

**CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**METODOLOGIA DE DESENVOLVIMENTO DE UMA API DE
SERVIÇO RESTFUL INTEGRADO COM UMA APLICAÇÃO MÓVEL
ANDROID PARA O SETOR IMOBILIÁRIO**

ALEXANDRE FERREIRA DOS SANTOS

Marília
2013

**CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**METODOLOGIA DE DESENVOLVIMENTO DE UMA API DE
SERVIÇO RESTFUL INTEGRADO COM UMA APLICAÇÃO MÓVEL
ANDROID PARA O SETOR IMOBILIÁRIO**

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Fundação de Ensino “Eurípides Soares da Rocha”, mantenedora do Centro Universitário Eurípides de Marília – UNIVEM, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof^o Ms. Ricardo Sabatine



CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

Alexandre Ferreira dos Santos

Desenvolvimento de uma aplicação móvel para o setor Imobiliário

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Sistemas de Informação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Sistemas de Informação.

Nota: 8.5 (oito e meio)

Orientador: Ricardo José Sabatine

1º. Examinador: Ildeberto de Gênova Bugatti

2º. Examinador: Fábio Dacêncio Pereira

Ricardo Sabatine

Ildeberto Bugatti

Fábio Dacêncio Pereira

Marília, 05 de dezembro de 2013.

AGRADECIMENTOS

Primeiramente Agradeço a Deus,
Por ter me dado a força necessária em todos os momentos da minha vida e sabedoria para concluir este trabalho.

Agradeço aos meus Pais Iracilda e Antônio,
Por sempre estarem ao meu lado, nos momentos mais difíceis e por apoiarem minhas decisões e por estarem sempre torcendo por mim.

Agradeço ao meu Irmão Adriano
Pela amizade e companheirismo de sempre, pelas orações feitas e motivação para o término do curso.

Agradeço a minha Tia Sônia,
Que sempre me ajudou nos momentos em que mais precisei, sempre me incentivando e dando motivação para concluir o curso.

Agradeço a minha namorada Carolina Pereira Veregue,
Que sempre esteve comigo nos momentos mais difíceis para a conclusão do trabalho, agradeço pelo apoio e pelas palavras de motivação que sempre foram importantes para o término do trabalho.

Agradeço ao meu orientador Ricardo Sabatine,
Que sempre me auxiliou para que o projeto caminhasse pelo melhor caminho, e com o seu vasto conhecimento me auxiliou para a conclusão deste trabalho.

Agradeço a Fundação Eurípides Soares da Rocha
Pela bolsa de estudo concedida para a realização deste sonho.

SANTOS, Alexandre Ferreira. **Metodologia de desenvolvimento de uma API de serviço RESTful integrado com uma aplicação móvel ANDROID para o setor imobiliário**. 2013. 61 f. Trabalho de Curso (Bacharelado em Sistemas de Informação) – Centro Universitário Eurípides de Marília, Fundação de Ensino “Eurípides Soares da Rocha”, Marília, 2013.

RESUMO

A evolução em desenvolvimento móvel e o acesso fácil à banda de internet proporcionou um grande aumento nas vendas de smartphones, o que aumentou proporcionou a geração de aplicativos para as mais diversas áreas do mercado. Com o crescimento do desenvolvimento móvel a cada dia mais o mercado oferece oportunidades para a implantação de aplicações em diferentes setores. Os aplicativos criados podem obter informações de maneira rápida e com uma consistência dos dados muito boa. Além disso, o ambiente mobile, cada dia mais cresce com várias inovações em aplicativos, unido com outras tecnologias como acesso a GPS se torna um grande aliado para o setor Imobiliário entre outros. No mercado de desenvolvimento Móvel temos grandes empresas que já estão bem à frente sobre essas tecnologias, dentre elas temos o *Android*, da *Google* e os *iOS* da *Apple*, são duas empresas que tem uma grande poder no desenvolvimento em ambientes mobile. Com isto foi desenvolvido uma API de serviço em RESTful em Java que concentra a regra de negócio para o mercado de venda e locação de empreendimentos imobiliários, integrado com uma aplicação móvel para o sistema operacional Android para clientes finais.

Palavras-chave: Desenvolvimento *Mobile*, *Android*, Setor Imobiliário e Aplicativos.

SANTOS, Alexandre Ferreira. **Metodologia de desenvolvimento de uma API de serviço RESTful integrado com uma aplicação móvel ANDROID para o setor imobiliário**. 2013. 61 f. Trabalho de Curso (Bacharelado em Sistemas de Informação) – Centro Universitário Eurípides de Marília, Fundação de Ensino “Eurípides Soares da Rocha”, Marília, 2013.

ABSTRACT

The evolution in mobile development and easy access to internet bandwidth provided a large increase in smartphone sales , which increased provided the generation of applications for many different market areas. With the growth of mobile development every day the market offers more opportunities for the deployment of applications in different industries . Applications built can get information quickly and with a consistency of very good data. Moreover, the mobile environment, increasingly grows with several innovations in applications , together with other technologies like GPS access becomes a great ally for the Real Estate sector among others . Mobile development on the market have large companies that are already well ahead on these technologies , among them we have Android, iOS Google and Apple are two companies that have a great power in the development of mobile environments . This was developed with an API service in RESTful Java that concentrates the business rule for market sale and lease of real estate projects, integrated with a mobile application for the Android operating system for end customers.

Key words: Mobile Development, Android, Real Estate Sector and Applications.

LISTA DE ILUSTRAÇÕES

Figura 1 – Arquitetura <i>Android</i>	16
Figura 2 – Mapeamento de Processos	19
Figura 3 – Mapeamento com Framework Hibernate	21
Figura 4 – Arquitetura da Aplicação	22
Figura 5 – Mapeamento da Classe Imóvel	24
Figura 6 – Satélites	25
Figura 7 – Portal das Imobiliárias.....	27
Figura 8 – Cadastro Usuário.....	28
Figura 9 – Cadastro Imóvel	28
Figura 10 – Processo de Pesquisa.....	29
Figura 11 – Diagrama de caso de uso.....	31
Figura 12 – Serviço Rest	35
Figura 13 – DER - Banco de dados do Serviço.....	36
Figura 14 - Arquitetura MVC.....	37
Figura 15 – MVC da aplicação Móvel	38
Figura 16 – Diagrama de Classes	40
Figura 17 – Diagrama Rest.....	41
Figura 18 – Diagrama de Sequência.....	42
Figura 19 – Tela de Apresentação da Aplicação	46
Figura 20 – Tela Menu Inicial	47
Figura 21 – Telas do Cadastro de Imóvel.....	48
Figura 22 – Tela de Pesquisa Imóvel Filtro.....	50
Figura 23 – Tela de Pesquisa por Mapa	51
Figura 24 – Tela Meus Imóveis.....	52

LISTA DE TABELAS

Tabela 1 – Vendas Global de Smartphones.....	10
Tabela 2 – Recursos Imóvel	43
Tabela 3 – Recursos Usuário	45
Tabela 4 – Resultados.....	53

LISTA DE ABREVIATURAS E SIGLAS

2D	Computer Graphics (Duas Dimensões)
3D	Computer Graphics (Três Dimensões)
ADT	Android Developer Kit
API	Application Programming Interface
BPM	Business Process Management
CRECI	Conselho Regional de Corretores
CSS	Cascading Style Sheets
DER	Diagrama Entidade Relacionamento
GPS	Global Position System
HTML	Hyper Text Markup Language
HTTP	Hyper Text Transfer Protocol
IDE	Integrated Development Environment
IMEI	Código de Identificação do aparelho celular
IOS	iPhone Operation System
JSON	JavaScript Object Notation
MVC	Model View Controller
ORM	Object Relational Mapping
REST	Representation State Transfer
SDK	Software Development Kit
SO	Sistema Operacional
SQL	Structured Query Language
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XHTML	Extensible Hyper Text Markup Language
XML	Extensible Markup Language
WEB	World Wide Web

SUMÁRIO

Introdução.....	10
1 Aplicação Móvel	12
1.1 Tipos de Aplicações Móveis.....	12
1.1.1 Aplicações Web.....	12
1.1.2 Aplicações Híbridas.....	13
1.1.3 Aplicações Nativas	13
1.2 Sistemas Operacionais para dispositivos móveis	14
1.2.1 Windows Phone.....	14
1.2.2 iOS.....	14
1.2.3 Android.....	15
2 Tecnologia de Desenvolvimento.....	19
2.1 Mapeamento de processo.....	19
2.2 ORM – Hibernate	20
2.3 REST	22
2.4 Geoposicionamento	24
2.4.1 Google Mapas.....	25
3 Implementação da Proposta	26
3.1 Estudo de Caso	26
3.2 Mapeamento de processos dos problemas.....	27
3.3 Descrições dos principais casos de uso	29
3.4 Desenvolvimento da aplicação	34
3.4.1 Arquitetura em layers	37
3.4.1.1 MVC.....	37
3.4.2.1 Descrição da camada da aplicação móvel	38
3.4.2.2 Descrição da camada API REST	40
3.4.2.3 Integração entre as aplicações	42
4 Resultados Obtidos.....	46
4.1 Definição dos critérios de avaliação da aplicação móvel	53

5 Conclusão	54
5.1 Trabalhos Futuros	54
Anexo 1 - Questionário de Qualidade	55
Referências	57

INTRODUÇÃO

Segundo pesquisa realizada pela consultoria Gartner o número de smartphones corresponde a 55% das vendas mundiais, trata -se da maior comercialização de telefones inteligentes já registrada, a Figura 1 ilustra bem este número de vendas de aparelhos smartphones.

Tabela 1 - Vendas Global de Smartphones

	2º trimestre 2012	3º trimestre 2013
Total	171,6 milhões	250,2 milhões
Samsung	55 milhões	80,3 milhões
Apple	24,6 milhões	30,3 milhões
Android	124,5 milhões	205 milhões
iOS	24,6 milhões	30,3 milhões

Fonte: Portal UOL, 2013.

Devido ao grande crescimento de aparelhos smartphones, o desenvolvimento de aplicações móveis ultrapassou os limites de serem apenas aplicativos para o auxílio de escritórios, como calculadora, sendo atualmente utilizados por vários segmentos.

Segundo o Conselho Regional de Corretores de Imóveis o crescimento de locação de imóveis vem crescendo gradativamente. O crescimento no setor imobiliário se destaca pela alavancagem dos negócios, por meio de facilidade em financiamentos bancários. (SÃO PAULO, 2013).

Propõe – se neste trabalho unir o crescimento de aplicações móveis ao setor imobiliário, que atualmente tem um mercado com um leque muito grande para investimentos.

Nos trabalho correlatos encontrados, atualmente para este tipo de negócios, são aplicativos que tem a capacidade apenas de fazer o processo de buscar por um empreendimento, o que se propõe é a criação de um serviço RESTful para gerenciar toda

regra de negócios obtida junto a usuários utilizando técnicas como mapeamento de processos (BPM) e UML e, uma aplicação Android que possa consumir esse serviço.

1.1 Objetivos

Desenvolvimento de um serviço aliado a uma aplicação móvel que seja útil tanto para o setor imobiliário, como para pessoas físicas que queriam efetuar o processo de divulgação de seu imóvel sem a necessidade de um intermediário, para a venda e locação do imóvel.

1.2 Organização do Documento

A monografia foi estruturada em 5 capítulos:

- No capítulo 1 é feita uma revisão bibliográfica sobre aplicações móveis.
- No capítulo 2 é feita uma análise das principais tecnologias e metodologia para o desenvolvimento móvel.
- No capítulo 3 são apresentados detalhes sobre desenvolvimento da proposta.
- No capítulo 4 são apresentados os resultados obtidos.
- No capítulo 5 é apresentada a conclusão do trabalho.

1 APLICAÇÃO MÓVEL

Segundo Fling (2009), as aplicações móveis são aplicativos construídos para funcionar em dispositivos móveis (*smartphones, tablets*) e tem como objetivo maior fornecer informações aos usuários ou grupo de clientes.

Essas aplicações podem ser tanto simples como o envio de *SMS*, ou até aplicativos robustos que com a capacidade de interação com os componentes nativos do aparelho tendo como objetivo facilitar a vida dos usuários, trazendo agilidade e facilidade.

Para obter os aplicativos contamos com duas grandes empresas que disponibilizam diversos aplicativos que são a *Google Play*, que disponibiliza aplicações para sistemas *Android* mantido pela *Google* e a *App Store da Apple* disponibiliza os aplicativos para *IOS*.

Nestas lojas existem diferentes aplicações para diversas áreas, como, por exemplo, entretenimentos, informações e até mesmo softwares que são utilizados por pequenas e grandes empresas, para vendas e guias turísticos, pode – se encontrar aplicações pagas e gratuitas.

1.1 Tipos de Aplicações Móveis

No desenvolvimento de aplicações móveis são encontrados diversos tipos de desenvolvimento de aplicações, como as que fornecem uma agilidade maior no tempo de resposta e outras que necessitam de uma capacidade maior do recurso da internet ou que tenham acesso por meio de um armazenando em nuvem, usando o conceito de cloud computing. Desta maneira, é possível dividir em 3(três) os tipos de aplicações: Web, Híbridas e Nativas. . (TOLEDO, 2012; CHARLAND, 2011).

1.1.1 Aplicações Web

Uma *web app* é projetada para operar em todas as plataformas, pois é acessível a partir de um navegador, ou seja, o aplicativo pode ser baixado da web cada vez que os usuários acessam o mesmo. Esse tipo de aplicativo é normalmente desenvolvido utilizando tecnologias como *HTML, CSS* e *Java Script*. (CHARLAND, 2011).

As vantagens de desenvolver aplicações web app são variadas, devido os *smartphones* atuais possuírem navegadores de internet próprios é possível ter acesso a vários

aplicativos de forma eficiente sem ter a necessidade de instalar nenhum tipo de aplicativo no dispositivo (CHARLAND, 2011), sem contar que não é necessário ter um conhecimento específico para cada tipo de navegador.

