

CAMILA GONDIM QUARESMA

DESENVOLVIMENTO DE APLICATIVO WEB MARKET SYSTEM

**Assis
2011**

CAMILA GONDIM QUARESMA

DESENVOLVIMENTO DE APLICATIVO WEB MARKET SYSTEM

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: Prof. Dr. Luiz Carlos Begosso.

Área de Concentração: Informática.

Assis
2011

FICHA CATALOGRÁFICA

QUARESMA, Camila Gondim

Desenvolvimento de aplicativo Web Market System/ Camila Gondim Quaresma.
Fundação Educacional do Município de Assis – FEMA – Assis, 2011.
““p.

Orientador: Prof. Dr. Luiz Carlos Begosso.

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis –
IMESA.

1. JSF.2. Google Maps.3. Tecnologia Java.4. Android.

CDD: 001.6
Biblioteca da FEMA

DESENVOLVIMENTO DE APLICATIVO WEB MARKET SYSTEM

CAMILA GONDIM QUARESMA

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Bacharelado em Ciência da Computação, analisado pela seguinte comissão examinadora:

Orientador: Prof. Dr. Luiz Carlos Begosso.

Analizador (1):

Assis
2011

DEDICATÓRIA

Dedico este trabalho a minha mãe Geni, aos meus irmãos Igor e Diogo, e ao meu namorado Eduardo, por sempre me apoiarem, incentivarem e me amarem incondicionalmente.

AGRADECIMENTOS

*Primeiramente agradeço a **Deus**, que me guiou a todo o momento, reavivando minha fé e força. Sempre sendo meu alicerce. Sem Ele nada seria possível.*

A minha mãe Geni, mulher guerreira que sempre esteve ao meu lado me motivando e me amparando. Alguém que me faz querer lutar cada dia mais. A quem devo a minha existência.

Ao meu irmão Igor, por ser ao ser humano maravilhoso que é, por ser a pessoa na qual quero me espelhar todos os dias da minha vida, por representar a figura de pai em minha vida.

Ao meu irmão Diogo, por sempre estar presente em todos os momentos, por ser além de um irmão; ser também um grande amigo, por ser meu exemplo de honestidade.

Ao meu namorado Eduardo, por entender minha ausência nesta época difícil, por todo apoio pessoal e técnico para realização deste trabalho. Por ser minha inspiração de persistência nos momentos mais críticos.

Ao meu orientador Luiz Carlos Begosso, por sempre ter me incentivado, e não ter me deixado desistir. Por ter me orientado da melhor maneira possível.

A todos os professores do Curso de Ciência da Computação da FEMA, pelos valiosos ensinamentos passados durante a minha jornada acadêmica. Em especial para professora Marisa Atsuko Nitto, pelos valiosos conselhos.

RESUMO

Com a escassez de tempo nos dias de hoje, as pessoas estão buscando uma forma de poupar tempo em atividades rotineiras, como fazer compras. Surge então a idéia de atender uma carência do mercado local: um sistema para otimizar o tempo gasto com compras em supermercado.

Com a carência de algo parecido na região, juntamente com a necessidade de maior conhecimento de novas tecnologias para a área acadêmica, surge então a motivação para construir um aplicativo, não somente comercial e que atenda essa necessidade da cidade residente, mas que também favoreça o conhecimento e implementações de novas tecnologias.

Este aplicativo terá enfoque em um sistema de compras de supermercado via web, totalmente voltado para implementações com diversas tecnologias, das quais são: Hibernate, Android para integração com dispositivos móveis e Google Maps com a utilização de suas API's. Usando a linguagem de programação Java para web (JEE), e assim, buscar maior contribuição em inovação para a área de atuação.

Palavras Chave: JSF; Google Maps; Tecnologia Java; Android.

ABSTRACT

Nowadays, with the short time, the people are looking for a way to save time in activities of their daily routine, like shopping. Then it comes the idea of meeting a lack of local market: a system to optimize the time spent shopping in the supermarket.

With the lack of something in the region and the need for better academic knowledge about new technologies, it emerges then the motivation to build an application, not only for business that meets the needs of city residents, but also to the encouragement of knowledge and implementation of new technologies.

This application will focus on a system of a grocery shopping via web, totally focused on coding with several technologies, like: Hibernate, Android for mobile devices and integration with Google Maps with the use of their API's. It will use the Java programming language for web (JEE) and thus seek greater contribution to the innovation for the area of interaction.

Keywords: JSF, Google Maps, Java Technology, Android.

LISTA DE ILUSTRAÇÕES

Figura 1 – Arquitetura Ferramenta <i>Eclipse</i> (<i>Eclipse.org</i>).....	17
Figura 2 – Arquitetura <i>Postgree</i> (<i>DBSeller</i>)	19
Figura 3 – Arquitetura <i>Android</i> (<i>Android Developer</i>)	20
Figura 4 – Arquitetura Java J2EE (Sun J2EE)	21
Figura 5 – Arquitetura e Funcionamento JSF.....	22
Figura 6 – Funcionamento de Templates.....	23
Figura 7 – Trajeto de rota (Aplicativo Google Maps)	25
Figura 8 – Trajeto alternativo de rota (Aplicativo <i>Google Maps</i>)	26
Figura 9 – Visão Geral da Arquitetura <i>Hibernate</i> (King, 2009)	28
Figura 10 – Visão de Funcionamento do <i>Hibernate</i>	29
Figura 11 – Arquitetura geral do Aplicativo.....	30
Figura 12 – Diagrama WBS.....	32
Figura 13 – Diagrama de Sequenciamento de Atividades.....	33
Figura 14 – Diagrama de Caso de Uso.....	34
Figura 15 – Diagrama de E-R.....	45
Figura 17 – Interface de Comppra.....	47
Figura 18 – Interface de Cadastramento de Não- Cliente.....	47
Figura 19 – Interface de Carrinho de Compras.....	47
Figura 20 – Interface da Lista de Clientes.....	49
Figura 21 – Interface do Cadastro de Clientes.....	49
Figura 22 – Interface da Lista de Produtos.....	50
Figura 23 – Interface da Lista de Categorias.....	50
Figura 24 – Interface da Lista de Pedidos de Compra.....	51
Figura 25 – Interface da Lista de Pedidos de Compra Individual.....	51
Figura 26 – Interface da realização do calculo de Rota, e suas devidas coordenadas.....	52

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
ER	Entidade Relacional
HTML	<i>Hypertext Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
HTTPD	<i>Hypertext Transfer Protocol Daemon</i>
HTTPS	<i>Hypertext Transfer Protocol Safe</i>
JEE	<i>Java Enterprise Edition.</i>
JSF	<i>JavaServer Faces</i>
RF	Requisito Funcional
RNF	Requisito Não Funcional
RUP	<i>Rational Unified Process</i>
SGBD	Sistema Gerenciador de Banco de Dados
SQL	<i>Structured Query Language</i>
SSL	<i>Secure Sockets Layer</i>
UC	<i>Use Case</i>
UML	<i>Unified Modeling Language</i>
WBS	<i>Work Breakdown structure</i>
A4F	<i>Ajax for JSF</i>

SUMÁRIO

1. INTRODUÇÃO	12
1.1 Objetivos	13
1.2 Justificativa.	13
1.3 Motivação	14
1.4 Perspectiva de Contribuição	14
1.5 Metodologia de Pesquisa.....	14
1.6 Estrutura do Trabalho.....	14
2. FUNDAMENTAÇÃO TEÓRICA	16
2.1 Eclipse	16
2.2 Software Editor de UML.....	17
2.3 PostgreSQL	18
2.4 Apache TomCat.....	19
2.5 Android	19
2.6 Linguagem Java	20
2.7 Google Maps API	23
2.8 Hibernate	27
3. DESENVOLVIMENTO DO APLICATIVO	30
3.1 Descrição do Problema	30
3.2 Modelagem do Problema.....	30
3.3 Desenvolvimento de Aplicativo	31
3.3.1 Especificação.....	31
3.3.2 Implementação	47
4. CRONOGRAMA.....	53
5. CONCLUSÃO.....	54
6. REFÊRENCIAS BIBLIOGRÁFICAS	55

1. INTRODUÇÃO

O momento em que se encontra nos dias de hoje, é o de tecnologia, de facilidade e praticidade. Com o grande desenvolvimento tecnológico e com a falta de tempo cada vez maior, as pessoas estão cada dia mais procurando por algo que lhe ocuparão menos tempo e que lhe consumirão menos esforços.

