



**LUCIANO BRASIL CARDOSO**

**MODELO ARQUITETÔNICO PARA  
INTEGRAÇÃO DO SOFTWARE SOCIAL  
PREÇO PÚBLICO E A PLATAFORMA PARA  
DISPOSITIVOS MÓVEIS ANDROID**

**LAVRAS - MG  
2011**

**LUCIANO BRASIL CARDOSO**

**MODELO ARQUITETÔNICO PARA INTEGRAÇÃO DO SOFTWARE  
SOCIAL PREÇO PÚBLICO E A PLATAFORMA PARA DISPOSITIVOS  
MÓVEIS ANDROID**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

Orientador:

Prof. Dr. Raphael Winckler de Bettio

**LAVRAS - MG  
2011**

**LUCIANO BRASIL CARDOSO**

**MODELO ARQUITETÔNICO PARA INTEGRAÇÃO DO SOFTWARE  
SOCIAL PREÇO PÚBLICO E A PLATAFORMA PARA DISPOSITIVOS  
MÓVEIS ANDROID**

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Ciência da Computação para obtenção do título de Bacharel em Ciência da Computação.

APROVADA em 22 de junho de 2011

  
Prof. Dr. Denilson Alves Pereira

DCC-UFLA

  
Prof. MSc. Eric Fernandes de Mello Araújo

DCC-UFLA

  
Prof. Dr. Raphael Winckler de Bettio  
Orientador

**LAVRAS - MG  
2011**

## AGRADECIMENTOS

Agradeço a minha família por todo amor e apoio incondicional. Por estarem sempre presentes quando eu precisei.

Ao professor e orientador Raphael, pelo aprendizado e conselhos dados ao longo do desenvolvimento desse trabalho e por ter aceitado o desafio de realizar esse trabalho em apenas um período.

A minha namorada Natália, pelo carinho, amor e atenção dados durante todos os momentos que precisei.

Aos meus amigos da Rep. Taverna, Elcio, Euler, Gustavo, Mário, Leandro, Luis Felipe, Lucas, Rafael e Wilker, muito obrigado por todos os momentos incríveis que passei com vocês.

A meus amigos de curso, que estiveram presentes nas horas de alegria e nas horas difíceis.

A todos os professores do DCC – UFLA, que durante estes quase cinco anos conseguiram fazer com que eu aprendesse muito.

Enfim, agradeço a todos que me ajudaram a concluir esse trabalho.

## RESUMO

O Preço Público é uma comunidade virtual *web* para compartilhamento do conhecimento dos preços de produtos de empresas tradicionais (que não utilizam e-commerce). Esse trabalho tem como objetivo principal definir um modelo arquitetônico capaz de integrar o software social Preço Público e a plataforma para dispositivos móveis Android, promovendo assim, uma nova forma de coleta de dados. Em um primeiro momento, é apresentado o referencial teórico a respeito de comunidades virtuais e também das seguintes tecnologias: Android, Java, Spring, Hibernate, MySql e Web Service. Em seguida, é apresentado o modelo arquitetônico e, por fim, o protótipo de software criado para validar o modelo.

**Palavras-chave:** Engenharia de *software*; Arquitetura de sistemas; Comunidades virtuais; Preço Público; Android; Integração de sistemas.

## **ABSTRACT**

The Preço Público is an online community web-based for sharing knowing of prices of products traditional companies (who do not use e-commerce). This work aims to define an architectural model that can integrate social software Preço Público and Android platform for mobile devices, promoting a new form of data collection. At first the work presents the theoretical framework about virtual communities and also the following technologies: Android, Java, Spring, Hibernate, MySQL e Web Service and then the architectural model is presented and ultimately created a prototype software to validate the model.

**Keywords:** Software engineering; Systems architecture; Virtual communities; Preço Público; Android; Systems integration.

## LISTA DE ILUSTRAÇÕES

Figura 1 Linha do tempo da evolução da Internet. ....	15
Figura 2 Usuários ativos do Facebook.....	18
Figura 3 Busca por nome do produto.....	22
Figura 4 Últimos preços publicados. ....	23
Figura 5 Cadastro de produto no Preço Público. ....	24
Figura 6 Acompanhar preços.....	24
Figura 7 Cadastrar novos produtos e/ou novas empresas. ....	25
Figura 8 Arquitetura da plataforma Android. ....	27
Figura 9 Exemplo de tela desenvolvido na plataforma Android.....	28
Figura 10 Arquivo XML com a definição da interface da Figura 9.....	29
Figura 11 Esquema do funcionamento da tecnologia Web Service.....	34
Figura 12 Fluxograma que representa o atual processo para publicar preço no Preço Público. ....	36
Figura 13 Fluxograma para o cadastro de preços no Preço Público através do Android.	38
Figura 14 Diagrama da arquitetura do subsistema para cadastro de preços no Preço Público através de dispositivos móveis. ....	39
Figura 15 Diagrama do fluxo das telas do aplicativo Preço Público <i>Mobile</i> . ....	41
Figura 16 Modelo de interface da tela inicial do aplicativo Preço Público <i>Mobile</i> . ....	42
Figura 17 Modelo de interface da tela de listagem dos preços cadastrados no aplicativo Preço Público para dispositivos móveis. ....	43
Figura 18 Modelo de interface da tela de <i>login</i> do aplicativo Preço Público para dispositivos móveis. ....	44
Figura 19 Modelo de dados do aplicativo Preço Público para dispositivos móveis. ....	45
Figura 20 Modelo de integração entre o aplicativo Android e o sistema Preço Público. ....	46
Figura 21 Parte do modelo de dados alterado do sistema Preço Público Web. ....	48
Figura 22 Diagrama de fluxo das páginas do sistema Preço Público web. ....	49
Figura 23 Modelo de interface representando a página inicial do sistema Preço Público web.....	50
Figura 24 Modelo de interface representando a página de listagem de preços cadastrados através do celular. ....	51

Figura 25 Modelo de interface representando a página para publicação dos produtos coletados através do dispositivo móvel. ....	52
Figura 26 Diagrama de classes do sistema Preço Público <i>mobile</i> . ....	54
Figura 27 Código para a criação de um botão para o Preço Público <i>mobile</i> . ....	54
Figura 28 Código para a criação de um campo texto para o Preço Público <i>mobile</i> . ....	55
Figura 29 Código para a criação de uma <i>label</i> para o Preço Público <i>mobile</i> . ....	56
Figura 30 <i>Print Screen</i> do Leitor de código de barras funcionando no emulador Android. ....	57
Figura 31 Código para a criação de fluxo entre telas no Preço Público <i>mobile</i> . ....	57
Figura 32 Código para inserir dados no banco de dados SQLite. ....	58
Figura 33 Código para criptografar a senha do usuário. ....	59
Figura 34 Código para transformar a lista de produtos no padrão JSON. ....	60
Figura 35 Serviço para realizar a integração entre o sistema <i>mobile</i> e <i>web</i> . ....	62
Figura 36 Serviço para verificar se o usuário é válido. ....	63
Figura 37 Serviço responsável por cadastrar uma lista de produtos no banco de dados MySQL do Preço Público <i>web</i> . ....	64
Figura 38 Exemplo do uso da <i>annotation</i> @Transaction. ....	64
Figura 39 Código para criar o <i>link</i> Mobile presente na página inicial do Preço Público <i>web</i> . ....	64
Figura 40 Serviço responsável por listar os registros da tabela <b>produtomob</b> . ....	65
Figura 41 Serviço responsável por verificar se o código de barras de um produto já existe no banco de dados. ....	66
Figura 42 Serviço responsável por excluir um registro da tabela <b>produtomob</b> . ....	67

## SUMÁRIO

1	INTRODUÇÃO .....	9
1.1	Contextualização e Motivação .....	9
1.2	Objetivos do Trabalho .....	11
1.2.1	Objetivo Geral.....	11
1.2.2	Objetivos Específicos.....	11
1.3	Metodologia .....	11
2	REFERENCIAL TEÓRICO .....	13
2.1	Internet .....	13
2.2	Comunidades virtuais .....	16
2.2.1	Facebook.....	17
2.2.2	Javafree .....	18
2.2.3	Mercado Livre.....	19
2.3	Compartilhamento de conhecimento .....	20
2.4	O Preço Público .....	21
2.5	Tecnologias empregadas .....	26
2.5.1	Android .....	26
2.5.2	Java .....	30
2.5.3	Spring.....	31
2.5.4	Hibernate e MySQL.....	32
2.5.5	Web Service.....	33
3	MÉTODO .....	35
3.1	Problema .....	35
3.2	Solução desenvolvida .....	39
3.2.1	Preço Público <i>mobile</i> .....	40
3.2.2	Diagrama de integração entre o sistema web e o aplicativo <i>mobile</i> do Preço Público .....	45
3.2.3	Preço Público <i>web</i> .....	47
3.3	Protótipo.....	53
3.3.1	Preço Público <i>mobile</i> .....	53
3.3.2	Integração do Preço Público <i>mobile</i> e <i>web</i> .....	58
3.3.3	Preço Público <i>web</i> .....	64
4	CONCLUSÃO .....	68
5	REFERÊNCIAS BIBLIOGRÁFICAS.....	71

## 1 INTRODUÇÃO

Esse capítulo irá introduzir o trabalho apresentando a contextualização, motivação, objetivos e a metodologia.

### 1.1 Contextualização e Motivação

A Internet surgiu na década de 50 através de esforços militares e acadêmicos, foi desenvolvida e cresceu mundialmente ao longo dos anos. Em consequência disso, nas duas últimas décadas está presente em todos os setores da Sociedade.

Na década de 90, as tecnologias mais utilizadas pelos usuários da Internet foram o e-mail para envio de mensagens, a Web para o comércio eletrônico, os mensageiros eletrônicos como o ICQ e o compartilhamento *peer-to-peer* de arquivos MP3 (KUROSE and ROSS, 2006).

A diversidade de tecnologias, serviços e produtos foi crescendo ao longo dos anos e atualmente a Internet está cada vez mais presente na rotina das pessoas.

As comunidades virtuais fazem parte do grupo de serviços que teve ascensão junto com a Internet (CHIU *et al*, 2006) e podem ser definidas como um conjunto de pessoas que possuem interesses ou objetivos em comum, e que se reúnem periodicamente utilizando a Internet como meio de comunicação (RIDINGS and GEFEN, 2004).

Entre os exemplos de comunidades virtuais que existem, pode-se citar: o Facebook ([www.facebook.com](http://www.facebook.com)), o JavaFree ([javafree.uol.com.br](http://javafree.uol.com.br)) e o Mercado Livre ([www.mercadolivre.com.br](http://www.mercadolivre.com.br)). O Facebook caracteriza-se por ser uma

comunidade virtual para a criação de redes de relacionamentos, o JavaFree tem como objetivo a discussão sobre a linguagem de programação Java e o Mercado Livre é uma comunidade virtual para venda e compra de produtos na Internet.

A Internet e as comunidades virtuais são responsáveis por facilitar o compartilhamento do conhecimento entre as pessoas. Cada comunidade citada anteriormente é responsável por compartilhar entre seus participantes um tipo de conhecimento específico.

Outros exemplos relevantes de comunidades virtuais são o BuscaPé ([www.buscapede.com.br](http://www.buscapede.com.br)) e o Preço Público ([www.precopublico.com.br](http://www.precopublico.com.br)), que possibilitam o compartilhamento do conhecimento sobre os preços de produtos.

A primeira é focada no compartilhamento do conhecimento sobre os preços de produtos de empresas que trabalham com comércio eletrônico (ou *e-commerce*) e a segunda é focada no mesmo tema, porém com empresas tradicionais. Outra diferença, é que no BuscaPé as empresas são responsáveis por divulgar o preço de seus produtos e no Preço Público os usuários da comunidade virtual que tem essa responsabilidade.

O objeto de estudo desse trabalho é o *software* social Preço Público e o desenvolvimento de uma nova aplicação para aumentar as possibilidades de compartilhamento de conhecimento sobre os preços dos produtos.

Será desenvolvido por esse trabalho um modelo arquitetônico, e em seguida um protótipo de *software*, para integrar o *software* social Preço Público e a plataforma de dispositivos móveis Android.

Com esse protótipo, é possível que o usuário armazene os preços dos produtos no Preço Público de maneira mais fácil e rápida. Um exemplo prático do uso do protótipo é dado a seguir: um usuário do Preço Público dirige-se a um comércio e, com o auxílio de um dispositivo móvel, tira foto do código de barras, digita a descrição e o preço de um produto que deseja enviar para a

comunidade virtual. Logo em seguida, o usuário envia esses dados para o Preço Público. Quando desejar, o usuário pode acessar a comunidade virtual e divulgar os preços dos produtos que foram coletados com a ajuda do dispositivo móvel.

## **1.2 Objetivos do Trabalho**

### **1.2.1 Objetivo Geral**

O objetivo deste trabalho é definir um modelo arquitetônico que possibilite a integração do *software* social Preço Público e a plataforma de dispositivos móveis Android.

### **1.2.2 Objetivos Específicos**

O trabalho possui os seguintes objetivos específicos:

- Estudar o modelo arquitetônico do *software* Preço Público.
- Estudar a plataforma Android.
- Definir um modelo arquitetônico que permita a integração dos *softwares* acima citados.
- Desenvolver um protótipo utilizando o modelo arquitetônico definido.

## **1.3 Metodologia**

Esta pesquisa pode ser classificada quanto à natureza como tecnológica, pois tem como objetivo a geração de um novo produto para o *software* social Preço Público. Além disso, pode ser considerada como uma pesquisa

exploratória, experimental e em laboratório, de acordo com seus objetivos, procedimentos e local de execução respectivamente (ZAMBALDE *et al*, 2008).

O trabalho está estruturado da seguinte maneira:

- O Capítulo 2 apresenta o Referencial Teórico, mostrando os principais conceitos utilizados no trabalho.
- O Capítulo 3 mostra o Desenvolvimento do trabalho, detalhando o problema em questão, a solução desenvolvida e o protótipo desenvolvido a partir do modelo proposto.
- O Capítulo 4 finaliza o trabalho apresentando as conclusões, discutindo os resultados obtidos e propondo trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Nesse capítulo serão abordados os conceitos que embasaram o desenvolvimento desse trabalho.

