

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO**

**SISTEMA WEB-MOBILE PARA LOCAÇÃO DE IMÓVEIS**

**HERMANO ROEPKE**

**BLUMENAU**  
**2014**

**2014/1-07**

**HERMANO ROEPKE**

## **SISTEMA WEB-MOBILE PARA LOCAÇÃO DE IMÓVEIS**

Trabalho de Conclusão de Curso submetido à Universidade Regional de Blumenau para a obtenção dos créditos na disciplina Trabalho de Conclusão de Curso II do curso de Sistemas de Informação — Bacharelado.

Prof. Alexander Roberto Valdameri, Mestre – Orientador

**BLUMENAU  
2014**

**2014/1-07**

# **SISTEMA WEB-MOBILE PARA LOCAÇÃO DE IMÓVEIS**

Por

**HERMANO ROEPKE**

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: \_\_\_\_\_  
Prof. Alexander Roberto Valdameri, Mestre – Orientador, FURB

Membro: \_\_\_\_\_  
Prof. Everaldo Artur Grahl, Mestre – FURB

Membro: \_\_\_\_\_  
Prof. Wilson Pedro Carli, Mestre – FURB

Blumenau, 07 de julho de 2014.

Dedico este trabalho a minha família, aos amigos, e especialmente aqueles que me ajudaram diretamente na realização deste.

## **AGRADECIMENTOS**

A Deus, pelo seu imenso amor e graça.

À minha família, que sempre esteve presente.

Aos meus amigos, pelos empurrões e cobranças.

Ao meu orientador, professor Alexander Roberto Valdameri, por ter acreditado na conclusão deste trabalho.

Aos professores do Departamento de Sistemas e Computação da Universidade Regional de Blumenau por suas contribuições durante os semestres letivos.

Se fosse fácil, todo mundo saberia explicar como fazer. Viver não é assim tão simples, mas que a gente complica não há dúvidas.

Caio Fernando Abreu

## RESUMO

Este trabalho apresenta o desenvolvimento de um sistema *web-mobile* para locação de imóveis. Devido à indisponibilidade de tempo das pessoas para procurar um imóvel, ou por elas não conhecerem a localidade do mesmo, a ausência de dados visuais e geográficos tornou-se necessário o desenvolvimento de um sistema que o auxilie. O sistema permite ao usuário visualizar todos os imóveis cadastrados disponíveis para locação e verificar suas características e imagens a partir da localização atual do usuário. Para o desenvolvimento foi utilizada a técnica *Create, Read, Update, Delete* (CRUD) do *Netbeans Integrated Development Environment* (IDE). O trabalho foi baseado na utilização API do Google Maps para mostrar a localização dos imóveis disponíveis para locação.

Palavras-chave: *Web-mobile*. Imóveis. API Google Maps. CRUD.

## **ABSTRACT**

This paper presents the development of a web-mobile system for rental properties. Due to lack of time for people to Find A Property, or they do not get to know the locality of it, the absence of visual and geographic data became necessary the development of a system to assist you. The system allows the user to view all registered properties available for lease and check its features and images as from the user's current location. For developing the technique Create, Read, Update, Delete (CRUD) Netbeans Integrated Development Environment (IDE) was used. The work was based on the utilization of the Google Maps API to show the location of properties available for lease.

**Keywords:** Web-mobile. Properties. API Google Maps. CRUD.



## LISTA DE FIGURAS

Figura 1 – Visualização do imóvel.....	18
Figura 2 – Visualização do Boteco Maps.....	19
Figura 3 – Tela do “boteco” para o usuário final .....	20
Figura 4 – Portal imobiliário .....	21
Figura 5 – Busca detalhada.....	22
Figura 6 – Exibição de vídeo (YouTube) .....	23
Figura 7 – Consulta de imóveis via <i>web</i> .....	24
Figura 8 – Tela de detalhes do imóvel.....	25
Figura 9 – Modelo de casos de uso.....	28
Figura 10 – Modelo entidade relacionamento .....	29
Figura 11 – Utilização da biblioteca do Primefaces .....	31
Figura 12 – Carregamento da API do Google Maps .....	32
Figura 13 – Iniciando um mapa.....	32
Figura 14 – Função que pega localização atual.....	33
Figura 15 – <i>Map clusterer</i> .....	34
Figura 16 – Classe <i>markercluster.js</i> .....	34
Figura 17 – Exibir configurações do servidor Glassfish 4.0 .....	35
Figura 18 – Configuração no servidor Glassfish 4.0 .....	36
Figura 19 – Configurações no <i>web.xml</i> .....	37
Figura 20 – Código tela de <i>login</i> .....	38
Figura 21 – Tela inicial pedindo se pode usar localização .....	39
Figura 22 – Tela inicial.....	40
Figura 23 – Tela inicial sem localização .....	41
Figura 24 – Tela com as características do imóvel.....	41
Figura 25 – Tela com as características do imóvel 2.....	42
Figura 26 – Tela página de <i>login</i> .....	42
Figura 27 – Tela página inicial .....	43
Figura 28 – Tela erro no <i>login</i> .....	43
Figura 29 – Tela lista de todos imóveis.....	43
Figura 30 – Tela criar novo imóvel .....	44
Figura 31 – Tela visualização de imóvel .....	45

Figura 32 – Tela criar nova imagem.....	46
Figura 33 – Tela lista de todas as imagens .....	46

## LISTA DE QUADROS

Quadro 1 – Requisitos funcionais.....	27
Quadro 2 – Requisitos não funcionais.....	27
Quadro 3 – Quadro comparativo do sistema com os trabalhos correlatos .....	47
Quadro 4 – Descrição do caso de uso manter bairros.....	53
Quadro 5 – Descrição do caso de uso manter cidades.....	53
Quadro 6 – Descrição do caso de uso manter localidades.....	53
Quadro 7 – Descrição do caso de uso manter logradouros.....	54
Quadro 8 – Descrição do caso de uso manter contatos .....	54
Quadro 9 – Descrição do caso de uso manter imóveis .....	54
Quadro 10 – Descrição do caso de uso manter imagens dos imóveis .....	55
Quadro 11 – Descrição do caso de uso alterar situação do imóvel .....	55
Quadro 12 – Descrição do caso de uso visualizar localização atual .....	56
Quadro 13 – Descrição do caso de uso visualizar imóveis disponíveis .....	56
Quadro 14 – Descrição do caso de uso visualizar informações do imóvel .....	56
Quadro 15 – Dicionário da tabela “bairro” .....	57
Quadro 16 – Dicionário da tabela “cidade” .....	57
Quadro 17 – Dicionário da tabela “localidade” .....	57
Quadro 18 – Dicionário da tabela “logradouro” .....	58
Quadro 19 – Dicionário da tabela “contato” .....	58
Quadro 20 – Dicionário da tabela “imovel” .....	58
Quadro 21 – Dicionário da tabela “imagem” .....	59

## **LISTA DE SIGLAS**

*AJAX – Asynchronous Javascript and XML*

*API – Application Programming Interface*

*CRUD – Create, Read, Update, Delete.*

*EJB – Enterprise JavaBeans*

*IDE - Integrated Development Environment*

*JPA – Java Persistence API*

*JSF – Java Server Faces*

*MER – Modelo Entidade Relacionamento*

*PHP – Hypertext Preprocessor*

*SQL – Structured Query Language*

*XML - eXtensible Markup Language*

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>12</b>
1.1 OBJETIVOS DO TRABALHO .....	13
1.2 ESTRUTURA DO TRABALHO .....	13
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>14</b>
2.1 ESCOLHENDO UM IMÓVEL .....	14
2.2 ALUGUEL .....	15
2.3 ASPECTOS ANALISADOS NA ESCOLHA DO IMÓVEL PARA RESIDÊNCIA .....	15
2.4 GOOGLE MAPS.....	16
2.5 WEB-MOBILE.....	17
2.6 TRABALHOS CORRELATOS .....	17
<b>3 DESENVOLVIMENTO.....</b>	<b>26</b>
3.1 LEVANTAMENTO DE INFORMAÇÕES .....	26
3.2 ESPECIFICAÇÃO .....	26
3.2.1 Requisitos do Sistema .....	27
3.2.2 Diagrama de Casos de Uso .....	28
3.2.3 Modelo Entidade Relacionamento .....	29
3.3 IMPLEMENTAÇÃO .....	30
3.3.1 Técnicas e ferramentas utilizadas .....	30
3.3.1.1 Netbeans IDE 7.4 e a técnica CRUD.....	30
3.3.1.2 Primefaces.....	31
3.3.1.3 API Javascript do Google Maps v3 .....	32
3.3.1.4 Map Clusterer .....	33
3.3.1.5 Protegendo aplicação WEB no Netbeans IDE usando servidor Glassfish 4.0 .....	35
3.3.2 Operacionalidade da implementação .....	38
3.3.2.1 Módulo de Consulta.....	39
3.3.2.2 Módulo de administração .....	42
3.4 RESULTADOS E DISCUSSÃO .....	46
<b>4 CONCLUSÕES.....</b>	<b>49</b>
4.1 EXTENSÕES .....	49
<b>REFERÊNCIAS .....</b>	<b>51</b>
<b>APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO.....</b>	<b>53</b>

<b>APÊNDICE B – DESCRIÇÃO DO DICIONÁRIO DE DADOS.....</b>	<b>57</b>
---	-----------

## 1 INTRODUÇÃO

Na última década o governo brasileiro tem oportunizado programas de habitação para que os cidadãos possam realizar o, até então distante, sonho da casa própria. Mesmo assim, a operação de locação ainda tem uma considerável fatia do mercado imobiliário. Milhares de novos imóveis são construídos com a finalidade de locação. Isto ocorre devido a vários fatores, sendo o principal o alto custo da construção civil. Dentre esses imóveis para locação residencial tem-se apartamentos, casas e quitinetes (CAIXA ECONÔMICA FEDERAL, 2013).

Consciente ou não das suas necessidades, em geral, o locatário encontra dificuldades para selecionar um bom local para viver, tendo em vista, a falta de tempo para procurar um imóvel. Acredita-se que, por vezes, o usuário encontre dificuldades em verificar/comparar imóveis para locação considerando, por exemplo, informações sobre a localização e proximidades das suas atividades do cotidiano como o trabalho, a família, o lazer, os supermercados, as farmácias e outros serviços.

Quando você compra uma residência, está levando junto sua localização. Preste muita atenção na distância que, diariamente, você e seu cônjuge terão de percorrer até o local de trabalho. Em grandes cidades, essa logística transformou-se em um critério chave na escolha da residência. (HALFELD, 2002, p. 105).

Muitas pessoas que vivem de aluguel não tem tempo suficiente para procurar um bom imóvel ou simplesmente não conhece a localidade para qual pretendem se mudar. A mudança muitas vezes ocorre devido a transferência de emprego ou venda do local onde residem. Conseqüentemente, muitas vezes faz-se uma má escolha da localidade levando em conta as proximidades com trabalho ou lazer, segurança, preço, e/ou tamanho do imóvel por ser muito pequeno ou grande dependendo de sua necessidade.

Ainda, pode-se afirmar que ausência de dados visuais (características do imóvel), bem como dados geográficos (localização e proximidades), podem tornar-se fatores determinantes para a escolha do imóvel desejado. Embora existam diversos sítios de imobiliárias na internet onde é possível se obter informações sobre imóveis, muitos não apresentam dados acerca da localização e proximidades sobre o imóvel, que para o locatário são importantes para uma comparação com outro imóvel.

Diante do exposto, este trabalho apresenta uma solução utilizando computação móvel aos usuários do serviço de locação imobiliária. Através deste sistema é possível consultar os

imóveis disponíveis, podendo dessa forma, selecionar previamente aqueles que mais lhe interessar. Com o sistema *web-mobile* de locação de imóveis o interessado pode analisar e filtrar previamente os imóveis que lhe interessam para possíveis visitas, onde a possibilidade de satisfação do usuário seria superior.

## 1.1 OBJETIVOS DO TRABALHO

O objetivo deste trabalho é apresentar um sistema *web-mobile* que auxilie o usuário a encontrar um imóvel para alugar a partir de sua localização contendo dados e imagens.

Os objetivos específicos do trabalho são:

- a) integrar informações ao Google Maps;
- b) apresentar graficamente a localização dos imóveis.

## 1.2 ESTRUTURA DO TRABALHO

No primeiro capítulo tem-se a introdução ao tema principal deste trabalho com a apresentação da justificativa e dos objetivos.

No segundo capítulo apresenta-se a fundamentação teórica pesquisada sobre tipos de imóveis para locação, aluguel, aspectos analisados na opção de escolha do imóvel para residência, Google Maps, *web-mobile* além de trabalhos correlatos.

O terceiro capítulo apresenta o desenvolvimento do sistema *web-mobile* Alugue Imóveis SC iniciando-se com o levantamento de informações, tendo na sequência as técnicas e ferramentas utilizadas bem como a elaboração de alguns diagramas para auxiliar na compreensão do sistema, a operacionalidade do mesmo e resultados e discussões.