Antigamente havia com maior intensidade a força dos costumes familiares que era tradicionalmente: a mulher fica em casa cuidando dos afazeres da família, enquanto seu marido iria para o trabalho conseguir o sustento. Com o grande avanço das conquistas das mulheres no mercado de trabalho, isso tem mudado significativamente.

Hoje em dia esse quadro foi alterado: homens e mulheres buscam, ao máximo, minimizar tempo gasto com rotinas domésticas, pois, na maioria dos casos, ambos estão envolvidos em suas atividades profissionais.

Com isso, surgiu a ideia do desenvolvimento de um sistema que pudesse integrar praticidade, comodidade e o mais importante uma grande otimização do tempo. Nesse sentido, será necessário somente um computador com acesso à internet para que as pessoas possam efetuar suas compras sem a necessidade de sair de casa. Teremos a comodidade e segurança de quando for entregue suas compras, o cliente poderá receber em seu celular a mensagem que sua compra foi entregue corretamente, com data e horário. Dessa forma, o mercado terá uma grande otimização no seu serviço de entrega, melhorando assim, seu tempo e custo de entrega das compras.

A ideia desse sistema é que suas compras possam ser feitas diretamente do conforto da sua casa. Com o intuito de não passar por situações incômodas, como grande maioria acaba tendo que enfrentar para fazer suas compras: trânsito para chegar ao mercado, falta de vagas no estacionamento, filas e mais filas, dificuldades em achar a gôndola que se encontra o produto que o consumidor procura, entre outros incômodos.

1.1 OBJETIVOS

Este trabalho tem por objetivo desenvolver a análise, o projeto e a implementação de um aplicativo *web* de supermercado, denominado de Web Market System, que possibilita que sejam efetuadas todas as transações de compra, pagamento e entrega de produtos através da internet.

O sistema oferece a funcionalidade de comunicação com cliente através de um dispositivo móvel. Oferece também, a funcionalidade de cálculo de rota, utilizando diretamente a API do aplicativo já existente, o Google Maps. Com enfoque nas seguintes tecnologias:

- JSF - *Java Server Faces*,
- Google Maps,
- Hibernate, e
- Android.

Será utilizada a linguagem de programação Java para web (JEE) e serão construídos três módulos principais: o módulo do cliente ou administrador, o módulo de entrega de mercadorias e o módulo de comunicação com o cliente.

1.2 JUSTIFICATIVA

Com o desenvolvimento do Web Market System será possível proporcionar aos clientes uma economia de tempo, pois toda a compra poderá ser realizada via *web*, e ainda receberão mensagem de texto, via celular, com o horário que suas compras foram entregues.

Para o supermercado, o sistema proporcionará um maior controle nas entregas das mercadorias. Os caminhões de entrega de compras terão suas rotas programadas para cada remessa de entrega. Com isso, o supermercado ganhará mais tempo, agilizará as entregas e economizará combustível.

1.3 MOTIVAÇÃO

Adquirir conhecimento na área de desenvolvimento, para facilitar uma situação que ainda não foi implementada na cidade residente e região. Perspectiva de atuação no segmento de *java web*, já que a maior parte para o mercado de trabalho hoje, se encontra em desenvolvimento *web*. Estudo de funcionalidade do Android, iteração com dispositivo móvel, o uso de Hibernate, e o manuseio dos métodos do Google Maps. Oportunidades profissionais promissoras no futuro.

1.4 PERSPECTIVA DE CONTRIBUIÇÃO

Este trabalho irá contribuir tanto para a área educacional como profissional. No campo educacional servirá como estudo das implementações de tecnologias na programação Java, enquanto que para o setor profissional servirá para futuros negócios, já que na cidade de Assis e região, nenhum supermercado utiliza esse tipo de aplicação.

1.5 METODOLOGIAS DE PESQUISA

A metodologia de desenvolvimento aplicada neste trabalho será a de pesquisa em bibliotecas, de trabalhos já publicados, livros, apostilas, vídeo-aula, tutoriais, uso de métodos do Google Maps e orientação com professores.

1.6 ESTRUTURA DO TRABALHO

O presente trabalho está dividido em 5 capítulos. O primeiro, essa Introdução, estabelece os objetivos, motivação, justificativa, perspectivas de contribuição e a metodologia para o seu desenvolvimento. O capítulo 2, estabelece a fundamentação teórica, apresentando todo conteúdo das tecnologias e ferramentas utilizadas para desenvolver o aplicativo. O capítulo 3, é formado pela modelagem do problema,

estabelece os requisitos iniciais, especificação contendo seus devidos diagramas e descrição de caso de uso. O capítulo 4, ilustra o cronograma da realização deste trabalho. O capítulo 5 , estabelece a conclusão do mesmo.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 ECLIPSE

O Eclipse é uma ferramenta de desenvolvimento baseada em Java que integra inúmeras outras ferramentas propícias ao desenvolvimento de aplicações corporativas. Possui facilidades que vão desde a rápida visualização de todos os arquivos contidos no projeto até ferramentas de gerenciamento de trabalho coletivo.

- Um *Class browser* ou *Package Explorer*, que permite visualizar toda a estrutura de diretórios e arquivos contidos no projeto
- *Wizards*, que são "programas" que auxiliam na construção rápida de aplicações com características iniciais já conhecidas. (Campos;Leal; Borba e Henrique,2008).

De modo geral pode-se dizer que o Eclipse é uma plataforma open-source desenvolvida para construir ambientes integrados que podem ser usados para o desenvolvimento de aplicações de diversos tipos, como por exemplo web sites, programas C++, java e J2EE. Foi criada pela OTI (Object Technology International Inc.) e IBM.É uma ferramenta genérica e com arquitetura aberta e extensível baseada em plug-ins.

A Figura 1 permite-nos uma visão geral da ferramenta:

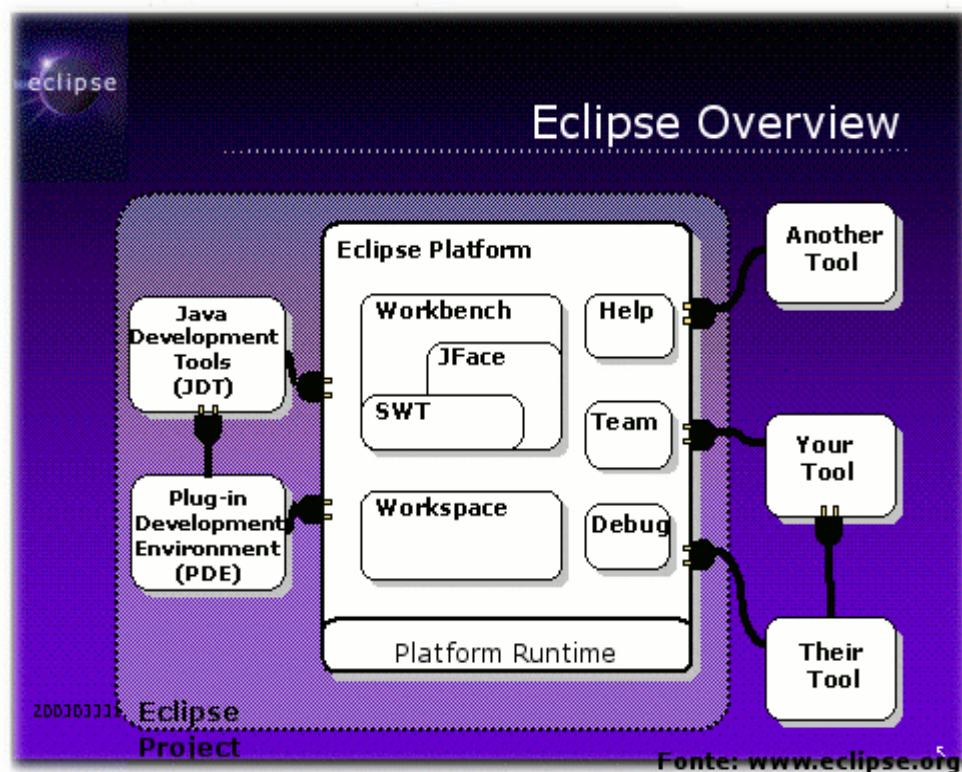


Figura 1 – Arquitetura Ferramenta Eclipse (Extraído eclipse.org)