Porém, necessita de uso frequente da internet, pois sem a conexão não será possível acessar os servidores que hospedam as aplicações.

1.1.2 Aplicações Híbridas

É uma aplicação que combina código nativo do sistema operacional com outros códigos, como por exemplo, o *HTML*. Com o uso de *frameworks* é possível ter um único código para ser executado em vários *devices*. (CHARLAND, 2011).

Para facilitar o desenvolvimento de aplicações *híbridas* existem diversos *frameworks* que facilitam a implementação como *PhoneGap* e *Titanium*. (CHARLAND, 2011).

Embora aplicações *híbridas* mesquem código nativo com código *HTML*, as aplicações ainda têm limitações de acesso físico do aparelho ou métodos do sistema operacional, ou seja, nem tudo que foi planejado em uma aplicação pode ficar semelhante à outra, como, por exemplo, os componentes do iOS é diferente do Android.

1.1.3 Aplicações Nativas

Uma aplicação nativa é desenvolvida para rodar diretamente em um sistema operacional (SO) em *tablets* e *smartphones*. Geralmente é compilado, o que é mais rápido do que linguagens interpretadas como *Java Script*. *WebViews* e navegadores (*HTML e CSS*) para criar interfaces para o usuário com diferentes graus de capacidade e sucesso. (CHARLAND, 2011).

Devido ao fato da aplicação ser nativa, ou seja, só é executada no sistema operacional para o qual foi desenvolvida, as grande empresas disponibilizam ferramentas como linguagem de programação e *SKD* para o desenvolvimento, tais como *Java* com *Eclipse* entre outras.

Uma das grandes vantagens de se utilizar uma aplicação nativa é o fato de ser possível ter um acesso amplo aos recursos do dispositivo utilizar o *GPS*, câmera, entre outros recursos que estão disponíveis pelo sistema operacional do dispositivo, desta maneira possibilitando um retorno mais eficiente do que uma *web app*. (CHARLAND, 2011;

TOLEDO, 2012). Com a característica de trabalhar *off-line*, é possível executar aplicativos nativos mesmo em localidade sem acesso a redes móveis ou com baixo nível de sinal.

Por outro lado, pode ser considerada desvantagem o fato de ser necessário possuir conhecimento específica em tecnologias, como *Java*, *C#* e *Objective-C*.

As aplicações nativas ficam disponíveis em lojas oficiais como *Google Play* e *App Store*.

Neste trabalho iremos desenvolver uma aplicação nativa devido a necessidade de otimização do uso de recursos de hardware (GPS) e pela obrigatoriedade de estar sempre disponível para o usuário independente da existência ou não de conexão de internet.

1.2 Sistemas Operacionais para dispositivos móveis

O desenvolvimento de uma aplicação demanda tempo e recursos financeiros, com isso, torna-se necessário uma análise de vários requisitos, como por exemplo, determinar qual será a plataforma móvel alvo. Entre os principais SO (Sistemas Operacional) disponíveis, atualmente podemos destacar *Windows Phone*, *Android* e *iOS*. Como a aplicação será desenvolvida para *Android*, iremos destacar características do mesmo.

1.2.1 Windows Phone

O *Windows Phone Mobile*, sistema operacional desenvolvido pela *Microsoft* até então (2013) ocupa o 3º lugar no ranking de sistemas operacionais mobile mais vendidos no mundo, ficando atrás apenas do *Android* e *IOS*, segundo pesquisa feita pelo IDC (*Analyze the Future*, 2013)

Além de facilmente conectar seu aparelho celular a uma rede sem fio, o *Windows Mobile* permite uma adaptação mais rápida de aplicativos desenvolvidos, a princípio, para o *Windows* do seu computador (*Word*, *Outlook* etc.). (MICROSOFT, 2013).

A *Microsoft* oferece uma plataforma de desenvolvimento gratuita com bastante recurso que usa a linguagem *C#*, fazendo com que desenvolvedores da plataforma desktop tenham grande facilidade em gerar aplicativos móveis por já estarem familiarizados com o *C#*. (MICROSOFT, 2013).

1.2.2 iOS

O *IOS* é o sistema operacional móvel da *Apple*, derivado do *Mac OS X*. Lançado primeiramente para iPhone, o sistema também é compatível com outros aparelhos da empresa, como *IPOD*, *IPAD*, *IPOD Touch* e *Apple TV*. (APPLE.COM, 2013)

O *IOS* da *Apple* tem o seu kit de desenvolvimento gratuito, que conta com a *IDE Xcode*, porém, está disponível apenas para o sistema operacional *Mac OS*. (APPLE.COM, 2013)

1.2.3 Android

Apresentaremos com mais detalhes a plataforma Android por ser o foco deste trabalho.

O *Android* é a plataforma *open-source* criada pelo *Google* para o desenvolvimento de aplicações para dispositivos móveis capazes de usufruir dos diversos utilitários que um dispositivo móvel possa oferecer. (PEREIRA; SILVA, 2009).

Por ser baseado no *Linux*, o mesmo se encarrega de gerenciar a memória e os processos, permitindo que várias aplicações possam ser executadas ao mesmo tempo sem que o usuário perceba, além disso, conta com uma interface rica em recursos já instalados, o que permite um desenvolvimento ágil e poderoso. (LECHETA, 2010).

- **Arquitetura, SDKs e Linguagens para desenvolvimento**

A linguagem utilizada é o Java, recomendada para o desenvolvimento de aplicativos nativos, porém é possível escrever aplicações utilizando a linguagem C.

Para o desenvolvimento de aplicações existe disponível um kit de desenvolvimento denominado *Android SDK*, que fornece bibliotecas e ferramentas de desenvolvimento necessárias para construir, testar e depurar aplicativos para o *Android*.

É recomendada a utilização do pacote *ADT (Android Developer Tools)* aliada ao *Eclipse IDE* que inclui os componentes essenciais do *Android SDK* para agilizar o desenvolvimento de aplicativos *Android*.

Pelo fato do *Android* ser de código aberto, o desenvolvimento pode ser feito por qualquer tipo de pessoa, não fica restrito a desenvolvedores de uma única empresa, por isso é possível disponibilizar seu aplicativo ‘caseiro’, para que várias pessoas tenham acesso a ele.

Arquitetura *Android* é baseada em quatro camadas (Aplicação, Bibliotecas, *Runtime*, *Kernel*) como ilustrado na Figura 1 (LECHETA, 2010).

Figura 1 – Arquitetura *Android*

Fonte: VIVASEMFIO.COM, 2013.

- **Camada de Aplicação:** É responsável por hospedar as aplicações nativas que o dispositivo disponibiliza, como agenda, navegadores, entre outros.

Essa camada tem a capacidade de interagir com todas as aplicações nativas do aparelho, dessa forma o mesmo se torna um gerenciador de tarefas. (FREIRE, 2012)

Nesta camada se localizam todos os aplicativos que são executados sobre o sistema operacional dentre outros.

- **Camada de Biblioteca:** Tem a responsabilidade de armazenar diferentes tipos de bibliotecas, ou seja, funciona como um recipiente de bibliotecas que são utilizadas no decorrer da ação do aplicativo.

Desta maneira, as bibliotecas tem a função de gerenciar os serviços de acesso a banco de dados, entre outras funcionalidades, ou seja, tem a capacidade de fazer com que o sistema se adapte a diferentes tipos de dispositivos móveis. (FREIRE, 2012).

O mesmo possui as bibliotecas *C/C++* que são utilizadas pelo sistema e também bibliotecas de multimídia, visualização de camadas *2D* e *3D*, funções para navegadores web, funções para gráficos, fontes *bitmap*, vetorizados e funções de acesso a banco de dados *SQLite*.

- **Camada de *RUnime*:** Nesta camada encontram-se as bibliotecas nativas do *Java*.

Devido a isso, nessa camada existem dois componentes que são essenciais: o *Core Libraries*, que se utiliza da *API* do *Java* para compilar os arquivos e o *Dalvik Virtual Machine*, que tem a tarefa de executar as aplicações por meio de uma máquina virtual. (VANI; FREITAS; SCHEMBERG, 2009).

- **Camada de *Kernel Linux* :**Esta camada tem a responsabilidade de gerenciar os processos, tendo um controle sobre a memória do dispositivo e de seus processos.

Esta camada se localiza no nível mais baixo da arquitetura do *Android*. . (VANI; FREITAS; SCHEMBERG, 2009).

O núcleo do sistema operacional *Android* é derivado do kernel do Linux herdando diversas características dessa plataforma.

O *Android SDK* é o software utilizado para desenvolver aplicações no Android que tem um emulador para simular o celular, ferramentas utilitárias e uma API completa para a linguagem *Java*, com todas as classes necessárias para desenvolver as aplicações.

Também é possível plugar um celular real na porta *USB* do computador e executar os aplicativos diretamente no celular. Isto sem dúvida facilita muitos os testes em aparelhos reais e torna o desenvolvimento bem mais produtivo. O único pré - requisito é instalar o driver *USB* do celular no computador. (LECHETA, 2010).

- **Componentes Android :**Para o desenvolvimento de uma aplicação móvel em *Android*, é fundamental compreender quatro componentes básicos, adaptados de Lecheta (2010) e *Android Developer* (2013):

✓ **Provedores de Conteúdo:** responsáveis pelo gerenciamento de dados compartilhados por meio de sistemas de arquivos como internet e de bancos de dados *SQLITE* do próprio dispositivo.

✓ **Broadcast Receiver:** componente que tem a responsabilidade de gerenciar as notificações de eventos do dispositivo ou da aplicação, como por exemplo, informar que o aparelho se apresenta em um estado crítico da bateria, ou como encontrar o posicionamento por meio do GPS integrado no dispositivo.

✓ **Atividades:** representa uma tela com uma interface para o usuário, ou seja, cria a integração do usuário com a aplicação.

✓ **Serviços:** é responsável por executar as tarefas em segundo plano que tenham uma duração maior, como por exemplo, quando o usuário está ouvindo uma música.

- **Boas práticas de desenvolvimento móvel**

O *Android* roda em uma variedade de dispositivos que oferecem diferentes tamanhos de tela e densidades, capacidade de processamento e de memória.

O desenvolvimento de uma aplicação móvel depende de muitas características para que ele possa permitir que o usuário final tenha todos os recursos que a aplicação possa oferecer.

No caso da utilização de uma página Web ser requisitada por um dispositivo móvel a primeira coisa que se deve fazer é a criação de uma página totalmente responsiva, ou seja, que se adeque a tela do aparelho.

Segundo a *W3C (Consórcio World Wide Web)*, para uma boa construção de uma aplicação, que seja utilizada em um aparelho móvel deve – se utilizar tecnologias de *CSS Móvel (Folhas de estilos)*, *SVG Tiny (Gráficos adequados)* e *XHTML* para celular (Definição de subconjuntos), essas tecnologias são de grande importância para o desenvolvimento de página para aplicativos móveis. (W3C, 2013)

Para o desenvolvimento de uma aplicação nativa que irá interagir juntamente com os componentes do aparelho, é fundamental um conhecimento elevado no tipo de linguagem de programação disponível em cada sistema operacional, para que a mesma possa tirar total proveito dos recursos que o aparelho possa fornecer.

O layout da aplicação é algo que deve ser projetado com certo cuidado, por se tratar da interface que o usuário irá interagir diariamente. Deve haver layout para diferentes tipos de aparelhos (*smartphone e tablets*), como a criação de layouts com orientação vertical e horizontal, para que o usuário possa usufruir de sua aplicação de posicionamentos diferentes sem perder suas características básicas.

2. TECNOLOGIA DE DESENVOLVIMENTO

Neste capítulo será apresentado as tecnologias ou técnicas utilizadas no desenvolvimento do projeto.

2.1 Mapeamento de processo

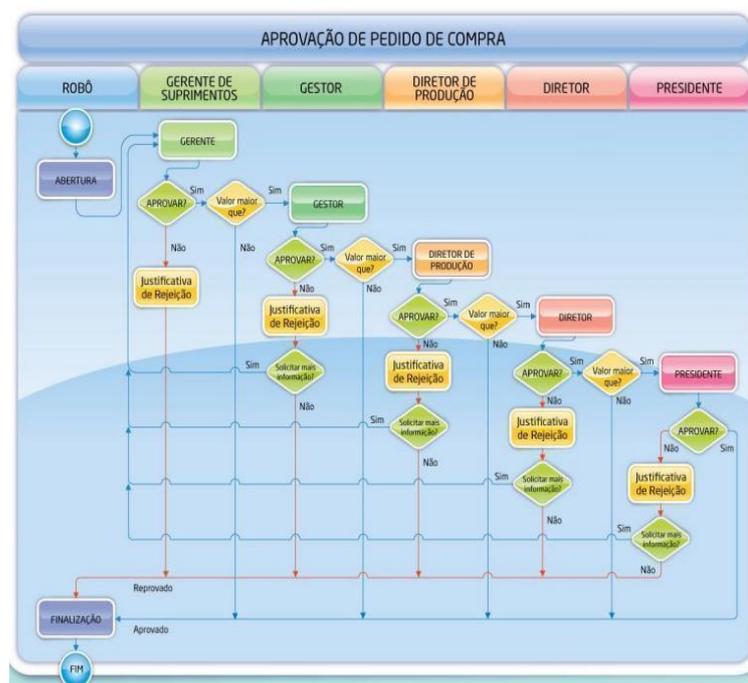
Segundo Cruz (2008) o *Business Process Management* (BPM) em português Mapeamento de Processos, é um conjunto formado por tecnologias e metodologias cujo objetivo é possibilitar que processos de negócios integrem, logica e cronologicamente, clientes e fornecedores e todo ou qualquer elemento com os quais eles possam interagir.

Desta forma, permite uma visão detalhada de todos os processos internos e externos, das operações e das atuações de cada membro no processo de negócios.