No quarto capítulo tem-se as conclusões deste trabalho bem como são apresentadas sugestões para trabalhos futuros.



## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda assuntos a serem apresentados nas seções a seguir, tais como escolhendo um imóvel, aluguel, aspectos analisados na escolha do imóvel para residência, Google Maps, *web-mobile*, além de trabalhos correlatos.

### 2.1 ESCOLHENDO UM IMÓVEL

O interessado normalmente já tem noção do que realmente quer alugar, porém a dificuldade na questão financeira as vezes os fazem tomar medidas diferentes, perante seus desejos e necessidades.

Na hora de procurar um imóvel, o comprador deve avaliar quais são suas necessidades e também tentar descobrir como são os vizinhos, se a rua é tranquila, se tem uma infraestrutura de serviços próxima do imóvel. “Quando for conhecer um apartamento, é preciso sondar com o zelador, de forma sutil, quem serão os vizinhos que moram em cima e embaixo. Se tem crianças e animais. O mesmo deve ser feito com a casa, é importante saber como é a vizinhança, se a rua é tranquila, se tem algum estabelecimento barulhento”, aconselha a gerente de Locação da Lello Imóveis, Márcia Vavassori. (UOL MULHER, 2011).

Segundo Primar Administradora de Imóveis (2013), apartamentos são preferíveis para casais ou pessoas solteiras de um poder aquisitivo médio ou alto, levando em conta a correria do dia-a-dia, essas pessoas preferem não ter muitos serviços domésticos, como cuidar de jardim. Casas são mais procuradas por casais, normalmente com filhos, com intenção que eles tenham uma área fora de casa para brincar, como um quintal, por exemplo, mantendo algum animal de estimação, pois dependendo do tamanho da casa o locatário terá vários serviços para serem feitos como aparar grama, cuidar do jardim, entre outros. As quitinetes são como apartamentos, porém com sala e cozinha conjugados. Geralmente são procuradas por pessoas de baixa renda financeira, e para aqueles que não querem ter muito serviço doméstico, haja vista que são bem menores que um apartamento.

## 2.2 ALUGUEL

O perfil das pessoas é de extrema importância quando se decide comprar um imóvel ou alugar. Segundo Halfeld (2002 p. 105), “comprar a casa própria requer algumas ponderações. Sugiro que solteiros, descasados ou profissionais sujeitos a mudanças de cidades tenham muita calma antes de adquirir o apartamento de seus sonhos.”

O aluguel é a opção de jovens e/ou pessoas que tenham uma rotatividade grande de emprego. Outro motivo pelo qual se decide pela locação é a falta de capital financeiro para adquirir imóvel próprio ou construir. Por vezes isto ocorre por falta de comprovação de renda ou até mesmo endividamento, o que impossibilita conseguirem financiamento junto ao governo e bancos.

Da mesma forma como um determinado imóvel possui um certo **valor de venda** no livre mercado imobiliário, esse mesmo imóvel possui um certo **valor locativo**, ou **aluguel**, que pode auferir no livre mercado. Ou seja, há um certo valor que, além de satisfazer o proprietário, é o máximo que os pretensos inquilinos se dispõem a pagar para ocupar o imóvel através de locação, ambos – inquilino e proprietário. (CANDELORO, 1991 p. 15, grifo do autor).

Sendo assim, o usuário do sistema deve buscar imóveis com preços que não lhe comprometa muito seu orçamento, comparando lugares próximos para verificar se não há abuso do proprietário ou da imobiliária perante o valor negociado para o aluguel. Em fim, é sugerida uma análise cuidadosa em alguns aspectos antes da escolha de um imóvel.

## 2.3 ASPECTOS ANALISADOS NA ESCOLHA DO IMÓVEL PARA RESIDÊNCIA

Várias são as características do imóvel que o usuário analisa ou deveria analisar, para fazer uma locação satisfatória, atendendo assim, todas as suas necessidades ou desejos. Essas características variam conforme o tipo do imóvel, pois o usuário que pretende locar uma casa tem um perfil diferente do que decide locar um apartamento ou quitinete por exemplo.

Encontrar um imóvel para alugar, baseado apenas na vizinhança onde ele se encontra e na aparência e conservação pode ser fácil, mas saber qual se encaixa no seu perfil, seja ele casa ou apartamento, é o mais difícil. “O imóvel perfeito é aquele que se adequa às necessidades do morador”, explica o advogado imobiliário e diretor de Condomínios da Primar Administradora

de Bens, Carlos Samuel de Oliveira Freitas. (UOL MULHER, 2011).

Em vista das necessidades e desejos do morador, destaca-se, algumas características importantes à se analisar na escolha de um lugar agradável para morar, evitando assim arrependimentos futuros. Em apartamentos as análises são o tamanho, o número de quartos, se está mobiliado ou não, se possui churrasqueira, elevador, a quantidade de elevadores, o valor do condomínio, as vagas na garagem, a área de festa, a fachada, o preço, as proximidades com trabalho/ faculdade/ família/ lazer, a tranquilidade e segurança. Em casas as análises são o tamanho, a quantidade de quartos, o quintal, as vagas de garagem, os riscos de desastres, como enchentes e desabamentos, as proximidades com trabalho/ escola/ lazer, preço, a tranquilidade e segurança. Em quitinetes é analisado o tamanho, a quantidade de quartos, as vagas de garagem coberta, o preço, as proximidades com trabalho/ lazer, a tranquilidade e segurança.

#### 2.4 GOOGLE MAPS

Segundo Google (2013), Google Maps é um serviço de pesquisa e visualização de mapas e imagens de satélite da Terra gratuito da *web* fornecido pela empresa Google. Ele redefiniu o mapeamento *on-line* oferecendo uma poderosa tecnologia de informações de mapas amigável e informações sobre empresas locais, incluindo o endereço das empresas, informações de contato e direções de tráfego.

O Google Maps disponibiliza também uma *Application Programing Interface* (API) chamada Google Maps JavaScript API v3. Desta forma os programadores não se atêm a detalhes de implementação, sendo utilizadas apenas funções desta API, compartilhando seu geoprocessamento. A API oferece diversos utilitários para manipulação de mapas e para a adição de conteúdo ao mapa por meio de múltiplos serviços, o que permite criar robustos aplicativos de mapas em seu *website*.

Os serviços do Google Maps estão disponíveis para qualquer *site* que seja gratuito para seus consumidores. Para usar de suas ferramentas é necessário se inscrever para uma chave que pode ser utilizada para um único “diretório” do seu servidor. A documentação necessária para utilizar-se dos serviços está disponível no Google Maps JavaScript API v3 (GOOGLE DEVELOPERS, 2013).

## 2.5 WEB-MOBILE

Segundo Oliveira (2010), “*web mobile* é o termo que define o simples conceito de acesso à internet, através de um dispositivo móvel, permitindo alcançar qualquer página da *web*, em qualquer lugar do mundo, a qualquer momento”. Para este mesmo autor, o conceito *web-mobile* ganhou força nos últimos anos, devido ao explosivo crescimento e aceitação das tecnologias móveis, tais como celulares, *smartphones*, *tablets*, *netbooks* e internet móvel. Não apenas em países ricos, mas também em países em desenvolvimento como o Brasil, configurando tal aceitação uma tendência mundial.

A aplicação *web-mobile* é uma aplicação *web* que é otimizada para uma ou mais plataformas de dispositivos móveis, podendo ser executada em todo aparelho que possua um navegador compatível com os padrões *web*.

## 2.6 TRABALHOS CORRELATOS

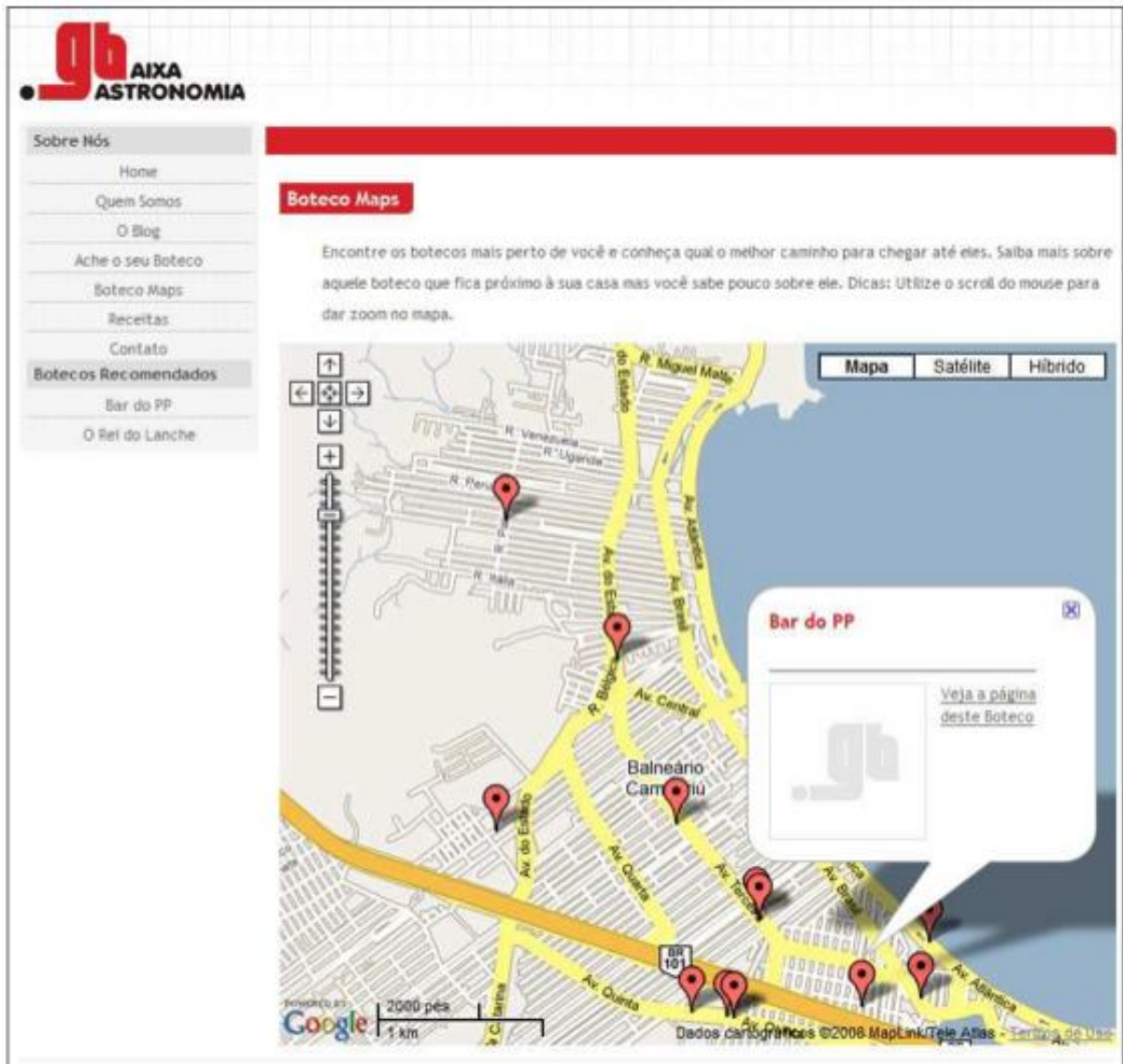
Pode-se considerar, dentre outros, os trabalhos de Cordeiro (2011), Correia (2008), Silva (2010) e Vicentini (2007).

O trabalho de Cordeiro (2011) descreve um Sistema para Identificação de Pontos de Referência Imobiliário, com o objetivo de fornecer e gerenciar os imóveis através da análise dos pontos de interesses, possibilitando a imobiliária ou até mesmo ao visitante da página à consulta do imóvel e seu arredor, sem precisar acessar uma nova página que possua o mapa. O trabalho foi desenvolvido na linguagem *Hypertext Preprocessor* (PHP) utilizando banco de dados MySQL.

Na Figura 1 é apresentada a visualização de um imóvel com suas respectivas características e o mapa de sua localização.



Figura 2 – Visualização do Boteco Maps



Fonte: Correia (2008).

Figura 3 – Tela do “boteco” para o usuário final

**gb AIXA ASTRONOMIA**

**Sobre Nós**

- Home
- Quem Somos
- O Site
- Ache o seu Boteco
- Boteco Maps
- Reservas
- Contato

**Botecos Recomendados**

- Bar do PP
- Quil do Lanche

**Bar do PP**

Endereço: **Rua 3300, 800**  
 Cidade: **Belém/Pará**  
 Encontre este boteco no Boteco Maps

**Descrição**

Boteco com mais de 20 anos de tradição em Belém/Pará.  
 As sextas sempre com as melhores receitas preparadas pelo Pedro Poffo (PP), com várias especialidades, mas sem dúvida o Kibe Crú é a grande diferencial da casa.  
 Público de querubões, de boa conduta, músicas de primeira qualidade, fazem do Bar do PP um lugar da Rua 3300.  
 Se der sorte, o que não é difícil, você pode participar de um dia em que a roda de violão toca, fogueira e cantada pelas frequentadoras do PP.  
 Vale a pena conferir:

1. Rodrigo Geovanilda
2. Kibe Crú (sob encomenda)
3. Culinária de Tefal (sob encomenda)

Além disso, o bar conta com o espaço Jantar, que é o rei das festividades de todo o Brasil.  
 Vale a pena conferir:

- GM - 3,00 (o que é GM?)