2.2 SOFTWARE EDITOR DE UML – *Unified Modeling Language*

UML é uma linguagem visual para especificação (modelagem) de sistemas orientados a objeto. A UML privilegia a descrição de um sistema seguindo três perspectivas:

1. Os diagramas de classes - (Dados estruturais);
2. Os diagramas de casos de uso (Operações funcionais);
3. Os diagramas de sequência, atividades e transição de Estados (Eventos temporais).

Os casos de uso de um projeto de *software* são descritos na linguagem UML através de Diagramas de Casos de Uso (Use Case). Diagrama de "Use Case": É um diagrama usado para se identificar como o sistema se comporta em várias situações que podem ocorrer durante sua operação. Descrevem o sistema, seu ambiente e a

relação entre os dois. Os componentes deste diagrama são os atores, os "Use Case" e os relacionamentos. Casos de uso e Relacionamentos (Macoratti, 2011).

Para este trabalho, será utilizada a ferramenta Argo/UML, este inclui os mesmos recursos de edição e geração de códigos encontrados em ferramentas CASE comerciais, mas foca sua atenção naqueles que melhoram a utilização e suporte de necessidades cognitivas de designers. Usa formatos baseados em XML como PGML e XMI e tem suporte a OCL, opção de repositório SQL e diagramas de Deployment e Collaboration.

2.3 POSTGRESQL

O PostgreSQL é um SGBD (Sistema Gerenciador de Bancos de Dados) relacional e orientado a objetos. Oferece mecanismos eficientes de segurança e integridade de dados, além de suportar quase todas construções SQL. É um software de livre distribuição e, em termos de recursos, pode ser comparado aos melhores bancos de dados comerciais existentes. (Niederauer, 2001).

A Figura 2 é apresentada a arquitetura da ferramenta, contendo como é realizado seu funcionamento.

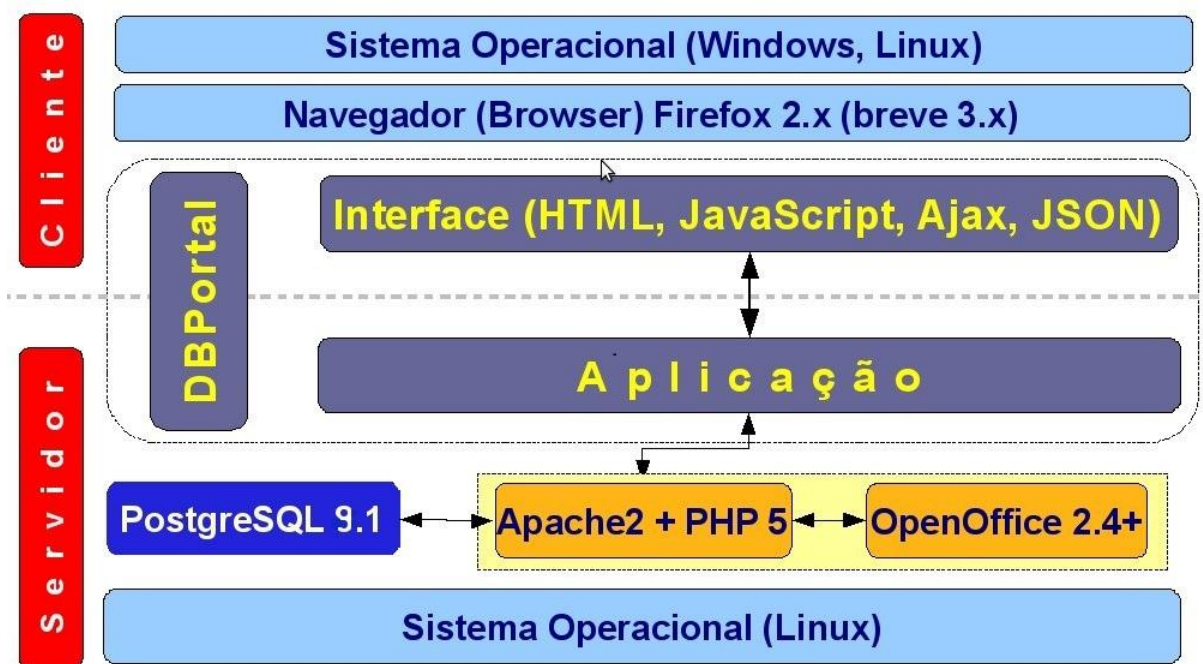


Figura 2 - Arquitetura Postgree (Extraído de DBSeller)

2.4 APACHE TOMCAT

Tecnicamente, o Tomcat é um container Web, parte da plataforma corporativa *Java Enterprise Edition* (J2EE ou Java EE) que abrange as tecnologias *Servlet* e *JSP*, incluindo tecnologias de apoio relacionadas como *Realms* e segurança, *JNDI Resources* e *JDBC DataSources*. O Tomcat tem a capacidade de atuar também como servidor *web/HTTP* autônomo, ou pode funcionar integrado a um servidor *web* dedicado, como o Apache *httpd* ou o Microsoft *IIS*, provendo a parte dinâmica de *Java Servlet* e *JavaServer Pages* (JSP) (Ávila,2003).

2.5 ANDROID

O Android é a plataforma open-source criada pelo Google para o desenvolvimento de aplicações para dispositivos móveis. Inclui um sistema operacional baseado no Linux e diversas aplicações, com uma rica interface gráfica, um browser para navegar na internet, integração com o Google Maps, suporte a multimídia, GPS,

banco de dados integrado, jogos em 3D e mais. A plataforma de desenvolvimento do Android permite desenvolver e integrar aplicações de forma simplificada utilizando a linguagem de programação Java e um ambiente de desenvolvimento de alto nível e produtividade como o Eclipse (Lecheta, 2010). A Figura 3 representa a arquitetura geral da tecnologia Android.