O BPM tem a capacidade de mapear graficamente todos os processos de negócios envolvidos dentro da empresa, tornando – se uma aplicação capaz de gerenciar todo um fluxo de trabalho, por meio de gráficos.

A Figura 2 ilustra um exemplo de mapeamento gráfico do fluxo de trabalho em uma empresa. Nota - se que o processo ilustrado é a realização de uma aprovação de um pedido de compra.

Figura 2 – Mapeamento de Processos



Fonte: LECOM S. A. , 2013.

Com a utilização do BPM é possível obter algumas vantagens em seus processos, como:

- ✓ Visibilidade
- ✓ Padronização de Controle
- ✓ Produtividade
- ✓ Rastreabilidade
- ✓ Documentação

2.2 ORM - Hibernate

Ferramentas para auxiliar na tarefa de manipulação manual de dados em *SQL* com *JDBC* tornaram-se populares entre os desenvolvedores Java e são conhecidas como ferramentas de mapeamento objeto-relacional (ORM).

O *Hibernate* é um Framework ORM *open source* em *Java* responsável pelo mapeamento das classes Java em tabelas de banco de dados e de tipos de dados Java para tipos de dados *SQL* (HIBERNATE, 2013). Além disso, ele oferece recursos para consulta de dados e simplifica a tarefa comum de traduzir conjuntos de resultados para objetos Java.

Objetivo do *Hibernate* é retirar do desenvolvedor 95% das tarefas de desenvolvimento relacionadas à persistência, permitindo reduzir significativamente o tempo de desenvolvimento gasto com manipulação manual de dados em *SQL* com *JDBC* (JBOSS, 2013).

O *Hibernate* não é indicado para aplicações que usam somente *stored-procedures* para implementar a lógica de negócio no banco de dados (JBOSS, 2013).

O *Hibernate* funciona no molde que outras especificações, ou seja, provê um conjunto de anotações/classes e deixa a cargo do framework o funcionamento propriamente dito. Essas anotações pertencem ao pacote *javax.persistence*.

Entre as principais anotações podemos citar (HIBERNATE, 2013):

- ✓ *@Entity*: indica que objetos dessa classe se tornem "persistíveis" no banco de dados.
- ✓ *@Id*: indica que o atributo id é nossa chave primária (você precisa ter uma chave primária em toda entidade)
- ✓ *@GeneratedValue*: indica ao banco de dados que seja utilizado o recurso de auto *increment* ou *sequence*.

Exemplo

A Figura 3 ilustra um mapeamento utilizando Anotações da classe *Imovel.java*.

Figura 3 - Mapeamento com Framework Hibernate

```

@XmlRootElement
@XmlAccessorType( XmlAccessType.FIELD )
@Entity
@Table(name = "imovel")
public class Imovel extends Model<Imovel> implements Serializable {

    /**
     *
     */
    private static final long serialVersionUID = 1L;

    @Column(name="nome", length=100)
    private String nome;

    @Column(name="visualizacao")
    private Long view;

    @Column(name="finalidade")
    private String finalidade = null;

    @OneToOne(cascade={CascadeType.ALL}, fetch = FetchType.EAGER)
    @JoinColumn(name = "enderecoid")
    private Endereco endereco;

    public Imovel(){
        super(Imovel.class);
    }

    @Override
    public void valida() throws ModelException {
        // TODO Auto-generated method stub
    }

    //getters setters
}

```

Fonte: Elaborado pelo autor, 2013.

2.3 REST

O *REST* é uma sigla que significa *Representational State Transfer*, é um modelo arquitetural (FIELDING, 2000) que define um conjunto de regras simples para implementar serviços *web*. Porém, a mesma não define tecnologias que devem ser utilizadas.

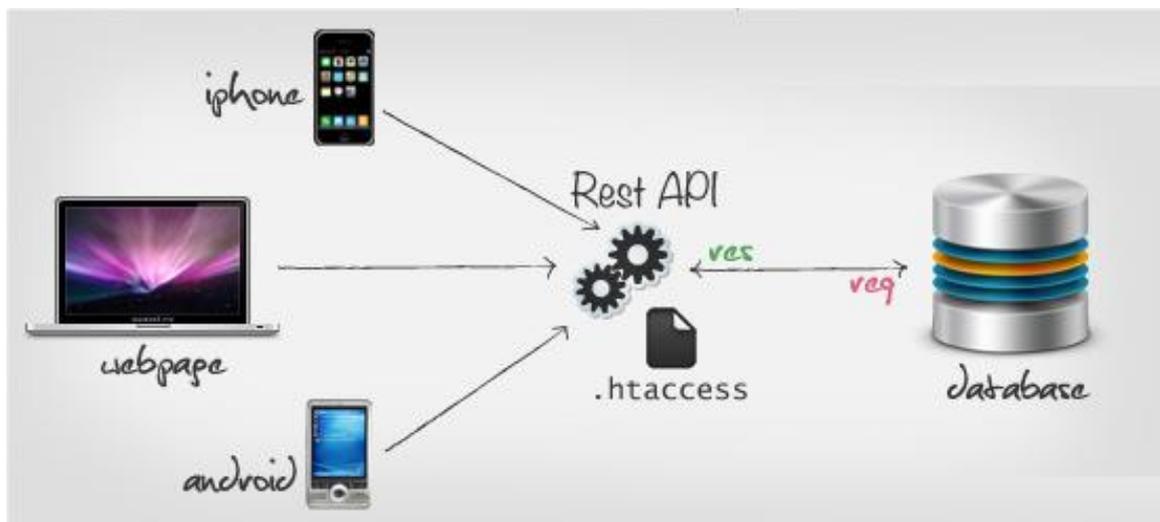
Uma forma de implementar *REST* é utilizar o conceito de *RESTful* que se baseia no protocolo da aplicação *HTTP*, no padrão de nomenclatura *URI* e na linguagem *XML/JSON*.

Podemos destacar os seguintes pontos:

- ✓ Fazer uso de diferentes métodos (ou verbos) de comunicação (*GET*, *POST*, *PUT*, *DELETE*, *HEAD*, *OPTIONS*) do *HTTP* para realizar solicitações para API;
- ✓ Utilização de *headers HTTP* (tanto padronizados quanto customizados);
- ✓ Definição de arquivos como recursos (ou seja, cada um com seu próprio endereço);
- ✓ Utilização de diferentes tipos de *media type*. Exemplo, *JSON* ou *XML* para representação de dados.

Utilizando esse conceito, os serviços se tornam disponíveis para diferentes aplicações clientes, ou seja, não há necessidade de mudança de uma aplicação para outra, pois o serviço se concentra em um servidor próprio, como ilustrado na Figura 4.

Figura 4 - Arquitetura da Aplicação



Fonte: EVOLUTION, 2013.

- ***RESTful em JAVA***

Atualmente, os serviços *REST* estão padronizados sob uma única especificação, em *Java*, chamada *JAX-RS*.

Esta especificação funciona no mesmo molde que outras especificações, ou seja, provê um conjunto de anotações/classes e deixa a cargo de vários frameworks o funcionamento propriamente dito.

Dentre os principais frameworks *JAX-RS*, é possível destacar o *framework open source Jersey*, que visa tornar mais fácil a construção de serviços *Web RESTful*.

Em geral, todo serviço *JAX-RS* precisa conter:

- ✓ Uma ou mais anotações *@Path*(que pode ser incluída tanto em classes quanto em interfaces e métodos);
- ✓ Uma ou mais anotações *@Consumes*(que segue a mesma regra de *@Path*);
- ✓ Um número qualquer de anotações *@Produces*;
- ✓ Definição dos métodos *HTTP* a serem utilizados, através das anotações *@GET*, *@POST*, *@PUT*, *@DELETE*, *@HEAD* ou *@OPTIONS*.

A Figura 5 ilustra um mapeamento utilizando Anotações da classe *Imovel.java*.

Figura 5 - Mapeamento da classe Imóvel

```

@Path("/imovel")
public class ImovelResource {

    @GET
    @Produces(MediaType.APPLICATION_JSON)
    public List<Imovel> getImovel() {
        //processa
        return null;
    }

    @Path("/{id}")
    @PUT
    @Consumes({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML})
    @Produces({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
    public Response atualizaImovel(Imovel imovel, @PathParam("id") Long id) {
        //processa
        return Response.ok().entity("Retorna Resposta").build();
    }

    @POST
    @Consumes({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML})
    @Produces({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
    public Response adicionarImovel(Imovel imovel) {
        //processa
        return Response.ok().entity("Retorna resposta").build();
    }

    @Path("/{id}")
    @DELETE
    @Produces({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
    public Response removeImovel(@PathParam("id") long id) {
        //processa
        return Response.ok().entity("Retorna resposta").build();
    }
}

```

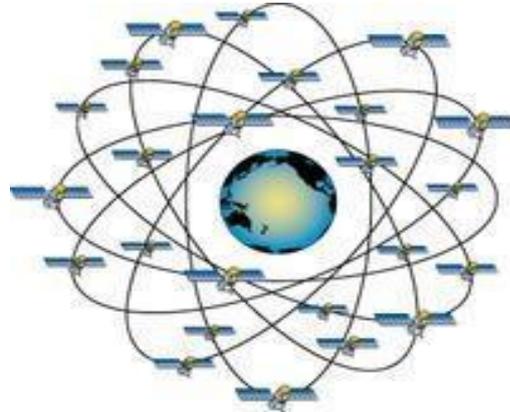
Fonte: Elaborado pelo autor, 2013.

2.4 Geoposicionamento

O GPS (*Global Position System*) vem sendo um componente utilizado em diversas aplicações móveis, o mesmo tem a capacidade de localizar pessoas por todo o mundo. (MONICO, 2000)

Para que o sistema de navegação seja atualizado rapidamente, ele necessita de informações rápidas que são captadas por 24 satélites espalhados na órbita terrestre, como pode ser visualizado na Figura 6.

Figura 6 - Satélites



Fonte: MONICO, 2000.

Seu princípio básico para a localização de diferentes coordenadas é a utilização da medida da distância entre o usuário e os satélites, desta forma sendo possível localizar aonde que o mesmo se encontra em determinado momento de sua pesquisa.

Com a utilização dos recursos de *GPS* integrados em diferentes aparelhos se transformando assim em um receptor de sinal, cada dia mais as aplicações desenvolvidas usufruem destes complementos para que suas aplicações se tornem atrativas e competitivas no mercado da tecnologia.

2.4.1 Google Maps

Para o desenvolvimento da aplicação móvel e do RESTful, será necessários uma integração com a API Google Maps, pois o mesmo terá a responsabilidade de responder as requisições feitas pela aplicação para determinar latitude e longitude dos imóveis cadastrados, ou seja fará todo mapeamentos dos imóveis cadastrados.

O *Google Maps* é uma das funcionalidades nas aplicações móveis que fazem uso do recurso do geoposicionamento, o mesmo destaca - se por interagir entre pessoas e lugares. (GOOGLE MAPS, 2013).

3. Implementação da proposta

Com base na necessidade do mercado, o desenvolvimento de uma aplicação móvel para finalidade de venda e locação pode ser de auxílio tanto para a divulgação (imobiliárias), como para aqueles que estão em busca de um empreendimento imobiliária.

Com o grande crescimento nas aplicações móveis e o setor imobiliário estar em constante aumento uma aplicação com a capacidade de interagir com o usuário na busca de um empreendimento pode ser de grande proveito para a população que busca determinado tipo de imóvel. (SÃO PAULO, 2013)

3.1 Estudo de caso

O desenvolvimento de uma aplicação imobiliária é de grande utilidade para o setor devido ao fato de estar em constante crescimento e auxiliando no processo de vendas.

As aplicações encontradas nas lojas de aplicações são distribuídas pelas próprias empresas imobiliárias, ou seja, são encontrados imóveis relacionados aos seus banco de dados internos, o mesmo não tem a funcionalidade de interagir com o usuário.

Um dos aplicativos pesquisados foi o Portal das Imobiliárias, desenvolvido por IIT Tecnologia, uma empresa licenciada para o Portal das Imobiliárias, o mesmo tem uma aplicação que é disponibilizada para diferentes imobiliárias, a única diferença é a questão dos layouts que cada empresa se adequa.

Os aplicativos criados tem a capacidade de realizar diferentes processos como:

- ✓ Filtragem: Tem a capacidade de usar filtros para a busca de imóveis conforme suas necessidades.
- ✓ Imóveis próximos: Com o GPS do aparelho ativo, aplicação tem a capacidade de localizar imóveis de um determinado raio.
- ✓ Realidade Aumentada: Com a câmera do aparelho ativada, os usuários poderão posicionar sua câmera e realizar a busca por imóveis cadastrados.
- ✓ Recente: Aplicação lista os imóveis que foram vistos pelo usuário, fazendo com que ele possa ir diretamente ao imóvel que o mesmo já selecionou.

A aplicação desenvolvida depende de uma boa conexão com a internet, pois se refere a uma *WebView*, com isso faz com que aplicação fique dependente de conexão.

As funcionalidades da aplicação podem ser visualizadas na Figura 7.

Figura 7 - Portal das Imobiliárias



Fonte: PORTAL das imobiliárias, 2013.

O acesso via Web da aplicação apresentada permite que os usuários possam cadastrar imóveis e alimentar a base de dados da imobiliária criando um vínculo com app móvel, algo que não é permitido dentro da aplicação móvel onde o usuário tem apenas acesso a pesquisa de imóveis.