### 2.1 Internet

A Internet pode ser definida, basicamente, como uma rede mundial de computadores que são interconectados obedecendo a um conjunto de regras e protocolos (MOWERY; SIMCOE, 2002).

A história do surgimento da Internet teve início com a criação da ARPA (*Advanced Research Projects Agency*) nos Estados Unidos em 1957. A ARPA foi criada pelo Departamento de Defesa norte-americano com a intenção de criar projetos para competir com o projeto soviético *Sputnik* (BEAMONT, 1999).

Em 1967, o pesquisador Lawrence Roberts, apresentou em uma conferência o primeiro esboço para uma rede de computadores por comutação de pacotes que foi denominado ARPAnet (LEINER *et al*, 2009).

Mowery e Simcoe (2002) explicam que a comutação de pacotes trabalha dividindo a informação em pequenos pacotes que são enviados pela rede. Quando chegam ao destino, esses pacotes são remontados e assim é estabelecida a configuração inicial da informação.

A ARPAnet foi instalada pela primeira vez em 1969 em quatro universidades norte-americanas (LEINER *et al*, 2009).

Em 1972, quinze universidades já estavam conectadas à ARPAnet (KUROSE; ROSS, 2006) e em 1973, as primeiras conexões internacionais foram estabelecidas, uma na Inglaterra e outra na Noruega (BEAMONT, 1999).

A origem da Internet só foi possível por esforços militares e acadêmicos, em pouco tempo de surgimento já era possível enviar mensagens através, por exemplo, de uma conexão entre um ponto nos Estados Unidos e outro na Inglaterra.

Segundo Kurose e Ross (2006), o período de 1972 até 1980 foi marcado pelo surgimento de outras redes além da ARPAnet. Uma dessas redes foi a Telenet, uma rede comercial pertencente à empresa BBN que utilizava a mesma tecnologia de comutação de pacotes que a ARPAnet.

Esse último fato mostra que o desenvolvimento da Internet passa a contar com outros centros de pesquisas e não mais apenas com a ARPA.

Na década de 1980 o crescimento das redes continuou acontecendo em ritmo acelerado, chegando ao número de cem mil máquinas ligadas a ARPAnet (KUROSE; ROSS, 2006).

Em 1983, a ARPAnet iniciou o uso do protocolo TCP/IP, como consequência disso, a ARPAnet foi dividida na rede MILNET para fins militares e ARPAnet para pesquisas acadêmicas (LEINER *et al*, 2009).

A NSF (*National Science Foundation*) criou em 1986 a NSFNET uma rede para conectar centros de pesquisas a supercomputadores que pertenciam à NSF (KUROSE; ROSS, 2006).

A década de 90 foi marcada pelos seguintes fatos:

- Crescente uso comercial da Internet;
- Criação da *World Wide Web* (WWW), que possibilitou o uso da Internet em níveis doméstico e empresarial no mundo inteiro;
- Fim da ARPAnet (KUROSE; ROSS, 2006);

- Início da Internet no Brasil através da Fundação de Amparo à Pesquisa do Estado de São Paulo (VIEIRA, 2003).

Para Kurose e Ross (2006), as tecnologias de maiores uso na década de 90 foram: o e-mail, a Web para o comércio eletrônico, os mensageiros eletrônicos (ICQ, MSN) e o compartilhamento *peer-to-peer* de arquivos MP3.

Segundo Mowery e Simcoe (2002), a junção da tecnologia *World Wide Web* com a Internet favoreceu o crescimento para as aplicações comerciais. Os autores explicam que esse desenvolvimento aconteceu pelo alto investimento dos Estados Unidos nessa área até meados de 2000.

A Figura 1 mostra a linha do tempo da evolução no âmbito mundial da Internet desde a sua origem em 1957 até a criação da WWW em 1991.

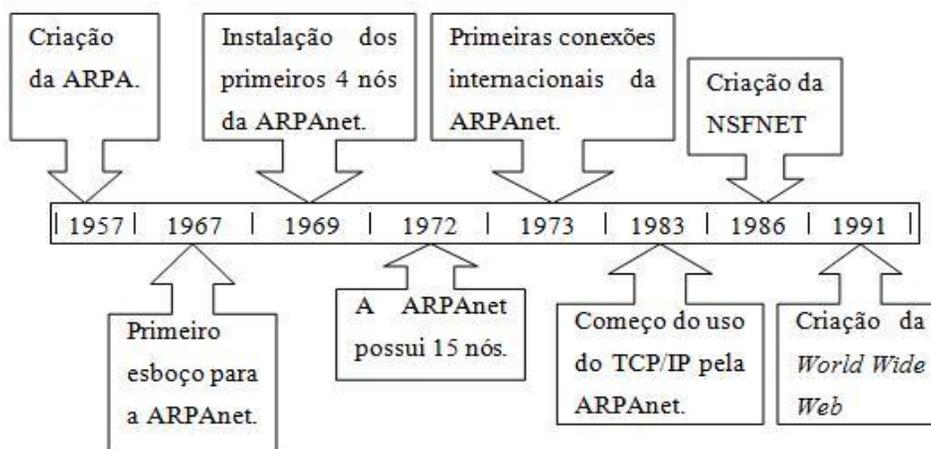


Figura 1 Linha do tempo da evolução da Internet (Fonte: próprio autor).

O enorme crescimento da Internet gerou uma variedade muito grande de serviços e tecnologias disponíveis (LEINER *et al*, 2009), uma das consequências disso, foi a ascensão do uso das comunidades virtuais pelos usuários da Internet (CHIU *et al*, 2006), que veremos a seguir.

## 2.2 Comunidades virtuais

Para melhor entendimento do que é uma comunidade virtual será apresentado antes o conceito geral de comunidade.

De acordo com Axt (2006), comunidade é um grupo de pessoas que possui um objetivo em comum e trabalham em conjunto para atingir esse objetivo. Segundo a autora esse conceito é bem antigo e pode ser aplicado inclusive no mundo animal.

As comunidades virtuais surgiram com a proliferação mundial do uso da Internet (CHIU *et al*, 2006) e são definidas por Ridings e Gefen (2004) como um conjunto de pessoas que possuem interesses ou objetivos em comum que se reúnem periodicamente utilizando a Internet como meio de comunicação.

Segundo Maggio *et al* (2009), a principal diferença entre as comunidades reais e virtuais está na dependência tecnológica das comunidades virtuais.

Essa diferença faz com que o desenvolvimento das comunidades virtuais seja mais difícil, mas também possibilita uma maior flexibilidade de espaço e tempo para seus usuários (MAGGIO *et al*, 2009).

Os tipos de interesses ou objetivos que uma comunidade virtual pode abordar são vastos, como por exemplo, relacionamento social, desenvolvimento de alguma linguagem de programação e comércio eletrônico (*e-commerce*).

No próximo tópico será apresentado um exemplo de comunidade virtual para cada tema citado anteriormente.

### 2.2.1 Facebook

O Facebook ([www.facebook.com](http://www.facebook.com)) é uma das maiores comunidades virtuais direcionadas para o relacionamento social que existe atualmente. Foi criada em 4 de fevereiro de 2004 por Mark Zuckerberg e alguns colegas de faculdade (PATRICIO; GONÇALVES, 2010).

O Facebook é um exemplo de comunidade virtual que tem como objetivo criar uma rede para relacionamento entre as pessoas.

Outras comunidades também possuem uma grande quantidade de usuários e competem com Facebook por novos usuários, entre elas estão: Orkut, MySpace e o Twitter.

Inicialmente o Facebook era utilizado somente para fins acadêmicos e tinha os seguintes objetivos: comunicação, compartilhamento de informações e publicação de fotografias entre alunos, professores e funcionários que participavam da comunidade acadêmica (PATRICIO; GONÇALVES, 2010).

Esses serviços que o Facebook possuía fizeram com que a comunidade virtual crescesse rapidamente na área acadêmica, a prova disso é que em um ano de funcionamento o Facebook já possuía mais de cinco milhões de usuários<sup>1</sup>.

Atualmente o Facebook é uma rede social gratuita e permite a participação de qualquer pessoa. Possui diversas ferramentas que facilitam a comunicação entre seus usuários, entre elas podemos citar: adicionar fotografias, vídeos, comentários, enviar mensagens, integração com outros *websites*, entre outras tecnologias (PATRICIO; GONÇALVES, 2010).

---

<sup>1</sup> <http://www.facebook.com/press/info.php?timeline>

O crescimento do Facebook é percebido através da quantidade de usuários ativos que o *site* possui, passando de um milhão de usuários ativos em dezembro de 2004 a 500 milhões em 2010<sup>1</sup>. A Figura 2 mostra o crescimento do Facebook ao longo do tempo.



Figura 2 Usuários ativos do Facebook (Fonte: próprio autor).

### 2.2.2 Javafree

A comunidade virtual brasileira Javafree ([javafree.uol.com.br](http://javafree.uol.com.br)) é uma das maiores do mundo sobre programação Java, foi fundada em dezanove de março de 2003 por Dalton Camargo<sup>2</sup>.

---

<sup>1</sup> <http://www.facebook.com/press/info.php?timeline>

<sup>2</sup> <http://javafree.uol.com.br/sobre.jbb>

O Javafree é um exemplo de comunidade virtual que aborda o tema de desenvolvimento da linguagem de programação Java. Existem outras comunidades virtuais que também abordam esse tema, entre elas estão: GUF (Grupo de Usuários Java), Java Programming Forums e SouJava.

O Javafree já realizou vários trabalhos, como tutoriais, artigos e entrevistas com o objetivo de desenvolver a linguagem Java e o *Software* livre<sup>2</sup>.

O portal possui também um fórum para discussão entre os usuários, uma sessão para notícias e outra para a oferta de empregos, todas essas funcionalidades abordam o mesmo tema: a linguagem Java.

Em junho de 2010, o grupo DevMedia adquiriu o portal Javafree.

### **2.2.3 Mercado Livre**

A comunidade virtual Mercado Livre ([www.mercadolivre.com.br](http://www.mercadolivre.com.br)) é uma das maiores da América Latina no ramo de vendas e compras pela Internet, oferece uma variedade de serviços e produtos para que o comércio eletrônico seja realizado da forma mais fácil, rápida e confiável possível<sup>3</sup>.

O Mercado Livre foi inaugurado em agosto de 1999 por Marcos Galperín e é importante para esse trabalho, pois é um exemplo de comunidade virtual de compra e venda pela Internet.

Outras comunidades virtuais que concorrem com o Mercado livre são: Ebay, SuperTube e TodaOferta.

O Mercado Livre já possui 50,2 milhões de usuários registrados confirmados até a data de três de novembro de 2010<sup>3</sup>.

---

<sup>3</sup> <http://www.mercadolivre.com.br/org-img/html/MLB/institucional>

### 2.3 Compartilhamento de conhecimento

Segundo Bartol e Srivastava (2002), conhecimento é o conjunto de informações, idéias e opiniões que um indivíduo ou grupo utiliza para realizar alguma ação.

De acordo com Argote e Ingram (2000), o compartilhamento de conhecimento pode ser definido como o processo em que uma unidade (indivíduo, empresa, departamento, etc.) tem seu conhecimento afetado por outra unidade.

Atualmente a Internet e as comunidades virtuais têm facilitado o compartilhamento de conhecimento entre os seus usuários, sendo assim, dois usuários pertencentes a uma comunidade virtual e que estão em lugares diferentes podem compartilhar seus conhecimentos. (KOH; KIM, 2004).

Segundo Koh e Kim (2004), as comunidades virtuais que utilizam de forma efetiva o compartilhamento de informações têm alguns resultados positivos para a comunidade como, por exemplo, coesão do grupo, sentimento de fazer parte da comunidade e lealdade dos membros da comunidade.

Das comunidades virtuais apresentadas na sessão 2.2 o Facebook tem o objetivo de criar uma rede de relacionamento entre as pessoas.

O Javafree tem a característica de compartilhar entre seus usuários o conhecimento na linguagem de programação Java.

O Mercado livre tem o objetivo de compartilhar o conhecimento sobre os produtos que são anunciados pelos vendedores no *site*.

O Yahoo Respostas (br.answers.yahoo.com) é outro exemplo interessante de compartilhamento de conhecimento em comunidades virtuais, pois possui um fórum onde o usuário faz uma pergunta sobre qualquer tema e

outros usuários respondem e qualificam a melhor resposta para aquela pergunta (ADAMIC *et al*, 2007).

No próximo tópico será apresentada a comunidade virtual Preço Público que tem o objetivo de compartilhar o conhecimento sobre preços de qualquer produto entre seus usuários e é a comunidade virtual alvo de estudo dessa monografia.

## 2.4 O Preço Público

O compartilhamento de conhecimento sobre preços é um dos temas que uma comunidade virtual pode abordar. O BuscaPé ([www.buscapede.com.br](http://www.buscapede.com.br)) é um exemplo de comunidade virtual que tem como objetivo compartilhar preços de produtos.

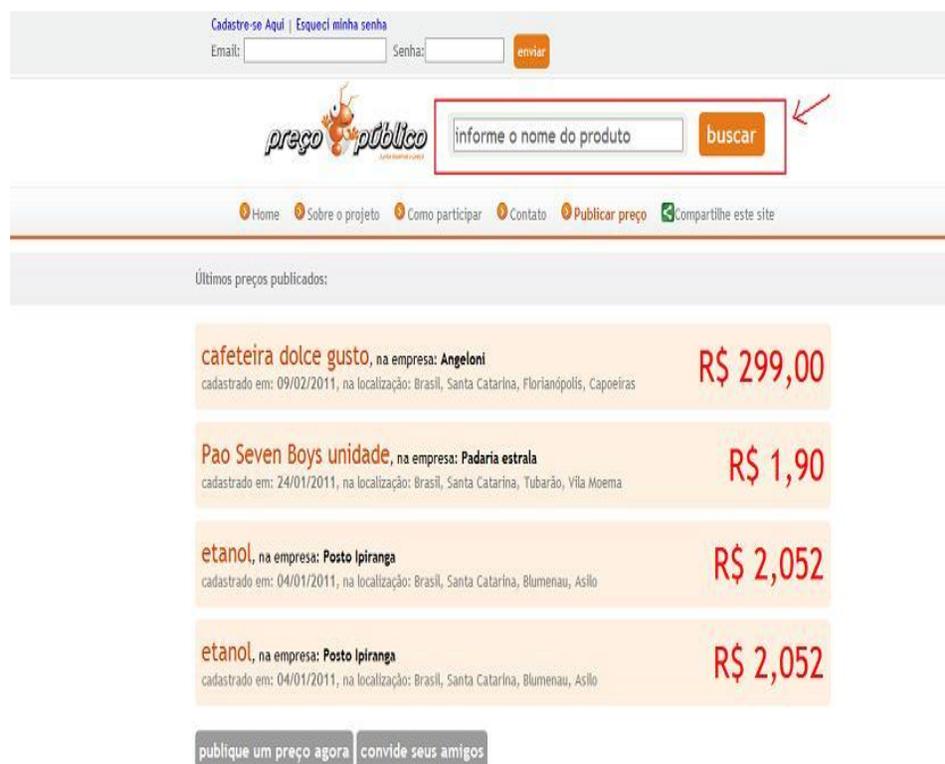
Nessa comunidade os anúncios dos preços dos produtos são feitos pelas empresas que em sua maioria trabalham com o *e-commerce*, e a busca pelos preços dos produtos pode ser realizada por qualquer usuário. Com isso, os usuários podem comparar os preços de várias lojas virtuais utilizando a Internet como facilitadora.