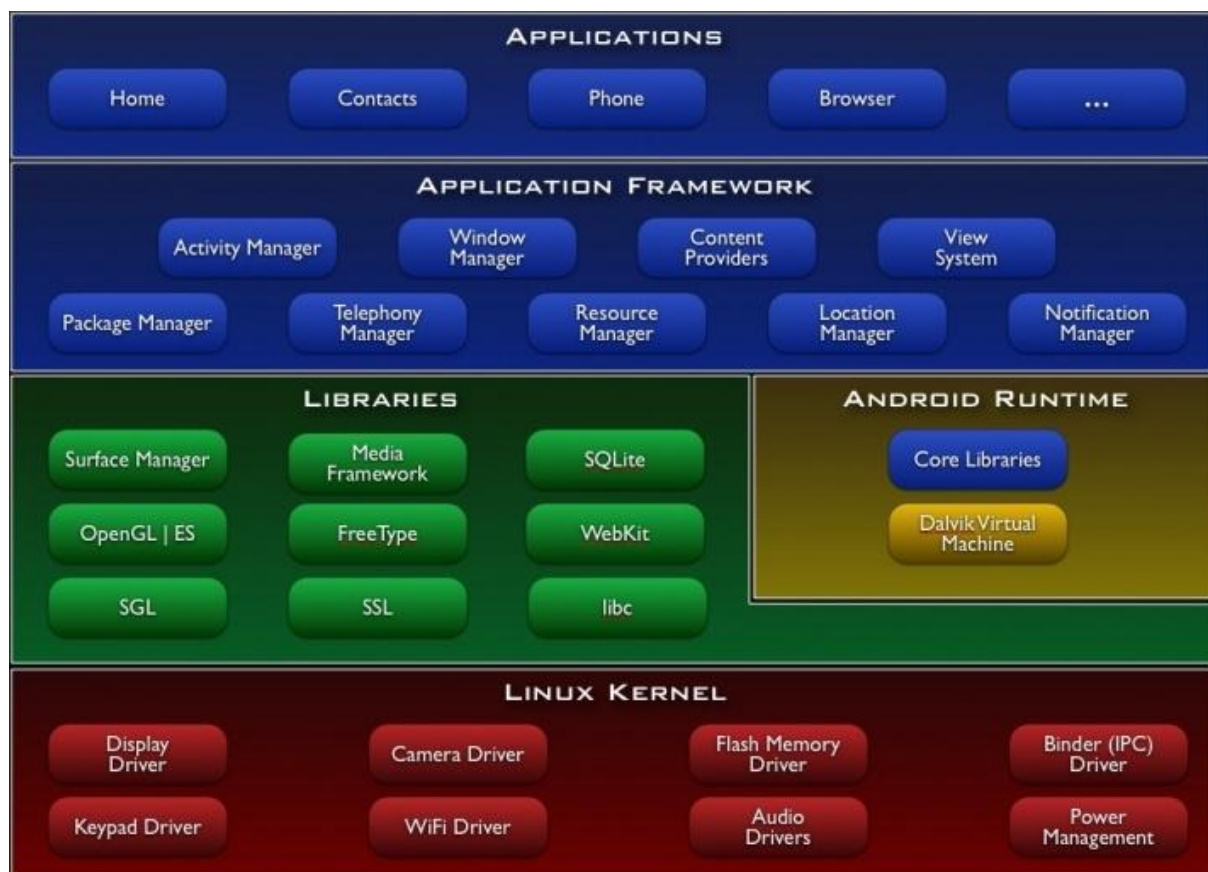


Figura 3 - Arquitetura Android (Extraído de Android Developers)

2.6 LINGUAGEM JAVA

A tecnologia Java está distribuída em três principais plataformas de aplicações para o desenvolvimento de softwares. Elas podem se comunicar através dos recursos da linguagem para que uma mesma aplicação ofereça seus recursos a diversos clientes (SUN, 2006b).

Plataformas existentes:

- Micro Edition - voltada para dispositivos móveis (celulares e demais dispositivos eletrônicos);
- *Standart Edition* - utilizada no desenvolvimento de aplicações *desktop*;
- *Enterprise Edition* - empregada no desenvolvimento de aplicações que são executadas em servidores.

Quando se efetuam comandos ou requisições ao servidor, através de *applets* ou páginas JSP, essas requisições são enviadas via protocolos HTTP ou HTTPS (SSL) para *containers*, que podem ser tanto o *web* container que transforma código JSP em *servlets* ou para o *container* EJB.

Tem-se na Figura 4, a ilustração da arquitetura Java J2EE baseando-se na J2SE e utilizando-se de recursos como as *Java Server Pages* (JSP).

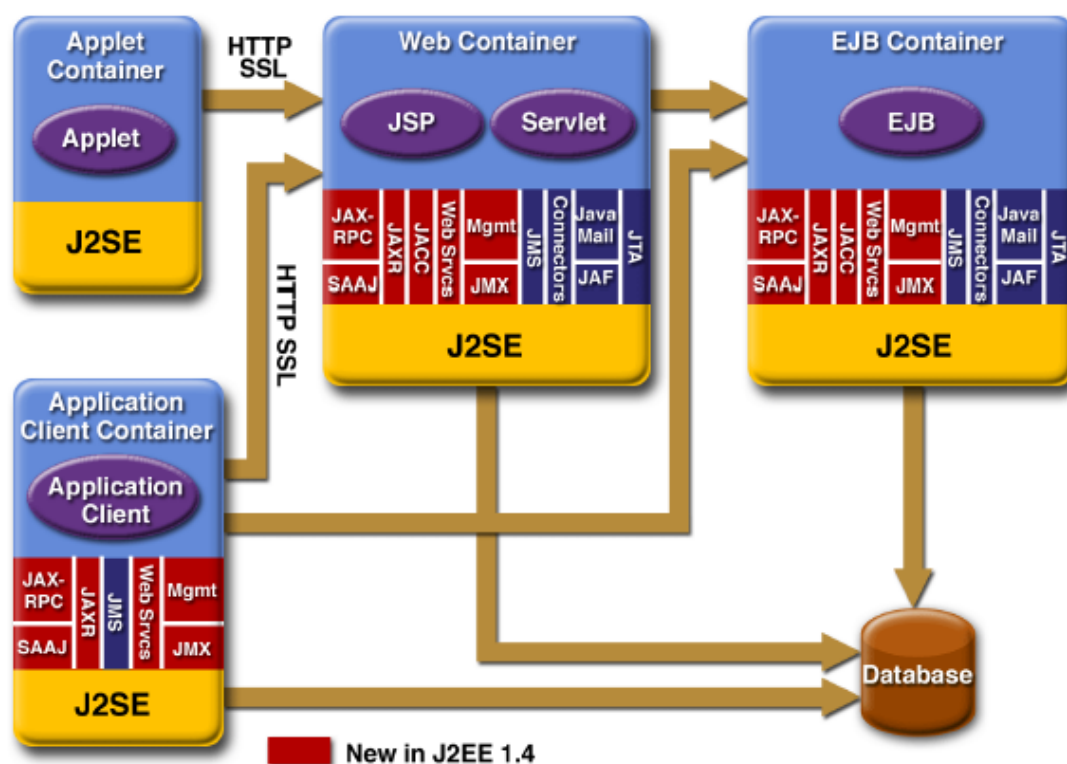


Figura 4 - Arquitetura Java J2EE (Extraído de Sun J2EE)

Na implementação deste aplicativo, também será utilizado o framework JSF JavaServer Face. No qual é apresentado sua estrutura e funcionamento na Figura 5.