3.2 Mapeamento de processo do problema

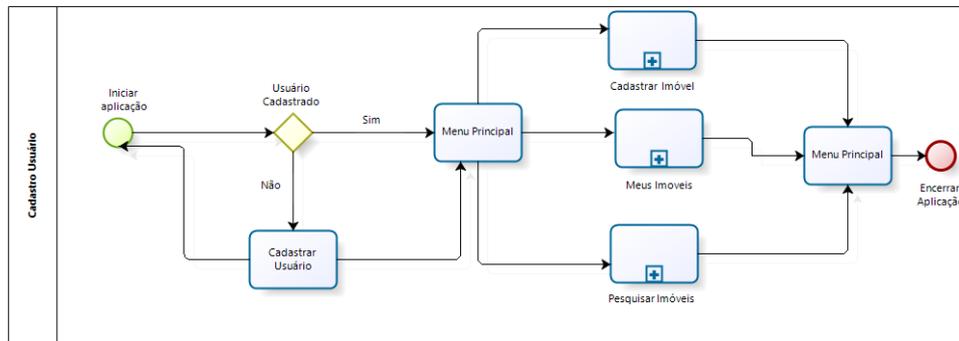
Aplicação será composta por alguns processos básicos para que o mesmo possa exercer toda funcionalidade de busca e cadastro de usuários e de imóveis.

- **Cadastro de usuário**

Ao iniciar a aplicação pela primeira vez o usuário (dono do aparelho) terá que realizar um cadastro para que o IMEI, código de identificação do celular seja gravado em um arquivo de configuração do mesmo.

Neste cadastro de usuário serão informados os seguintes campos obrigatórios para preenchimento: nome do usuário, e-mail, como ilustrado na Figura 8.

Figura 8 - Cadastro Usuário



Fonte: Elaborado pelo autor, 2013.

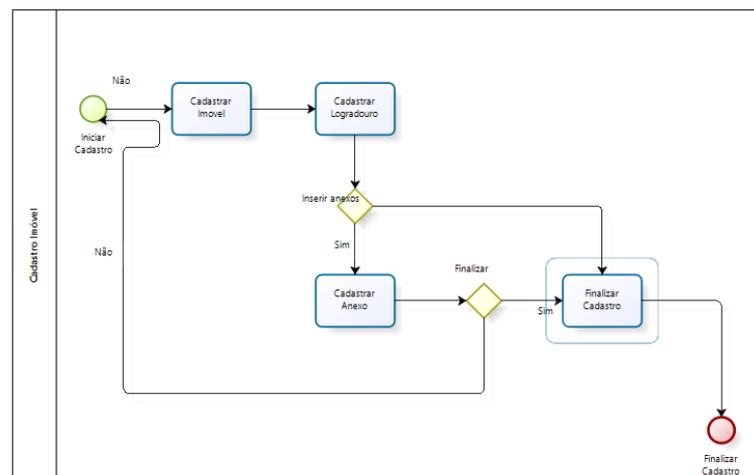
- **Cadastro de Imóvel**

O processo de cadastramento de um imóvel será praticamente semelhante ao cadastro de usuários, serão inseridas informações sobre o contato do representante do imóvel, telefone para contato, finalidade do imóvel entre outras informações adicionais.

Devido ao fato da aplicação ser disponibilizada para diversos públicos será possível anunciar imóveis de outras pessoas em outras cidades de maneira remota, precisando apenas de informações obrigatórias como nome e telefone de contato.

O mapeamento do cadastro pode ser visualizado na Figura 9.

Figura 9 - Cadastro Imóvel



Fonte: Elaborado pelo autor, 2013.

O mapeamento das tarefas de cadastro tanto de usuários como de imóveis são primordiais devido ao fato da aplicação depender totalmente dos dados inseridos no mesmo.

Para a realização de busca dos imóveis serão utilizados dois tipos de pesquisa.

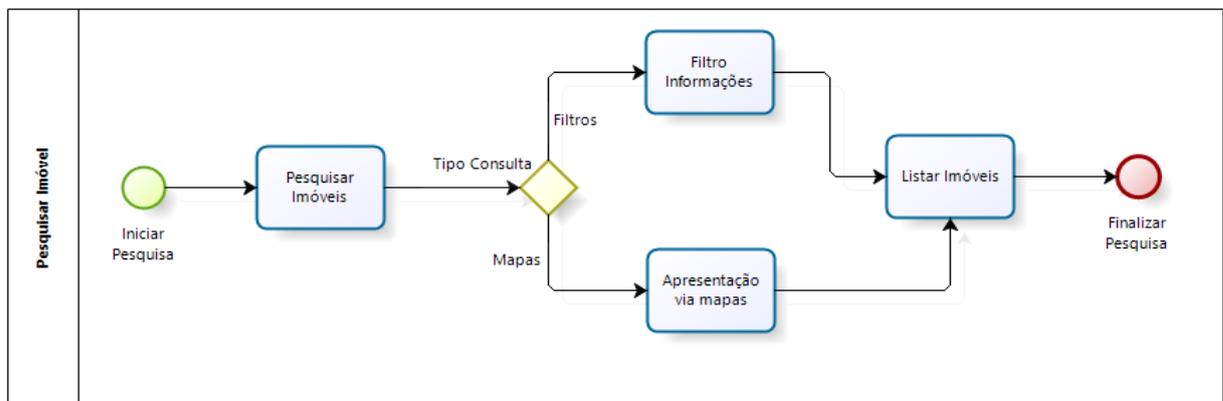
- **Pesquisa via Filtro** - Esta pesquisa será feita por meio de filtragem, ou seja, selecionar filtros que atendam critérios especificados pelos usuários.

Neste processo os usuários terão uma tela na qual será inserida alguns filtros padrões para a busca, por exemplo, buscar por valor do imóvel.

- **Pesquisa via Mapa** - Uma das grandes funcionalidades da aplicação será a busca de imóveis feita pelos Mapas. Um ponto importante da pesquisa via mapa é que será possível localizar ambientes próximos aos imóveis cadastrados.

Os mapeamentos dos processos são ilustrados na Figura 10.

Figura 10 - Processo de Pesquisa



Fonte: Elaborado pelo autor, 2013.

3.3 Descrições dos principais casos de uso

Para especificar as funcionalidades do sistema foi utilizada a linguagem unificada de modelagem (UML). Segundo Booch (2005) UML é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software.

O vocabulário UML abrange três tipos de blocos de construção: Itens (abstrações identificadas), Relacionamentos (reúnem esses itens) e diagramas (agrupam coleções de itens interessantes). (BOOCH et al., 2005).

Os diagramas representam uma visão parcial dos elementos que compõem o sistema, sendo que o mesmo elemento pode ser visualizado em vários diagramas. Entre os diagramas utilizados neste projeto podemos destacar:

Diagrama de classes: Ilustra um conjunto de classes, interfaces e colaborações e seus relacionamentos. Graficamente, um diagrama de classes é uma coleção de vértices e arcos. O diagrama é utilizado para fazer a modelagem da visão estática do projeto do sistema. (BOOCH et al., 2005).

Diagrama de casos de uso: O diagrama é utilizado para a modelagem de comportamento de aspectos dinâmicos de sistemas. O diagrama costuma conter os seguintes itens: Assunto, Casos de Uso, Atores e Relacionamentos, ou seja, o mesmo ilustra um conjunto de casos de uso e atores e seus relacionamentos. (BOOCH et al., 2005).

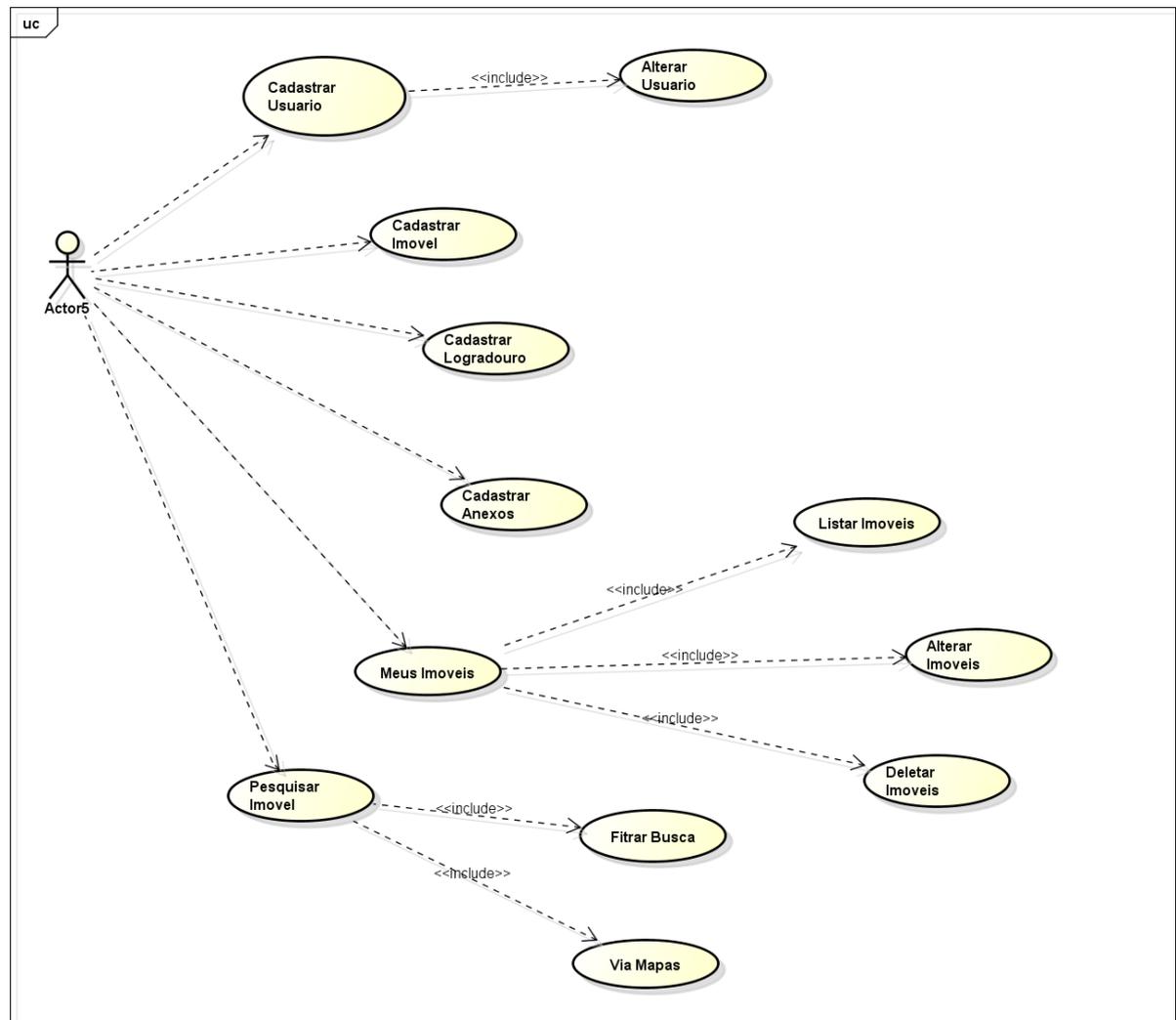
Diagrama de Pacote: O diagrama é utilizado para organizar seus elementos de modelagem em conjunto maiores que possam ser manipulados como grupos, o mesmo pode ser empregado para apresentar diferentes visões da arquitetura de seu sistema. (BOOCH et al., 2005).

Diagrama de Sequência: É um diagrama de interação que dá ênfase a ordenação temporal de mensagens. Graficamente ilustra uma tabela que mostra objetos distribuídos no eixo X e mensagens em ordem crescente no tempo, no eixo Y. (BOOCH et al., 2005).

O caso de uso de uma aplicação serve para especificar as funcionalidades a serem desenvolvidas pelos softwares a serem produzidos, isto é, descrevem um conjunto de ações que produzem algum resultado.

Para o desenvolvimento da aplicação a ser gerada foi construída o diagrama de caso de uso, o mesmo pode ser visualizado na Figura 11.

Figura 11 - Diagrama de caso de uso



powered by astah

Fonte: Elaborado pelo autor, 2013.

Descrição: Cadastrar Usuário: O caso de uso especifica a ação de realizar o primeiro cadastro do usuário com aplicação. Apenas usuários cadastrados podem realizar pesquisas e cadastros dentro da aplicação.

(a) Ator: Usuário.

(b) Pré-condições:

1. O ator deve estar cadastrado no sistema.

(c) Pós-condições:

1. O ator fica habilitado a realizar ações na aplicação.

(d) Requisitos funcionais:

1. RF28 - A aplicação deve prover uma interface para o clientes se autenticarem no mesmo.

(e) Requisitos não funcionais

1. RNF01 - A aplicação deve registrar as operações realizadas no sistema para permitir auditoria.

2. RNF02 - Impedir acesso ao cadastro de imóvel, se o usuário não estiver cadastrado.

3. RNF03 - O código do celular deve ser armazenado em um arquivo, de preferência do Sistema Operacional *Android*.

Descrição: Cadastrar Imóvel: O caso de uso especifica a ação de cadastrar diferentes tipos de imóveis. Esta funcionalidade servirá para que a aplicação seja alimentada por diferentes usuários.

(a) Ator: Usuário.

(b) Pré - condições:

1. O ator deve estar cadastrado no sistema.

(c) Pós-condições:

1. O ator fica habilitado a realizar o cadastro de vários imóveis.

(d) Requisitos funcionais:

1. RF28 - A aplicação deve prover uma interface para que o usuário possa realizar o cadastro do imóvel de maneira ágil e eficiente.

(e) Requisitos não funcionais

1. RNF01 - A aplicação deve registrar os dados inseridos para que o mesmo gere uma lista, para ser visualizada por outros usuários da aplicação.

2. RNF02 - Impedir o cadastro de imóveis inadequados para a aplicação.

Descrição: Cadastrar Logradouro: O caso de uso especifica a ação de cadastrar o logradouro do imóvel. Esta funcionalidade servirá para que a aplicação utilize as informações para direcionar o imóvel no mapa no momento da pesquisa do imóvel.

(a) Ator: Usuário.

(b) Pré - condições:

1. O ator deve estar cadastrado no sistema e ter cadastro um imóvel.

(c) Pós-condições:

1. O ator fica habilitado a realizar o cadastro do logradouro, após o preenchimento dos dados do imóvel.

(d) Requisitos funcionais:

1. RF28 - A aplicação deve prover uma interface para que o usuário possa realizar o cadastro do logradouro de maneira ágil e eficiente.