O Preço Público ([www.precopublico.com.br](http://www.precopublico.com.br)) é outro exemplo de comunidade virtual de compartilhamento de preços de produtos.

Na comunidade Preço Público, diferentemente do BuscaPé, os responsáveis por anunciar os preços dos produtos são os próprios usuários da comunidade virtual. Além disso, é possível compartilhar preços de empresas tradicionais e das empresas que praticam o *e-commerce*, porém o foco são as empresas tradicionais.

Dentre as funcionalidades que o usuário do Preço Público pode acessar serão descritas as seguintes: buscar os preços dos produtos pelo nome, ver os últimos preços publicados no sistema, publicar os preços dos produtos, acompanhar o preço dos produtos e cadastrar novos produtos e/ou novas empresas.

A funcionalidade **buscar os preços dos produtos por nome** possibilita o usuário realizar uma consulta por um produto. A Figura 3 mostra em destaque a área que o usuário pode acessar essa opção no Preço Público.



The screenshot shows the top section of the 'preço público' website. At the top, there is a login area with links for 'Cadastro-se Aqui' and 'Esqueci minha senha', followed by input fields for 'Email:' and 'Senha:', and an 'enviar' button. Below this is the website logo and a search bar with the placeholder text 'informe o nome do produto' and a 'buscar' button. A red box highlights the search bar and button, with a red arrow pointing to the 'buscar' button. Below the search bar is a navigation menu with links: Home, Sobre o projeto, Como participar, Contato, Publicar preço, and Compartilhar este site. The main content area is titled 'Últimos preços publicados:' and lists four items:

Produto	Empresa	Preço
cafeiteira dolce gusto	Angeloni	R\$ 299,00
Pao Seven Boys unidade	Padaria estrela	R\$ 1,90
etanol	Posto Ipiranga	R\$ 2,052
etanol	Posto Ipiranga	R\$ 2,052

At the bottom of the list, there are two buttons: 'publique um preço agora' and 'convide seus amigos'.

Figura 3 Busca por nome do produto (Fonte: www.precopublico.com.br).

A funcionalidade **ver os últimos preços publicados no sistema** possibilita o usuário ver na página inicial do sistema os últimos preços lançados

por outros usuários. A Figura 4 mostra em destaque onde está localizada na página inicial essa funcionalidade.

The screenshot shows the homepage of 'Preço Público'. At the top, there is a login section with fields for 'Email' and 'Senha' and an 'enviar' button. Below this is the site's logo and a search bar with the placeholder text 'informe o nome do produto' and a 'buscar' button. A navigation menu includes links for 'Home', 'Sobre o projeto', 'Como participar', 'Contato', 'Publicar preço', and 'Compartilhe este site'. The main content area is titled 'Últimos preços publicados:' and contains a list of four items:

Produto	Empresa	Preço
cafeteira dolce gusto	Angeloni	R\$ 299,00
Pao Seven Boys unidade	Padaria estrala	R\$ 1,90
etanol	Posto Ipiranga	R\$ 2,052
etanol	Posto Ipiranga	R\$ 2,052

At the bottom of the highlighted list, there are two buttons: 'publique um preço agora' and 'convide seus amigos'.

Figura 4 Últimos preços publicados (Fonte: www.precopublico.com.br).

A funcionalidade **publicar os preços dos produtos** permite aos usuários cadastrar no sistema os preços de produtos. A Figura 5 mostra a página de cadastro de produtos do Preço Público.

A funcionalidade **acompanhar o preço do produto** tem como objetivo relatar ao usuário, através do e-mail pessoal, quando um preço menor de um produto de seu interesse foi cadastrado no sistema.

Bem-vindo, Luciano Brasil Cardoso  
Caso este usuário não seja você clique aqui Brasil  
Alterar a minha localização

**preço público** informe o nome do produto

Home Sobre o projeto Como participar Contato **Publicar preço** Compartilhe este site

Cadastro de pesquisa de compra:

Preencha o formulário e clique no botão enviar dados.

Produto:  Estabelecimento:

Por exemplo: Lava-roupas 11 kg brastemp modelo bwl11A ou Leite longa vida integral 1 litro tirol

Preço:

Figura 5 Cadastro de produto no Preço Público (Fonte: www.precopublico.com.br).

Além disso, é possível filtrar para que seja recebido somente cadastro de preços de determinada localidade.

A Figura 6 mostra em destaque onde está localizada essa funcionalidade no Preço Público.

Bem-vindo, Luciano Brasil Cardoso  
Caso este usuário não seja você clique aqui Brasil  
Alterar a minha localização

**preço público** informe o nome do produto

Home Sobre o projeto Como participar Contato **Publicar preço** Compartilhe este site

Cadastro de pesquisa de compra:

As informações sobre o produto estão abaixo:

**Padaria estrala / Senac Pao Seven Boys unidade**

Brasil, Santa Catarina, Tubarão, Vila Moema  
[Ver mapa](#)

R\$ 1,90

Postivações:0  
Negativações:0

Esta informação é confiável?

Você comprou esse produto?

sabe de um preço melhor? **acompanhar preços**

Figura 6 Acompanhar preços (Fonte: www.precopublico.com.br).

Outra funcionalidade presente na Figura 6 é a classificação da confiabilidade do preço publicado. O usuário da comunidade pode determinar se o preço publicado é confiável ou não. Isso poderá ser feito através das opções que estão do lado direito do preço.

As funcionalidades de **cadastro de novos produtos e/ou novas empresas** só estarão habilitadas para uso quando o usuário for publicar algum preço de produto. Caso o produto e/ou empresa que o usuário digitar ainda não tiver sido cadastrado por algum outro usuário será possível cadastrar esses novos dados. A Figura 7 mostra como é possível ter acesso a essas opções.

Bem-vindo, Luciano Brasil Cardoso  
Caso este usuário não seja você clique aqui

Brasil  
Alterar a minha localização

**preço público**  
informe o nome do produto

Home Sobre o projeto Como participar Contato **Publicar preço** Compartilhar este site

Cadastro de pesquisa de compra:

Preencha o formulário e clique no botão enviar dados.

Produto:  Estabelecimento:

Selecione um produto abaixo ou refine a sua busca:

Pousada (ar condicionado, café da manhã, tv a cabo e Internet) - mensal

... ou cadastre um novo produto

Preço:

A empresa rex não foi encontrada, deseja cadastrá-la?

Figura 7 Cadastrar novos produtos e/ou novas empresas (Fonte: [www.precopublico.com.br](http://www.precopublico.com.br)).

O acesso às funcionalidades **buscar os preços dos produtos por nome** e **ver os últimos preços publicados no sistema** pode ser realizado por qualquer

usuário que acessa o sistema, mas as funcionalidades, **publicar os preços dos produtos**, e **cadastro de novos produtos e/ou novas empresas**, só podem ser utilizadas por usuários que estejam cadastrados no sistema.

O Preço Público fornece aos visitantes e aos participantes da comunidade virtual, outras funcionalidades, como por exemplo, cadastro de usuário, envio de convites para outras pessoas participarem da comunidade, entrar em contato com os administradores do sistema, determinar se os preços publicados por outros usuários são confiáveis e acesso a canais de comunicação como Twitter, Orkut, Facebook e Blog.

## **2.5 Tecnologias empregadas**

Esse tópico tem como objetivo apresentar as tecnologias que serão utilizadas neste trabalho.

### **2.5.1 Android**

O uso de aparelhos celulares tem crescido ao longo dos anos, com isso, têm surgido sistemas operacionais especialmente para a área de dispositivos móveis, entre eles está o sistema operacional Android que será descrito a seguir.

O Android é uma plataforma *open source* para desenvolvimento de aplicações para dispositivos móveis que foi criada em 2007 (MACARIO *et al*, 2009). A plataforma possui “*um sistema operacional, um middleware, aplicativos e interface do usuário*” (PEREIRA; SILVA, 2009).

Segundo Macario *et al* (2009), a arquitetura da plataforma Android é composta por cinco partes:

- *Kernel Linux*;
- Bibliotecas;
- *Android runtime*;
- *Framework* de aplicações;
- Aplicativos.

A Figura 8 mostra a arquitetura da plataforma Android e alguns recursos que possibilitam o desenvolvimento de aplicativos móveis.

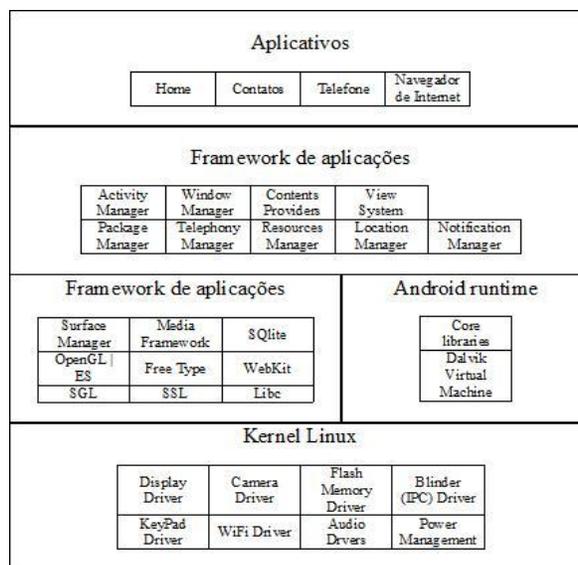


Figura 8 Arquitetura da plataforma Android (Fonte: adaptado pelo autor de <http://developer.android.com/guide/basics/what-is-android.html>).

A camada *Kernel Linux*, por exemplo, possui os recursos necessários para realizar a comunicação entre o *hardware* e o *software*<sup>4</sup>.

<sup>4</sup> <http://developer.android.com/guide/basics/what-is-android.html>

A plataforma Android não é a única opção para os desenvolvedores de aplicativos móveis. As plataformas iPhone e BlackBerry são exemplos de concorrentes do Android.

No desenvolvimento de aplicativos na plataforma Android é possível a criação de telas a partir de um arquivo XML. A Figura 9 mostra um exemplo de tela criada e a Figura 10 mostra o arquivo XML desse exemplo.



Figura 9 Exemplo de tela desenvolvido na plataforma Android (Fonte: próprio autor).

Também é possível o desenvolvimento de aplicativos que utilizem banco de dados e *web services*. No primeiro caso pode ser usado SQLite como ferramenta para a persistência de dados, e para o segundo caso existe o *framework* kSOAP2 que auxilia na criação de *web services*.

O SQLite é uma biblioteca que trabalha com o banco de dados SQL e já está presente nos dispositivos móveis Android. O *framework* kSOAP é direcionado para a criação de *web services* em dispositivos móveis