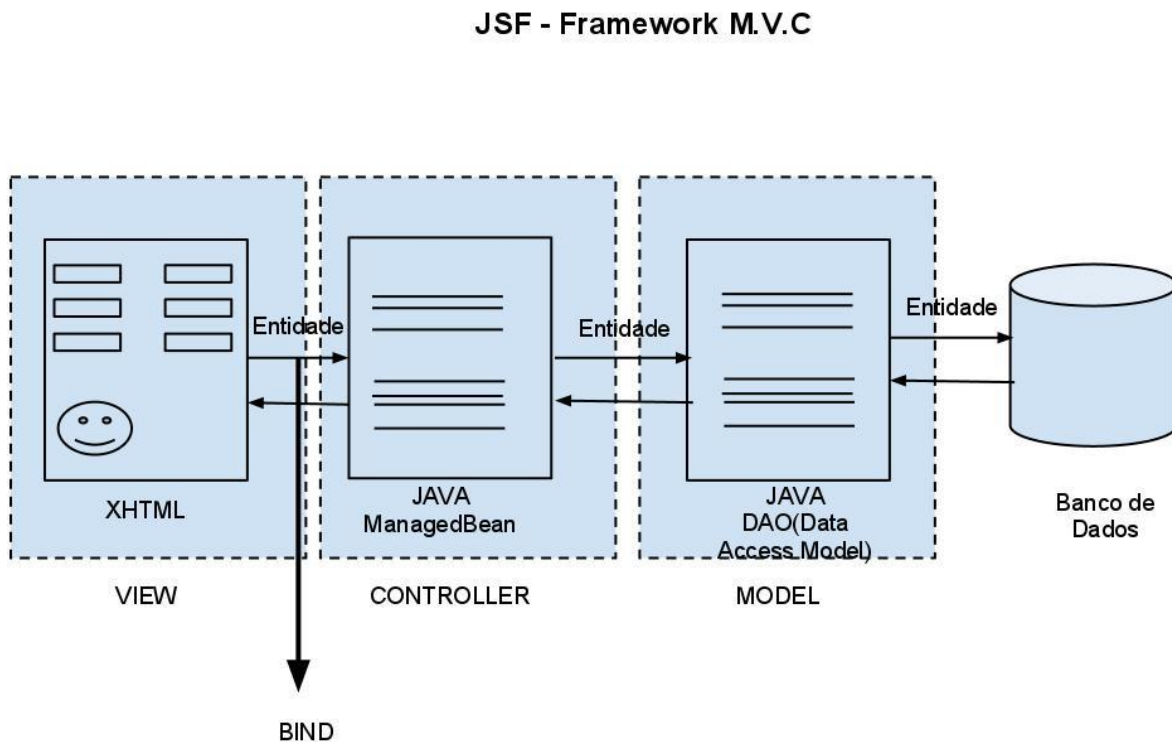


Figura 5 - Arquitetura e Funcionamento JSF

Dentro do framework JSF, será utilizado o recurso *Facelets* que dá origem ao *Template*, no qual possui as páginas integrantes do layout do sistema, são basicamente as interfaces comuns a serem incluídas por todas as páginas, como por exemplo, o cabeçalho e o rodapé. Elas serão mantidas por todas as páginas da aplicação através do framework Tiles.

É possível ver como será utilizado este recurso, na apresentação da figura 6.

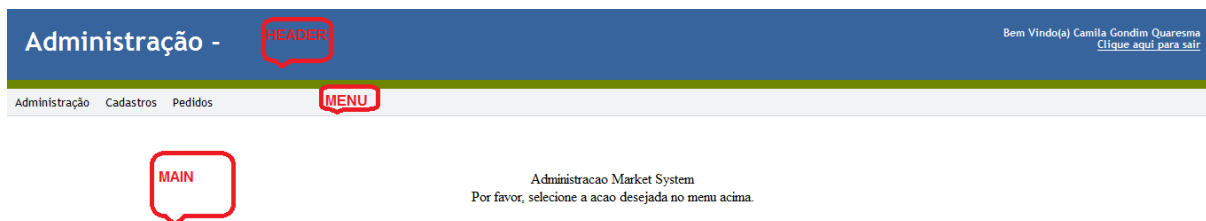


Figura 6 – Funcionamento de *Templates*

Será utilizado também, o recurso A4J – Ajax for JFS. Que é uma tecnologia AJAX desenvolvido especialmente para JSF – JavaServer Face.

2.7 GOOGLE MAPS API

Quando o GoogleMaps foi lançado, em fevereiro de 2005, ainda em sua versão beta, tornou-se rapidamente uma referência em serviços de mapas na Internet. Com uma interface rica e intuitiva, a aplicação permitia acesso a uma enorme base de dados contendo inúmeras imagens de satélite, mapas de cidades, bairros, ruas e avenidas dos Estados Unidos. Com o tempo, novas localidades foram sendo adicionadas no sistema até que em meados de maio de 2007, a Google finalmente disponibilizou consultas de endereços no Brasil, sendo que, em outubro de 2007, uma versão estendida para o público brasileiro foi oferecida, com a possibilidade de se localizar restaurantes, hotéis, traçar rotas, dentre outras utilidades (AZEVEDO, 2010). É um serviço gratuito fornecido pela empresa Google, usado para realizar pesquisas, visualização de mapas e imagens de satélite da Terra. Possibilita a produção de um mapa personalizado completo, sendo possível marcar locais,

adicionar vídeos, fotos e compartilhar todo este conteúdo na rede mundial de computadores (GOOGLE MAPS, 2010)

Em um navegador, e com um simples clique o usuário pode dar a volta virtual pelo mundo, desfrutando de fotos aéreas de grande qualidade em algumas zonas, e do mapeamento vetorial completo de outras. Porém, o interessante do serviço é que Google oferece uma API (Application Programming Interface) para todos aqueles interessados em usar este serviço e integra-lo em sua própria página, de forma simples e gratuita (Chaure, 2007).

A API do Google Maps é uma interface de desenvolvimento para aplicações baseadas no próprio Google Maps, permitindo criar aplicativos inovadores de mapeamento para a plataforma da Web. Essa API ajuda a integrar mapas e geocodificação

aos sites, possibilitando que as aplicações com conteúdo geo60 referenciado sejam facilmente apresentadas em qualquer navegador (GOOGLE DEVELOPER GUIDE, 2010).

Com os recursos dos aplicativos é possível calcular a distância de cada percurso, que é apresentado em várias opção de rota.

Como segue nas próximas figuras, que é representada três opções diferentes de percurso.

A Figura 7 mostra a primeira opção de rota sugerida pelo aplicativo.

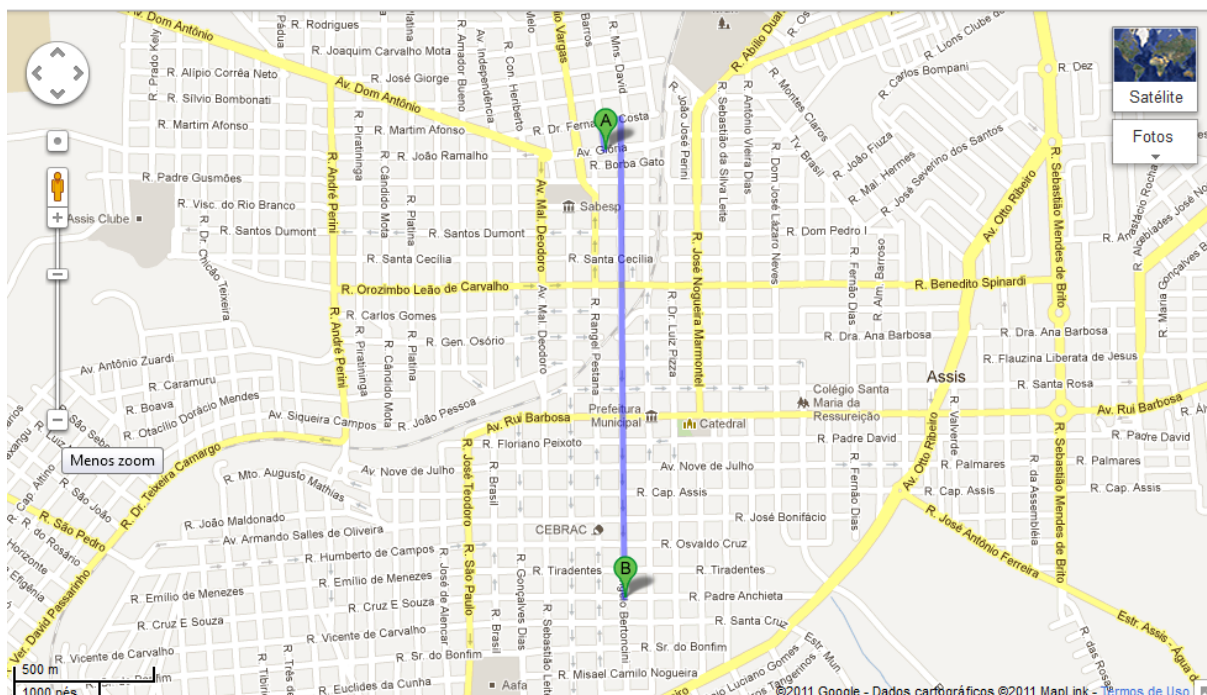


Figura 7 – Trajeto de rota (Extraído do Aplicativo Google Maps)

Neste primeiro mapa, é mostrada a primeira opção de rota calculada pelo Google Maps, do ponto A ao ponto B, ou seja, de Vila Glória, Assis – São Paulo à Rua Padre Anchieta, Assis – São Paulo, respectivamente. Neste primeiro trajeto sugerido, foi calculado 1,9 km de distância, com o tempo de 7 minutos para chegar ao destino.