Nível: **A**  
 Status: **Operante**  
 Última Atualização: 07/06/2008

**No Bar do PP tem**

- Atendimento de Primeira
- Dom Pappo
- Cerveja Gelada
- Jogo do Bicho
- Lanches
- Petiscos
- TV (Jogos / Novela)

**Fotos do Bar do PP**

Home Quem Somos O Site Ache o seu Boteco Boteco Maps Reservas Contato

Copyright 2008 - Todos os direitos reservados. Desenvolvido por J&B Web

Fonte: Correia (2008).



O trabalho de Silva (2010) descreve um Sistema de Gestão e Divulgação de Imóveis via *web*, para a cidade de Itapema no Estado de Santa Catarina. Este trabalho foi desenvolvido utilizando PHP e AJAX, integrado com a API do Google Maps e a API de dados do YouTube.

Na Figura 4 é apresentado o portal do sistema onde pode-se verificar lançamentos, fazer uma busca rápida e demais funcionalidades.

Figura 4 – Portal imobiliário

The screenshot displays the 'Its Imóveis' website interface. At the top, there's a navigation bar with 'Home', 'Imóveis', 'Parceiros', and 'Contato'. Below the navigation, a status bar indicates 'Atualmente possuímos mais de 10 propriedades: 1 para Aluguel, 10 para Venda, 1 para Temporada' and the date 'Segunda-Feira - 10 de Julho de 2010'. The main content area features a large banner for 'Splendour of the Sea RESIDENCE' with the text 'PRONTO PARA MORAR' and 'O empreendimento se situa no melhor plano da região. J.A. Russi'. To the right of the banner is a vertical stack of four smaller images showing different views of the building. Below the banner, there are three main sections: 'SOBRE O NOSSO SITE' with placeholder text, 'LANÇAMENTOS' with a grid of four interior photos, and 'BUSCA RÁPIDA' with dropdown menus for 'Finalidade' and 'Tipo de Imóvel', and a 'Pesquisar' button. The footer contains 'ÚLTIMOS IMÓVEIS CADASTRADOS' with a grid of four photos, 'ENCONTRE-NÓS' with social media icons for Twitter, Facebook, and Flickr, and 'TEXTO NO RODAPÉ' with contact information: 'Copyright © 2010 - Itapema Imóveis. Todos os direitos reservados. Cel: (47) 9139-5235 / Tel: (47) 3289-3285'.

Fonte: Silva (2010).



Na Figura 5 é mostrada uma busca detalhada dos imóveis no sistema.

Figura 5 – Busca detalhada

The screenshot displays the 'Its Imóveis' website interface. At the top, there is a navigation bar with 'Home', 'Imóveis', 'Parceiros', and 'Contato'. Below the header, a search bar shows filters for 'Apartamento', 'Casa', 'Terreno', and 'Temporário'. The main section is titled 'TODOS OS IMÓVEIS' and lists four properties:

- Splendor of the Sea:** Av. Marechal Deodoro, 3117. Sob consulta. Tipo: Apartamento. Categoria: Praia popular. Dormitórios: 4 Quartos. Garagem: 2 Vagas Cobertas. Agência: Andreinha.
- Possidon Residential Club:** Av. Marechal Deodoro, 3117. Sob consulta. Tipo: Apartamento. Categoria: Praia Popular. Dormitórios: 4 Quartos. Garagem: 2 Vagas Cobertas. Agência: Andreinha.
- Premier Tower:** Av. Marechal Deodoro, 3117. Sob consulta. Tipo: Apartamento. Categoria: Praia Popular. Dormitórios: 4 Quartos. Garagem: 2 Vagas Cobertas. Agência: Dom Bosco.
- Summer Villa:** Av. Marechal Deodoro, 3117. Sob consulta. Tipo: Apartamento. Categoria: Praia Popular. Dormitórios: 4 Quartos. Garagem: 2 Vagas Cobertas. Agência: Dom Bosco.

Each property card includes a photo, a 'Sob consulta' status, and links for 'DETALHES', 'VER NO MAPA', and 'REQUERER INFORMAÇÕES'. A right sidebar titled 'RESULTADO' shows 'Foram encontrados 10 imóveis referentes aos seus filtros' and a 'BUSCA AVANÇADA' section with various filters like 'Localização', 'Preço', 'Área', 'Garagem', 'Dormitórios', and 'Quartos'. A 'Pesquisar' button is at the bottom of the sidebar. The footer contains 'TODOS IMÓVEIS CATEGORIAS', 'CONTATE-NOS' (with WhatsApp, Facebook, and Email icons), 'TODOS OS SERVIÇOS', and contact details: 'Copyright © 2019. Todos os direitos reservados. Contato: (41) 3253-8289 / (41) 3253-3255'.

Fonte: Silva (2010).

A Figura 6 possibilita a exibição de um vídeo do imóvel hospedado no YouTube e verifica características e o mapa da localização do imóvel.

Figura 6 – Exibição de vídeo (YouTube)



Fonte: Silva (2010).

O trabalho de Vicentini (2007) descreve um sistema de processamento de transações que auxilie no processo de vendas de imóveis, permitindo maior controle e agilidade nas negociações. O foco principal é possibilitar o armazenamento e a consulta de informações num único sistema de maneira organizada e de fácil acesso.

O trabalho foi desenvolvido na ferramenta Genexus, com geradores de código Visual Basic e .NET e banco de dados *Microsoft Data Engine (MSDE)*. Na Figura 7 é apresentada a consulta de imóveis via *web*.

Figura 7 – Consulta de imóveis via web

**Consulta Imóvel**

Tipo de imóvel:

Cidade:

Bairro:

Estado:

Valor mínimo:

Valor máximo:

Clique no imóvel para ver detalhes

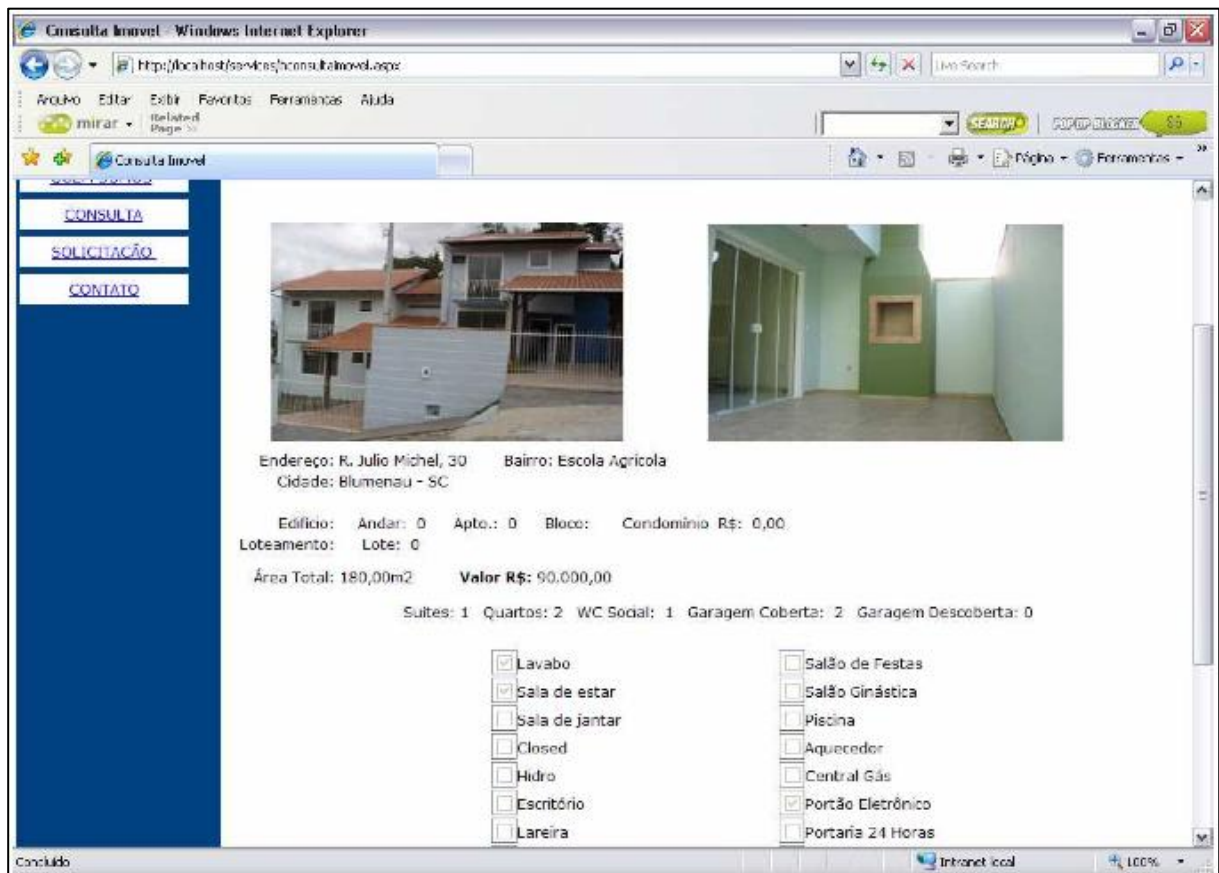
Descrição tipo	Bairro	Cidade	UF	Valor
<a href="#">CASA</a>	<a href="#">Garcia</a>	<a href="#">Blumenau</a>	<a href="#">SC</a>	<a href="#">100.000,00</a>
<a href="#">CASA</a>	<a href="#">Roupevazinha</a>	<a href="#">Blumenau</a>	<a href="#">SC</a>	<a href="#">75.000,00</a>
<a href="#">APARTAMENTO</a>	<a href="#">Escola Agrícola</a>	<a href="#">Blumenau</a>	<a href="#">SC</a>	<a href="#">90.000,00</a>
<a href="#">APARTAMENTO</a>	<a href="#">Roupeva Norte</a>	<a href="#">Blumenau</a>	<a href="#">SC</a>	<a href="#">90.000,00</a>
<a href="#">SOBRADO</a>	<a href="#">Escola Agrícola</a>	<a href="#">Blumenau</a>	<a href="#">SC</a>	<a href="#">90.000,00</a>
<a href="#">APARTAMENTO</a>	<a href="#">Roupeva Norte</a>	<a href="#">Blumenau</a>	<a href="#">SC</a>	<a href="#">85.000,00</a>

[1º Primeiro < Anterior](#) [Próximo >](#) [Último >>](#)

Fonte: Vicentini (2007).

A Figura 8 apresenta as características do imóvel onde pode-se notar que não há integração com o Google Maps.

Figura 8 – Tela de detalhes do imóvel



Fonte: Vicentini (2007).

### 3 DESENVOLVIMENTO

Neste capítulo estão descritos as particularidades técnicas do sistema proposto tais como a descrição do mesmo e a apresentação dos requisitos funcionais e não funcionais, diagrama de casos de uso e suas descrições, diagrama de entidade relacionamento, implementação, principais ferramentas utilizadas e, também, operacionalidade.

#### 3.1 LEVANTAMENTO DE INFORMAÇÕES

O sistema desenvolvido fundamenta-se em uma plataforma *web-mobile* para auxiliar na locação de imóveis tais como apartamentos, casas e quitinetes, que auxilie o usuário a escolher o local a partir de sua localização atual utilizando-se da API do Google Maps.

Com base nas necessidades básicas do locatário e informações padrões nos sítios de imobiliárias obteve-se os requisitos desenvolvidos. O sistema tem a funcionalidade de mostrar as informações sobre os locais disponíveis a locação que são pertinentes na hora de comparar um imóvel com outro, auxiliando assim o usuário a verificar os imóveis que mais o atraírem para visitaç o ou fechar neg cio, obtendo assim a satisfa o do usu rio.

Para a constru o da aplica o foram utilizadas as seguintes ferramentas:

- a) Netbeans 7.4, como plataforma de desenvolvimento;
- b) ferramenta DBDesigner, para constru o do Modelo de Entidade e Relacionamento (MER);
- c) MySQL 10.7, como banco de dados.

#### 3.2 ESPECIFICA O

Nesta se o s o apresentados os principais requisitos funcionais e n o funcionais, sua rastreabilidade com casos de uso e o MER.

### 3.2.1 Requisitos do Sistema

O Quadro 1 apresenta os requisitos funcionais do sistema e sua rastreabilidade, ou seja, vinculação com o(s) caso(s) de uso associado(s).

