF10:** O sistema deverá organizar a lista de compras de acordo com o mercado inserido pelo usuário aproximando as seções por ordem de distância.
- **RF11:** Para as compras em andamento o usuário poderá mudar o status dos produtos, marcando os como pegas ou não pegas.

- **RF12:** O cliente deverá ter acesso a visualização dos itens inseridos no carrinho, os que não foram inseridos e os que estão aguardando a serem inseridos no carrinho de compras.
- **RF13:** O usuário deverá ter acesso as compras finalizadas.
- **RF14:** O usuário deverá ter acesso aos mercados próximos a ele.

A.0.2 Análise de requisitos não funcionais

A seguir são listados os requisitos não funcionais da aplicação.

- O aplicativo deverá ser desenvolvido seguindo o padrão do material design do Google.
- O aplicativo será desenvolvido na linguagem de programação Java.
- O aplicativo deverá pegar a localização geografia através do uso do GPS, para saber em qual mercado o cliente se encontra.
- O aplicativo deverá ser suportado em versões do Android a partir da versão 5.
- O aplicativo deverá armazenar as informações de categorias e produtos oriundos de um serviço externo localmente no SQLite.
- O aplicativo atualizará seu banco local de acordo com as atualizações do serviço externo.
- O aplicativo deverá armazenar as informações das listas e das compras localmente no SQLite.
- O sistema não apresentará aos usuários quaisquer dados de cunho privativo.

APÊNDICE B – Casos de uso

Essa seção tem como objetivo listar e explicar cada um dos casos de uso propostos nesse trabalho. Na [Figura 12](#) são apresentados os casos de uso que fazem interação com o usuário que está utilizando o aplicativo.

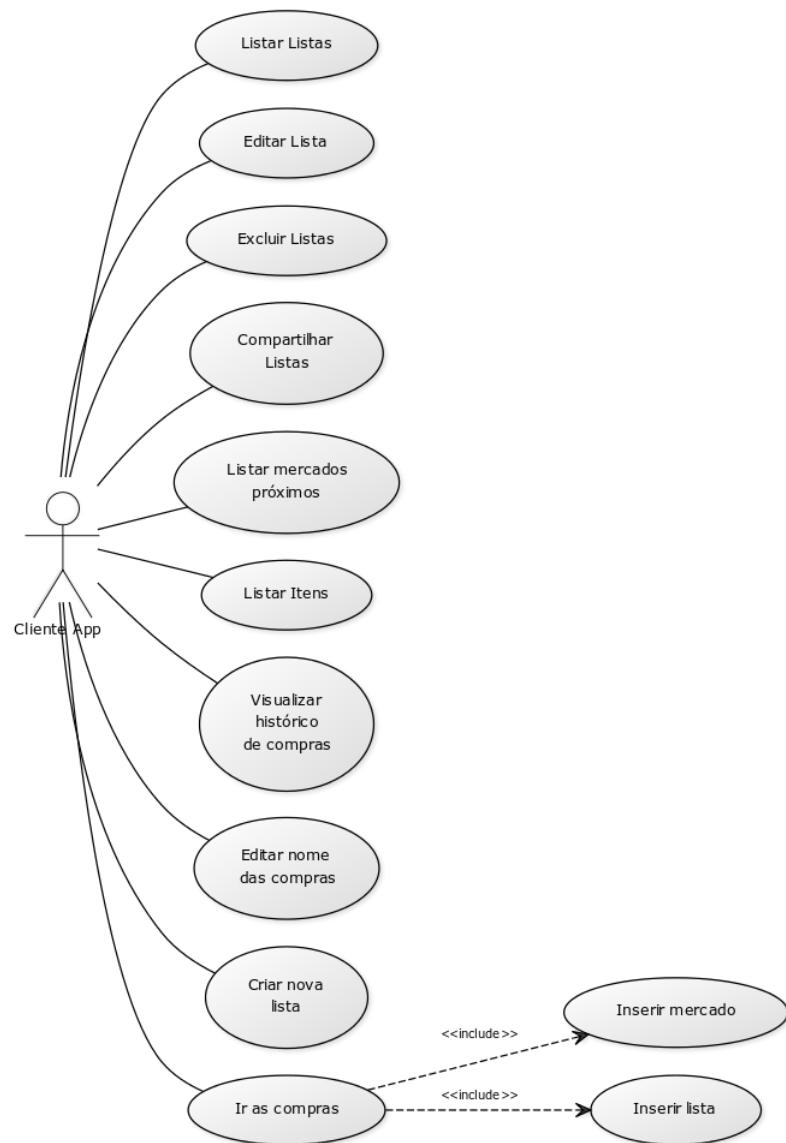


Figura 12 – Diagrama de casos de uso.