A Figura 8 representa a rota alternativa apresentada pelo aplicativo.

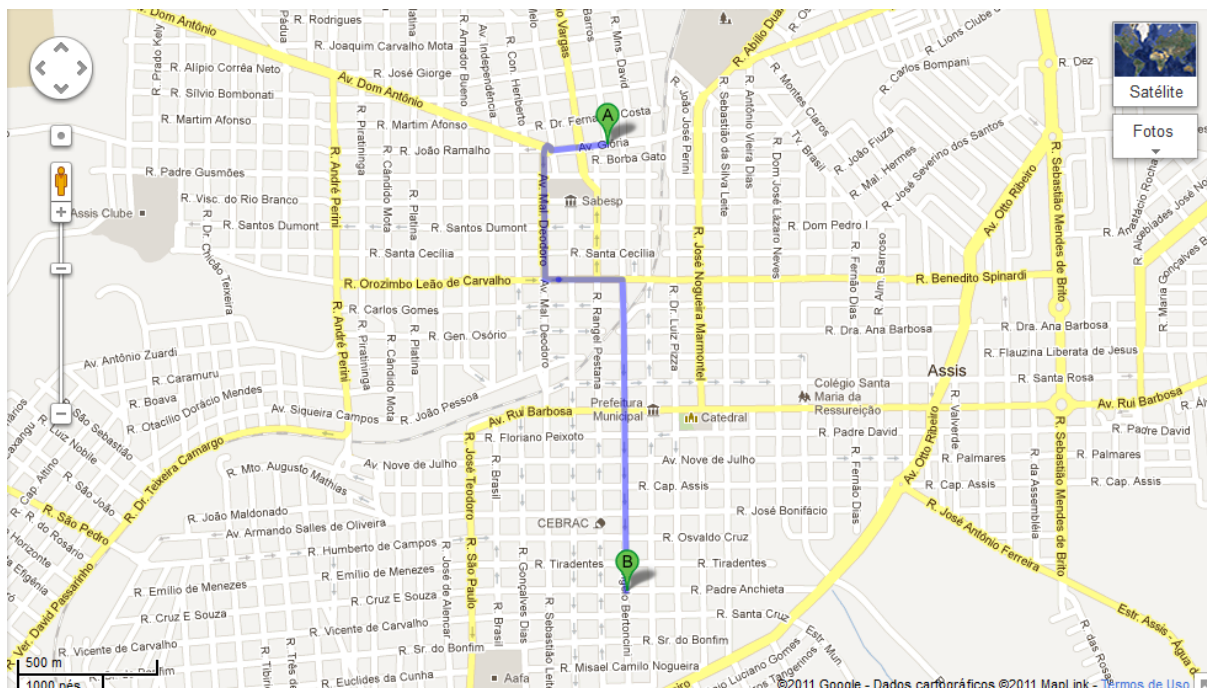


Figura 8 – Trajeto alternativo de rota (Extraído do Aplicativo Google Maps)

Neste segundo mapa é mostrado a segunda trajetória oferecida, dos mesmos pontos, porém é mostrado uma rota alternativa, e nesta, tem a distância de 2,1 km, e com o tempo de 7 minutos para chegar ao destino pretendido.

Concluindo, é possível calcular várias rotas pelo aplicativo, porém cada rota tem uma distância e/ou tempo diferente, sendo assim, é possível escolher o que é mais viável para a necessidade de cada momento.

Visando explorar os recursos da API do Google Maps, foi utilizado para o cálculo uma fórmula de trigonometria esférica que utiliza as localizações geográficas das cidades, latitude e longitude, que calcula a distância entre dois pontos no globo terrestre. A latitude e longitude são recuperadas diretamente da base de dados do Google Maps por meio do endereço do ponto que pode ser uma rua, cidade, estado ou país.

A API do Google Maps foi escolhida para representação gráfica do grafo pois a quantidade de recursos georeferenciais que esse serviço possui são vastos e incrivelmente simples de serem utilizados. A API disponibilizada duas formas de manipulação, uma em JavaScript e outra em Adobe Flash, a adotada será em JavaScript.

2.8 HIBERNATE

Trabalhar com software orientado a objetos e um banco de dados relacional pode ser dispendioso e consumir demasiado tempo em ambientes de desenvolvimento de software corporativos.

Hibernate é um *framework* de mapeamento Objeto-Relacional (MOR) que se refere à técnica de se mapear a representação de dados de um modelo de objetos para um modelo de dados de um banco de dados relacional, com um esquema baseado em SQL.

Hibernate não cuida somente do mapeamento de classes Java para tabelas de banco de dados, mas também fornece formas de facilitar consultas ao banco. Com isso, ele pode também reduzir significativamente o tempo de desenvolvimento gasto com dados de outra forma com a instrumentação manual em SQL e JDBC.

Sua arquitetura "compreensiva" abstrai a aplicação do JDBC/JTA e APIs adjacentes e deixa o Hibernate tomar conta dos detalhes (King, 2009).

Conforme mostra a figura 09, a arquitetura do hibernate dispõe de uma camada de abstração de dados entre a camada de aplicação e banco de dados. Desta maneira, ele interage e traduz o modelo de objetos da aplicação para o modelo de dados do banco, ou vice-versa

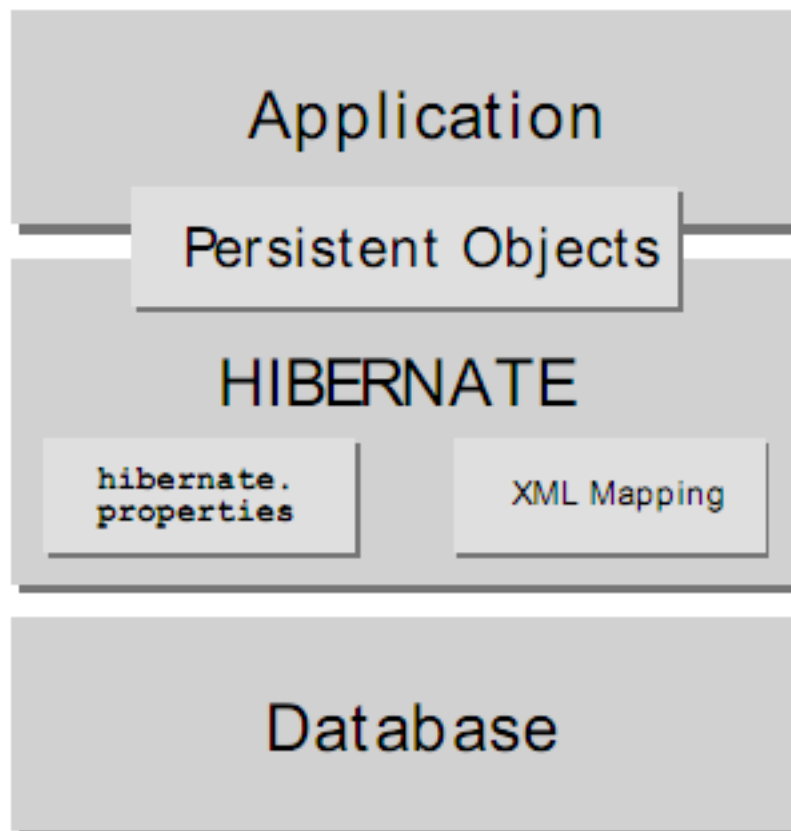


Figura 9 - Visão Geral da Arquitetura Hibernate (King, 2009)

Como será utilizado o banco de dados PostgreSQL, o Hibernate fará todo o *Bind*, para a conversação de uma linguagem orientada a objeto para um banco relacional. Esse processo pode ser visto, através da figura 10.