Quadro 1 – Requisitos funcionais

<b>Requisitos Funcionais</b>	<b>Caso de Uso</b>
RF01: O sistema deve permitir ao administrador manter bairros.	UC01
RF02: O sistema deve permitir ao administrador manter cidades.	UC02
RF03: O sistema deve permitir ao administrador manter localidades.	UC03
RF04: O sistema deve permitir ao administrador manter logradouros.	UC04
RF05: O sistema deve permitir ao administrador manter contatos de imóveis.	UC05
RF06: O sistema deve permitir ao administrador manter imóveis.	UC06
RF07: O sistema deve permitir ao administrador manter imagens dos imóveis.	UC07
RF08: O sistema deve permitir ao administrador alterar a situação do imóvel de disponível para indisponível.	UC08
RF09: O sistema deve permitir ao usuário visualizar sua localização atual.	UC09
RF10: O sistema deve permitir ao usuário visualizar a localização dos imóveis disponíveis para locação.	UC10
RF11: O sistema deve exibir ao usuário as características do imóvel quando o mesmo requerer.	UC11

O Quadro 2 lista os requisitos não funcionais do sistema.

Quadro 2 – Requisitos não funcionais

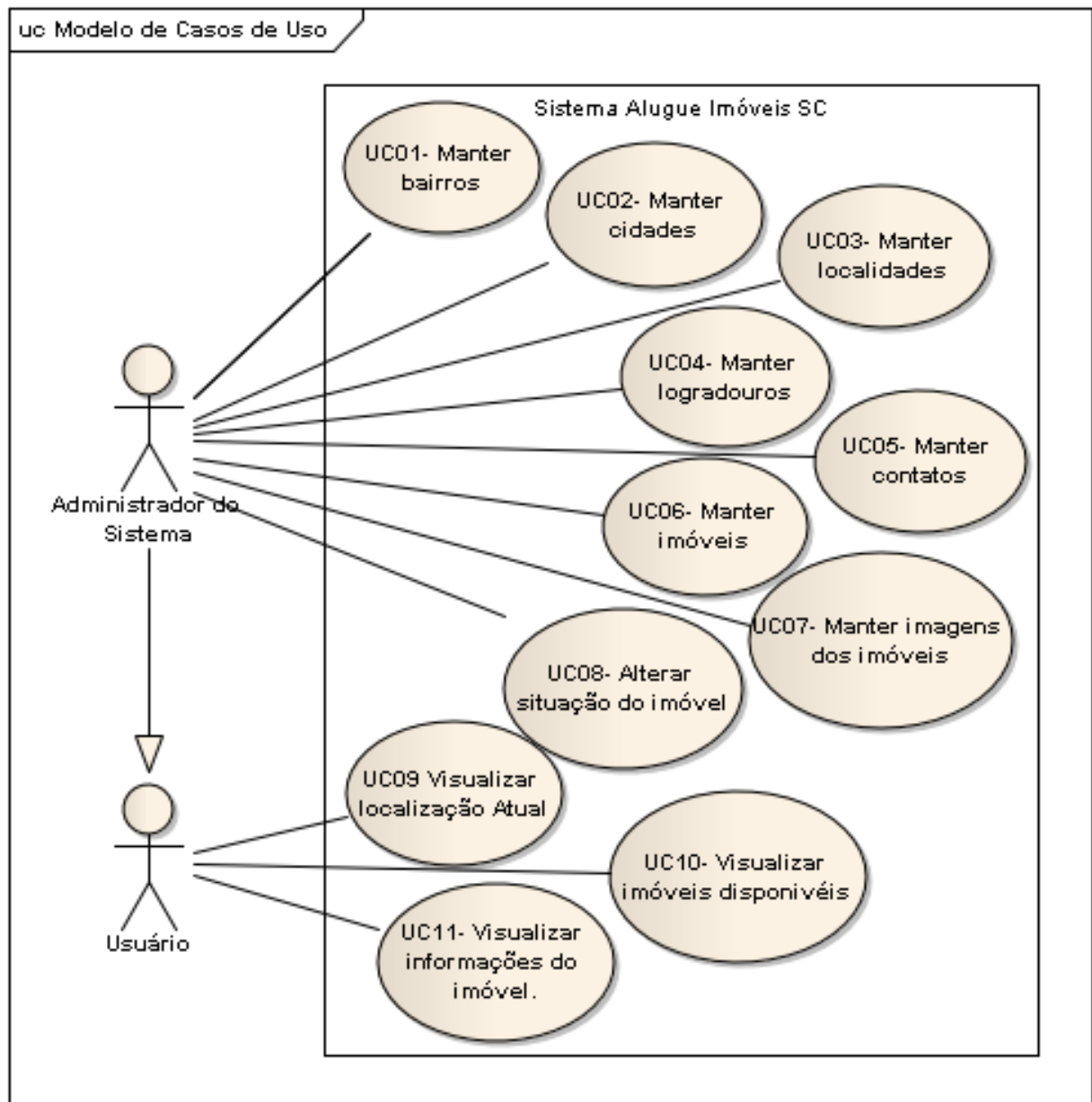
<b>Requisitos Não Funcionais</b>
RNF01: O sistema deve utilizar a localização do usuário para iniciar o programa.
RNF02: O sistema deverá ser desenvolvido em JavaScript.
RNF03: O sistema deverá utilizar banco de dados MySQL 10.7.
RNF04: O sistema deve utilizar a integração com Google Maps.
RNF05: O sistema deverá ser utilizado por no mínimo de uma tela de 9' para a parte

administrativa.

### 3.2.2 Diagrama de Casos de Uso

A Figura 9 contempla o papel de cada ator envolvido no sistema através de um diagrama de casos de uso. A descrição detalhada dos casos de Uso é apresentada no Apêndice A.

Figura 9 – Modelo de casos de uso

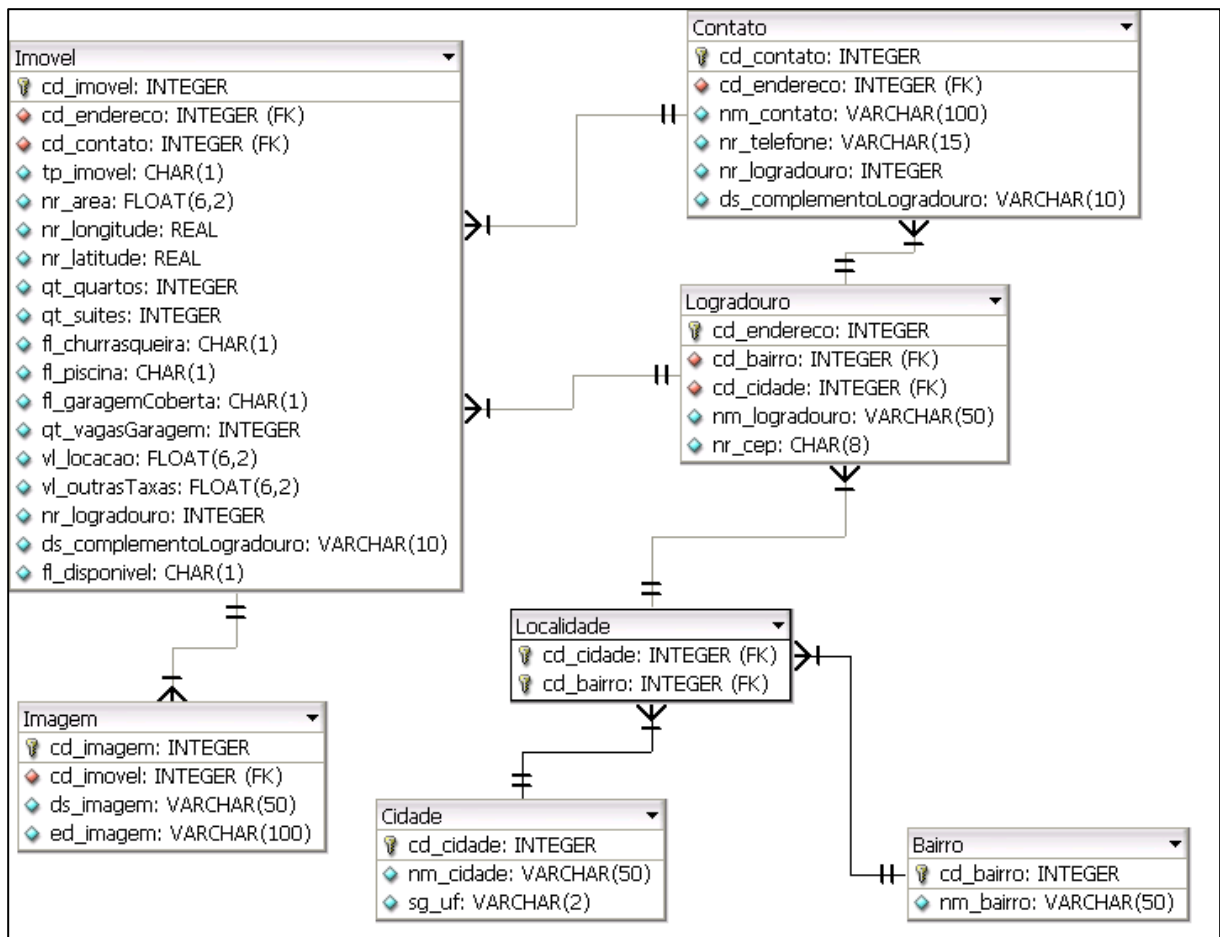




### 3.2.3 Modelo Entidade Relacionamento

A Figura 10 contempla o MER do sistema. O dicionário de dados é apresentado no Apêndice B.

Figura 10 – Modelo entidade relacionamento



A função de cada classe de entidade está descrita a seguir:

- classe Imóvel – classe para instanciar um imóvel;
- classe Imagem – classe para instanciar uma imagem de um imóvel;
- classe Contato – classe para instanciar um contato do imóvel;
- classe Logradouro – classe para instanciar um logradouro;
- classe Localidade – classe para instanciar uma localidade;
- classe Cidade – classe para instanciar uma cidade;
- classe Bairro – classe para instanciar um bairro.



### 3.3 IMPLEMENTAÇÃO

A seguir são mostradas as técnicas e ferramentas utilizadas e a operacionalidade da implementação.

#### 3.3.1 Técnicas e ferramentas utilizadas

Para o desenvolvimento do sistema foi utilizada a linguagem de programação Java com a técnica *Create, Read, Update, Delete* (CRUD) por meio do Netbeans IDE. Foi utilizada também a biblioteca do PrimeFaces, da API JavaScript do Google Maps v3 e a técnica de proteção e autenticação de acesso às páginas pelo servidor do Glassfish 4.0.

##### 3.3.1.1 Netbeans IDE 7.4 e a técnica CRUD

O Netbeans IDE oferece a capacidade de exibir e modificar os dados contidos no banco de dados, conhecida como recurso *Create, Read, Update, Delete* (CRUD). A aplicação desenvolvida contará com as seguintes tecnologias:

- a) JavaServer Faces (JSF) 2.x: para páginas *web* de *frontend*, manipulação de validação e gerenciamento do ciclo de solicitação-resposta;
- b) Java Persistence API (JPA) 2.0: utilizando EclipseLink para gerar classes de entidade a partir do banco de dados e gerenciar transações (EclipseLink é a implementação de referência para JPA e é o provedor de persistência default para o GlassFish Server);
- c) Enterprise JavaBeans (EJB) 3.1: que oferece EJBs sem estado que acessam as classes de entidade e contém a lógica de negócios da aplicação.

O IDE oferece dois assistentes que geram todos os códigos da aplicação. O primeiro é o assistente para Classes de Entidade do Banco de Dados que permite gerar as classes de entidade a partir do banco de dados fornecido. Após criar as classes de entidade, você utiliza o assistente de Páginas JSF de Classes de Entidade para criar EJBs e *beans* gerenciados pelo

JSF para as classes de entidade, bem como um conjunto de páginas de Facelets para manipular as *views* dos dados da classe de entidade (NETBEANS, 2014a).