(e) Requisitos não funcionais

1. RNF01 - A aplicação deve registrar os dados inseridos para que o mesmo gere uma lista, para ser visualizada por outros usuários da aplicação.

2. RNF02 - Impedir o cadastro de logradouros inadequados para a aplicação.

Descrição: Cadastrar Anexos: O caso de uso especifica a ação do usuário ter a capacidade de anexar arquivos de imagens ou de vídeos para agregar um valor maior para o seu imóvel a ser cadastrado.

(a) Ator: Usuário.

(b) Pré - condições:

1. O usuário já deve estar cadastrado no sistema, e já deve ter realizado o cadastro do imóvel.

(c) Pós-condições:

1. O ator fica habilitado a realizar o cadastro de anexos, com imagens e vídeos.

(d) Requisitos funcionais:

1. RF28 - A aplicação deve prover uma interface para que o usuário possa realizar o cadastro de vários anexos.

(e) Requisitos não funcionais

1. RNF01 - A aplicação deve salvar os arquivos para que possam estar disponíveis para o seu cadastro.

Descrição: Meus Imóveis: Esta especificação de caso de uso irá permitir visualizar uma lista de imóveis cadastrados por ele.

(a) Ator: Usuário.

(b) Pré - Condições:

1. O usuário já deve estar cadastrado na aplicação.
2. O usuário já deve ter um imóvel cadastrado na aplicação.

(c) Pós - Condições:

1. O ator fica habilitado a realizar mudanças em seus imóveis, como alterar dados do imóvel e imagens do mesmo.

(d) Requisitos funcionais:

1. RF28 - A aplicação deve prover uma interface para o que o usuário possa realizar as alterações

(e) Requisitos não funcionais

1. RNF01 - A aplicação deve salvar os imóveis alterados na lista de imóveis.

Descrição: Pesquisar Imóveis: Esta especificação de caso de uso irá permitir ao usuário realizar pesquisas por imóveis em duas categorias: Filtro ou Mapa.

(a) Ator: Usuário.

(b) Pré - Condições:

1. O usuário já deve estar cadastrado na aplicação.

(c) Pós - Condições:

1. O ator fica habilitado a realizar pesquisa de imóveis por meio de filtros ou via mapa.

(d) Requisitos funcionais:

1. RF28 - A aplicação deve prover uma interface para o que o usuário possa realizar a pesquisa via mapa e por meio de filtragem.

(e) Requisitos não funcionais

1. RNF01 - A aplicação deve salvar seus imóveis selecionados em uma lista de favoritos.

3.4 Desenvolvimento da aplicação

O ambiente é formado por uma aplicação *Webservice* (API REST) responsável pelas regras de negócio, terá a responsabilidade de atender e processar requisições efetuadas pela aplicação móvel cliente desenvolvida.

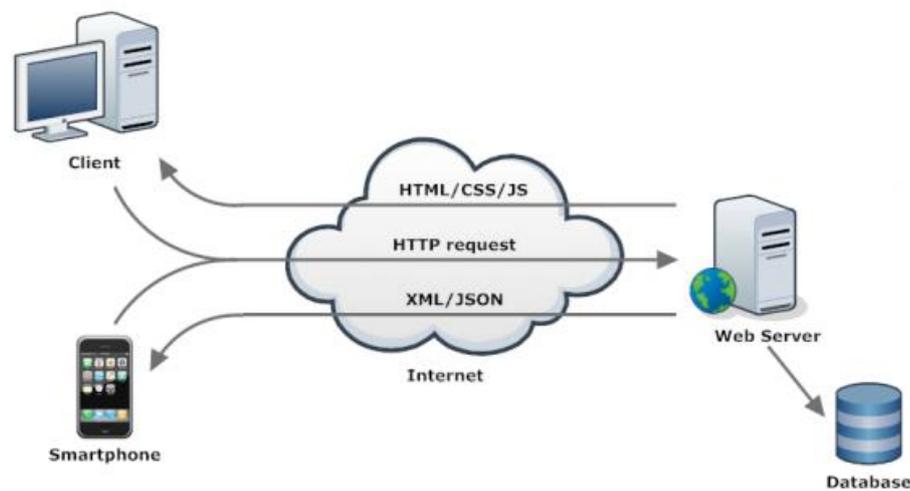
Com o desenvolvimento em camada, toda a regra fica concentrada em um determinado repositório, caso seja necessário mudar alguma regra, o mesmo será alterado

apenas uma vez, o que irá possibilitar a integração em vários outros ambientes, com uma mesma regra.

A aplicação móvel consome os serviços fornecidos pela API REST, ou seja, a parte de interface gráfica é direcionada para a plataforma *Android*. Com isso, caso futuramente queira ser implantado em outro sistema operacional a regra será a mesma.

A Figura 12 ilustra um escopo de como o trabalho de integração entre as aplicações será feita:

Figura 12 - Serviço REST



Fonte: ATZENI, 2013.

Os componentes físicos da aplicação são:

- **API REST**

É um serviço (Web Services) que gerencia as informações do sistema, ou seja, as consultas a serem realizadas passarão pela API REST que irá gerenciar toda a regra de negócios implantada na aplicação.

Esta API foi desenvolvida utilizando a linguagem de programação Java, aliada ao framework Hibernate para persistência de dados e o framework Jersey para Restful.

- **Entidade de Armazenamento (Banco de dados do Serviço)**

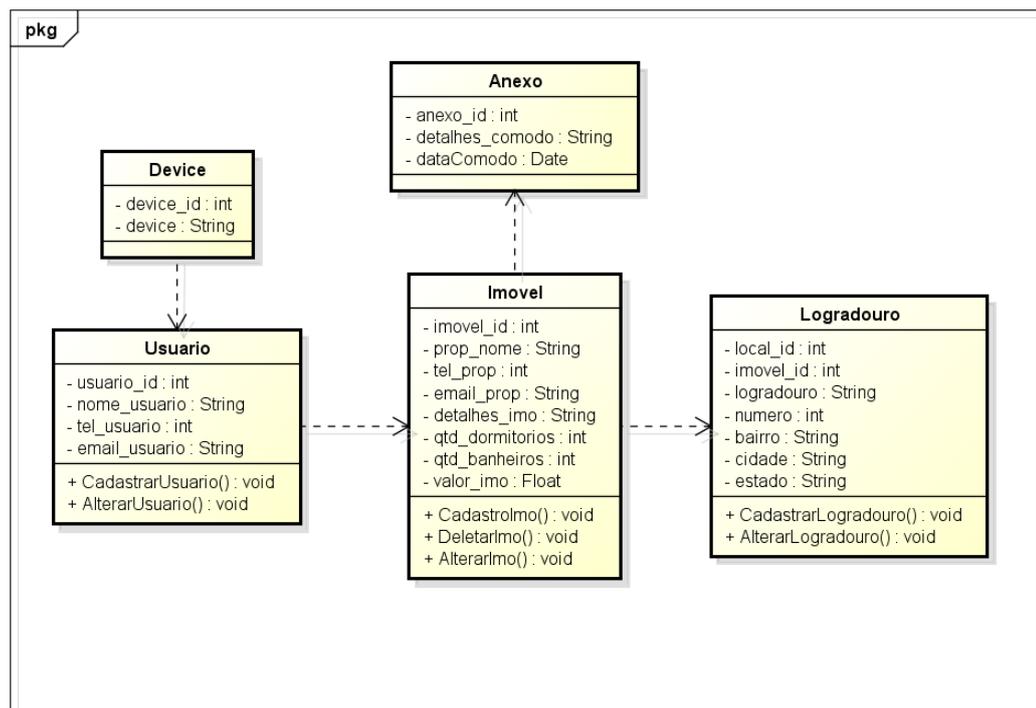
O banco de dados utilizado para aplicação será o MySQL um banco relacional e que apresenta um grande poder de manipulação e armazenamento de dados.

O banco de dados da aplicação terá acesso direto com a API REST desenvolvida.

Foi utilizado o Framework ORM Hibernate para gerenciamento das requisições no banco de dados.

Na Figura 13 ilustra o diagrama utilizado na aplicação.

Figura 13 - DER - Banco de dados do Serviço



Fonte: Elaborado pelo autor, 2013.

- **Aplicação móvel**

A aplicação foi desenvolvida com a linguagem Java, utilizando a Android SDK que fornece ferramentas de desenvolvimento necessárias para construir, testar e depurar aplicativos para o Android.

A aplicação é composta por uma lista de imóveis com duas finalidades: venda ou locação, além da capacidade de informar empreendimentos imobiliários que estão disponíveis para a comercialização, a mesma tem a capacidade de realizar os cadastros do usuário que realizará o download da aplicação, e outra funcionalidade importante é poder publicar anúncios de imóveis.

O objetivo da aplicação é fazer com que não só corretores de imóveis tenham acesso a essas informações, mas sim todo um público que esteja à procura de um empreendimento imobiliário de forma rápida, ágil e eficiente; Por meio de um aparelho *smartphone* o usuário poderá realizar diversas consultas e cadastro de diferentes tipos de imóveis e finalidades.

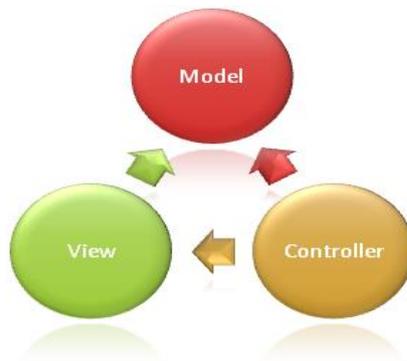
3.4.1 Arquitetura em layers

Layers se refere à camada lógica de organização dos componentes do sistema, as aplicações deste projeto foram organizadas utilizando o modelo de arquitetura MVC.

3.4.1.1 MVC

O MVC (*Model, View e Controller*) permite que a aplicação a ser desenvolvida possa ser particionada em funcionalidades, ou seja, cada processo do desenvolvimento fica separado, como pode ser visualizado na Figura 14. (GONÇALVES, 2007)

Figura 14 - Arquitetura MVC



Fonte: Fonte: GAMA, 2011.

As funcionalidades estão divididas em: *Model, View e Controller* são as atividades básicas que o MVC pode assumir em cada aplicação a ser desenvolvida.

A **View** ou as interfaces de visualização, é a camada responsável pela interação com o usuário, ela terá a responsabilidade de fazer toda a inserção e apresentação das informações fornecidas na aplicação. (GONÇALVES, 2007).

O **Model** responsável pela a manipulação e geração dos dados envolvidos na aplicação. (GONÇALVES, 2007).

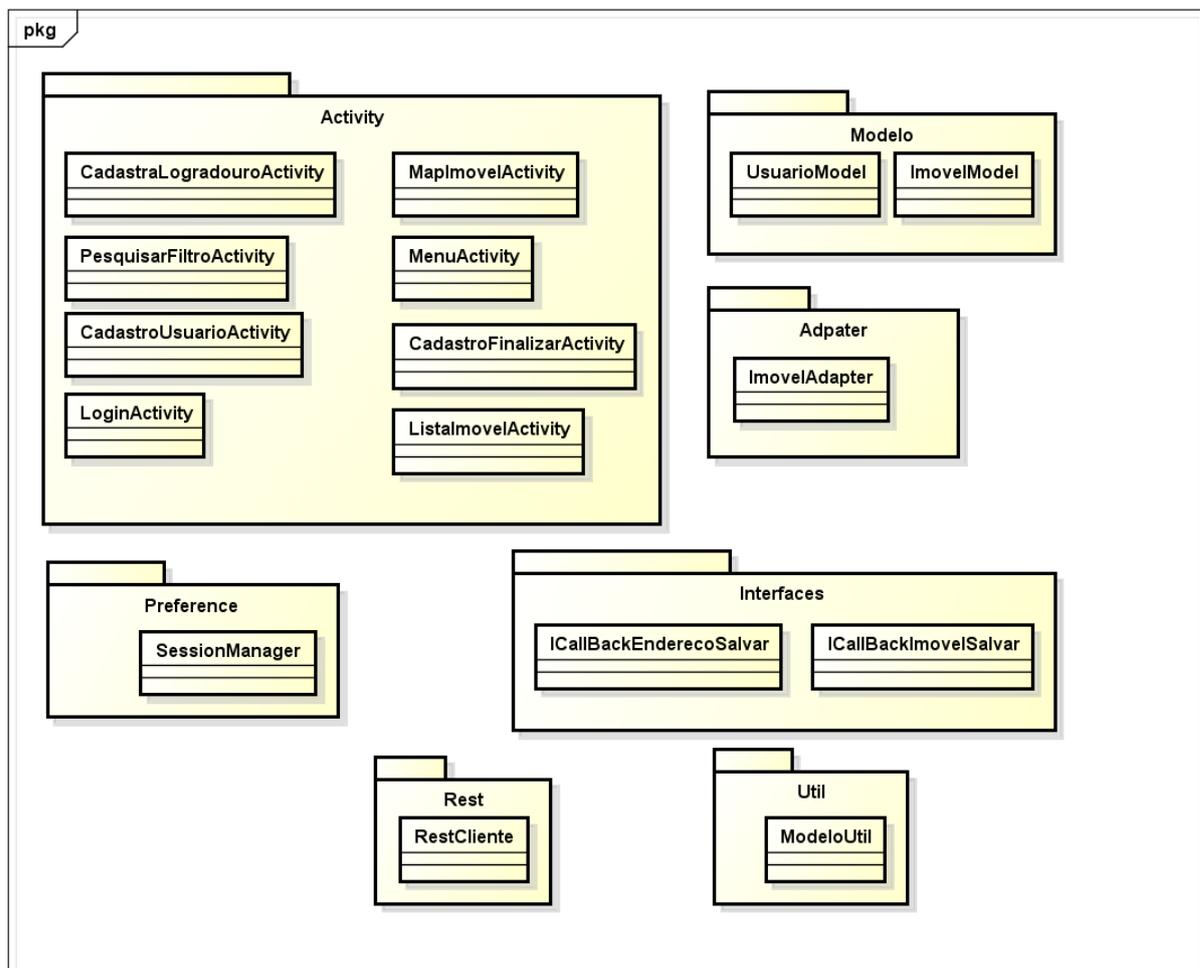
O **Controller** é a camada que irá controlar os processos que serão incluídos no fluxo de informações que serão transmitidas pela aplicação. Nesta camada é que acontecem as validações das informações, ou seja, definem toda a regra de negócios que a aplicação deve possuir. (GONÇALVES, 2007).