HIBERNATE → Framework ORM

O: Object
R: Relacional
M: Mapping

MAPEAMENTO OBJETO - RELACIONAL

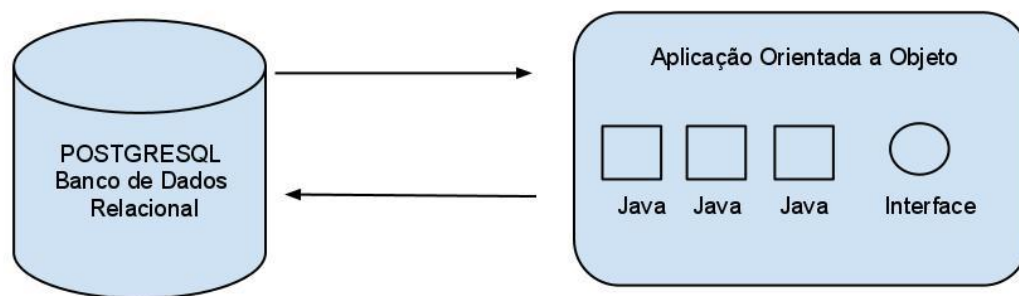


Figura 10 – Visão de Funcionamento do Hibernate

3. DESENVOLVIMENTO DO APLICATIVO

3.1 DESCRIÇÃO DO PROBLEMA

Tendo em vista a oportunidade de mercado, pois na cidade de Assis e região não existe um sistema, com essa funcionalidade. Surgiu a ideia do sistema, através da necessidade que existe, hoje em dia, em fazer tudo com o menos esforço possível. Também, a maior necessidade de economia em quilometragem, para entrega das compras. Como no sistema poderá ser visto, é possível calcular rota para uma entrega com caminhos mais otimizado, resultando assim, em menor gasto com combustível, e menor tempo de entrega.

3.2 MODELAGEM DO PROBLEMA

A Figura 11 representa a estrutura em que o aplicativo foi desenvolvido.

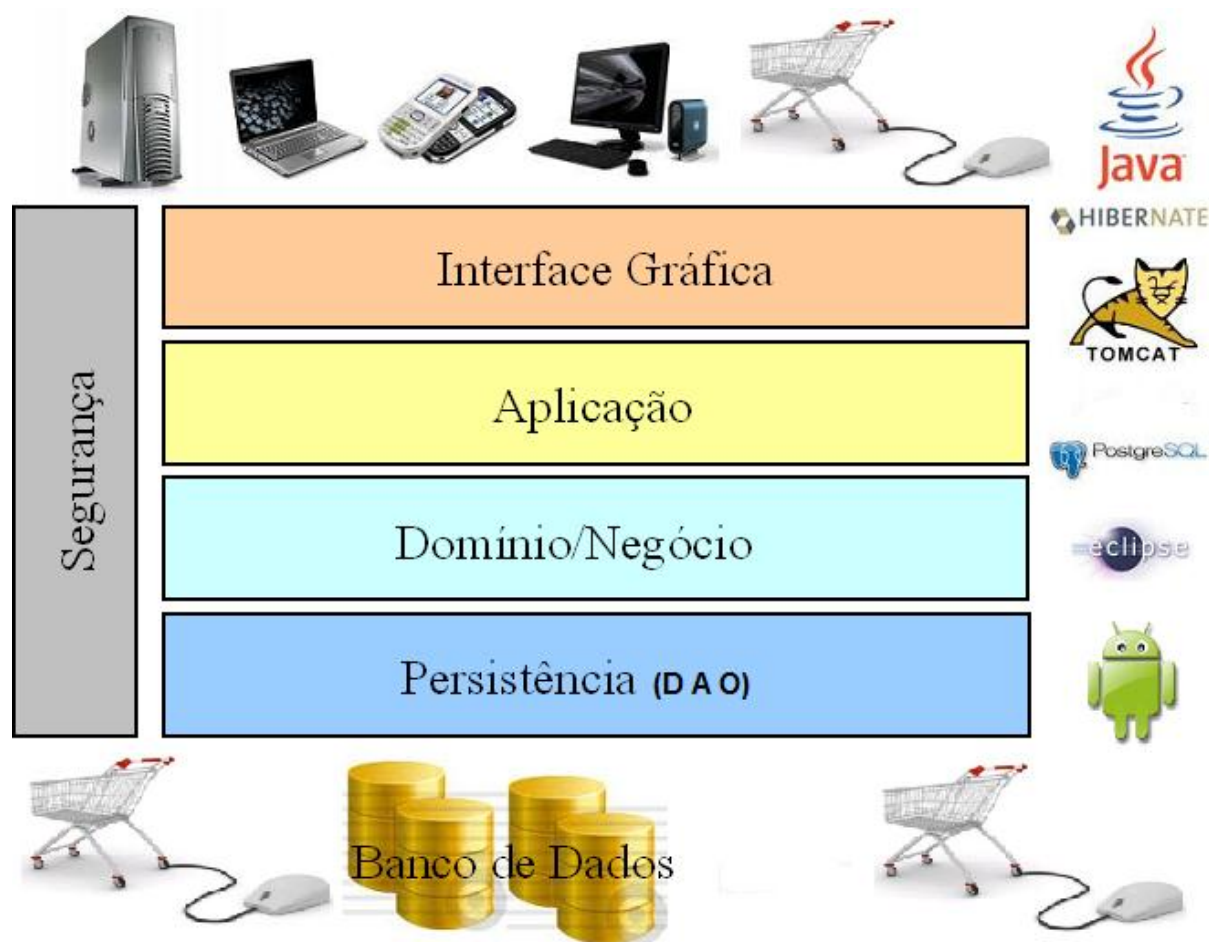


Figura 11 - Arquitetura geral do Aplicativo

3.3 DESENVOLVIMENTO DE APLICATIVO

3.3.1 Especificação

Nesta seção serão apresentados a os requisitos iniciais já levantados, e que define grande parte do escopo do sistema.

Onde, conforme nomenclatura usada pelo RUP - *Rational Unified Process* (ou Processo Unificado Racional), RF significa Requisito Funcional e RNF significa Requisito Não-Funcional. São eles:

- **Requisitos Funcionais**

RF01: para realizar as compras, o usuário necessita realizar um cadastro no sistema, informando seus dados pessoais.

RF02: Para cada tipo de usuário existirá um nível de acesso.

RF03: para realizar as compras, o usuário deverá efetuar o pagamento com cartão de crédito ou por boleto bancário.

RF04: para realizar as compras, o pagamento do usuário deve primeiramente ser confirmado.

RF05: A entrega da mercadoria somente será efetuada após ser calculado pelo sistema, as possíveis rotas para todas entregas do dia.

RF06: o sistema deve prover envio de msgs sms assim que a compra for entregue na casa do usuário.

RF07: envio de email de ofertas do dia, para usuários cadastrados.

RF08: geração da lista de mercadoria para retirada dos produtos do depósito, diretamente para o caminhão de entrega.

RF09: Geração da melhor rota para entrega das mercadorias.