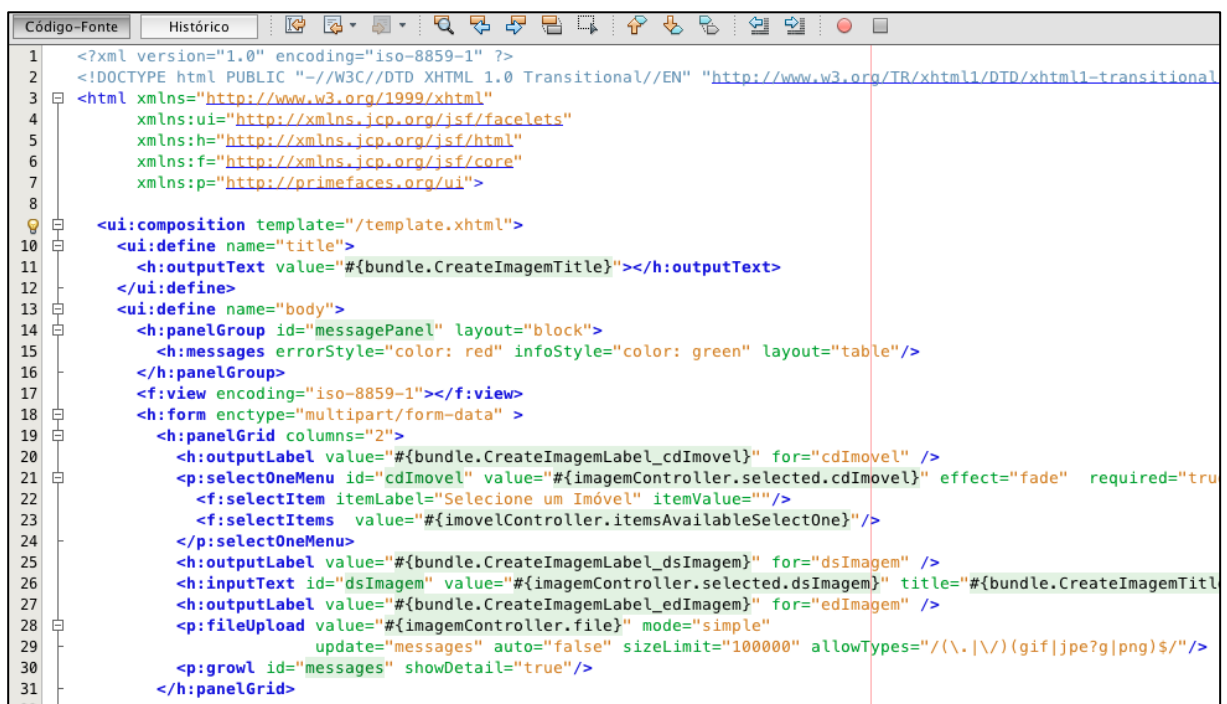
Mais informações sobre a técnica CRUD podem ser encontradas no link “<[https://netbeans.org/kb/docs/web/jsf20-crud\\_pt\\_BR.html](https://netbeans.org/kb/docs/web/jsf20-crud_pt_BR.html)>” (NETBEANS, 2014a).

### 3.3.1.2 Primefaces

A biblioteca do Primefaces é livre com código aberto, biblioteca de componentes para o *framework* Java Server Faces (JSF), desenvolvido pela Prime Teknoloji. Ela fornece um conjunto de componentes principalmente visuais, que podem ser usados por programadores de JSF, além do pequeno conjunto de componentes básicos que vem com a plataforma central JSF para compor a interface para uma aplicação *web*. (PRIMEFACES, 2014).

Na Figura 11 demonstra-se a utilização da biblioteca Primefaces para a inserção de imagens no sistema.

Figura 11 – Utilização da biblioteca do Primefaces



```

1 <?xml version="1.0" encoding="iso-8859-1" ?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
3 <html xmlns="http://www.w3.org/1999/xhtml"
4       xmlns:ui="http://xmlns.jcp.org/jsf/facelets"
5       xmlns:h="http://xmlns.jcp.org/jsf/html"
6       xmlns:f="http://xmlns.jcp.org/jsf/core"
7       xmlns:p="http://primefaces.org/ui">
8
9 <ui:composition template="/template.xhtml">
10 <ui:define name="title">
11 <h:outputText value="#{bundle.CreateImagemTitle}"></h:outputText>
12 </ui:define>
13 <ui:define name="body">
14 <h:panelGroup id="messagePanel" layout="block">
15 <h:messages errorStyle="color: red" infoStyle="color: green" layout="table"/>
16 </h:panelGroup>
17 <f:view encoding="iso-8859-1"></f:view>
18 <h:form enctype="multipart/form-data" >
19 <h:panelGrid columns="2">
20 <h:outputLabel value="#{bundle.CreateImagemLabel_cdImovel}" for="cdImovel" />
21 <p:selectOneMenu id="cdImovel" value="#{imagemController.selected.cdImovel}" effect="fade" required="tru
22 <f:selectItem itemLabel="Selecione um Imóvel" itemValue="" />
23 <f:selectItems value="#{imagemController.itemsAvailableSelectOne}" />
24 </p:selectOneMenu>
25 <h:outputLabel value="#{bundle.CreateImagemLabel_dsImagem}" for="dsImagem" />
26 <h:inputText id="dsImagem" value="#{imagemController.selected.dsImagem}" title="#{bundle.CreateImagemTitl
27 <h:outputLabel value="#{bundle.CreateImagemLabel_edImagem}" for="edImagem" />
28 <p:fileUpload value="#{imagemController.file}" mode="simple"
29               updates="messages" auto="false" sizerLimit="100000" allowTypes="(\\.|\/)(gif|jpe?g|png)$"/>
30 <p:growl id="messages" showDetail="true"/>
31 </h:panelGrid>

```

### 3.3.1.3 API Javascript do Google Maps v3

Todos os aplicativos\* da API do Google Maps devem carregar a API do Google Maps usando uma chave de API. O uso de uma chave de API permite monitorar a utilização da API do Google Maps por parte de seu aplicativo e garante que o Google pode entrar em contato com você com relação a seu aplicativo, se necessário. Se a utilização da API do Google Maps de seu aplicativo ultrapassar os limites de uso, você deverá carregar a API do Google Maps usando uma chave de API para adquirir cota adicional. (GOOGLE DEVELOPERS, 2013).

O principal aspecto da aplicação é a integração com a API do Google Maps onde o usuário pode verificar a real latitude e longitude do imóvel. Na Figura 12, é demonstrado como se faz o carregamento da API do Google Maps.

Figura 12 – Carregamento da API do Google Maps

```
<html>
  <head>
    <script type="text/javascript"
      src="http://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&sensor=SET_TO_TRUE_OR_FALSE">
    </script>
```

O parâmetro *key* contém a chave de API de seu aplicativo. O parâmetro *sensor* do URL deve ser incluído e indica se esse aplicativo usa um sensor (como um localizador de GPS) para determinar a localização do usuário. Na Figura 13 pode ser verificado como se inicializa um mapa utilizando da API do Google Maps.

Figura 13 – Iniciando um mapa

```
function initialize() {
  var mapOptions = {
    center: new google.maps.LatLng(-26.9187523, -49.0660252),
    zoom: 10,
    mapTypeId: 'roadmap'
  };
  map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);
```

Na Figura 14, é demonstrada a função da API do Google Maps que identifica a sua localização atual. Se não for possível encontrar a posição atual em função do *browser* não suportar geoposicionamento ou pelo usuário não permitir que o sistema utilize da sua localização atual, a aplicação iniciará mostrando a cidade de Blumenau.

Figura 14 – Função que pega localização atual

```

// Tentar a geolocalização com HTML5
if (navigator.geolocation) {
  navigator.geolocation.getCurrentPosition(function(position) {
    var pos = new google.maps.LatLng(position.coords.latitude,
    position.coords.longitude);
    var contentString = '<div id="content">' +
      '<div id="siteNotice">' +
      '</div>' +
      '<div id="bodyContent">' +
      '<p>Localização atual usando HTML5</p>' +
      '</div>' +
      '</div>';
    infoWindowPosition.setContent(contentString);
    marker = new google.maps.Marker({
      map: map,
      position: pos,
      title: 'Localização atual usando HTML5.',
      icon: 'resources/icon/boneco.png'
    });
    google.maps.event.addListener(marker, 'click', function() {
      infoWindowPosition.open(map, marker);
    });
    google.maps.event.addDomListener(window, 'load', initialize);
    map.setCenter(pos);
  }, function() {
    handleNoGeolocation(true);
  });
} else {
  // Se o Browser não suportar geolocalização
  handleNoGeolocation(false);
}

function handleNoGeolocation(errorFlag) {
  if (errorFlag) {
    var content = 'Erro: O serviço de geolocalização falhou. Posição Atual Blumenau.';
  } else {
    var content = 'Erro: Seu browser não suporta geolocalização.';
  }
  var options = {
    map: map,
    position: new google.maps.LatLng(-26.9187523, -49.0660252), //se falhar direciona para Blumenau
  };
  contentString = '<div id="content">' +
    '<div id="siteNotice">' +
    '</div>' +
    '<div id="bodyContent">' +
    '<p>Localização atual falhou. Posição atual Blumenau</p>' +
    '</div>' +
    '</div>';
  infoWindowPosition.setContent(contentString);
  marker = new google.maps.Marker(options);
  //evento para abrir a infowindow
  google.maps.event.addListener(marker, 'click', function() {
    infoWindowPosition.open(map, marker);
  });
  google.maps.event.addDomListener(window, 'load', initialize);
  map.setCenter(options.position);
}

```

### 3.3.1.4 Map Clusterer

Com o *Map Clusterer* é possível juntar os marcadores próximos para que se tornem um grupo, e a cada *zoom in* do usuário ele vai transformando os marcadores individualmente como demonstrado na Figura 15.

Figura 15 – *Map clusterer*

Fonte: Google Developers (2013).

Para que o agrupamento dos marcadores próximos uns dos outros seja possível é utilizado uma classe JavaScript que faz o tratamento desta aproximação através de cálculos chamada `markerclusterer.js`. A Figura 16 demonstra essa classe.

Figura 16 – Classe `markerclusterer.js`

```

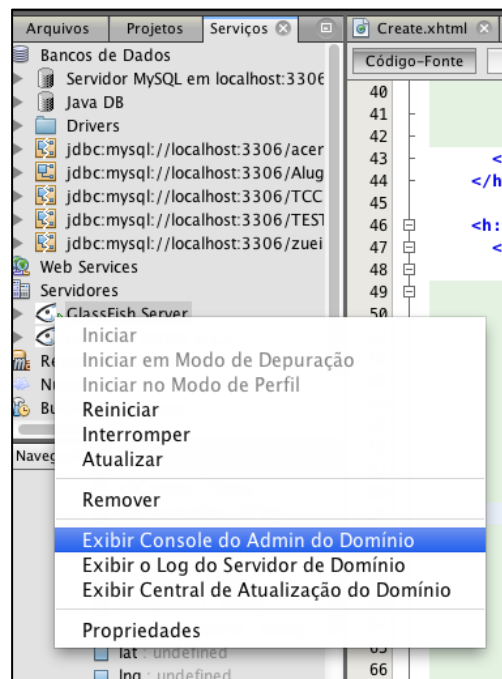
1  eval(function(p, a, c, k, e, r) {
2  e = function(c) {
3      return(c < a ? '' : e(parseInt(c / a)) + ((c = c % a) > 35 ? String.fromCharCode(c + 29) : c.toString(36)));
4  };
5  if (!''.replace(/~/, String)) {
6      while (c--)
7          r[e(c)] = k[c] || e(c);
8      k = [function(e) {
9          return r[e];
10         }];
11     e = function() {
12         return '\\w+!';
13     };
14     c = 1;
15 }
16 ;
17 while (c--)
18     if (k[c])
19         p = p.replace(new RegExp('\\b' + e(c) + '\\b', 'g'), k[c]);
20 return p;
21 }('5 J(b,a){b.1c().X(J,o.n.2Y);4.I=b;4.N=a;4.B=s;4.q=s;4.1x=s;4.1m=t;4.L(b.z())}J.6.2J=5(){7 c=4;7 f;7 d;4.q=30.3H

```

### 3.3.1.5 Protegendo aplicação WEB no Netbeans IDE usando servidor Glassfish 4.0

Para que a parte dos cadastros do sistema esteja segura é necessário implementar um método de autenticação para acessar determinadas páginas. Na Figura 17 mostra-se como abrir as configurações do servidor Glassfish 4.0 para se fazer os passos necessários para a autenticação.

Figura 17 – Exibir configurações do servidor Glassfish 4.0



A Figura 18 demonstra que é necessário criar um *Realm* do tipo *file* onde se define usuário e senha para o sistema.

Figura 18 – Configuração no servidor Glassfish 4.0

The screenshot shows the Glassfish 4.0 administration console. The top navigation bar includes 'Home', 'About...', and 'Help'. Below this, the user is identified as 'admin', the domain as 'domain1', and the server as 'localhost'. The main title is 'GlassFish™ Server Open Source Edition'.

On the left, a 'Tree' view shows the configuration hierarchy. Under 'Security', the 'Realms' folder is expanded, showing 'admin-realm', 'certificate', and 'file'. The 'file' realm is selected.