Caso de Uso: Criar Lista	
Fluxo Principal: Criar Lista	
1.	Cliente abre o aplicativo
2.	Sistema fornece um menu com as funções disponíveis
3.	Cliente escolhe criar lista
4.	Cliente adiciona os itens através das imagens.
5.	Cliente finaliza a criação da lista e volta para menu principal.

Tabela 9 – Caso de Uso: Criar Lista

Caso de Uso: Listar Lista	
Fluxo Principal: Listar Lista	
1.	Cliente abre o aplicativo.
2.	Sistema fornece um menu com as funções disponíveis.
3.	Cliente escolhe a opção “Listas”.
4.	Sistema apresenta uma tela com todas as listas criadas, ordenadas pela ordem de modificação.

Tabela 10 – Caso de Uso: Listar Lista

Caso de Uso: Editar Lista	
Fluxo Principal: Editar Lista	
1.	Cliente abre o aplicativo
2.	Sistema fornece um menu com as funções disponíveis.
3.	Cliente escolhe a opção “Listas”
4.	Sistema apresenta uma tela com todas as listas criadas.
5.	Cliente abre uma lista.
6.	Cliente poderá alterar o nome da lista, excluir a lista ou compartilhá-la.
7.	Sistema salva as alterações.

Tabela 11 – Caso de Uso: Editar Lista

Caso de Uso: Excluir Lista	
Fluxo Principal: Excluir Lista	
1.	Cliente abre o aplicativo.
2.	Sistema fornece um menu com as funções disponíveis.
3.	Cliente abre clica na opção “Listas”.
4.	Sistema apresenta uma tela com todas as listas criadas pelo usuário.
5.	Cliente escolhe a lista que deseja excluir e clica na opção excluir lista.
6.	Sistema salva as alterações.
7.	Sistema volta para a tela listar listas.

Tabela 12 – Caso de Uso: Excluir Lista

Caso de Uso: Compartilhar Lista	
Fluxo Principal: Compartilhar Lista	
1.	Cliente abre o aplicativo.
2.	Sistema fornece um menu com as funções disponíveis .
3.	Cliente escolhe a opção, “Listas”.
4.	Sistema apresenta uma tela listando as listas disponíveis.
5.	Cliente escolhe a lista e clica no ícone que representa a funcionalidade compartilhar.
6.	Sistema apresenta os aplicativos na qual o cliente poderá utilizar para compartilhar a lista: Gmail, Facebook Whatsapp.
7.	Cliente escolhe um dos aplicativos, e endereça para as pessoas com quem deseja compartilhar a lista.
8.	Sistema envia a lista e volta para a tela onde lista as listas disponíveis.

Tabela 13 – Caso de Uso: Compartilhar Lista

Caso de Uso: Listar Mercados	
Fluxo Principal: Listar Mercados	
1.	Cliente abre o aplicativo.
2.	Sistema fornece um menu com as funções disponíveis.
3.	Cliente clica na opção “Mercados”.
4.	Sistema lista todos os mercados próximos ao cliente.

Tabela 14 – Caso de Uso: Listar Mercados

Caso de Uso: Ir as Compras	
Fluxo Principal: Ir as Compras	
1.	Cliente abre o aplicativo e escolhe a opção “Ir as compras”.
2.	Sistema apresenta uma tela com todas as listas disponíveis.
3.	Cliente escolhe uma das listas, e clica no ícone que representa ir as compras.
4.	Sistema apresenta uma tela com sugestões de mercados para o cliente, com base na geolocalização.
5.	Cliente seleciona o mercado em que se encontra.
6.	Sistema adiciona mercado a lista.
7.	Sistema apresenta a lista com as seções organizadas de acordo com o mercado selecionado anteriormente.
8.	Cliente marca os itens que pegou ou não encontrou.
9.	Sistema armazenará os itens pegos na aba de itens pegos e os itens não pegos na aba dos itens não pegos.