A plataforma Android foi escolhida por dar suporte à linguagem Java, sendo que o Preço Público foi desenvolvido também nessa linguagem, com isso a integração das duas aplicações será facilitada.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent">
<TextView
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Digite o primeiro número"/>
<EditText
android:id="@+campo/num1"
android:layout_width="fill_parent"
android:layout_height="wrap_content"/>
<TextView
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Digite o segundo número"/>
<EditText
android:id="@+campo/num2"
android:layout_width="fill_parent"
android:layout_height="wrap_content"/>
<TextView
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Resultado"/>
<EditText
android:id="@+campo/res"
android:layout_width="fill_parent"
android:layout_height="wrap_content"/>
<Button
android:id="@+botao/btmostrar"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Mostrar"/>
</LinearLayout>
```

Figura 10 Arquivo XML com a definição da interface da Figura 9 (Fonte: próprio autor).

### 2.5.2 Java

A linguagem Java foi apresentada pela empresa Sun em maio de 1995 (DEITEL; DEITEL, 2005). Nasceu com o objetivo de ser utilizada para desenvolvimento de aplicativos embutidos, mas acabou ganhando espaço em outra área, o desenvolvimento Web (VAREJÃO, 2004).

A linguagem de programação Java possui uma especificação técnica que padroniza a sintaxe e a semântica. Além disso, disponibiliza uma API (*Application Program Interface*) que determina as classes e interfaces para o desenvolvimento de sistemas em Java. A especificação técnica e a API são responsáveis por definir a linguagem Java de programação (LIANG, 2009).

A empresa Oracle comprou a Sun em 2009, e atualmente é responsável por manter e atualizar os produtos relacionados à linguagem Java.

A linguagem Java possui as seguintes plataformas integradas: Java SE, Java ME, Java EE.

A plataforma Java ME permite o desenvolvimento de aplicações para dispositivos móveis (COSTA *et al*, 2010). A Java SE é mais abrangente que a Java ME, e permite a criação de *softwares* de pequeno porte específicos para *Desktops* e servidores. A Java EE é utilizada para o desenvolvimento de sistemas empresariais de grande porte<sup>5</sup>.

Segundo Andreão (2007), a linguagem Java possui as seguintes características:

---

<sup>5</sup><http://www.oracle.com/us/technologies/java/index.html>

- Simplicidade;
- Orientada a objetos;
- Portabilidade;
- Possui uma grande quantidade de bibliotecas;
- Desalocação automática de memória através do *Garbage Colector*.

Essas características motivaram o uso da linguagem Java no desenvolvimento de sistemas computacionais de vários tipos (GONÇALVES, 2007).

O crescente uso da linguagem Java proporcionou o surgimento de conjuntos de códigos responsáveis por prover uma solução para uma família de problemas (ou *frameworks*), para auxiliar o desenvolvimento de sistemas nessa linguagem. No próximo tópico será apresentado o *framework* Spring.

### 2.5.3 Spring

O Spring é um *framework* de código aberto que auxilia o desenvolvimento de aplicações para a plataforma Java EE (MELA *et al*, 2008). Foi criado em 2004 por Rod Johnson<sup>6</sup> e atualmente é mantido sob a licença Apache 2.0.

---

<sup>6</sup><http://static.springsource.org/spring/docs/current/spring-framework-reference/html/index.html>

Segundo Arthur e Azadegan (2005), o Spring tem como objetivo agregar testabilidade, simplicidade e portabilidade ao desenvolvimento da aplicação. Para que isso seja possível, esse *framework* possui as seguintes características<sup>7</sup>:

- Possui um *container* leve e completo: esse *container* facilita a testabilidade e escalabilidade do sistema, permitindo que partes da aplicação sejam desenvolvidas e testadas de maneira isolada antes da implantação do sistema.
- Camada comum para gerenciamento de transações: essa camada facilita o controle de transações no banco de dados.
- Camada de abstração JDBC (*Java Database Connectivity*): facilita o tratamento de erros e diminui a quantidade de código para o uso de JDBC.
- Integração com o *framework* Hibernate: essa característica facilita a persistência de dados no sistema.

No próximo tópico será apresentado o *framework* de persistência de dados Hibernate e o sistema de gerenciamento de banco de dados MySQL.

#### 2.5.4 Hibernate e MySQL

O Hibernate é um *framework* responsável por mapear objetos da linguagem Java para dados de modelo relacional usando SQL (*Structured Query Language*) (KING *et al*). Originalmente era possível somente o mapeamento de objetos da linguagem Java, mas atualmente o Hibernate tem o suporte para outras linguagens.

---

<sup>7</sup><http://www.springsource.org/about>

A configuração do Hibernate pode ser feita de maneira simples e esse fato faz com que o desenvolvedor economize tempo com a configuração do banco de dados e preocupe-se com a lógica de negócio (GONÇALVES, 2007).

O MySQL é um sistema gerenciador de banco de dados. Além de ser portátil para várias plataformas, tem alto desempenho, confiabilidade e facilidade de uso<sup>8</sup>.

A união do Hibernate e o MySQL permite o desenvolvedor de *software* criar aplicações que conectem ao banco de dados utilizando a programação orientada a objetos.

### 2.5.5 Web Service

Web Service é uma tecnologia que utiliza padrões para a integração entre diferentes sistemas (AGARWAL *et al*, 2007). Utilizando essa tecnologia, sistemas podem ser integrados mesmo que sejam desenvolvidos em linguagem diferentes e que estejam distribuídos em uma rede interna ou externa.

A padronização utilizada pela tecnologia Web Service para a integração entre os sistemas é realizada através de linguagens e protocolos, como por exemplo, UDDI (*Universal Description, Discovery and Integration*), WSDL (*Web Services Description Language*) e SOAP (*Simple Object Access Protocol*) (BENATALLAH *et al*, 2003).

A Figura 11 mostra o funcionamento básico da tecnologia Web Service utilizando as linguagens e os protocolos citados anteriormente.

---

<sup>8</sup><http://www.mysql.com/why-mysql/>

Essa figura mostra uma aplicação cliente e um Web Service localizado no servidor. O funcionamento é o seguinte: o cliente invoca o serviço para o servidor e o Web Service retorna a resposta para o cliente, tal comunicação é realizada através dos protocolos HTTP e SOAP.

A facilidade do uso de Web Services utilizando os protocolos HTTP e SOAP, e a linguagem WSDL é que os sistemas que serão integrados por essas tecnologias são independentes de linguagens, ou seja, uma aplicação cliente desenvolvida na linguagem C# pode ser integrada a uma aplicação servidor escrita na linguagem Java.

Essa monografia utiliza a tecnologia Web Service para integrar uma aplicação desenvolvida em Android com a comunidade virtual Preço Público.

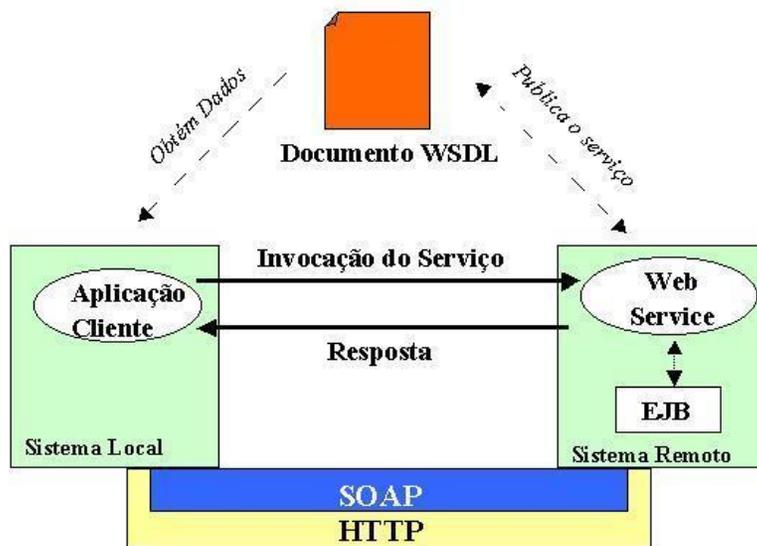


Figura 11 Esquema do funcionamento da tecnologia Web Service (Fonte: [http://devedge-temp.mozilla.org/viewsource/2002/soap-overview/index\\_pt\\_br.html](http://devedge-temp.mozilla.org/viewsource/2002/soap-overview/index_pt_br.html)).

### 3 MÉTODO

Nesse capítulo será descrito o problema abordado por esse trabalho, a solução e o protótipo desenvolvidos.

#### 3.1 Problema

A funcionalidade **publicar preço dos produtos** que *software* social Preço Público possui e que foi descrito brevemente na seção 2.4, será detalhado a seguir.

Para que o usuário da comunidade virtual Preço Público possa publicar algum preço de produto, é necessário em primeiro lugar fazer o *login* no sistema, logo em seguida, ir até ao *link* **Publicar preço** na parte superior da página (Figura 12). Nessa página é solicitado que o usuário entre com o nome do produto, o estabelecimento que foi encontrado e o preço. Se o produto ou o estabelecimento nunca foram cadastrados no sistema, o usuário tem a possibilidade de fazer esses cadastros nesse passo. Para finalizar o processo, o usuário deve pressionar o botão **Enviar dados**.