The main content area is titled 'Manage Users' and contains the following configuration details:

- Configuration Name:** server-config
- Realm Name:** file
- Class Name:** com.sun.enterprise.security.auth.realm.file.FileRealm

Below these are 'Properties specific to this Class':

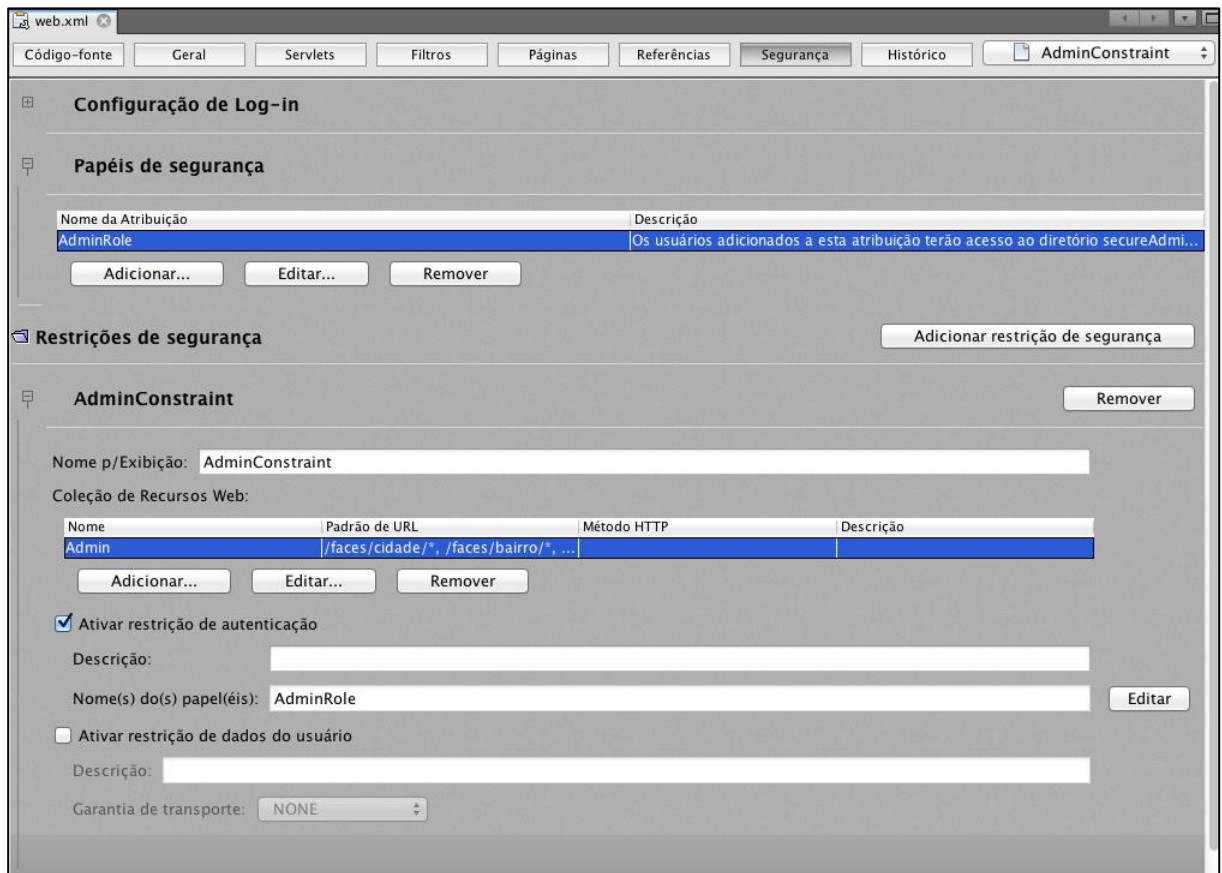
- JAAS Context:** \* fileRealm (Identifier for the login module to use for this realm)
- Key File:** \* \${com.sun.aas.instanceRoot}/config/keyfile (Full path and name of the file where the server will store all user, group, and password information for this realm)
- Assign Groups:** (Comma-separated list of group names)

At the bottom, there is an 'Additional Properties (1)' section with a table:

Select	Name	Value	Description
<input type="checkbox"/>	admin	adminpw1	

Na Figura 19 são mostradas as configurações de segurança do arquivo `web.xml`, onde na coleção de recursos `web` menciona as páginas que necessitam de autenticação para funcionar.

Figura 19 – Configurações no web.xml



A Figura 20 mostra o código da tela de *login* da aplicação para que só permita o acesso às páginas se o usuário estiver logado.



Figura 20 – Código tela de *login*

```

<html>
<head xmlns="http://www.w3.org/1999/xhtml">
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
  <title>Alugue Imóveis SC</title>
  <style>
    body {
      font-family: Arial, Helvetica, sans-serif;
      color: #3a4f54;
      background-color: #dfecf1;
      font-size: 20;
    }
    button, input {
      font-size: 30;
    }
  </style>
</head>
<body align="center" >
  <br><br>
  <h1>Acesso Restrito</h1>
  <form action="j_security_check" method="POST">

    <h2>Login: <input type="text" name="j_username"></input></h2>
    <h2>Senha: <input type="password" name="j_password"></input></h2>

    <button type="submit">
      
    </button>
  </form>
</body>
</html>

```

O procedimento mais detalhado de como proteger uma aplicação *web* no Netbeans IDE usando servidor Glassfish 4.0 pode ser encontrado no link “<a href="https://netbeans.org/kb/docs/web/security-webapps\_pt\_BR.html?print=yes#Deploy\_run">https://netbeans.org/kb/docs/web/security-webapps\_pt\_BR.html?print=yes#Deploy\_run</a>” (NETBEANS, 2014b).

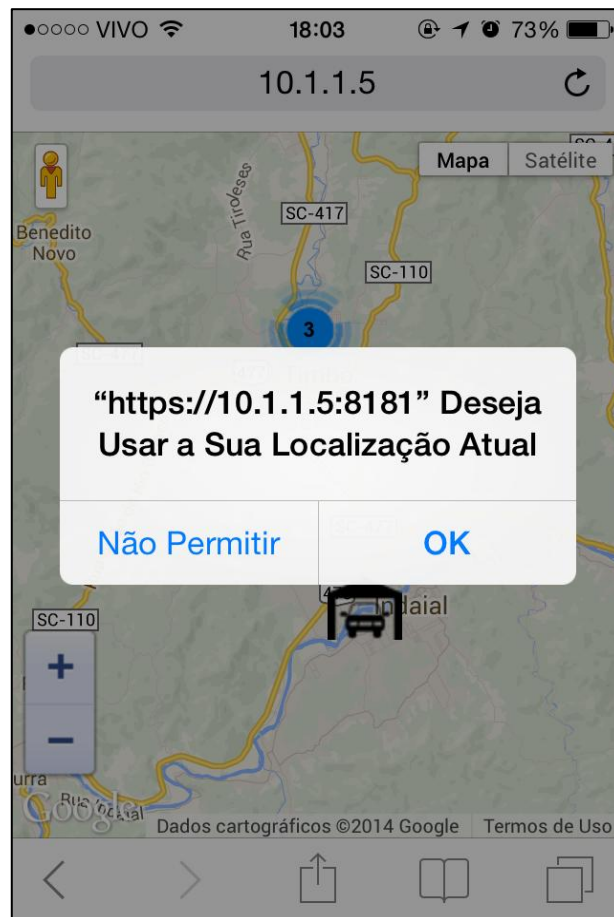
### 3.3.2 Operacionalidade da implementação

Nesta seção é demonstrado o funcionamento da implementação através da utilização do sistema por meio do usuário, e na inserção de um imóvel e imagens deste por meio do administrador.

### 3.3.2.1 Módulo de Consulta

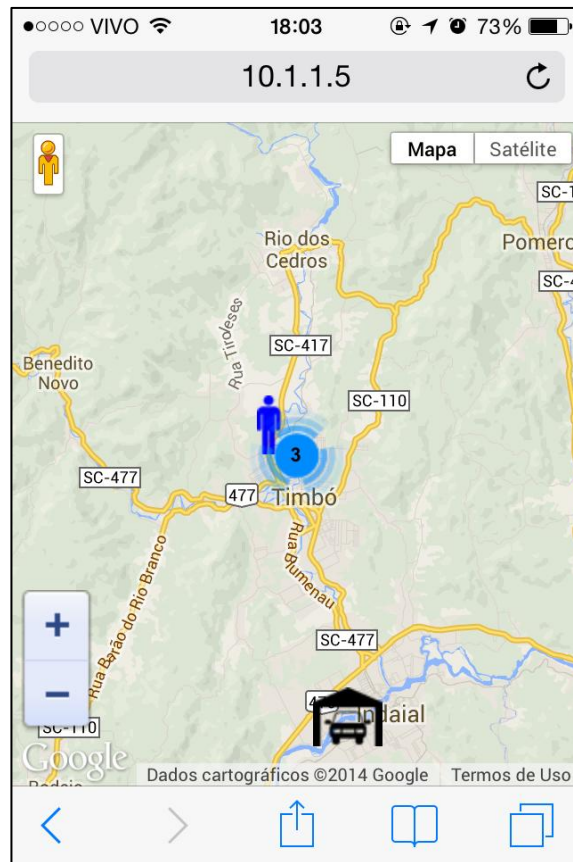
Denominou-se de “Alugue Imóveis SC” o sistema criado. Ao acessar o sistema será solicitada a permissão para utilizar-se da localização atual, conforme Figura 21.

Figura 21 – Tela inicial pedindo se pode usar localização



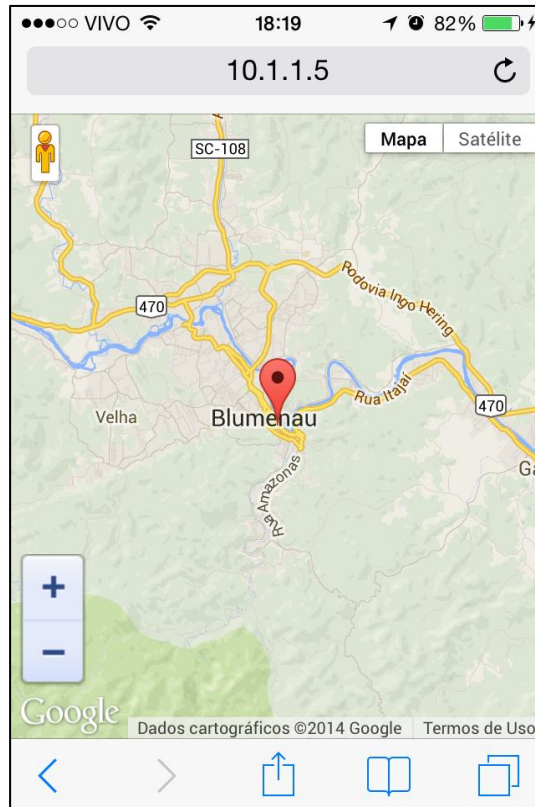
Na Figura 22 pode-se ver a localização do usuário após permitir que usa-se a localização.

Figura 22 – Tela inicial



Na Figura 23 o usuário não permitiu a localização atual e o mapa foi direcionado para Blumenau.

Figura 23 – Tela inicial sem localização



Ao clicar sobre um marcador pode-se visualizar as características do imóvel, conforme as Figuras 24 e 25.

Figura 24 – Tela com as características do imóvel



Figura 25 – Tela com as características do imóvel 2



### 3.3.2.2 Módulo de administração

Ao tentar acessar qualquer endereço da aplicação da parte administrativa o administrador é direcionado para tela de *login*, conforme Figura 26.

Figura 26 – Tela página de *login*

Ao informar as credenciais, se o *login* e senha estiverem corretos o administrador será direcionado para página inicial. Na Figura 27 a página inicial tem os ícones representando bairros, cidades, localidades, logradouros, contatos, imóveis e imagens respectivamente.

Figura 27 – Tela página inicial



Caso contrário ele será direcionado para uma página avisando que houve erro no *login*, conforme Figura 28.

Figura 28 – Tela erro no *login*

Ao clicar no ícone mostrar todos imóveis da Figura 27 é apresentada a lista de imóveis cadastrados, conforme Figura 29. Para se obter uma melhor visualização das informações é aconselhável que para acesso a parte administrativa o dispositivo móvel tenha uma tela de no mínimo 9 polegadas.

Figura 29 – Tela lista de todos imóveis

Lista de Todos os Imóveis															
(1 of 1) 1 10															
Tipo	Disponível	Valor da Locação	Valor das Outras Taxas	Endereço	Número do Logradouro	Descrição do Complemento do Logradouro	Área	Latitude	Longitude	Quantidade de Quartos	Quantidade de Suítes	Garagem Coberta	Quantidade de Vagas na Garagem	Churrasqueira	Piscina
Casa	Sim	R\$: 500.0	R\$: 0.0	Rua Marechal Deodoro da Fonseca, Nações - Timbó SC	Nº 1750		92.0 m²	-26.8106098	-49.2728943	3	0	Sim	4	Sim	Não
Quitinete	Sim	R\$: 700.0	R\$: 0.0	Rua Caracas, Tapaços - Indaial SC	Nº 89		0.0 m²	-26.910832	-49.252759	3	0	Sim	1	Sim	Sim
Apartamento	Sim	R\$: 450.0	R\$: 0.0	Rua Grevilêia, Estados - Timbó SC	Nº 421	Apt. 200	0.0 m²	-26.8427424	-49.286899	2	0	Sim	1	Não	Não
Apartamento	Sim	R\$: 900.0	R\$: 150.0	Rua Marechal Deodoro da Fonseca, Nações - Timbó SC	Nº 1694	Apt. 601	80.0 m²	-26.8110893	-49.2728161	3	3	Sim	2	Sim	Sim
Casa	Sim	R\$: 700.0	R\$: 0.0	Rua Grevilêia, Estados - Timbó SC	Nº 135		323.0 m²	-26.799297	-49.2732708	4	1	Sim	4	Sim	Não

Cadastrar Imóvel


(1 of 1) 1 10

Para se cadastrar um novo imóvel o usuário deve clicar no ícone criar novo imóvel. Ao fazer isso é gerada uma página para o cadastro contendo as informações do imóvel como pode ser visto na Figura 30.