Tabela 15 – Caso de Uso: Ir as Compras

Caso de Uso: Consultar Compras	
Fluxo Principal: Consultar Compras	
1.	Cliente abre o aplicativo.
2.	Sistema fornece um menu com as funções disponíveis.
3.	Cliente cliente clica na opção “Histórico”.
4.	Sistema fornece todas as compras já finalizadas e em andamento.
5.	Cliente procura a compra que lhe interessa.
6.	Ao achar clica na compra desejada.
7.	Sistema abre a compra.
8.	Cliente consulta o que lhe interessa (data, itens não encontrados, itens que não foram pegos).

Tabela 16 – Caso de Uso: Consultar Compras

Caso de Uso: Listar itens e categorias	
Fluxo Principal: Listar itens e categorias	
1.	Cliente abre o aplicativo.
2.	Sistema fornece um menu com as funções disponíveis.
3.	cliente clica na opção “Produtos”.
4.	Sistema lista todas as categorias disponíveis no sistema.
5.	Cliente expande a categoria.
6.	Sistema lista os itens, por ordem alfabética, disponíveis dentro da categoria.

Tabela 17 – Caso de Uso: Listar itens e categorias

APÊNDICE C – Modelo Entidade Relacionamento

O banco de dados desempenha um papel de armazenamento das listas de compras criadas e finalizadas pelo usuário, como também um papel importante no algoritmo de ordenamento que é mencionado na [seção 3.3](#).

Na figura [Figura 13](#) é mostrada uma visão geral do modelo de banco de dados utilizado nesse aplicativo, são oito tabelas que desempenham papéis distintos mas ao mesmo tempo auxiliares entre si. A seguir estão descritas as informações que cada tabela é responsável por armazenar.

A tabela **seção** é responsável por armazenar as categorias dos produtos existentes no aplicativo. Enquanto a tabela **produto** é responsável por armazenar os produtos disponíveis no aplicativo. Tendo as informações de produtos e categorias o usuário já é capaz de criar uma lista de produtos e informações como: nome da lista e data de criação ficarão armazenadas na tabela **lista_de_produtos**, enquanto os produtos inseridos na lista pelo usuário ficarão armazenados na tabela **produtos_na_lista**.

Ao iniciar as compras o usuário é capaz de alterar o status dos produtos para: pego, não pego ou aguardando, essas informações ficam armazenadas na tabela **status_do_produto**. Informações gerais, como: se a compra foi excluída do aplicativo, id do mercado onde a compra está sendo realizada e id da lista que contém os produtos da compra, ficam armazenadas na tabela **lista_de_compras**. Os mercados que já tenham compras realizadas, terão seus dados armazenados na tabela **mercado**.

Por fim, ao finalizar as compras é realizado o algoritmo para a otimização das compras, através do relacionamento das seções. Essas informações ficam armazenadas na tabela **relacionamento_entre_seções**.

No desenvolvimento do banco de dados foi adicionado ainda uma classe de controle chamada *Contrato* como recomendado na documentação de boas práticas do Google para banco de dados. O objetivo dessa classe é prover uma forma segura para manter a integridade do banco, servindo como uma variável global de configuração dos nomes das tabelas e colunas presentes no aplicativo. Na [Listagem C.1](#) é apresentado um exemplo de como é feita a configuração dessa classe para a tabela produtos e suas colunas. Essa mesma configuração foi feita para todas as demais tabelas apresentadas na [Figura 13](#).

Para criar e gerenciar o banco de dados foi utilizado a classe **SQLiteOpenHelper** provida pelo Google e disponível nativamente no Android. Essa classe proporciona aos