Com a utilização de *MVC* nas aplicações de grande complexidade, o projeto poderá ser dividido em blocos para uma diminuição de erros, gerando assim uma menor complexidade em cada tarefa a ser realizada.

3.4.2.1 Descrição das camadas da aplicação móvel

A Figura 15 ilustra a arquitetura MVC da aplicação móvel.

Figura 15 - MVC da aplicação Móvel



Pacote Activity: (Controle) O pacote activity tem o objetivo de determinar qual será a View que irá ser mostrada ao usuário e realizar o controle das interações do usuário.

Pacote Adapter: O pacote Adapter tem a responsabilidade de fornecer acesso aos itens de dados, o mesmo permite definir o modelo de uma lista dentro da aplicação.

O pacote faz a interação entre uma View e os dados a serem exibidos.

Pacote Rest: O pacote é responsável por obter uma instancia dos objetos java.net, que por si tem a funcionalidade de dar suporte aos recursos básicos do HTTP.

Pacote Interfaces: O pacote é utilizado para definir ações que devem ser executadas dentro das classes.

Pacote Util: O pacote tem funcionalidade de auxiliar no desenvolvimento e reutilização de códigos, utilizado para a criação de métodos estáticos dentro do desenvolvimento, pode ser utilizado como pacote *Helper* em outras aplicações.

Pacote Preference: O pacote tem a responsabilidade de gravar a sessão do usuário que esta logado.

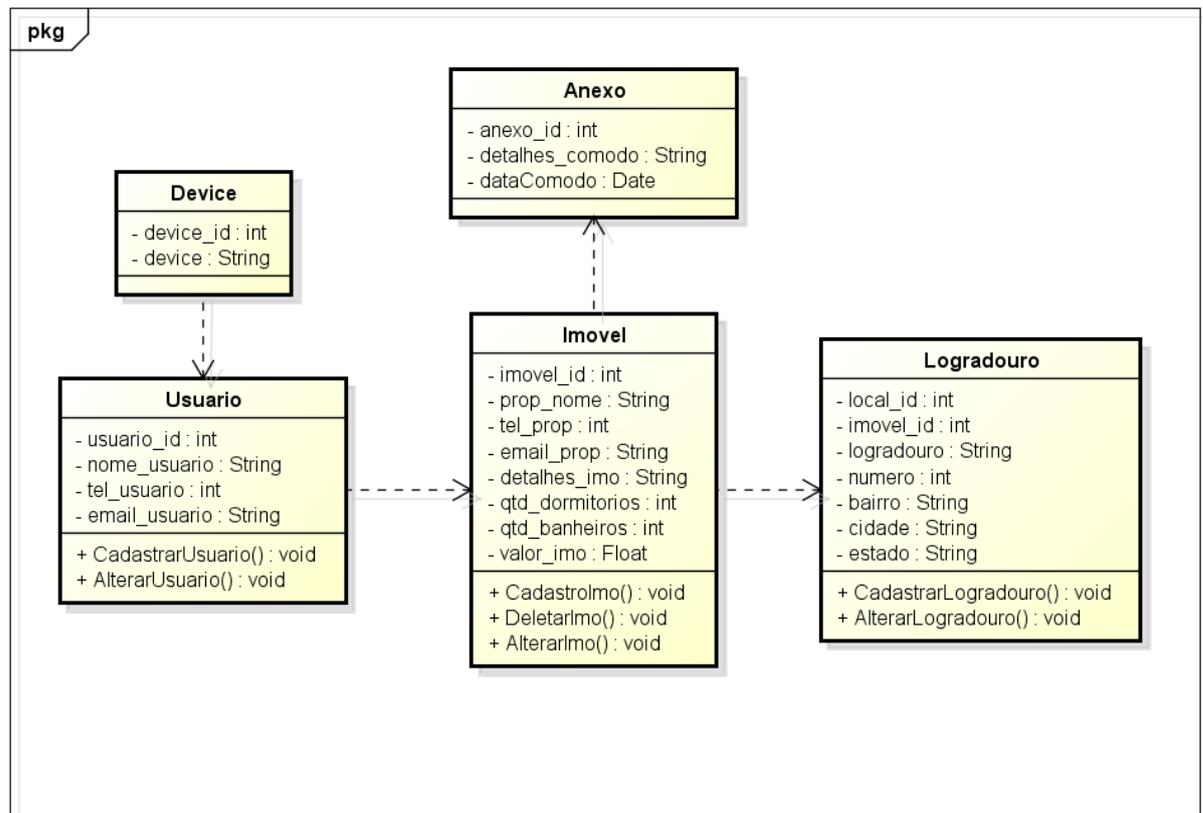
Pacote Model: O pacote é utilizado para mapear as regras de negócios.

Os modelos de negócios a serem criados são:

- ✓ Device – Irá fornecer informações sobre o número do aparelho para ser salvo em seu próprio arquivo de preferência.
- ✓ Usuários – Irá conter as informações de cada usuário.
- ✓ Imóvel – Irá fornecer informações sobre os imóveis, se os mesmos estão para a venda ou locação. Além de informar um detalhamento sobre os cômodos do imóvel.
- ✓ Localização – Irá informar o logradouro do imóvel que será pesquisado ou cadastrado.
- ✓ Anexos – Será anexada junto ao imóvel uma foto do imóvel para que possa agregar valor para o mesmo, podendo o imóvel ser visualizado.

A Figura 16 ilustra a modelagem utilizada no desenvolvimento da aplicação.

Figura 16 - Diagrama de Classes



Fonte: Elaborado pelo autor, 2013.

3.4.2.2 Descrição da camada API REST

A aplicação foi estruturada com os seguintes pacotes:

Pacote DAO: É responsável pelo acesso as informações contidas no banco de dados. Tem a responsabilidade de encontrar todas as funcionalidades no desenvolvimento de acesso e trabalho com dados em um único pacote.

Pacote Model: Classes com as regras de negócio. Além disso, representam as entidades mapeadas no banco de dados.

Pacote Exception: Classes que representam as exceções própria da aplicação. Tem a responsabilidade de tratar os erros dos modelos e da view no momento de apresentação da aplicação ao serviço.

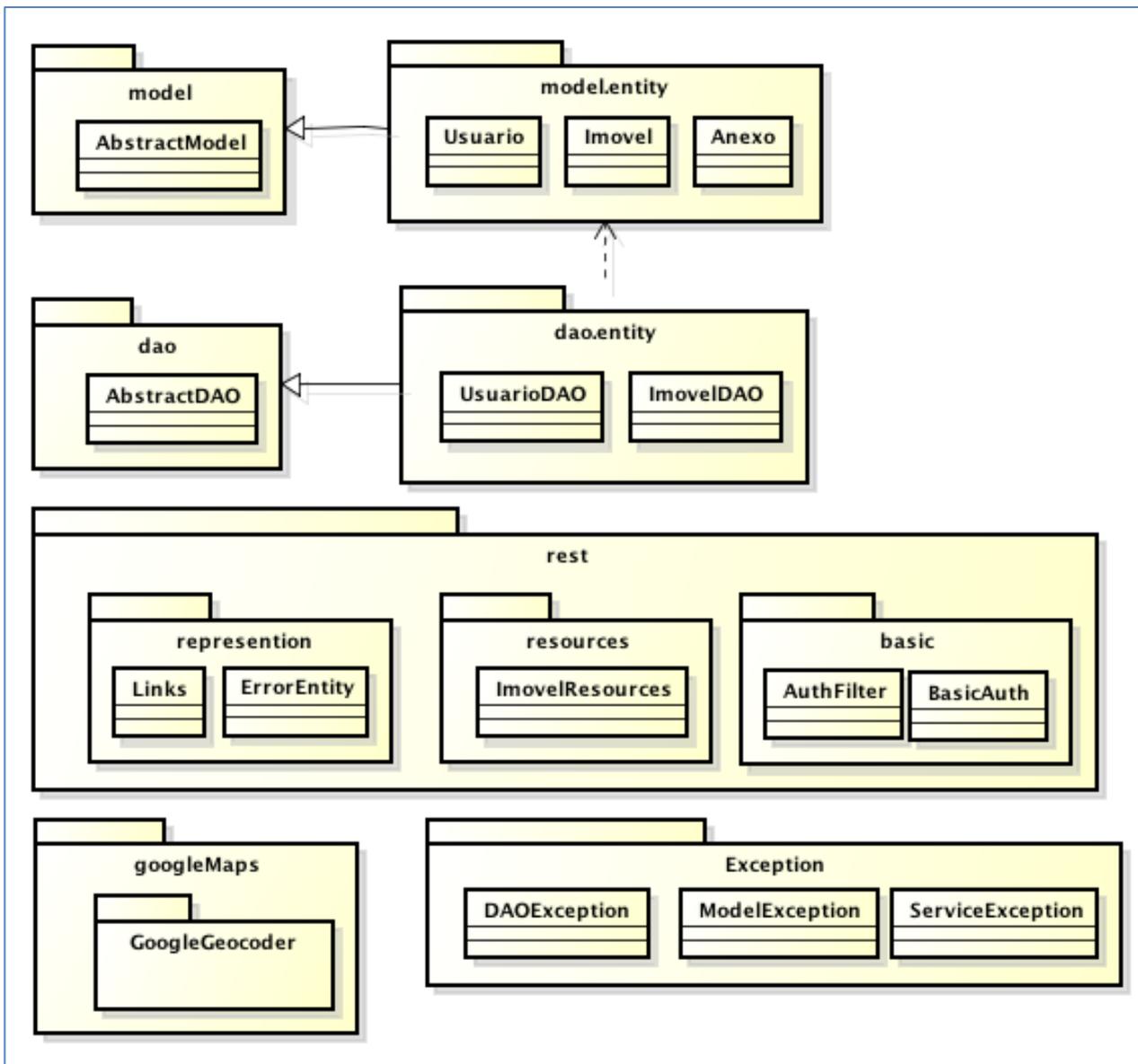
Pacote GoogleMaps: Tem a responsabilidade de obter as coordenadas dos endereços dos imóveis.

Pacote Helper: Utilizado para a criação de métodos estáticos de auxílio.

Pacote Resource: Recursos que serão disponibilizado pelo serviço.

A Figura 17 ilustra a arquitetura MVC da aplicação de integração (REST API).

Figura 17 - Diagrama Rest



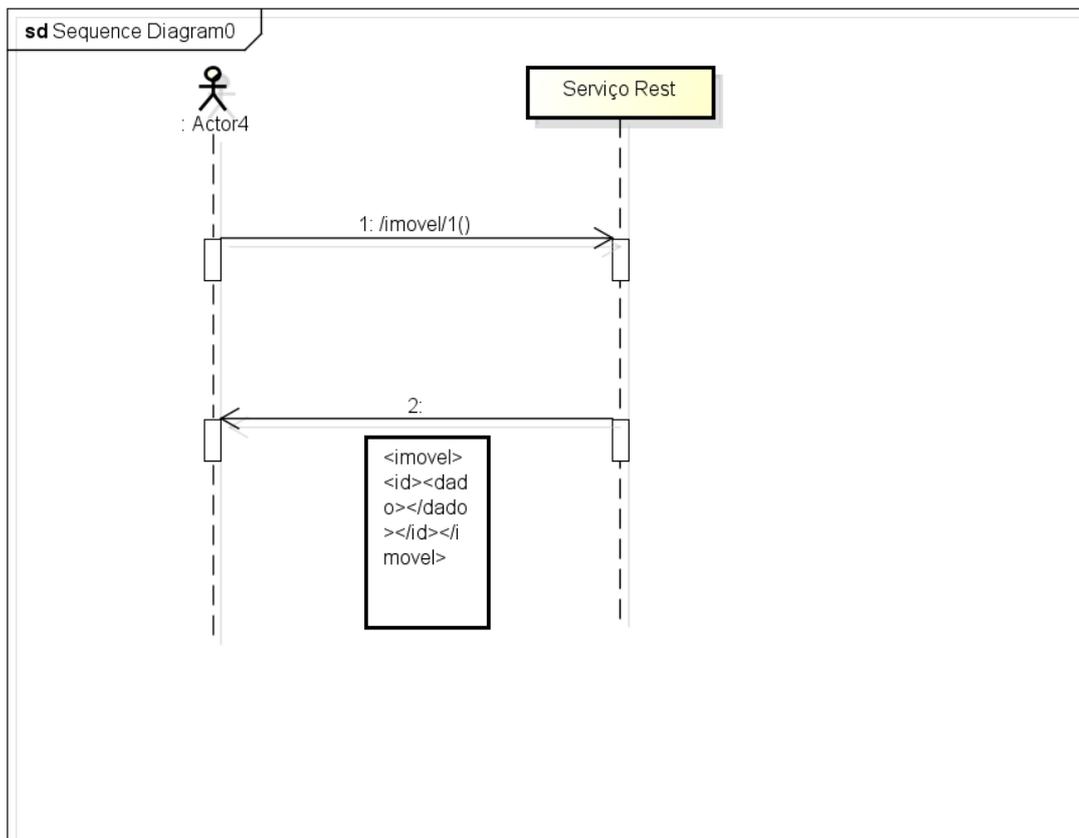
3.4.2.3 Integração entre as aplicações

A aplicação móvel é ligada com o sistema de integração (API REST) que é responsável pela gestão das informações que servirão para alimentar a aplicação móvel.

A app será composta por alguns recursos que servirão de auxílio para que o usuário tenha um melhor aproveitamento.