Figura 30 – Tela criar novo imóvel

## Criar Novo Imóvel

Disponível:	<input type="button" value="Sim"/>
Tipo do Imóvel:	<input type="button" value="Casa"/>
Valor da Locação:	<input type="text" value="500"/>
Valor das Outras taxas:	<input type="text" value="0.0"/>
Endereço:	<input type="text" value="Rua Marechal Deodoro da Fonseca - Nações, Timbó - SC"/>
Número do Logradouro:	<input type="text" value="1750"/>
Descrição do Complemento do Logradouro:	<input type="text"/>
Área do Imóvel:	<input type="text" value="92"/>
Latitude do Imóvel:	<input type="text" value="-26.8106098"/>
Longitude do Imóvel:	<input type="text" value="-49.2728943"/>
Quantidade de Quartos:	<input type="button" value="3"/>
Quantidade de Suites:	<input type="button" value="0"/>
Garagem Coberta:	<input type="button" value="Sim"/>
Quantidades de Vagas na Garagem:	<input type="button" value="4"/>
Churrasqueira:	<input type="button" value="Sim"/>
Piscina:	<input type="button" value="Não"/>
Contato:	<input type="text" value="Hermano Roepke - Telefone: (47) 9114-4444"/>




Depois do cadastro do imóvel é necessário cadastrar suas imagens. Para criar uma nova imagem deve-se clicar no ícone criar nova imagem na visualização do imóvel, conforme Figura 31.

Figura 31 – Tela visualização de imóvel

## Visualizar Imóvel

Tipo:	Casa
Disponível:	Sim
Valor da Locação:	R\$: 500.0
Valor das Outras Taxas:	R\$: 0.0
Endereço:	Rua Marechal Deodoro da Fonseca, Nações - Timbó SC
Nº do Logradouro:	Nº 1750
Complemento do Logradouro:	
Área:	92.0 m <sup>2</sup>
Latitude:	-26.8106098
Longitude:	-49.2728943
Quantidade de Quartos:	3
Quantidade de Suites:	0
Garagem Coberta:	Sim
Quantidade de Vagas na Garagem:	4
Churrasqueira:	Sim
Piscina:	Não
	Hermano Roepke - (47) 9114-4444

[Cadastrar Imagem](#)



Então será mostrada ao administrador a tela para criar uma nova imagem, conforme Figura 32.



Figura 32 – Tela criar nova imagem

### Criar Nova Imagem

Imóvel:

Descrição:


Imagem:  












  

Para efetuar o cadastro de mais imagens ao imóvel repete-se o procedimento apresentado tantas vezes quanto necessário. Caso seja necessária a manutenção das imagens de um determinado imóvel, isto é possível através do recurso apresentado na Figura 33.

Figura 33 – Tela lista de todas as imagens

### Lista de Todas as Imagens

(1 of 2) 

Descrição	Imagem	Imóvel	
Fachada		Rua Marechal Deodoro da Fonseca, Nações nº 1750 Timbó SC. Contato: Hermano Roepke - (47) 9114-4444	
Sala		Rua Marechal Deodoro da Fonseca, Nações nº 1750 Timbó SC. Contato: Hermano Roepke - (47) 9114-4444	
Banheiro		Rua Marechal Deodoro da Fonseca, Nações nº 1750 Timbó SC. Contato: Hermano Roepke - (47) 9114-4444	
Cozinha		Rua Marechal Deodoro da Fonseca, Nações nº 1750 Timbó SC. Contato: Hermano Roepke - (47) 9114-4444	
Quarto		Rua Marechal Deodoro da Fonseca, Nações nº 1750 Timbó SC. Contato: Hermano Roepke - (47) 9114-4444	
			

### 3.4 RESULTADOS E DISCUSSÃO

O Quadro 3 apresenta a comparação entre o Sistema Alugue Imóveis SC com os trabalhos correlatos, apresentados na seção 2.6.

Quadro 3 – Quadro comparativo do sistema com os trabalhos correlatos

	<b>ALUGUE IMÓVEIS SC</b>	<b>CORDEIRO (2011)</b>	<b>CORREIA (2008)</b>	<b>SILVA (2010)</b>	<b>VICENTINI (2007)</b>
Possui mapas	SIM	SIM	SIM	SIM	SIM
Mostra o mapa com todos marcadores	SIM	NÃO	SIM	NÃO	NÃO
Possui marcador com informação	SIM	NÃO	SIM	NÃO	NÃO
Possui agrupamento de marcadores	SIM	NÃO	NÃO	NÃO	NÃO
Tecnologia(s)	JAVA	PHP	WebM e AJAX	PHP	GENEXUS com Visual Basic
Banco de dados	MySQL	MySQL	MySQL	MySQL	MSDE

Os trabalhos correlatos e o desenvolvimento neste trabalho possuem a mesma finalidade, auxiliar ao usuário na escolha de um imóvel ou no caso de Correia (2008) um “boteco“ e todos desenvolvidos para o ambiente *web*. Apesar de semelhantes, podem-se destacar algumas diferenças importantes no sistema desenvolvido e seus sistemas correlatos tais como: melhor visualização de todos os imóveis disponíveis no mapa, localização atual do usuário e a mobilidade que o torna acessível em qualquer dispositivo móvel conectado a internet.

Convém destacar as diferenças entre os trabalhos apresentados e o Alugue Imóveis SC quanto ao uso de tecnologias. Os trabalhos de Cordeiro (2011) e Silva (2010) foram desenvolvidos utilizando PHP e banco de dados MySQL, já Correia (2008) utilizou WebM e AJAX e banco de dados MySQL, integrando parte dos recursos do Google Maps. Já Vicentini (2007) utilizou Genexus sendo o código gerado para Visual Basic e banco de dados MSDE. Todos esses trabalhos tinham como plataforma de acesso um navegador de internet em equipamentos do tipo computador pessoal.

Já o Alugue Imóveis SC foi desenvolvido em Java com bibliotecas do Primefaces e banco de dados MySQL. Sua principal contribuição está focada no uso por meio de dispositivos móveis (*smartphones* e *tablets*) com acesso a internet.

## 4 CONCLUSÕES

Neste trabalho apresentou-se o desenvolvimento de um sistema na plataforma *web-mobile* para auxiliar os usuários a encontrar um imóvel para alugar a partir de sua localização atual contendo dados e imagens, em que seus objetivos inicialmente foram alcançados com sucesso.

As ferramentas utilizadas para a elaboração do trabalho foram adequadas. As utilizações das bibliotecas contribuíram na agilidade no aprimoramento do sistema. O maior desafio na produção do trabalho foi compreender o funcionamento da API do Google Maps v3, bem como a utilização de algumas classes como o `markerclusterer.js` que tornou possível o agrupamento dos marcadores, tendo assim uma melhor visualização do sistema. A API do Google Maps v3 oferece diversos serviços por isso é necessário ter um prévio conhecimento com programação JavaScript e conceitos de programação voltados a objetos.

Acredita-se que o sistema possa auxiliar o usuário em encontrar imóveis rapidamente a partir de sua localização atual, propiciando assim uma avaliação prévia antes do contato para agendamento de visita. A integração do Google Maps proporcionou ao usuário uma maneira fácil de encontrar imóveis próximos de diferentes tipos (apartamentos, casas e quitinetes).

Além de identificar imóveis a partir de sua localização atual, o sistema permite que qualquer pessoa de qualquer localização que pode ser, por exemplo, na locomoção diária para o trabalho de ônibus, em uma sala de espera ou em alguma fila, e utilizar esse tempo que não serviria para nada, consiga identificar imóveis no entorno de um determinado endereço. Isto é um diferencial do trabalho pois integra as informações do imóvel com sua localização e proximidades.

### 4.1 EXTENSÕES

A partir deste projeto, como sugestão para trabalhos futuros tem-se:

- a) implementar uma maneira de contar as visitas feitas por usuários em cada marcador para poder gerar um relatório de acessos ao contato;
- b) desenvolver um módulo para venda de imóveis além do módulo de locação;

- c) desenvolver filtros, para a questão de distância do local onde se encontra;
- d) desenvolver um sistema onde o próprio contato cadastre o seu imóvel;
- e) integrar a API do Google Directions para mostrar o caminho à ser percorrido até o imóvel;
- f) integrar a API do Google Street View para verificar a rua em que o imóvel se encontra;
- g) transformar o sistema *web mobile* em aplicativo para dispositivo móvel sendo para Android, IOS e Windows Phone.

## REFERÊNCIAS

CAIXA ECONÔMICA FEDERAL. **Construção Civil**. [S.l.], out. 2013. Disponível em: <<http://www1.caixa.gov.br/construcaocivil/index.asp>>. Acesso em: 25 out. 2013.

CANDELORO, Milton. **Avaliação de Aluguéis**. São Paulo: Pini, 1991.

CORDEIRO, Aline. **Sistema para Identificação de Pontos de Referência Imobiliário**. 2011. 62 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

CORREIA, Rion B. **Sistema para Gestão e Divulgação de Ambientes Gastronômicos**. 2008. 62 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

GOOGLE. **Google Maps**. [S.l.], [2013?]. Disponível em: <<https://support.google.com/maps/answer/3092426?hl=pt-BR>>. Acesso em: 25 de out. 2013.

GOOGLE DEVELOPERS. **API Javascript do Google Maps v3**. [S.l.], fev. 2013. Disponível em: <<https://developers.google.com/maps/documentation/javascript/?hl=pt-BR>>. Acesso em: 25 out. 2013.

HALFELD, Mauro. **Seu imóvel: Como comprar bem**. São Paulo: Fundamento Educacional, 2002.

OLIVEIRA, Leandro. **Web mobile: a internet como unificadora de apps móveis**. [S.l.], nov. 2010. Disponível em: <<http://www.mobileeasy.com.br/dotheevolution/2010/11/25/web-mobile-a-internet-como-unificadora-de-apps-moveis/>>. Acesso em: 25 out. 2013.

NETBEANS. **Gerando uma Aplicação CRUD JavaServer Faces 2.x Usando um Banco de Dados**. [S.l.], [2014a?]. Disponível em: <[https://netbeans.org/kb/docs/web/jsf20-crud\\_pt\\_BR.html](https://netbeans.org/kb/docs/web/jsf20-crud_pt_BR.html)>. Acesso em: 20 fev. 2014.

NETBEANS. **Protegendo uma aplicação Web no NetBeans IDE**. [S.l.], [2014b?]. Disponível em: <[https://netbeans.org/kb/docs/web/security-webapps\\_pt\\_BR.html?print=yes#Deploy\\_run](https://netbeans.org/kb/docs/web/security-webapps_pt_BR.html?print=yes#Deploy_run)>. Acesso em: 20 fev. 2014.

PRIMAR ADMINISTRADORA DE IMÓVEIS. **A preferência dos jovens: imóveis pequenos e bem localizados**. Rio de Janeiro, ago. 2013. Disponível em: <<http://www.pautas.incorporativa.com.br/a-mostra-release.php?id=21851>>. Acesso em: 25 out. 2013.

PRIMEFACES. **Why PrimeFaces**. [S.l.], fev. 2014. Disponível em: <<http://www.primefaces.org/whyprimefaces>>. Acesso em: 20 fev. 2014.

SILVA, Marcos V. B. **Sistema de gestão e divulgação de imóveis.** 2010. 92 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

UOL MULHER. **Aluguel:** qual imóvel se encaixa melhor ao seu perfil, casa ou apartamento? São Paulo, nov. 2011. Disponível em: <<http://mulher.uol.com.br/casa-e-decoracao/noticias/infomoney/2011/11/14/aluguel-qual-imovel-se-encaixa-melhor-ao-seu-perfil-casa-ou-apartamento.htm>>. Acesso em: 30 ago. 2013.

VICENTINI, Diego. **Sistema de informações para controle de vendas em imobiliária.** 2007. 85 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

## APÊNDICE A – DESCRIÇÃO DOS CASOS DE USO

Este Apêndice apresenta a descrição dos principais casos de uso descritos na seção de especificação deste trabalho. Nos Quadros de 4 até 13 são demonstrados todos os casos de uso definidos para o desenvolvimento do sistema.