Após realizar o cadastro do preço do produto, é possível ver na página inicial do Preço Público a atualização realizada.

A Figura 12 mostra o fluxograma atual do processo para publicar preços de produtos no Preço Público.

Tendo em vista o atual modelo de cadastro de preços de produtos no Preço Público e o aumento do uso de dispositivos móveis, essa monografia tem o objetivo de incluir uma nova forma de cadastro de preços de produtos no Preço Público utilizando dispositivos móveis que utilizam o sistema operacional Android.

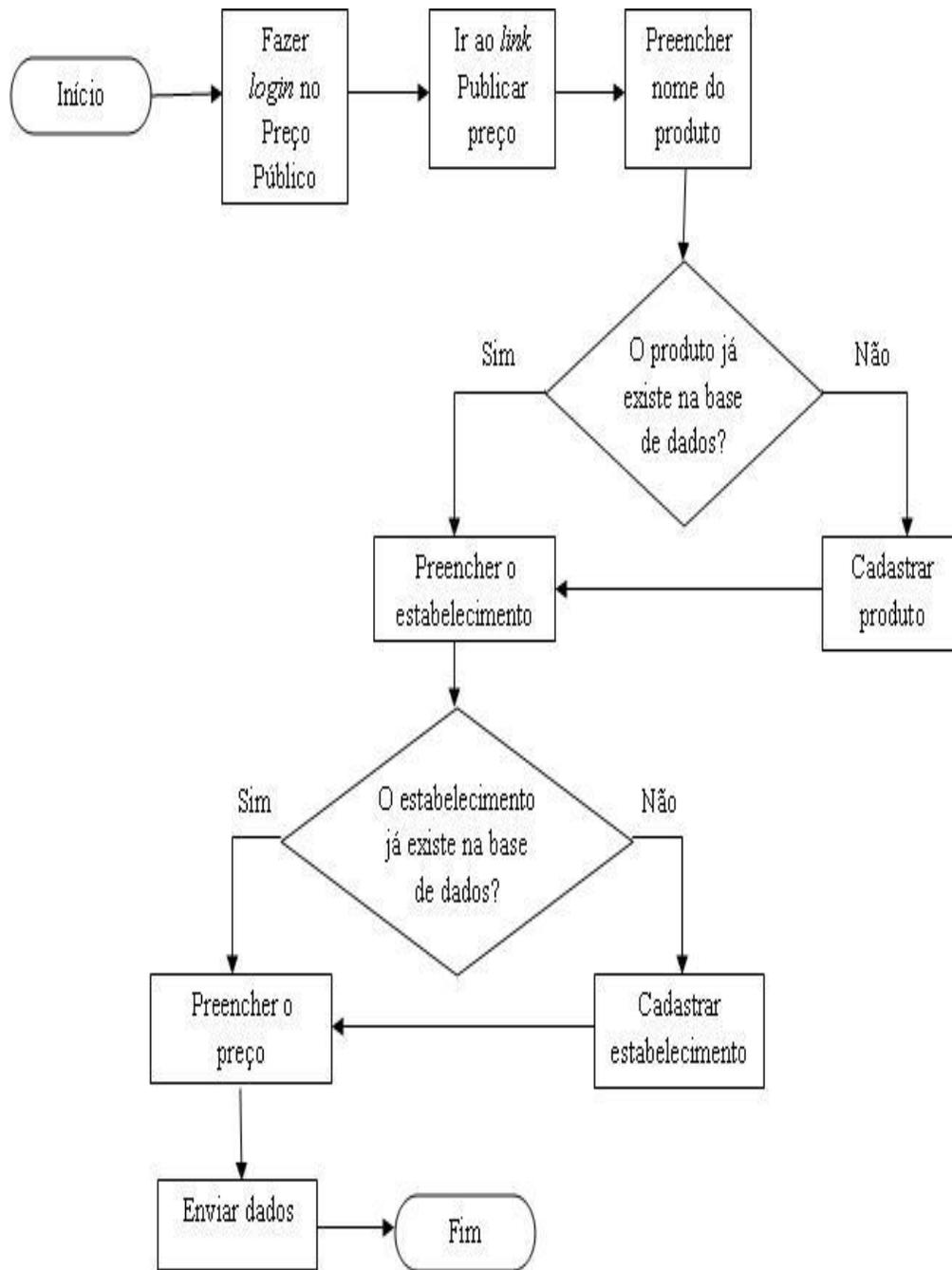


Figura 12 Fluxograma que representa o atual processo para publicar preço no Preço Público (Fonte: próprio autor).

Do ponto de vista do usuário, a nova forma de cadastro de preços de produtos no Preço Público será feita da seguinte maneira: o usuário da comunidade virtual Preço Público acessa o aplicativo do Preço Público para o Android, onde será possível armazenar o código de barras do produto através de um *scanner* (o funcionamento do *scanner* será explicado na seção 3.3.1) ou manualmente, uma descrição e o preço do produto. Esse processo pode ser repetido para vários produtos, sendo que todos esses dados são gravados no banco de dados do próprio aparelho celular.

Logo em seguida, é necessário enviar todas essas informações para o servidor do Preço Público. Para que isso seja possível, o dispositivo móvel precisa estar conectado à Internet. Essa é a etapa 1 que está presente na Figura 13.

Quando desejar, o usuário pode fazer o *login* no Preço Público, clicar no *link Mobile*, em seguida será listado todos os produtos que foram armazenados através do dispositivo móvel.

Os produtos que já possuem o código de barras cadastrado na base de dados do Preço Público serão listados através do nome, caso contrário, aparecerá uma mensagem avisando que o código de barras ainda não foi cadastrado.

Para publicar determinado preço, o usuário deverá acessar o *link Editar*, que irá redirecioná-lo para a página onde é possível editar o nome do produto (caso o código de barras não tenha sido cadastrado), o estabelecimento e o preço. Por fim poderá enviar os dados para o Preço Público. Essa é a etapa 2 que está presente na Figura 13

A Figura 13 mostra o fluxograma da nova forma de cadastro de preços utilizando dispositivos móveis Android, dando ênfase nas duas etapas que o processo possui.

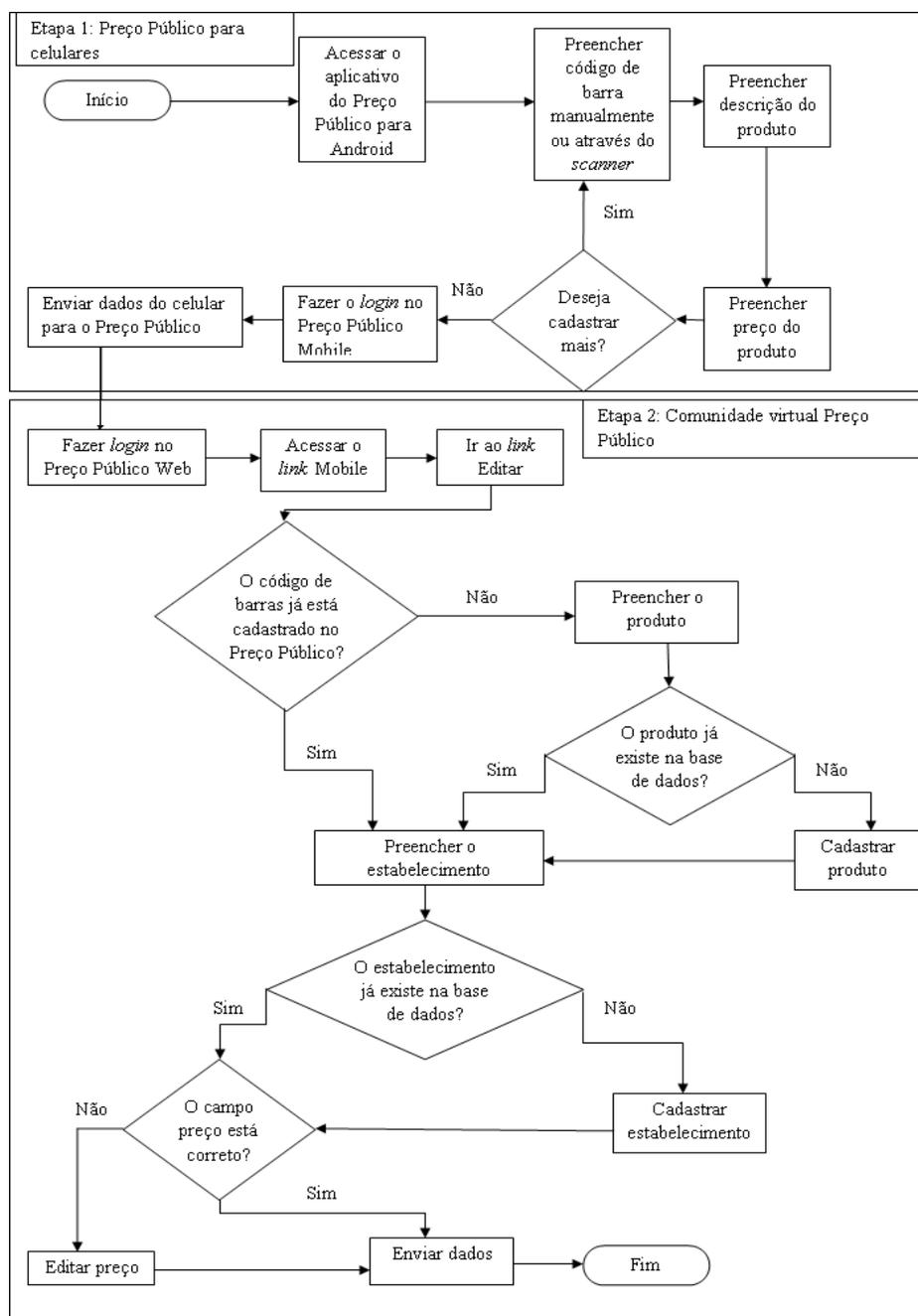


Figura 13 Fluxograma para o cadastro de preços no Preço Público através do Android (Fonte: próprio autor).

### 3.2 Solução desenvolvida

Nessa seção será apresentada a solução para o problema descrito anteriormente.

O primeiro passo desenvolvido foi o diagrama da arquitetura do subsistema para cadastro de preços de produtos através de dispositivos móveis. A Figura 14 mostra esse diagrama.

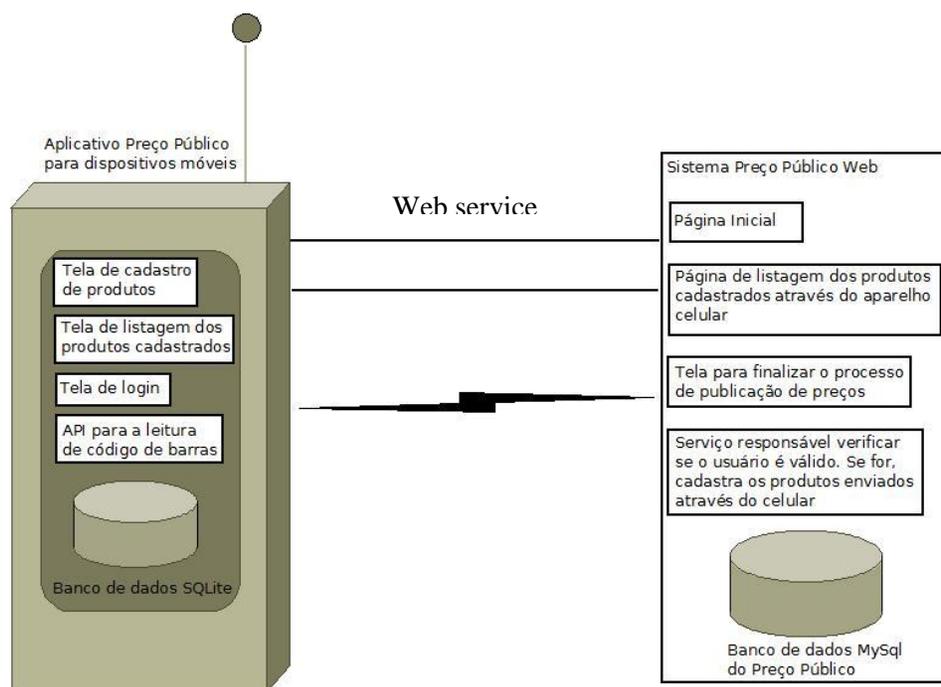


Figura 14 Diagrama da arquitetura do subsistema para cadastro de preços no Preço Público através de dispositivos móveis (Fonte: próprio autor).

Esse diagrama pode ser dividido em duas partes: sistema Preço Público *mobile* e *web*.

O sistema Preço Público *mobile* é um aplicativo desenvolvido para celulares que possui sistema operacional Android e tem o objetivo de armazenar preços de produtos.

Os dados coletados pelo dispositivo móvel são enviados para o sistema *web* através do uso da tecnologia *Web Service* e inseridos no banco de dados do sistema de destino (Preço Público *web*).

Para publicar os dados enviados, o usuário deverá acessar o sistema Preço Público *web* e finalizar o processo de cadastro dos preços.

No próximo tópico serão apresentados os seguintes itens relacionados ao Preço Público *mobile*: o diagrama de fluxo das telas, o modelo de interface de cada tela e o modelo de dados.

### **3.2.1 Preço Público *mobile***

O aplicativo Preço Público *mobile* possui três telas e o diagrama de fluxo é apresentado na Figura 15.

Cada botão presente nas telas do aplicativo possui a responsabilidade de redirecionar o usuário para alguma outra tela ou para ativar alguma funcionalidade. A seguir, será detalhado como cada botão funciona:

- Botão Leitura: faz a leitura de código de barras.
- Botão Cadastrar: insere o produto atual e possibilita cadastrar outro.
- Botão Enviar para o Preço Público: redireciona para a tela de listagem dos produtos.
- Botão Voltar: volta para a tela de cadastro de produtos.

- Botão Confirmar: confirma o envio dos dados cadastrados através do celular para o sistema Preço Público *web*.
- Botão Efetuar login: confirma o usuário e envia os dados para o Preço Público *web*.
- Botão Cancelar: cancela o *login* e retorna para a página de listagem dos produtos cadastrados.

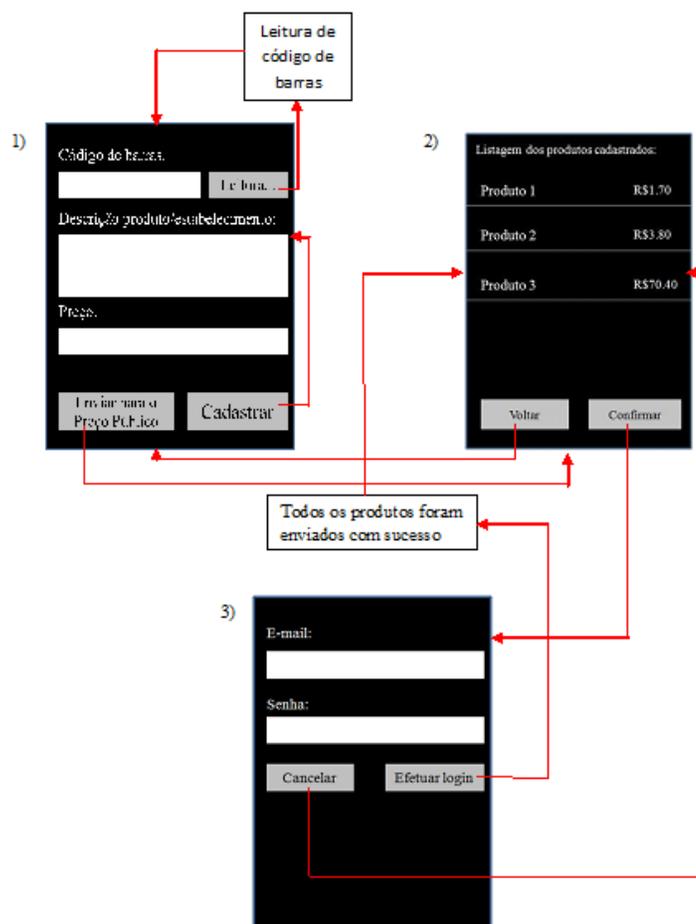


Figura 15 Diagrama do fluxo das telas do aplicativo Preço Público *Mobile* (Fonte: próprio autor).

Após essa visão geral do fluxo das telas do sistema *mobile*, será detalhada cada tela do diagrama.

A Figura 16 mostra o modelo de interface da tela inicial do aplicativo responsável pelo cadastro de produtos.

1)

Código de barras:

Leitura...

Descrição produto/estabelecimento:

Preço:

Cadastrar

Enviar para o Preço Público

Figura 16 Modelo de interface da tela inicial do aplicativo Preço Público *Mobile* (Fonte: próprio autor).

Esse modelo mostra três campos, um para a entrada do código de barras, outro para a descrição do produto e/ou estabelecimento e, por fim, um para o preço. Existem três botões, um para ativar a ação da leitura do código de barras, além desse, um para confirmar o cadastro e outro pra enviar os dados para o Preço Público.

A Figura 17 mostra o modelo de interface da tela de listagem dos preços cadastrados.

2)

Listagem dos produtos cadastrados:	
Produto 1	R\$1.70
Produto 2	R\$3.80
Produto 3	R\$70.40

Voltar      Confirmar

Figura 17 Modelo de interface da tela de listagem dos preços cadastrados no aplicativo Preço Público para dispositivos móveis (Fonte: próprio autor).

Esse modelo mostra uma listagem de todos os preços cadastrados no dispositivo móvel e possui dois botões, um para voltar para a tela de cadastro de produtos e outro para confirmar o envio dos dados listados para o sistema Preço Público.

A Figura 18 mostra o modelo de interface da tela de *login* do Preço Público *mobile*.

3)

O modelo de interface da tela de login do aplicativo Preço Público para dispositivos móveis apresenta um fundo preto com um contorno azul. No topo, o rótulo "E-mail:" precede um campo de entrada branco. Abaixo dele, o rótulo "Senha:" precede outro campo de entrada branco. Na base da interface, há dois botões retangulares cinza: "Cancelar" à esquerda e "Efetuar login" à direita.

Figura 18 Modelo de interface da tela de *login* do aplicativo Preço Público para dispositivos móveis (Fonte: próprio autor).

Esse modelo mostra dois campos, um para a entrada do e-mail e outro para a senha do usuário, além disso, possui um botão para confirmar o usuário e outro para cancelar o *login*.

Além dos modelos de interface mostrados anteriormente, foi desenvolvido o modelo de dados do aplicativo do Preço Público para dispositivos móveis, a Figura 19 mostra esse modelo.

Produto	
🔑	id_produto: INTEGER
◆	codigobarra: BIGINT
◆	descricao: TEXT
◆	preco: DOUBLE

Figura 19 Modelo de dados do aplicativo Preço Público para dispositivos móveis (Fonte: próprio autor).

O modelo de dados apresentado possui somente a tabela **Produto**, para armazenar os dados relativos aos produtos cadastrados. Essa tabela contém as seguintes colunas: **id\_produto**, **codigobarra**, **descricao**, **preco**.