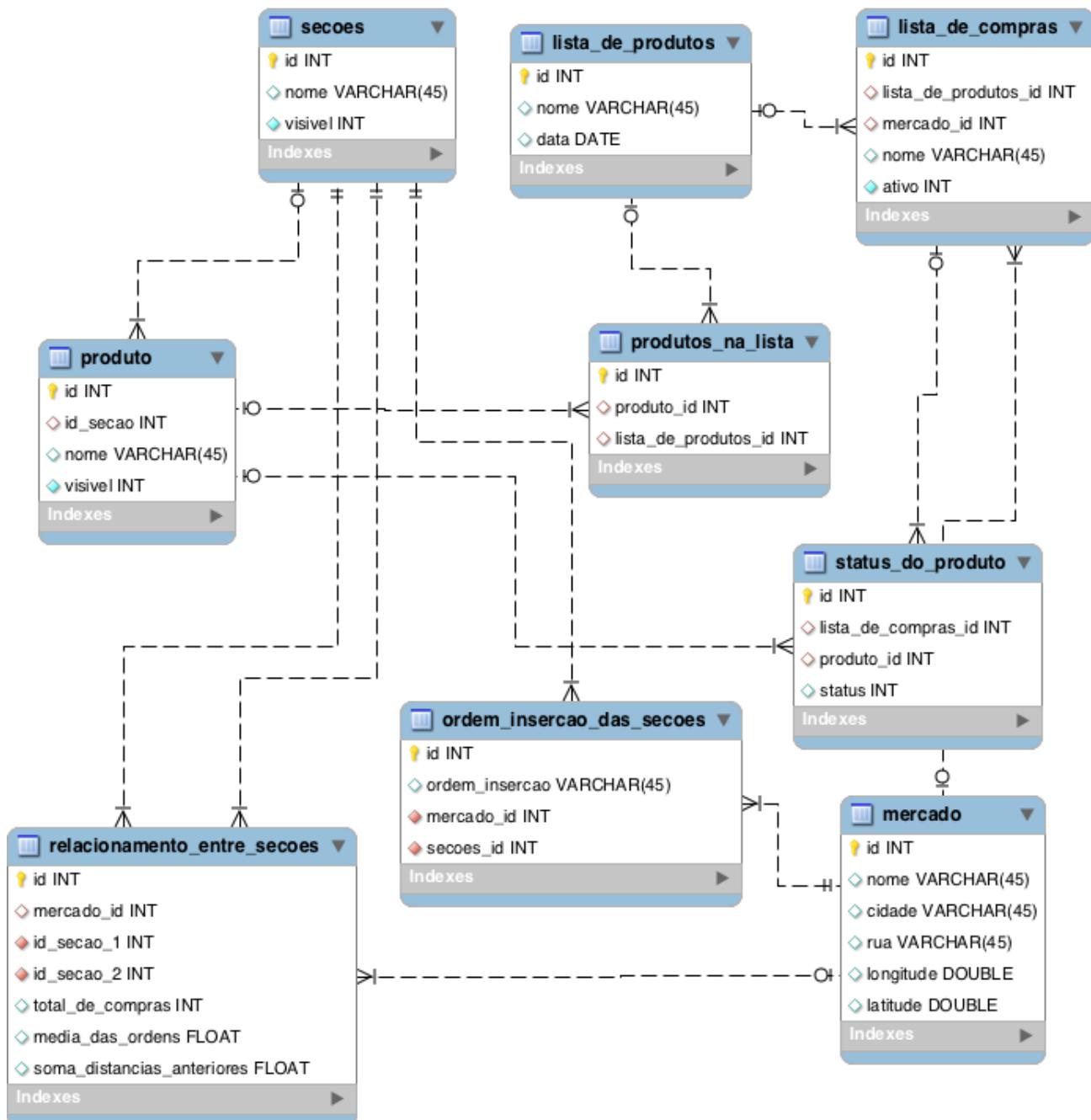


Figura 13 – Diagrama Entidade Relacionamento

desenvolvedores diversos métodos para criação e manutenção do banco de dados tais como *create*, *update* e *delete*. O SQLiteOpenHelper também auxilia na inicialização do aplicativo, atrasando o carregamento do banco até o primeiro uso de alguma tabela, e por fim auxilia no gerenciamento do banco de dados em futuras versões do aplicativo, provendo um mecanismo de atualização e migração do banco de dados.

```

1 public class Contract {
2
3     public static abstract class ProductTable {
4         public static final String TABLE_NAME = "produto";

```

```
5     public static final String _ID = "id";
6     public static final String PRODUCT_NAME = "nome";
7     public static final String PRODUCT_CATEGORY_ID = "id_secao";
8     public static final String PRODUCT_VISIABLE = "visivel";
9 }
```

Listagem C.1 – Classe Contract