Quadro 4 – Descrição do caso de uso manter bairros

**Caso de uso UC01 – Manter bairros.**

**Ator:** Administrador.

**Objetivo:** Cadastrar um bairro de uma ou mais localidades.

**Pré-condições:** Nenhuma.

**Pós-condições:** O administrador cadastrou um bairro de uma ou mais localidades.

**Cenário Principal:**

1. O administrador vai na opção de mostrar todos os bairros;
2. O administrador vai na opção cadastrar novo bairro;
3. O sistema apresenta a tela de cadastro de bairro;
4. O administrador informa os dados do bairro;
5. O administrador clica no botão salvar;
6. O sistema salva as informações do novo bairro.

Quadro 5 – Descrição do caso de uso manter cidades

**Caso de uso UC02 – Manter cidades.**

**Ator:** Administrador.

**Objetivo:** Cadastrar uma cidade de uma ou mais localidades.

**Pré-condições:** Nenhuma.

**Pós-condições:** O administrador cadastrou uma cidade de uma ou mais localidades.

**Cenário Principal:**

1. O administrador seleciona a opção de mostrar todas as cidades;
2. O administrador seleciona a opção cadastrar nova cidade;
3. O sistema apresenta a tela de cadastro de cidade;
4. O administrador informa os dados da cidade;
5. O administrador clica no botão salvar;
6. O sistema salva as informações da nova cidade.

Quadro 6 – Descrição do caso de uso manter localidades

**Caso de uso UC03 – Manter localidades.**

**Ator:** Administrador.

**Objetivo:** Cadastrar uma localidade de um endereço com bairro e cidade.

**Pré-condições:** Bairro e cidade cadastrados.

**Pós-condições:** O administrador cadastrou uma localidade de um ou mais imóveis e/ou contatos.



**Cenário Principal:**

1. O administrador seleciona a opção de mostrar todas as localidades;
2. O administrador seleciona a opção cadastrar nova localidade;
3. O sistema apresenta a tela de cadastro de localidade;
4. O administrador seleciona o bairro e a cidade da localidade;
5. O administrador clica no botão salvar;
6. O sistema salva as informações da nova localidade.

## Quadro 7 – Descrição do caso de uso manter logradouros

**Caso de uso UC04 – Manter logradouros.****Ator:** Administrador.**Objetivo:** Cadastrar um logradouro de um endereço com nome, localidade e cep.**Pré-condições:** Localidade cadastrada.**Pós-condições:** O administrador cadastrou um logradouro de um ou mais imóveis.**Cenário Principal:**

1. O administrador seleciona a opção de mostrar todos os logradouros;
2. O administrador seleciona a opção cadastrar novo logradouro;
3. O sistema apresenta a tela de cadastro de logradouro;
4. O administrador informa os dados do logradouro e seleciona a localidade;
5. O administrador clica no botão salvar;
6. O sistema salva as informações do novo logradouro.

## Quadro 8 – Descrição do caso de uso manter contatos

**Caso de uso UC05 – Manter contatos****Ator:** Administrador**Objetivo:** Cadastrar um contato de imóvel com nome, telefone, endereço, nº logradouro e descrição do logradouro.**Pré-condições:** Conter todos os dados do contato**Pós-condições:** O administrador cadastrou um contato de um ou mais imóveis**Cenário Principal:**

1. O administrador seleciona a opção mostrar todos os contatos;
2. O administrador seleciona a opção cadastrar novo contato;
3. O sistema apresenta a tela de cadastro de contato;
4. O administrador informa os dados do contato;
5. O administrador clica no botão salvar;
6. O sistema salva as informações do novo contato.

## Quadro 9 – Descrição do caso de uso manter imóveis

**Caso de uso UC06 – Manter imóveis.****Ator:** Administrador.**Objetivo:** Este caso de uso tem como objetivo fazer cadastro de imóvel contendo o tipo de imóvel, endereço, nº logradouro, complemento, latitude, longitude, área, valor da locação, valor de outras taxas, quantidade de quartos, quantidade de suítes, se possui garagem coberta, quantas vagas de garagem, se possui churrasqueira, se possui piscina e contato.

**Pré-condições:** Conter todos os dados referentes ao imóvel e o contato cadastrado.  
**Pós-condições:** Administrador cadastrou um imóvel.

**Cenário Principal:**

1. O administrador seleciona a opção mostrar todos os imóveis;
2. O administrador seleciona a opção cadastrar novo imóvel;
3. O sistema apresenta os tipos possíveis de imóveis para cadastro;
4. O administrador seleciona o tipo de imóvel;
5. O sistema apresenta uma tela contendo lista de características do imóvel e um campo para inserir um novo contato;
6. O administrador seleciona e informa características do imóvel;
7. O administrador clica no botão salvar;
8. O sistema salva as informações do novo imóvel..

Quadro 10 – Descrição do caso de uso manter imagens dos imóveis

**Caso de uso UC07 – Manter imagens dos imóveis.**

**Ator:** Administrador.

**Objetivo:** Cadastrar imagem de um imóvel com descrição, endereço da imagem e o imóvel.

**Pré-condições:** Imóvel estar cadastrado.

**Pós-condições:** O administrador cadastrou uma imagem de um imóvel.

**Cenário Principal:**

1. O administrador seleciona a opção de mostrar todas as imagens;
2. O administrador seleciona a opção cadastrar imagem;
3. O sistema apresenta a tela de cadastro de imagens;
4. O administrador informa os dados da imagem e seleciona o imóvel;
5. O administrador clica no botão salvar;
6. O sistema salva as informações da nova imagem do imóvel.

Quadro 11 – Descrição do caso de uso alterar situação do imóvel

**Caso de uso UC08 – Alterar situação do imóvel.**

**Ator:** Administrador.

**Objetivo:** Este caso de uso tem como objetivo alterar a situação do imóvel de disponível para indisponível quando o mesmo for locado e sair das pesquisas do usuário ou quando o imóvel locado voltar a estar disponível.

**Pré-condições:** Imóvel estar cadastrado.

**Pós-condições:** Situação do imóvel alterada.

**Cenário Principal:**

Quando o contato do imóvel avisar que um imóvel foi locado.

1. O administrador procura o imóvel no banco de dados;
2. O administrador altera a situação do imóvel de disponível para indisponível para que o cliente não enxergue o imóvel;
3. O sistema salva as informações.

Quadro 12 – Descrição do caso de uso visualizar localização atual

**Caso de uso UC09** – Visualizar localização atual.

**Ator:** Usuário.

**Objetivo:** Mostra no sistema a localização atual do usuário no mapa.

**Pré-condições:** Acesso a internet com 3G ou WI-FI.

**Pós-condições:** O sistema mostra a localização atual do usuário ou a localidade de Blumenau.

**Cenário Principal:**

1. O usuário acessa o sistema;
2. O sistema pergunta se pode usar sua localização atual;
3. O usuário responde que sim;
4. O sistema mostra a localização atual do usuário.

**Cenário Alternativo:**

No passo 3 caso o usuário responde que não...

O sistema mostra a localização de Blumenau e informa que não pode encontrar a localização atual.

Quadro 13 – Descrição do caso de uso visualizar imóveis disponíveis

**Caso de uso UC10** – Visualizar imóveis disponíveis.

**Ator:** Usuário.

**Objetivo:** Mostrar no sistema a localização dos imóveis disponíveis a locação ao redor da posição atual do usuário.

**Pré-condições:** Acesso a internet com 3G ou WI-FI.

**Pós-condições:** Sistema mostra imóveis disponíveis.

Quadro 14 – Descrição do caso de uso visualizar informações do imóvel

**Caso de uso UC11** – Visualizar informações do imóvel.

**Ator:** Usuário.

**Objetivo:** Mostra no sistema todas as informações cadastradas sobre o imóvel ao usuário.

**Pré-condições:** Acesso a internet com 3G ou WI-FI

**Pós-condições:** Sistema mostra informações do imóvel.

## APÊNDICE B – DESCRIÇÃO DO DICIONÁRIO DE DADOS

Este Apêndice apresenta a descrição das tabelas do banco de dados apresentadas na seção de especificação deste trabalho. Os tipos de dados utilizados nos atributos são:

- a) *integer*: armazena numéricos inteiros de 32 bits;
- b) *varchar*: armazena caracteres alfanuméricos até 255 caracteres;
- c) *float*: armazena caracteres alfanuméricos decimais com precisão simples;
- d) *char*: armazena caracteres alfanuméricos de tamanho fixo;
- e) *real*: armazena caracteres alfanuméricos decimais com precisão dupla.

Nos Quadros de 15 até 21 são demonstrados o dicionário de dados das tabelas.

Quadro 15 – Dicionário da tabela “bairro”

<b>Entidade:</b> Bairro		
<b>Descrição:</b> Armazena bairros do sistema		
<b>Atributo</b>	<b>Tipo</b>	<b>Descrição</b>
cd_bairro	<i>integer</i>	Chave primária
nm_bairro	<i>varchar(50)</i>	Nome do bairro

Quadro 16 – Dicionário da tabela “cidade”

<b>Entidade:</b> Cidade		
<b>Descrição:</b> Armazena cidades do sistema		
<b>Atributo</b>	<b>Tipo</b>	<b>Descrição</b>
cd_cidade	<i>integer</i>	Chave primária
nm_cidade	<i>varchar(50)</i>	Nome da cidade
sg_uf	<i>varchar(2)</i>	UF da cidade

Quadro 17 – Dicionário da tabela “localidade”

<b>Entidade:</b> Localidade		
<b>Descrição:</b> Armazena localidades do sistema		
<b>Atributo</b>	<b>Tipo</b>	<b>Descrição</b>
cd_bairro	<i>integer</i>	Chave estrangeira, entidade bairro
cd_cidade	<i>integer</i>	Chave estrangeira entidade cidade

No Quadro 18 tem-se o dicionário da tabela “logradouro”.

Quadro 18 – Dicionário da tabela “logradouro”

<b>Entidade:</b> Logradouro		
<b>Descrição:</b> Armazena logradouros do sistema		
<b>Atributo</b>	<b>Tipo</b>	<b>Descrição</b>
cd_endereco	<i>integer</i>	Chave primária
cd_bairro	<i>integer</i>	Chave estrangeira, entidade bairro
cd_cidade	<i>integer</i>	Chave estrangeira, entidade cidade
nm_logradouro	<i>varchar(50)</i>	Nome do logradouro
nr_cep	<i>char(8)</i>	CEP do logradouro

Quadro 19 – Dicionário da tabela “contato”

<b>Entidade:</b> Contato		
<b>Descrição:</b> Armazena contatos do sistema		
<b>Atributo</b>	<b>Tipo</b>	<b>Descrição</b>
cd_contato	<i>integer</i>	Chave primária
cd_endereco	<i>integer</i>	Chave estrangeira, entidade logradouro
nm_contato	<i>varchar(100)</i>	Nome do contato
nr_telefone	<i>varchar(15)</i>	Telefone do contato
nr_logradouro	<i>integer</i>	Nº do logradouro do contato
ds_complementoLogradouro	<i>varchar(10)</i>	Descrição do logradouro do contato

Quadro 20 – Dicionário da tabela “imovel”

<b>Entidade:</b> Imovel		
<b>Descrição:</b> Armazena imóveis do sistema.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descrição</b>
cd_imovel	<i>integer</i>	Chave primária
cd_endereco	<i>integer</i>	Chave estrangeira, entidade logradouro
cd_contato	<i>integer</i>	Chave estrangeira, entidade contato

tp_imovel	<i>char(1)</i>	Tipo do imóvel
nr_area	<i>float(6,2)</i>	Área do imóvel
nr_longitude	<i>real</i>	Coordenada de longitude do imóvel
nr_latitude	<i>real</i>	Coordenada de latitude do imóvel
qt_quartos	<i>integer</i>	Quantidades de quartos do imóvel
qt_suites	<i>integer</i>	Quantidade de suítes do imóvel
fl_churrasqueira	<i>char(1)</i>	Se o imóvel possui churrasqueira
fl_piscina	<i>char(1)</i>	Se o imóvel possui piscina
fl_garagemCoberta	<i>char(1)</i>	Se o imóvel possui garagem coberta
qt_vagasGaragem	<i>integer</i>	Quantidade de vagas na garagem do imóvel
vl_locacao	<i>float(6,2)</i>	Valor da locação do imóvel
vl_outrasTaxas	<i>float(6,2)</i>	Valor das outras taxas do imóvel
nr_logradouro	<i>integer</i>	Nº do logradouro do imóvel
ds_complementoLogradouro	<i>varchar(10)</i>	Descrição do logradouro do imóvel
fl_disponivel	<i>char(1)</i>	Se o imóvel está disponível para locação

Quadro 21 – Dicionário da tabela “imagem”

<b>Entidade:</b> Imagem		
<b>Descrição:</b> Armazena imagens dos imóveis do sistema		
<b>Atributo</b>	<b>Tipo</b>	<b>Descrição</b>
cd_imagem	<i>integer</i>	Chave primária
cd_imovel	<i>integer</i>	Chave estrangeira, entidade imóvel
ds_imagem	<i>varchar(50)</i>	Descrição da imagem
ed_imagem	<i>varchar(100)</i>	Endereço da imagem