O campo **id\_produto** tem a característica de identificar os produtos, e é definido como *autoincrement* e não pode ser nulo, o tipo aceito é o Integer. A coluna **codigobarra** aceita só valores BigInt e é responsável por armazenar os códigos de barras cadastrados no celular. A **descricao** aceita somente textos e é responsável por guardar as descrições dos produtos e/ou estabelecimentos. No campo **preço** é permitida a entrada somente de valores Double e tem a responsabilidade de armazenar os preços dos produtos cadastrados.

Os dados inseridos nessa tabela são enviados para o Preço Público *web*, e logo em seguida são excluídos. Caso ocorra algum erro no envio esses dados não são excluídos e o usuário pode tentar enviar novamente.

O próximo tópico irá apresentar o modelo de integração responsável por enviar os dados do aplicativo *mobile* para o sistema *web* do Preço Público.

### 3.2.2 Diagrama de integração entre o sistema *web* e o aplicativo *mobile* do Preço Público

A Figura 20 apresenta o diagrama de integração dos dois sistemas envolvidos no problema desse trabalho.

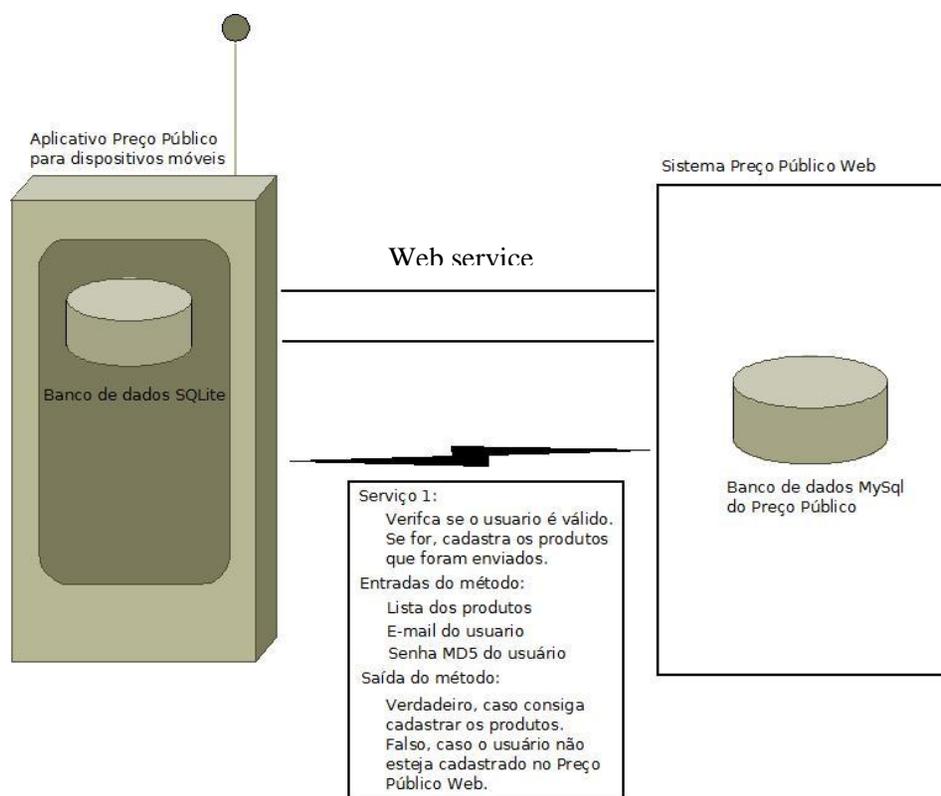


Figura 20 Modelo de integração entre o aplicativo Android e o sistema Preço Público (Fonte: próprio autor).

No desenho é possível ver que a integração é realizada através de um serviço.

Esse serviço recebe três parâmetros:

- Uma lista contendo preço, código de barras e descrição de cada produto;
- O e-mail do usuário;
- A senha MD5 do usuário.

A lista é recuperada do banco de dados SQLite do sistema *mobile* do Preço Público. O e-mail e a senha foram informados pelo usuário na tela mostrada na Figura 18, porém, por questões de segurança no envio dos dados, a senha foi criptografada utilizando o algoritmo MD5.

Após o envio dos dados para o sistema Preço Público *web*, o serviço é responsável por verificar se o usuário está cadastrado no sistema. Se estiver, é feito a inserção da lista de produtos no banco de dados MySQL. Caso contrário, o serviço não cadastra os dados.

No próximo tópico serão apresentados os seguintes itens relacionados ao sistema Preço Público *web*: o modelo de dados alterado, o diagrama de fluxo das telas, o modelo de interface de cada tela.

### 3.2.3 Preço Público *web*

O sistema Preço Público *web* teve seu modelo de dados alterado como mostrado na Figura 21.

Esse modelo mostra somente as tabelas importantes para o desenvolvimento do trabalho.

As partes em destaque na figura mostram as alterações que foram feitas no modelo de dados. A seguir é explicada cada alteração que foi realizada:

- A tabela **produtomob** tem o objetivo de armazenar os dados enviados pelo celular e possui os seguintes campos: **id**, **codigobarra**, **descricao**, **preco**, **moderado**, **id\_usuario**.
- A tabela **produto** foi alterada com a inclusão da coluna **codigobarra**, tal modificação foi necessária, pois o Preço Público *web* não possuía suporte para armazenar o código de barras do produto.

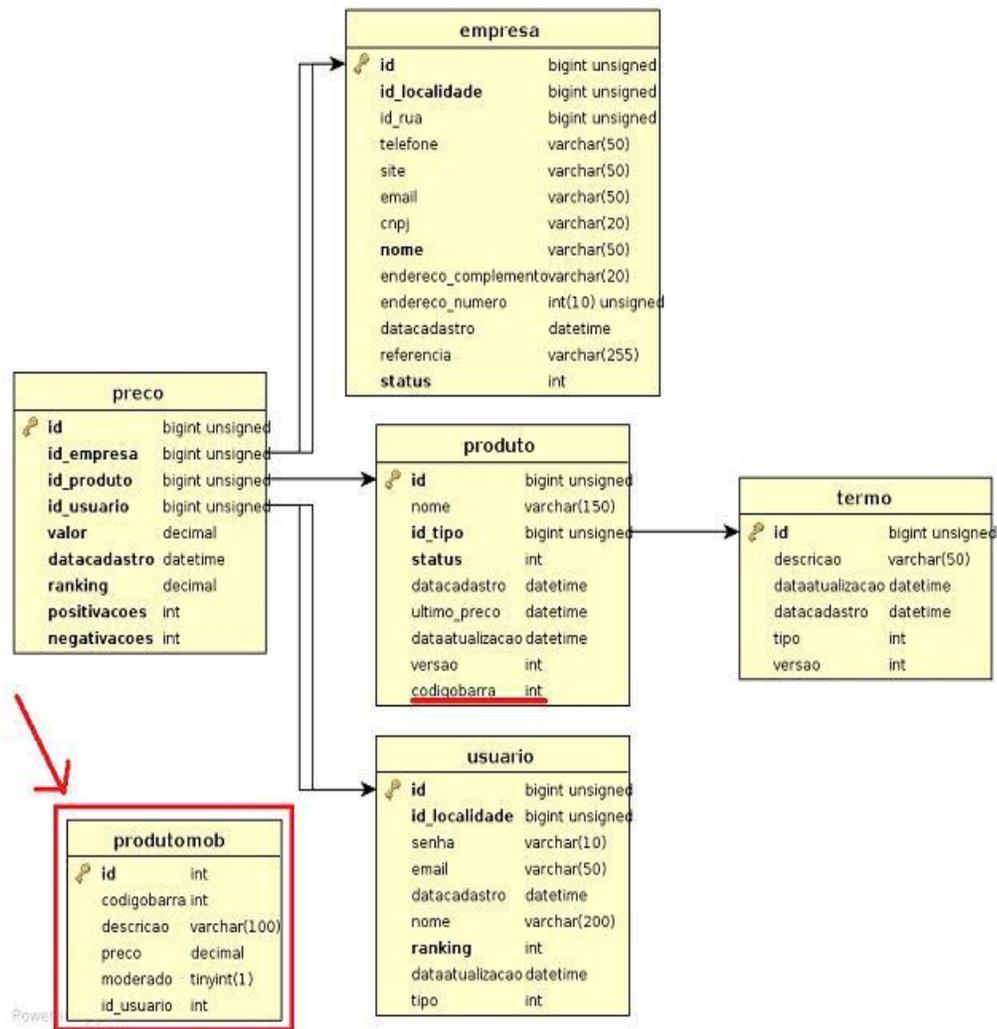


Figura 21 Parte do modelo de dados alterado do sistema Preço Público Web (Fonte: próprio autor).

Além da alteração no modelo de dados, foi criado um diagrama do fluxo das páginas para que o usuário da comunidade virtual Preço Público pudesse acessar os dados enviados pelo celular. A Figura 22 mostra esse diagrama de fluxo.



Figura 22 Diagrama de fluxo das páginas do sistema Preço Público web (Fonte: próprio autor).

Esse diagrama mostra a funcionalidade dos seguintes *links*:

- *Mobile*: redireciona o usuário para a página que lista os preços que foram enviados do celular.
- *Editar*: ao acessar esse *link* o usuário é redirecionado para a página de publicação do preço.
- *Clique aqui se não for esse produto*: Se o nome do produto estiver errado, significa que em algum momento o código de barras foi vinculado de maneira errada para aquele produto. Sendo assim, o usuário pode consertar esse erro acessando esse *link*.
- *Enviar dados*: tem a responsabilidade de publicar o preço.

Após essa visão geral do fluxo das páginas do sistema web, será detalhada cada tela do diagrama.

A seguir será mostrada a Figura 23 que representa o modelo de interface da página inicial do Preço Público.

1)



Figura 23 Modelo de interface representando a página inicial do sistema Preço Público web (Fonte: próprio autor).

No modelo está destacado o *link Mobile* que foi acrescentado para que seja possível o acesso aos preços cadastrados através do dispositivo móvel.

O modelo de interface da página de listagem de preços cadastrados através do celular é mostrado a seguir na Figura 24.

## 2) Preço Público



Figura 24 Modelo de interface representando a página de listagem de preços cadastrados através do celular (Fonte: próprio autor).

Esse modelo mostra a listagem de todos os preços que foram cadastrados através do celular e que podem ser publicados no Preço Público. Quando o código de barras do produto já foi cadastrado na comunidade virtual, o preço é listado pelo nome, caso contrário, é informado para o usuário que o código de barras ainda não foi cadastrado. Outras informações que são mostradas são: o número do código de barras, a descrição e o preço do produto.

O *link Excluir* será utilizado caso o usuário desista de publicar determinado produto na comunidade virtual Preço Público. Quando o *link* é acessado ocorre à exclusão do produto da tabela **produtomob**.

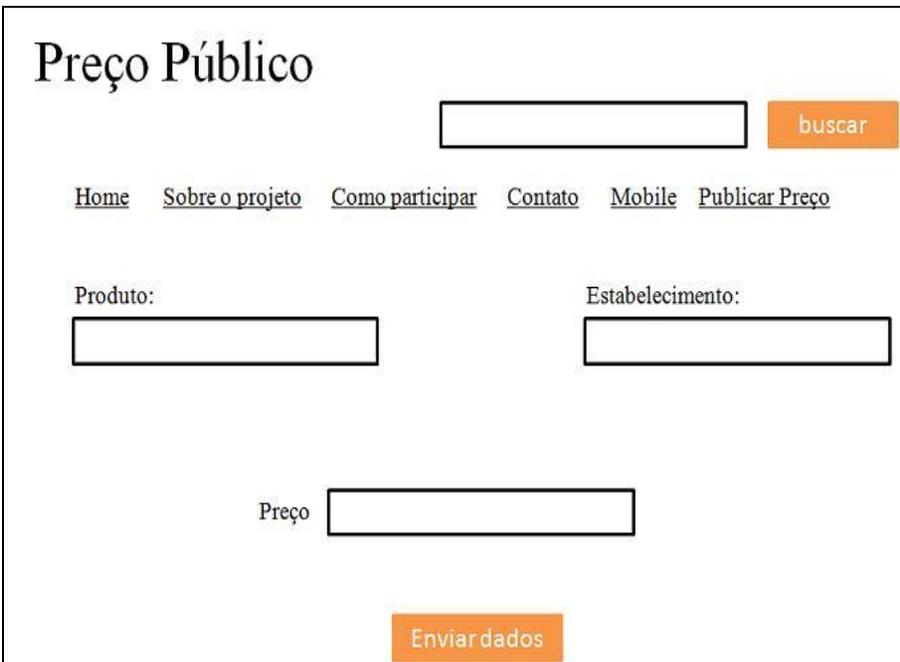
O *link Clique aqui para baixar o aplicativo* possibilita ao usuário fazer o *download* do arquivo de instalação do aplicativo Preço Público para dispositivos móveis.

Para publicar determinado preço o usuário deve acessar o *link* **Editar**, e será redirecionado para a página de cadastro de preço que é mostrada a seguir na Figura 25.

Esse modelo possui os campos produto, estabelecimento e preço. Caso o produto e/ou o estabelecimento necessitem serem cadastrados, a interface será igual à mostrada na Figura 8.

Enquanto o usuário vai entrando com o nome do produto ou do estabelecimento o Preço Público realiza uma busca semântica em seu banco de dados e mostra as palavras mais parecidas que são encontradas.

3)



O modelo de interface para publicação de preços apresenta o seguinte layout:

- Um campo de busca com o botão "buscar" em um botão laranja.
- Um menu de navegação com links: [Home](#), [Sobre o projeto](#), [Como participar](#), [Contato](#), [Mobile](#) e [Publicar Preço](#).
- Dois campos de texto rotulados "Produto:" e "Estabelecimento:".
- Um campo de texto rotulado "Preço".
- Um botão laranja "Enviar dados" na base da interface.

Figura 25 Modelo de interface representando a página para publicação dos preços dos produtos coletados através do dispositivo móvel (Fonte: próprio autor).

Ao preencher os campos o usuário pressiona o botão **Enviar dados** para finalizar o processo de publicação de preços. Por fim, o usuário é redirecionado para a página inicial.

Esse modelo foi reaproveitado do sistema antigo do Preço Público, portanto, quando o usuário clicar no botão **Enviar dados**, as informações seguem o fluxo normal de cadastro e são armazenadas nas seguintes tabelas mostradas na Figura 21: **preço, empresa, produto, usuário e termo**.

A próxima seção irá descrever o protótipo que foi desenvolvido a partir dessa solução.

### 3.3 Protótipo

Esse tópico tem como objetivo descrever o protótipo desenvolvido a partir do modelo descrito anteriormente.

#### 3.3.1 Preço Público *mobile*

O diagrama abaixo representa a estrutura das classes que foram criadas para o Preço Público *mobile*.

O desenvolvimento do protótipo teve início com a criação das telas do Preço Público *mobile*. Os componentes criados nessa fase foram os botões, os campos texto e os *labels*.

Na Figura 26 é mostrado a relação das classes **MainActivity**, **ListaProdutoActivity** e **LoginActivity** com um arquivo XML, que é responsável por definir os componentes que estarão presentes nas telas do protótipo.

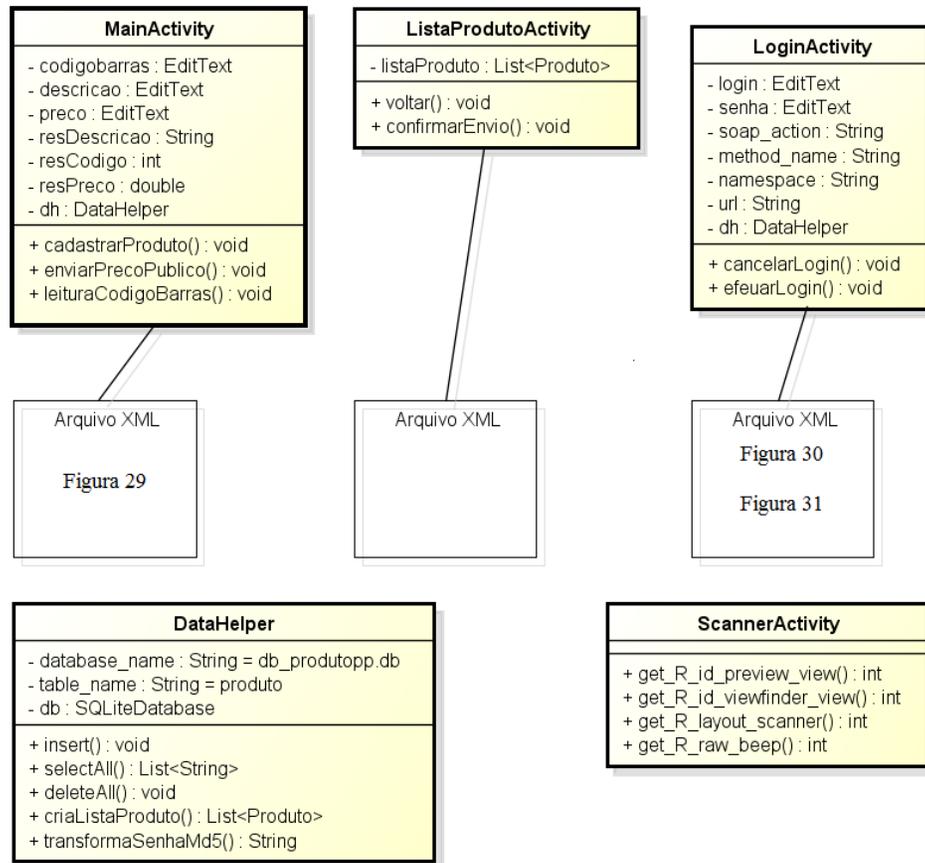


Figura 26 Diagrama de classes do sistema Preço Público *mobile* (Fonte: próprio autor)..

Um botão é criado como mostra o código da Figura 27.