Uma breve visualização dos recursos de funcionamento dos métodos utilizados para o cadastro de pesquisa de do imóveis pode ser visualizado na Figura 18, onde o usuário realiza uma requisição e é retornado um JSON ou XML.

Figura 18 - Diagrama de Sequência



Fonte: Elaborado pelo autor, 2013.

Recurso Imóvel

Os métodos fornecidos terão a capacidade de realizar tanto o cadastro de um determinado imóvel como a realização de consultas de imóveis por meio de visualizações

feitas pela integração com API GOOGLE MAPS que responderá as requisições feitas pela aplicação móvel, fornecendo informações sobre latitude e longitude do imóvel cadastrado.

Tabela 2 - Recursos Imóvel

Recursos Imóvel		
URL	Recurso HTTP	Descrição
/imovel/	GET	Lista de imóveis.
/imovel/{idimovel}	GET	Seleciona um imóvel específico
/imovel/{idimovel}	PUT	Altera um imóvel específico
/imovel/{idimovel}	DELETE	Remove um imóvel específico
/imovel/{idusuario}	POST	Cadastro Imóvel de um determinado usuário.
/imovel/{idimovel}/endereco/	POST	Cadastro de localização Imóvel
/imovel/{idimovel}/anexos/	POST	Cadastro de Anexos
/imovel/busca	GET	Lista de imóveis por critérios
/imovel/coordenada	GET	Listar de imóveis por coordenadas e raio.

Fonte: Elaborado pelo autor, 2013.

- **URI:** /imovel/

Verbo HTTP: GET

Descrição: Este recurso tem a finalidade de listar todos os imóveis cadastrados no banco de dados da aplicação.
- **URI:** /imovel/{idimovel}/

Verbo HTTP: GET

Descrição: O recurso consulta apenas o imóvel selecionado, possibilitando ao usuário obter maiores informações sobre o mesmo, como por exemplo, valor do imóvel e quantidade de dormitórios.
- **URI:** /imovel/{idusuario}/

Verbo HTTP: POST

Descrição: O recurso disponibiliza ao usuário poder cadastrar diversos tipos de imóveis.
- **URL:** /imovel/{idimovel}/endereço

Verbo HTTP: POST

Descrição: O recurso é utilizado para cadastrar ou alterar o logradouro, como número do imóvel, bairro e cidade, além de outras informações que são necessárias para facilitar a procura pelo imóvel. A localização do imóvel é necessária na aplicação, pois baseado nessas informações fornecidas pelo usuário, será possível realizar as requisições para o Google Maps, retornando os ponto de latitude e longitude do imóvel.

- **URL:** /imóvel/{idimovel}/anexos

Verbo HTTP: POST

Descrição: Este recurso tem a finalidade de registrar imagens do imóvel a ser cadastrado.

- **URL:** /imovel/busca

Verbo HTTP: GET

Descrição: Este recurso tem a finalidade de buscar informações sobre imóveis que estão à venda ou disponíveis para locação, conforme sua localização no momento da busca por um empreendimento imobiliário.

- **URL:** /imovel/coordenadas

Verbo HTTP: GET

Descrição: Este recurso tem a finalidade de buscar imóveis que estão à venda ou disponíveis para locação, conforme uma localização e o raio.

Parâmetros: latitude, longitude e raio (em km).

Exemplo: /imovel/coordenadas?latitude=23.33&longitude=49.00&raio=1.0

Recurso Usuário

Os recursos dos usuários não serão extensos como visto no cadastro do Imóvel devido ao fato de que o usuário realizará apenas um cadastro simples para autenticação da aplicação, a outra funcionalidade é a de alteração do perfil pessoal.

Os recursos a serem utilizados podem ser visualizados na Tabela 3.

Tabela 3 – Recursos Usuário

Recursos Usuário		
URL	Recurso HTTP	Descrição
/usuario/	GET	Listar Usuarios
/usuario/{idusuario}	PUT	Alterar Cadastro Usuário
/usuario/	POST	Incluir um novo usuário

Fonte: Elaborado pelo autor, 2013.

A descrição completa de cada recurso está descrito da seguinte forma:

- **URL:** /usuario/

Verbo HTTP: GET

Descrição: Retorna os usuário cadastrados.
- **URL:** /usuario/{idusuario}

Verbo HTTP: PUT

Descrição: O recurso tem a finalidade de alterar as informações cadastradas no primeiro momento.
- **URL:** /usuario

Verbo HTTP: POST

Descrição: Permite incluir um novo usuário.

4. Resultados Obtidos

O resultado deste projeto foi o desenvolvimento de uma aplicação móvel nativa, devido ao fato de usufruir recurso de geoposicionamento do aparelho, ativação de recurso nativos do mesmo, e a criação de um serviço (API REST) para realizar o processo de integração entre as informações. Como resultado, será apresentado neste capítulo o uso da aplicação em imagens.

O serviço é responsável por realizar toda integração da plataforma com o ambiente de desenvolvimento da aplicação nativa no SO.

A aplicação móvel consumirá frequentemente a API REST.

No ícone da aplicação no dispositivo fica ilustrado as funcionalidades que a aplicação irá fornecer, ou seja ter a capacidade de procurar por imóveis na palma de suas mãos.

A tela da apresentação da aplicação tem apenas a característica de demonstrar que o usuário terá a capacidade de construir sua própria imobiliária online em seu dispositivo, como pode ser visualizado na Figura 19.

Figura 19 - Tela de Apresentação da Aplicação



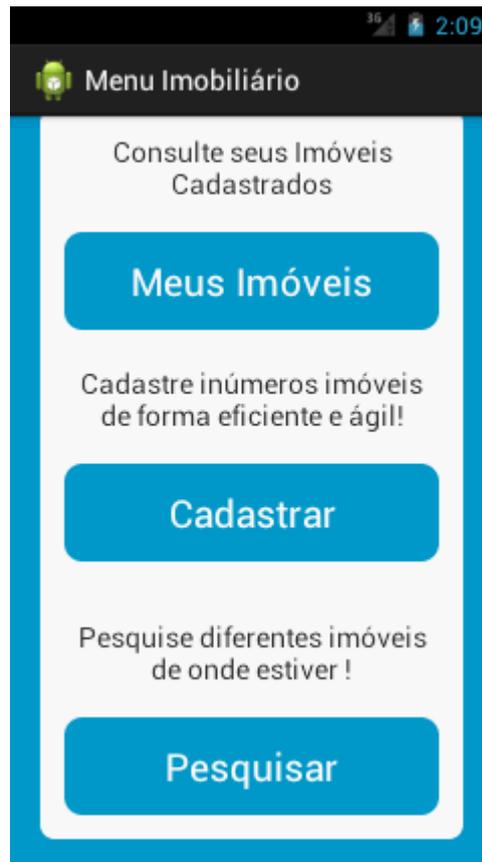
Fonte: Elaborado pelo autor, 2013.

Ao pressionar a tela do dispositivo, o usuário será encaminhado para o menu inicial da aplicação.

Neste momento o usuário terá a possibilidade de cadastrar um novo imóvel ou realizar busca por um empreendimento imobiliário.

É apresentando um pequeno trecho explicativo de cada funcionalidade inicial do menu, como pode ser visto na Figura 20.

Figura 20 - Tela Menu Inicial



Fonte: Elaborado pelo autor, 2013.

Caso o usuário queira cadastrar um imóvel, ao selecionar a opção de cadastro, o mesmo se depara com uma tela de cadastro, que segue uma ordem de dados a serem preenchidos, como pode ser visualizado na Figura 21.

Figura 21 - Telas de Cadastro de Imóvel

Cadastro Imóvel

DADOS GERAIS LOCALIZAÇÃO ANEXOS

Telefone:

Tipo Imóvel:

Finalidade:

Alugar Vender

Valor Imóvel

Dormitórios:

Banheiros:

Detalhes Imóvel:

Salvar

(A)

Cadastro Imóvel

DADOS GERAIS LOCALIZAÇÃO ANEXOS

Endereço:

Número Imóvel:

Bairro:

CEP:

Cidade:

Estado:

Salvar

(B)

Cadastro Imóvel

LOCALIZAÇÃO ANEXOS FINALIZAR

Anexar Imagens

Selecione imagens do seu próprio Smartphone para o cadastro do Imóvel !

Abrir Galeria

Tire diversas fotos para anexar ao cadastro !

Tirar Foto

(C)

Cadastro Imóvel

LOCALIZAÇÃO ANEXOS FINALIZAR

Confirmar Cadastro

Proprietário:

Telefone:

E-mail:

Tipo Imóvel:

Finalidade:

Valor Imóvel:

Dormitório(s):

Banheiro(s):

Detalhes Imóvel:

Endereço:

Número Imóvel:

Bairro:

CEP:

Cidade:

Estado:

Imagem(s):

Cadastrar

(D)

Fonte: Elaborado pelo autor, 2013.

- A. Dados Gerais - Serão registrados os dados do proprietário do imóvel, para que no caso de negociação o usuário tenha a possibilidade de entrar em contato com o proprietários do imóvel. Neste layout serão incluídas informações sobre quantidade de cômodos do imóvel e valor do mesmo, são dados de grande importância para agregar valor o seu imóvel.
- B. Cadastro de Logradouro – O layout de logradouro é importantíssimo para que os usuários possam localizar e realizar uma pesquisa melhor sobre o imóvel a ser vendido ou locado, será informado dado sobre a localização do imóvel.
- C. Cadastro de Anexos - O cadastro de anexo, servirá para agregar um valor maior para o seu imóvel o mesmo será utilizado para que o usuário possa inserir imagens de cômodos do imóvel, como ilustra a Figura 21 (C).
- D. Para o cadastro ser finalizado o usuário deve ir à aba de finalização e confirmar os dados que foram inseridos até o momento. Após a confirmação dos dados o mesmo poderá ser visualizado pelo usuário em Meus Imóveis.

Caso o usuário queira apenas visualizar os imóveis que estão cadastrados na aplicação, o mesmo irá selecionar a opção de pesquisa, como na vista na Figura 20 que ilustra a Tela Menu Inicial.

A pesquisa pelo imóvel poderá ser feita de duas maneiras como pode ser visualizado na Figura 22.

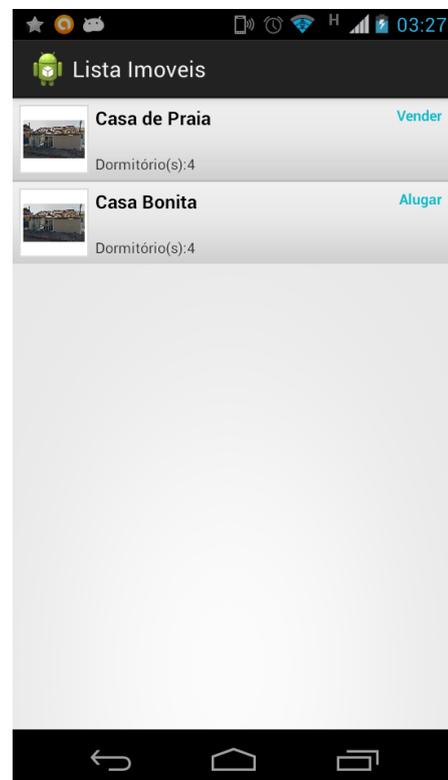
Figura 22 – Pesquisa de Imóvel Filtro



(A)



(B)



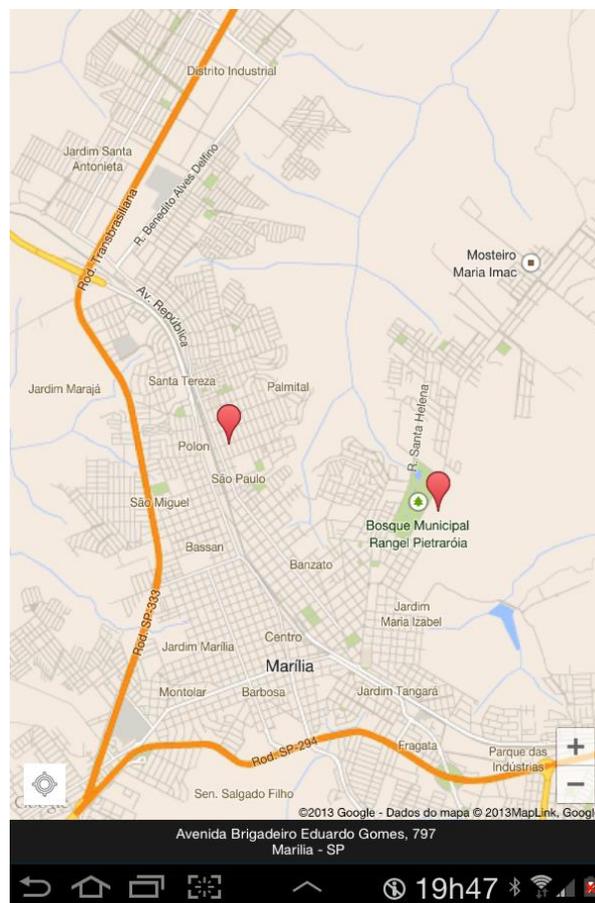
(C)

Fonte: Elaborado pelo autor, 2013.

- A. Duas opções de busca por imóveis.
- B. Filtro de Busca: O usuário irá selecionar o tipo de imóvel que ele deseja procurar, o valor correspondente ao que o mesmo queira pagar, e selecionar a quantidade de determinados cômodos do imóvel, como quantidade de quartos.
- C. Ao término da pesquisa o usuário será redirecionado a uma lista de imóveis, Na lista de consulta será exibida uma imagem do imóvel e alguns detalhes sobre o mesmo como o seu valor para venda ou locação. Caso não seja selecionado nenhuma opção no item (A) a lista de imóveis será composta por todos os imóveis cadastrados.

Quando o usuário selecionar a opção de pesquisa por “Próximos a mim”, o mesmo será direcionado a uma tela com acesso a visualização dos imóveis via mapa, como pode ser visualizado na Figura 23.