```

<Button
    android:id="@+botao/cadastro"
    android:layout_height="wrap_content"
    android:layout_width="fill_parent"
    android:text="Cadastrar"
    android:onClick="cadastrarProduto"
    android:layout_weight="1"
/>
  
```

Figura 27 Código para a criação de um botão para o Preço Público *mobile* (Fonte: próprio autor).

Os atributos que foram definidos para o botão são: o identificador, largura, altura e o texto que estará no botão. Nesse exemplo, o botão que foi criado é igual ao botão **Cadastrar** desenhado na Figura 16.

O atributo **onClick** determina qual método será executado quando o usuário clicar nesse botão, no caso, será chamado o método **cadaststrarProduto**.

O diagrama de classes da Figura 26 mostra que a implementação do método **cadaststrarProduto** está presente na classe **MainActivity**. E que o arquivo XML que contém o código da Figura 27 está associado a essa classe.

Um componente campo texto é criado através do código mostrado na Figura 28.

```
<EditText
    android:id="@+campo/login"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:maxLines="1"
/>
```

Figura 28 Código para a criação de um campo texto para o Preço Público *mobile* (Fonte: próprio autor).

Os atributos que foram definidos para o campo texto são: identificador, largura, altura, tamanho do texto e o número máximo de linhas que suporta o campo texto.

Esse código foi desenvolvido para a criação do campo texto **login** que foi mostrado na Figura 18, e está posicionado logo abaixo da *label* **E-mail**.

O diagrama de classes mostrado na Figura 26 apresenta a classe **LoginActivity** que possui o atributo **login**, responsável por receber os dados do campo texto criado através do código da Figura 28.

Um rótulo é criado a partir do código mostrado na Figura 29.

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Senha:"
/>
```

Figura 29 Código para a criação de uma *label* para o Preço Público *mobile* (Fonte: próprio autor).

Os seguintes atributos pertencem a esse rótulo: largura, altura e texto. Esse exemplo é igual ao rótulo **Senha:** mostrada na Figura 18.

Os códigos das Figuras 28 e 29 estão no arquivo XML que está associado à classe **LoginActivity**, como mostrado na Figura 26.

Dessa maneira, foram desenvolvidos os modelos de interface apresentados nas Figuras 16, 17 e 18.

O botão **Leitura...** (Figura 17) ativa a opção de leitura por código de barra, o desenvolvimento dessa funcionalidade foi possível com o uso da biblioteca *biggu* que realiza a leitura de código de barras. A Figura 30 mostra o funcionamento do *scanner* no emulador do Android.

Não foi possível tirar um *print screen* do leitor de código de barras funcionando no aparelho celular.

O segundo passo no desenvolvimento do protótipo foi realizar o fluxo entre as telas já desenvolvidas.

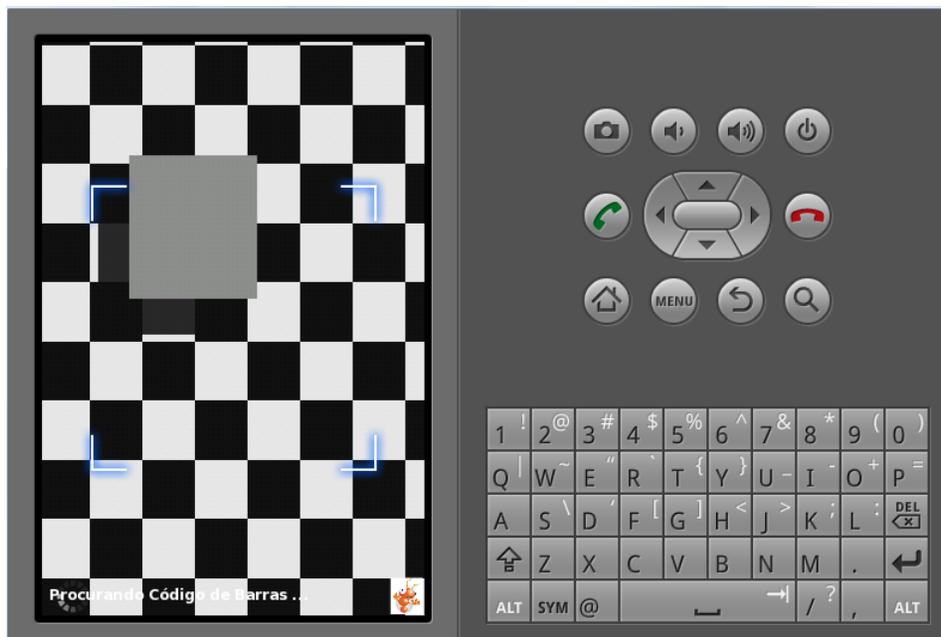


Figura 30 *Print Screen* do Leitor de código de barras funcionando no emulador Android (Fonte: próprio autor).

Para essa fase, o código mostrado na Figura 31 foi desenvolvido.

```
public void voltar(View v){
    Intent intent = new Intent(v.getContext(),org.me.precopublicomobile.MainActivity.class);
    startActivity(new Intent(intent));
}
```

Figura 31 Código para a criação de fluxo entre telas no Preço Público *mobile* (Fonte: próprio autor).

Essa figura mostra o método **voltar** pertencente à classe **ListaProdutoActivity** (mostrado no diagrama de classes da Figura 26), que é executado quando o usuário clica no botão **voltar** presente no modelo de interface mostrado na Figura 17.

O código da Figura 31 cria uma intenção para retornar à página inicial do aplicativo e inicia essa atividade. Sempre que for necessário realizar o fluxo

de telas será necessário o desenvolvimento dessas linhas de código, alterando somente a classe de destino.

Com isso, foi desenvolvido o fluxo entre telas que foi mostrado na Figura 15.

A terceira fase do desenvolvimento do protótipo está relacionada com o armazenamento dos produtos cadastrados. Na Figura 26 é mostrada a classe **DataHelper**, que é responsável por realizar a persistência dos dados no banco de dados SQLite embutido no aparelho celular.

O método **cadastrarProduto** pertencente à classe **MainActivity**, é responsável por receber os dados (código de barras, descrição e preço) do produto, criar uma instância da classe **DataHelper** e enviar os dados para o método **insert**. Esse método é responsável por armazenar os dados recebidos no banco de dados SQLite (utilizando a linguagem SQL padrão). A Figura 32 mostra o código do método **insert**.

```
public void insert(int codigobarra, String descricao, double valor){  
    db.execSQL("INSERT INTO produto (codigobarra,descricao,valor) VALUES "  
        + "("+codigobarra+";"+descricao+"; "+valor+");");  
}
```

Figura 32 Código para inserir dados no banco de dados SQLite (Fonte: próprio autor).

Após gravar os dados dos produtos no banco de dados do aparelho celular, é necessário enviar todas essas informações para o sistema Preço Público *Web*, essa função é realizada através da implementação de um Web Service. O próximo tópico irá descrever como foi utilizada essa tecnologia.

### 3.3.2 Integração do Preço Público *mobile* e *web*

Antes de enviar os dados para o banco de dados do Preço Público *web* foi necessário criptografar a senha do usuário, utilizando o algoritmo MD5 (algoritmo que tem como objetivo criptografar dados), para garantir a segurança dos dados durante o envio das informações através do Web Service. E também foi efetuado um processamento para que a lista de produtos ficasse no padrão JSON (formato de dados que transforma um objeto em um determinado padrão de String), essa abordagem só foi utilizada porque o programa dava erro ao enviar o objeto do tipo *List* serializado que estava definido para a lista de produtos.

A figura abaixo mostra o método responsável por realizar a criptografia da senha do usuário.

```
public String transformaSenhaMd5(String s) {
    try {
        // Cria MD5 Hash
        MessageDigest digest = java.security.MessageDigest.getInstance("MD5");
        digest.update(s.getBytes());
        byte messageDigest[] = digest.digest();
        // Cria Hex String
        StringBuffer hexString = new StringBuffer();
        for (int i=0; i<messageDigest.length; i++)
            hexString.append(Integer.toHexString(0xFF & messageDigest[i]));
        return hexString.toString();
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
    return "";
}
```

Figura 33 Código para criptografar a senha do usuário (Fonte: próprio autor).

Esse método recebe como entrada a senha normal do usuário e retorna a senha criptografada no formato MD5. Após o uso desse método, a senha do usuário poderá ser enviada com segurança para Preço Público *web*.

A figura abaixo mostra o código que foi desenvolvido para que a lista de produtos pudesse ficar no padrão JSON.

```

for (Produto produto:listaProduto){
    try {
        jsonOne = new JSONObject();
        jsonOne.put("codigo", produto.getCodigo());
        jsonOne.put("codigobarra", produto.getCodigobarra());
        jsonOne.put("descricao", produto.getDescricao());
        jsonOne.put("valor", produto.getValor());
        array.put(jsonOne);
    } catch (JSONException ex) {
        t = Toast.makeText(v.getContext(), "Erro ao enviar os dados.\nEntre em contato
com o administrador do sistema.", Toast.LENGTH_LONG);
        t.setGravity(0, 1, 80);
        ex.printStackTrace();
    }
}

```

Figura 34 Código para transformar a lista de produtos no padrão JSON (Fonte: próprio autor).

Esse código tem como objetivo criar uma String no padrão JSON contendo o identificador do produto, o código de barras, a descrição e o preço.

A estrutura inicial da lista de produtos pode ser representada da seguinte maneira:

- [Produto1, Produto2]

Sendo que, cada item representa um objeto do tipo Produto. Com a utilização do JSON, a estrutura da lista ficará da seguinte maneira:

- “[{“codigo”:1,“codigobarra”:123,“descricao”:”produto exemplo1”,“valor”:1.45},{“codigo”:2,“codigobarra”:145,“descricao”:”produto exemplo2”,“valor”:3.68}]”

Esse é um exemplo de String que poderia ser enviada através do Web Service para o sistema Preço Público *web*.

Após o processamento dessas informações, os dados (login e senha MD5 do usuário e a lista dos produtos em JSON) são enviados para o Preço Público *web* utilizando o *framework* XFire<sup>9</sup> para criação do Web Service.

O *framework* XFire é responsável por toda a definição da linguagem e do protocolo de comunicação entre os sistemas, sendo que o programador só fica responsável por configurar o XFire para encontrar o serviço na máquina servidora.

Para realizar a integração dos sistemas é necessário que no sistema Preço Público *web* seja implementado um serviço responsável por receber os dados que foram enviados.

A Figura 35 mostra o código do serviço que foi desenvolvido. Esse serviço recebe a *String* da lista de produtos no formato JSON, o *login* e a senha no formato MD5 para o acesso do usuário. E realiza os seguintes passos:

- 1) Verifica se o usuário é válido, isso é feito chamando o serviço **logarUsuário** em destaque na Figura 34 e passando como parâmetro o *login* e a senha MD5. Caso o usuário seja válido, o serviço **logarUsuario** retorna seu identificador, caso contrário, retorna *null*.
- 2) Se o usuário for válido, a lista que foi passada por parâmetro e está no formato JSON é transformada novamente no padrão *List* do Java e é nomeada de **listaProdutosMobile**. Ao final, é chamado o serviço **salvarDadosMobile** (em destaque na Figura 34) recebendo como parâmetro a lista de produtos já no padrão *List* e o identificador do usuário que deseja cadastrar esses produtos. Caso consiga cadastrar, retorna o valor **true**, caso o usuário não exista ou ocorra algum erro na gravação dos produtos, o retorno é **false**.

---

<sup>9</sup><http://xfire.codehaus.org/>

```

public String cadastrarProdutos(String produtos, String loginAcesso, String senhaMd5Acesso) {
    ApplicationContext applicationContext = AppContext.getApplicationContext();
    MobileService mobileService = (MobileService)applicationContext
        .getBean("mobileService", MobileService.class);
    Long idUsuario = mobileService.logarUsuario(loginAcesso, senhaMd5Acesso);
    Boolean cadastrou;
    String cadastrouString;
    if (idUsuario != null) {
        ProdutoMobDTO produtomobDTO = null;
        List<ProdutoMobDTO> listaProdutosMobile =
            new ArrayList<ProdutoMobDTO>();
        Object obj = JSONValue.parse(produtos);
        JSONArray array1 = (JSONArray)obj;
        JSONObject obj1 = null;
        for(int i=0; i<array1.size(); i++){
            produtomobDTO = new ProdutoMobDTO();
            obj1 = (JSONObject)array1.get(i);
            produtomobDTO
                .setCodigoBarra(Long.parseLong(obj1.get("codigoBarra").toString()));
            produtomobDTO.setDescricao(obj1.get("descricao").toString());
            produtomobDTO.setPreco(BigDecimal.valueOf(Double
                .parseDouble(obj1.get("valor").toString())));
            listaProdutosMobile.add(produtomobDTO);
        }
        cadastrou = mobileService.salvarDadosMobile(listaProdutosMobile, idUsuario);
        cadastrouString = cadastrou.toString();
        return cadastrouString;
    } else {
        return Boolean.FALSE.toString();
    }
}

```

Figura 35 Serviço para realizar a integração entre o sistema *mobile* e *web* (Fonte: próprio autor).

Os dois serviços destacados na Figura 35 terão suas implementações mostradas a seguir. A Figura 36 mostra o código do serviço **logarUsuario**.

```

public Long logarUsuario (String email, String senhaMD5Mob){
    Diversos diversos = new Diversos();
    try{
        List<Usuario> usuarios = (List<Usuario>)
        getManager().createNamedQuery("Usuario.findByEmail").setParameter("email
        ", email).getResultList();
        String senhaMD5;
        Long idEncontrado = null;
        for (Usuario usuario:usuarios){
            senhaMD5 = diversos.transformaSenhaMd5(usuario.getSenha());
            if (senhaMD5.equals(senhaMD5Mob)){
                idEncontrado = usuario.getId();
            }
        }
        return idEncontrado;
    } catch (NoResultException ex){
        return null;
    }
}

```

Figura 36 Serviço para verificar se o usuário é válido (Fonte: próprio autor).