Figura 23 - Pesquisa por Mapa



Fonte: Elaborado pelo autor, 2013.

Ao selecionar a opção por “Próximos a mim”, como pode ser visualizado na Figura 24, o usuário receberá um alerta para que ele ative sua permissão de acesso, para localizar o seu ponto no momento da consulta, e assim posteriormente o mapa com os pontos de imóveis que estão a vendas serão visualizados.

Na opção “Meus Imóveis”, o usuário será direcionado para uma lista de imóveis que o mesmo cadastrou.

Figura 24 - Meus Imóveis



- A. Tela do “Menu Iniciar” com a opção de Meus Imóveis.
- B. Tela com a lista de imóveis cadastrados pelo usuário. Nesta lista o usuário pode selecionar os imóveis e realizar alterações e excluir caso não queira mais a informação.

4.1 Definições dos critérios de avaliação da aplicação móvel

Para avaliação da aplicação proposta neste trabalho, foi definido uma metodologia de avaliação de qualidade da aplicação móvel com base nos seguintes critérios: Usabilidade, Funcionalidade, Eficiência e Confiabilidade.

- Usabilidade: Conjunto de características que auxiliam o usuário na integração com aplicação móvel, como layouts atraentes e legíveis e simplicidade para a realização de todos os processos que o mesmo queira utilizar.
- Funcionalidade: Conjunto de características que permite ao usuário realizar exclusão, alteração e inclusão de informações de forma ágil e eficiente.
- Eficiência: Capacidade da aplicação em tornar seu imóvel visível para diferentes usuários.
- Confiabilidade: Conjunto de características que tornam a aplicação segura, na inserção de dados pessoais do usuário.

O conceito de avaliação dado foi de Ruim, Bom e Excelente, o mesmo foi realizado por meio de uma pesquisa de satisfação por dois tipos de usuários:

- Usuário Comum: Usuário que apenas testou aplicação para procurar um imóvel.
- Usuário do Setor Imobiliário: Com a capacidade de visualizar valores na aplicação e verificar o que ela pode auxiliar no seu trabalho.

Com base nos 4 critérios de avaliação, a pesquisa foi realizada com 4 usuários, e os resultados obtidos foram:

Tabela 4 - Resultados

	Critérios Avaliados		
	RUIM	BOM	EXCELENTE
Usabilidade	()	(X)	()
Funcionalidade	()	(X)	()
Eficiência	()	()	(X)
Confiabilidade	()	(X)	()

Fonte: Elaborado pelo autor, 2013.

5. Conclusão

O objetivo deste trabalho foi o desenvolvimento de uma aplicação para o setor imobiliário e para pessoas físicas que estejam à procura de um imóvel para locação ou venda, mas também com a capacidade de realizar cadastros avulsos de imóveis.

O desenvolvimento foi composto por uma aplicação móvel para dispositivos smartphones com o Sistema Operacional Android e um serviço de integração em (*RESTful*) entre as aplicações.

Com o desenvolvimento da aplicação e dos estudos feitos, foi possível perceber o grande crescimento entre as tecnologias móveis e o grande crescimento em desenvolvimento para aplicações móveis em smartphones e o alto volume de usuários interessados em diversos tipos de aplicações.

No desenvolvimento do serviço de integração (*REST*), foi possível perceber que grandes empresas de diferentes segmentos estão usufruindo destas tecnologias para que suas soluções estejam cada vez mais agregadas em um único ponto, e que o usuário(cliente) esteja sempre desfrutando de sua tecnologia em qualquer tipo de ambiente em que esteja utilizando.

Algo notável nas pesquisas para desenvolvimento do projeto foi o grande crescimento no mercado de aplicações móveis, em diferentes segmentos tanto de entretenimento, como gastronomia, setor Imobiliário, entre outros.

5.1 Trabalhos Futuros

Como trabalho futuro é possível destacar:

- Implementação no módulo de pesquisa: Pode – se acrescentar o recurso de realidade aumentada para o modo de pesquisa.
- Alertas Inteligentes: Quando o usuário permanecer um determinado tempo no seu cadastro, a aplicação possa notificar o proprietário do imóvel, informando que seu imóvel esta sendo acessado.
- Traçar Rota: Traçar uma rota de onde o usuário esteja até o ponto onde o imóvel esta localizado, informando o melhor caminho para o usuário que deseja conhecer o imóvel.

Anexo 1 – Questionário de Qualidade

Anexo 1 – Questionário de Qualidade de Software

Nome: _____

Telefone: (____) _____ - _____

Profissão: _____

Em relação a utilização do sistema móvel, assinale a melhor alternativa que melhor representa a sua opinião quanto o uso da aplicação imobiliária.

Usabilidade			
	RUIM	BOM	EXCELENTE
Interface	()	()	()
Legibilidade	()	()	()
Navegabilidade	()	()	()

Funcionalidade			
	RUIM	BOM	EXCELENTE
Processo de Busca	()	()	()
Cadastro de Imóvel	()	()	()

Eficiência			
	RUIM	BOM	EXCELENTE
Performance	()	()	()
Acessibilidade	()	()	()

Confiabilidade			
	RUIM	BOM	EXCELENTE
Validação dos dados	()	()	()

Confiabilidade			
	RUIM	BOM	EXCELENTE
Validação dos dados	()	()	()

Observações: _____

Assinatura

Marília, ____, Novembro de 2013.

REFERÊNCIAS

- ANDROID DEVELOPER. **Componentes de Aplicação**. Disponível em: <<http://developer.android.com/guide/components/fundamentals.html>>. Acesso em: 19 maio 2013.
- ANDROID.COM. **Developers**. Disponível em: <<http://developer.android.com/index.html>>. Acesso em: 12 maio 2013.
- APPLE.COM. **Develop of OS X**. Disponível em: <<https://developer.apple.com/technologies/mac/#xcode>>. Acesso em: 12 maio 2013.
- ATZENI, P. **REST webservice with symfony**. Disponível em: <<http://di-side.com/di-side/services/web-solutions/rest-webservice-symfony/>>. Acesso em: 10 maio 2013.
- BOOCH, G.; JACOBSON, I.; RUMBAUGH, J. **UML: guia do usuário**. 2. ed. Rio de Janeiro: Elsevier, 2006.
- CHARLAND, A.; LEROUX, Brian. Mobile application development: web vs. native. **Communications of the ACM**, New York, v. 54, n. 5, p. 49-53, May 2011. Disponível em: <<http://dl.acm.org/citation.cfm?doid=1941487.1941504>>. Acesso em: 10 maio 2013
- COMPANY WEB. **Mapeamento de Processos**. Disponível em: <<http://www.companyweb.com.br/treinamento/analista-de-negocios-processos/mapeamento-de-processos/>>. Acesso em: 10 jun. 2013.
- CONSTRUDEIA. **Imobiliária**. Disponível em: <<http://construdeia.com/imobiliaria/>>. Acesso em: 10 maio 2013.
- CRUZ, Tadeu. **BPM e BPMS: Business Process Management & Business Process Management Systems**. Rio de Janeiro: Brasport, 2008. Disponível em: <http://books.google.com.br/books?hl=pt-BR&lr=&id=9MnIpS2fI-0C&oi=fnd&pg=PA1&dq=mapeamento+de+processo+bpm&ots=Kg-Dm9Ku8P&sig=BkJHJq_rgjwKl-22kutcqdHXwk#v=onepage&q=bpm&f=false>. Acesso em: 2 Ago. 2013.
- EVOLUTION. **RESTful API in PHP for iPhone, Android and Web**. Disponível em: <<http://www.smartsell.nl/restful-api-in-php-for-iphone-android-and-web/>>. Acesso em: 10 maio 2013.
- EXAME.COM. **Imóveis valorizam 13,7% em 2012; veja onde a alta foi maior**. Disponível em: <<http://exame.abril.com.br/seu-dinheiro/imoveis/noticias/imoveis-valorizam-13-7-em-2012-veja-onde-a-alta-foi-maior>>. Acesso em: 10 maio 2013.
- FIELDING, R. T. F. **Architectural Styles and the Design of Networkbased Software Architectures**. Tese. (Doutorado) – University of California, Irvine, 2000.
- FLING, B. **Mobile Design and Development**. Sebastopol: O Reilly, 2009.
- FREIRE, F. G. J. **Análise de desempenho plataformas para desenvolvimento com Sistema**

Operacional Android. Disponível em: <<http://www.cin.ufpe.br/~tg/2012-2/jgff.pdf>>. Acesso em: 28 maio 2013.

GAMA, A. **MVC simples e prático, Parte I.** 2011. Disponível em: <<http://alexandregama.wordpress.com/2011/03/24/mvc-simples-e-pratico-parte-i/>>. Acesso em: 10 maio 2013.

GO ANDROID. **Que tal um app para o seu Android que te ajuda a parar de dormir no ônibus e não passar do ponto?** Disponível em: <<http://letsgodroid.blogspot.com.br/2012/07/que-tal-um-app-para-o-seu-android-que.html>>. Acesso em: 6 jun. 2013.

GONÇALVES, E. **Desenvolvendo aplicações web com JSP, Servlets, Javasever faces, Hibernate, EJB 3 persistence e Ajax.** Rio de Janeiro: Ciência Moderna, 2007.

GOOGLE. **Conheça o novo Google Maps.** Disponível em: <<http://www.google.com/intl/pt-BR/maps/about/explore/>>. Acesso em: 6 jun. 2013.

HAZAËL-MASSIEUX, Dominique. **Mobile Web.** Disponível em: <<http://www.w3.org/standards/webdesign/mobilweb#summary>>. Acesso em: 15 maio 2013.

IDC, **Android Pushes Past 80% Market Share While Windows Phone Shipments Leap 156.0% Year Over Year in the Third Quarter, According to IDC**, Disponível em: <<http://www.idc.com/getdoc.jsp?containerId=prUS24442013>>. Acesso em: 14 nov. 2013.

JBOSS COMMUNITY. **Hibernate Documentation.** Disponível em <<http://www.hibernate.org/docs>>. Acesso em: 6 jun. 2013.

LECHETA, R. R. **Google Android: aprenda a criar aplicações para dispositivos moveis com o Android SDK.** 2. ed. São Paulo: Novatec, 2010.

LECOM S.A. **Série: BPM e suas aplicações, parte II.** Disponível em: <<http://www.bloglecom.com.br/2013/03/20/serie-bpm-e-suas-aplicacoes-parte-ii/>>. Acesso em: 6 jun. 2013.

LOMBARDO, J. et al. **Android Application Development.** Sebastopol: O Reilly, 2009.

MICROSOFT. **Windows Phone.** Disponível em: <<http://developer.windowsphone.com/en-us>>. Acesso em: 12 maio 2013.

MONICO, J. F. G. **Posicionamento pelo NAVSTAR GPS: descrição, fundamentos e aplicações.** São Paulo: Editora Unesp, 2000. Disponível em: <<http://books.google.com.br/books?id=n3q4ypMODpEC&printsec=frontcover&hl=pt-BR#v=onepage&q&f=false>>. Acesso em: 10 maio 2013.

PEREIRA, L. C. O.; SILVA, M. L. **Android para desenvolvedores.** Rio de Janeiro: Brasport, 2009. Disponível em <http://books.google.com.br/books?hl=en&lr=lang_en|lang_pt&id=8u9wJowXfdUC&oi=fnd&pg=PA1&dq=componentes+do+android&ots=LSip0aZot3&sig=Uhg4eS-ByMj-pbWHtbruLvGU1gY#v=onepage&q=componentes%20do%20android&f=false>. Acesso em 19 maio 2013.

PORTAL das imobiliárias. Disponível em: <<http://www.portaldasimobiliarias.com.br/>>. Acesso em: 10 maio 2013.

PORTAL TERRA. **Google atualiza o site Google Maps para celulares e aproxima o site do aplicativo.** Disponível em: <<http://www.techguru.com.br/google-atualiza-o-site-google-maps-para-celulares-e-aproxima-o-site-do-aplicativo/>>. Acesso em: 10 jul. 2013.

PORTAL UOL. **Smartphones somam 55% das vendas mundiais de celulares no 3º trimestre.** Disponível em: <<http://tecnologia.uol.com.br/noticias/reuters/2013/11/14/smartphones-somam-55-das-vendas-mundiais-de-celulares-no-3-trimestre.htm>>. Acesso em: 20 maio 2013.

RICHARDSON, L.; RUBY, S. **RESTful Web Services.** Sebastopol: O Reilly, 2007.

SÃO PAULO. Conselho Regional de Corretores de Imóveis de São Paulo. **Locação de imóvel residencial cresce pelo 3º mês seguido no Estado de SP.** Disponível em: <http://www.crecisp.gov.br/arquivos/pesquisas/estadual/2013/pesquisa_estadual_julho_2013.pdf>. Acesso em: 10 maio 2013.

TOLEDO, M.J. **Desenvolvimento em Smartphones: aplicativos Nativos e Web.** Disponível em: <<http://www.cpgls.ucg.br/7mostra/Artigos/AGRARIAS%20EXATAS%20E%20DA%20TERRA/Desenvolvimento%20em%20Smartphones%20-%20Aplicativos%20Nativos%20e%20Web.pdf>>. Acesso em: 10 maio 2013.

VANI, R; FREITAS, I; SCHEMBERG, E. E. **Plataforma Android.** Disponível em <http://www.jornaltech.com.br/wp-content/uploads/2009/09/Artigo_Android.pdf>. Acesso em: 28 maio 2013.

VIVASEMFIO.COM. **Arquitetura do Android.** Disponível em: <<http://www.vivasemfio.com/blog/arquitetura-do-android/>>. Acesso em: 10 maio 2013.

WAHAB'S BLOG. **Web Service Types: SOAP/XML-RPC/Restful.** Disponível em: <<http://hellowahab.wordpress.com/2012/12/06/web-service-types-soapxml-rpcrestful/>>. Acesso em: 6 jun. 2013.