O serviço mostrado na Figura 36 tem o objetivo de verificar se o e-mail e a senha criptografada que são passados como parâmetro, pertencem a algum usuário cadastrado na base de dados do Preço Público *web*.

O código da Figura 37 mostra o serviço **salvarDadosMobile**. Esse serviço recebe como entrada a lista de produtos e o identificador do usuário, e cadastra esses dados na tabela **produtomob**.

Por fim, o Spring utiliza *annotation* (@Transactional) para garantir o controle transacional de todas as operações que utilizam banco de dados. Isso significa que, se houver uma falha no meio de uma operação no banco de dados, o processamento retorna ao início, sem a perda de dados.

```

public boolean salvarDadosMobile(List<ProdutoMobDTO> dados, Long idUsuario){
    ProdutoMob produtoMobEntidade = null;
    Query q = getManager().createNamedQuery("Usuario.findById")
    .setParameter("id", idUsuario);
    Usuario usuario = (Usuario) q.getSingleResult();
    for (ProdutoMobDTO produto : dados){
        produtoMobEntidade = new ProdutoMob();
        produtoMobEntidade.setCodigoBarra(produto.getCodigoBarra());
        produtoMobEntidade.setDescricao(produto.getDescricao());
        produtoMobEntidade.setPreco(produto.getPreco());
        produtoMobEntidade.setModerado(Boolean.FALSE);
        produtoMobEntidade.setUsuario(usuario);
        getManager().persist(produtoMobEntidade);
    }
    return true;
}

```

Figura 37 Serviço responsável por cadastrar uma lista de produtos no banco de dados MySQL do Preço Público *web* (Fonte: próprio autor).

A Figura 38 mostra um exemplo de uso dessa *annotation*.

```

@Transactional
public boolean salvarDadosMobile(List<ProdutoMobDTO> dados, Long idUsuario);

```

Figura 38 Exemplo do uso da *annotation* @Transaction (Fonte: próprio autor).

A integração entre os sistemas *mobile* e *web* foi descrita nessa seção. Na próxima, será mostrado o protótipo desenvolvido para o Preço Público *web*.

### 3.3.3 Preço Público *web*

O protótipo para o Preço Público *web* começou a ser desenvolvido com a criação do *link Mobile* na página inicial, igual ao que está em destaque na Figura 23. A Figura 39 mostra o código para criar esse *link*.

```

<a href="mobile.jsp">Mobile</a>

```

Figura 39 Código para criar o *link* Mobile presente na página inicial do Preço Público *web* (Fonte: próprio autor).

Logo em seguida, foi desenvolvido o modelo de interface igual ao mostrado na Figura 24.

A principal tarefa dessa página é mostrar os registros da tabela **produtomob** (Figura 21), para que isso fosse possível, foi implementado o serviço mostrado na figura a seguir:

```
public List<ProdutoMobDTO> getProdutos(Long idUsuario){
    List<ProdutoMob> produtos = (List)getManager()
        .createNamedQuery("ProdutoMob.findByModerado")
        .setParameter("moderado", Boolean.FALSE).setParameter("id_usuario",idUsuario)
        .getResultList();
    List<ProdutoMobDTO> produtosMobDTO = new ArrayList();
    for (ProdutoMob produtoMobEntidade : produtos) {
        ProdutoMobDTO produtoMobDTO = new ProdutoMobDTO();
        produtoMobDTO.setId(produtoMobEntidade.getId());
        produtoMobDTO.setCodigoBarras(produtoMobEntidade.getCodigoBarras());
        produtoMobDTO.setDescricao(produtoMobEntidade.getDescricao());
        produtoMobDTO.setPreco(produtoMobEntidade.getPreco());
        produtosMobDTO.add(produtoMobDTO);
    }
    return produtosMobDTO;
}
```

Figura 40 Serviço responsável por listar os registros da tabela **produtomob** (Fonte: próprio autor).

Esse serviço recebe o identificador do usuário, e tem a responsabilidade de buscar os dados que foram cadastrados por esse usuário na tabela **produtomob** (Figura 21), e retornar a lista desses dados.

Ao receber a lista, a página deve chamar um serviço para verificar se cada produto pertencente à lista, já possui o código de barras cadastrado no banco de dados do Preço Público *web*. A figura a seguir mostra o código do serviço que realiza essa função:

```

public Produto getNomeProdutos(Long codigodebarra){
    Query q = getManager().createNamedQuery("Produto.findCodigoBarra")
        .setParameter("codigobarra",codigodebarra);
    try{
        Produto produto = (Produto) q.getSingleResult();
        return produto;
    } catch(NoResultException ex){
        return null;
    }
}

```

Figura 41 Serviço responsável por verificar se o código de barras de um produto já existe no banco de dados (Fonte: próprio autor).

Esse serviço realiza a busca pelo produto na tabela **produto** (Figura 21). Essa pesquisa é realizada através do código de barras passado por parâmetro, se for encontrado algum resultado, retorna o objeto do produto, caso contrário, retorna *null*.

O *link* **Editar** foi desenvolvido utilizando a mesma *tag* da Figura 38, ao acessá-lo o usuário é redirecionado para a página que possui o modelo de interface igual ao da Figura 25. A implementação desse modelo de interface não foi necessária, pois o sistema Preço Público *web* já possuía essa página desenvolvida.

Para melhorar a usabilidade do *link* **Editar**, ao acessá-lo, os campos nome (caso o produto tenha sido encontrado através do serviço da Figura 40) e o preço do produto já são preenchidos na página que tem responsabilidade de finalizar o processo de publicação de preços no Preço Público *web* (Figura 25).

O *link* **Clique aqui se não for esse produto** só irá aparecer caso o produto tenha sido encontrado pelo serviço da Figura 41, esse *link* só deve ser acessado se vinculação do código de barras com o produto esteja errada.

Outro *link* que está presente no modelo de interface da Figura 24 é o **Excluir** que ao ser acessado chama o seguinte serviço:

```
public void deleteProduto(Long idProdutoMob) {  
    Query q = getManager().createNamedQuery("ProdutoMob.findById")  
        .setParameter("id", idProdutoMob);  
    ProdutoMob produtoMob = (ProdutoMob) q.getSingleResult();  
    getManager().remove(produtoMob);  
}
```

Figura 42 Serviço responsável por excluir um registro da tabela **produtomob** (Fonte: próprio autor).

Esse serviço recebe o identificador do produto e apaga seu registro na tabela **produtomob** (Figura 21).

Por fim, o *link* **Clique aqui para baixar o aplicativo** (Figura 24) permite o usuário realizar o *download* do aplicativo Preço Público *mobile* para ser instalado em seu celular.

## 4 CONCLUSÃO

A abordagem estabelecida inicialmente pelo trabalho de criar um modelo arquitetônico e, logo em seguida, desenvolver um protótipo para a integração da plataforma Android e o *software* social Preço Público foi realizada ao longo do desenvolvimento do trabalho.

Essa abordagem mostrou-se eficaz, pois o trabalho obteve:

- Um modelo arquitetônico que descreve a integração entre o Preço Público *mobile* e *web*;
- Um protótipo de software que contém todos os requisitos iniciais funcionando e que será implantado para integrar o Preço Público *mobile* e *web*.

Sendo assim, o objetivo geral e os objetivos específicos do trabalho foram atingidos.

Ao longo do desenvolvimento da parte prática do trabalho os seguintes problemas foram enfrentados:

- Problema de integração: serialização do objeto lista de produtos não funcionou, foi necessário usar o padrão JSON para enviar os dados através do Web Service.
- Interface gráfica do aplicativo para dispositivos móveis: inicialmente a interface gráfica foi criada no emulador Android, ao implantar o aplicativo em um celular, a interface não ficou da mesma maneira. Para resolver o problema, foi necessário modificar o arquivo XML de cada interface que tinha sido criada.

- Problemas ao implantar o aplicativo para dispositivos móveis: quando o aplicativo foi implantado em um celular apareceram diversos problemas que não ocorriam quando o programa estava sendo emulado no computador. Esses problemas já foram solucionados utilizando a ferramenta de *debugger* que os celulares Android possuem.

Apesar de estar pronto para ser implantado, o protótipo tem algumas melhorias que podem ser realizadas, entre elas estão: melhoria da interface do aplicativo Preço Público *mobile* e realização de testes por outras pessoas além do desenvolvedor.

Com isso é possível definir os seguintes trabalhos futuros dessa monografia:

- Melhorar a usabilidade dos sistemas Preço Público *mobile* e *web*;
- Utilizar técnicas de qualidade de software no protótipo;
- Utilizar GPS no protótipo do Preço Público *mobile* para encontrar estabelecimentos;
- Buscar os estabelecimentos que usuário mais frequenta para o Preço Público *mobile*.
- Implantar o protótipo.



## 5 REFERÊNCIAS BIBLIOGRÁFICAS

ADAMIC, L. A.; ZHANG, J.; BAKSHY, E.; ACKERMAN, M. S. Knowledge sharing and Yahoo answers: Everyone knows something. **Proceeding of the XVII international conference on World Wide Web**. 2008.

AGARWAL, V.; CHAFLE, G; MITTAL, S.; SRIVASTAVA, B. Understanding Approches For Web Services Composition and Execution. **Proceedings of the 1<sup>st</sup> Bangalore Annual Compute Conference**. 2007.

ANDREÃO, C. F. **Conceitos básicos das plataformas Java e J2ME**. Disponível em: <http://www.devmedia.com.br/post-6484-CONCEITOS-BASICOS-DAS-PLATAFORMAS-JAVA-E-J2ME.html>. Consultado em: 30/03/2011.

ARGOTE, L.; INGRAM, P. Knowledge transfer: A basic for competitive advantage in firms. **Organization Behavior and Human Decision Processes**. 82(1):150-169, 2000.

ARTHUR, J.; AZADEGAN, S. Spring Framework for rapid open source J2EE Web Application Development: A case study. **Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05)**. 2005.

AXT, M. Comunidades virtuais de aprendizagem e interação dialógica: do corpo, do rosto e do olhar. **Filosofia Unisinos**. 7(3):256-268, 2006.

BARTOL, K. M.; SRIVASTAVA, A. Encouraging knowledge sharing: The role of organization reward systems. **Journal of Leadership and Organization Studies**. 9(1):64-77, 2002.

BEAUMONT, R. **History of the Internet**. Disponível em <http://www.fhi.rcsed.ac.uk/rbeaumont/virtualclassroom/chap6/internet/hist/hist1.htm>. Consultado em 15 de fevereiro de 2011.

BENATALLAH, B.; CASATI, F.; TOURNANI, F.; HAMADI, R.; Conceptual Modeling of Web Service Conversations. **Lecture Notes in Computer Science**. 2681(2003):449-467, 2003.

CHIU, C. M; HSU, M. H; WANG, E. T. G. Understanding knowledge sharing in virtual communities: An integration of social capital and social cognitive theories. **Decision Support Systems**. 42(3):1872-1888, 2006.

COSTA, G.; MARTINELLI, F.; MORI, P.; SCHAEFER, C.; WALTER, T. Runtime monitoring for next generation Java ME platform. **Computers and Security**. 29(1):74-87, 2010

DEITEL, H. M.; DEITEL, P. J. **Java Como Programar**. 6ª Edição. Pearson Prentice Hall, 2005.

GONÇALVES, E. **Desenvolvendo aplicações web com JSP, Servlets, JavaServer Faces, Hibernate, EJB3 Persistence e Ajax**. 1ª Edição. Editora Ciência Moderna, 2007

KING, G.; BAUER, C.; ANDERSEN, M. R.; BERNARD, E.; EBERSOLE, S.; FERENTSCHIK, H. Hibernate Reference Documentation.

KOH, J.; KIM, Y. G. Knowledge sharing in virtual communities: an e-business perspective. **Expert Systems with Applications**. 26(2):155-166, 2004.

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a Internet**. 3a Edição, Person Addison Wesley, 1981.

LAINER, B. M.; CERF, V. G.; CLARK, D. D.; KAHN, R. E.; KLEINROCK, L.; LYNCH, D. C.; POSTEL, J.; ROBERTS, L. G.; WOLFF, S. A brief history of the Internet. **ACM SIGCOMM Computer Communication Review**. 39(5), 2009 .

LIANG, D.; **Introduction to Java Programming**. 7ª Edição, Pearson Prentice Hall, 2009.

MACARIO, G.; TORCHIANO, M.; VIOLANTE, M. An In-Vehicle Infotainment Software Architecture Based on Google Android. **Industrial Embedded Systems, 2009. SIES '09. IEEE International Symposium on**. 257-260, 2009.

MAGGIO, M. D.; GLOOR, P. A.; PASSIANTE, G. Collaborative innovation networks, virtual communities and geographical clustering. **International Journal of Innovation and Regional Development**. 1(4):387-404, 2009.

MELA, M.; SAKOWICZ, B.; CHLAPINSKI, J. Advertising Service Based On Spring Framework. **Proceeding of International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science**. 2008.

MOWERY, D. C.; SIMCOE, T. Is the Internet a US invention?-an economic and technological history of computer networking. **Research Policy**. 31(8-9):1369-1387, 2002.

PATRICIO, M. R.; GONÇALVES, V. Utilização educativa do facebook no ensino superior. **Proceedings of the I Conference learning and teaching in higher education**. 2010.

PEREIRA, L. C. O.; SILVA, M. L. **Android para desenvolvedores**. 1ª Edição, Brasport, 2009.

RIDINGS, C. M.; GEFEN, D. Virtual community attraction: Why people hang out online. **Journal of Computer-Mediated Communication**. 10(1), 2004.

VAREJÃO, F. **Linguagens de Programação: Conceitos e Técnicas**. 2ª Edição, Elsevier, 2004.

VIEIRA, E. **Os bastidores da Internet no Brasil**. 1ª Edição, Manole, 2003.

ZAMBALDE, A. L.; PÁDUA, C. I. P. S.; ALVES, R. M. **O documento científico em Ciência da computação e Sistemas de Informação**. Lavras/MG: DCC/UFLA, 2008. Notas de aula – rascunho – texto em construção sem revisão de português e citações.