- **Requisitos não Funcionais**

RNF01: as páginas do sistema não deverão demorar mais que 5 segundos para serem carregadas.

RNF02: o sistema deverá comportar o envio de até 3 sms simultâneos.

3.3.1.1

WBS - Work Breakdown Structure

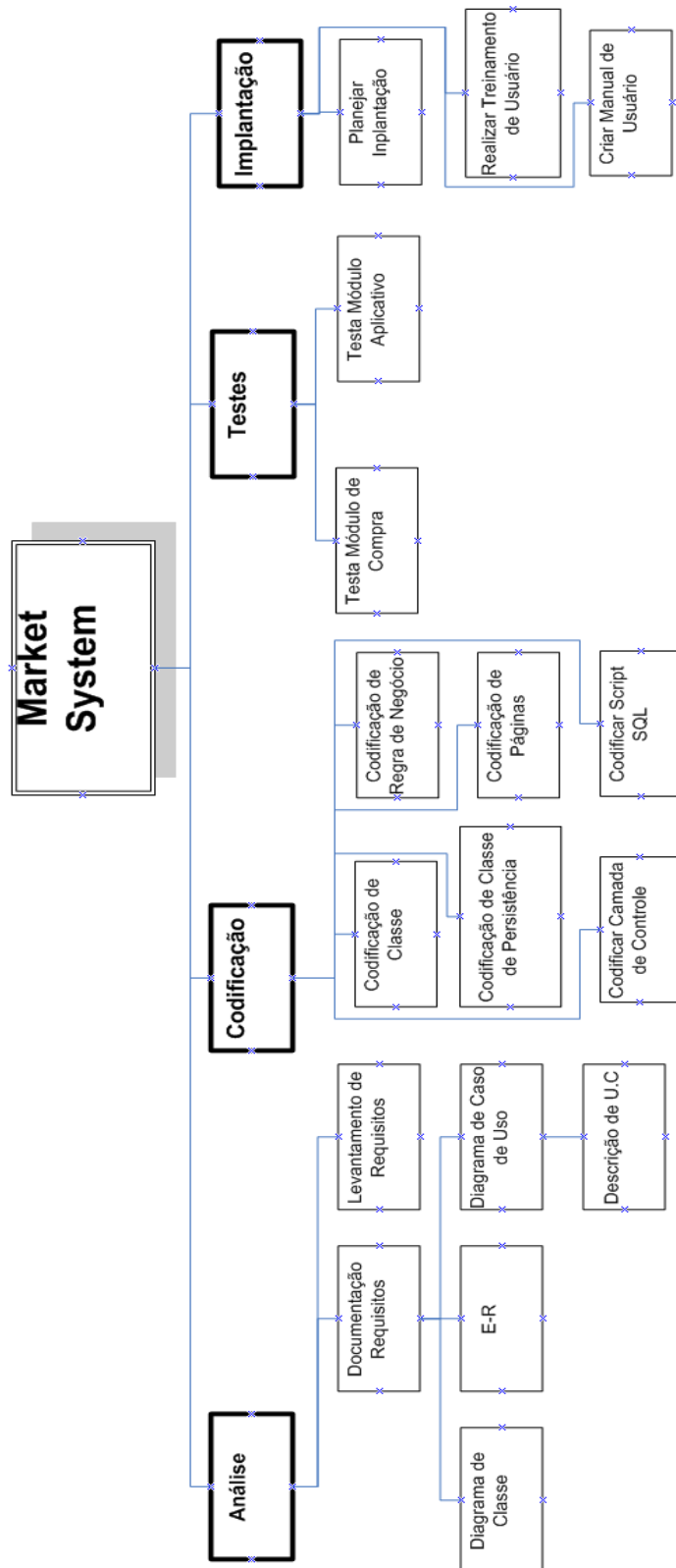


Figura 12 – Diagrama WBS

3.3.1.2 Sequenciamento de Atividades

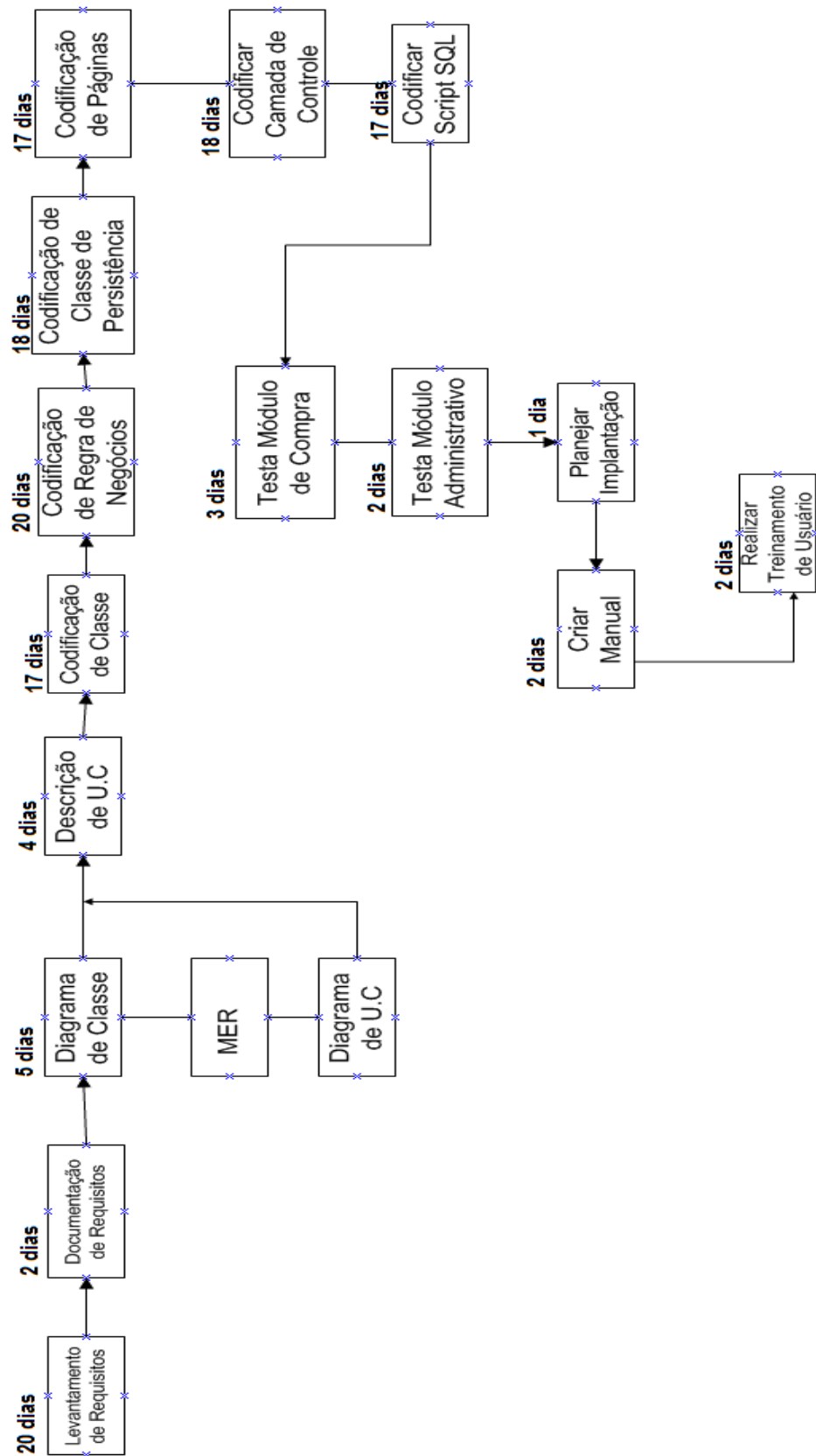


Figura 13 – Diagrama de Sequenciamento de Atividades

3.3.1.3 Diagrama de Caso de Uso

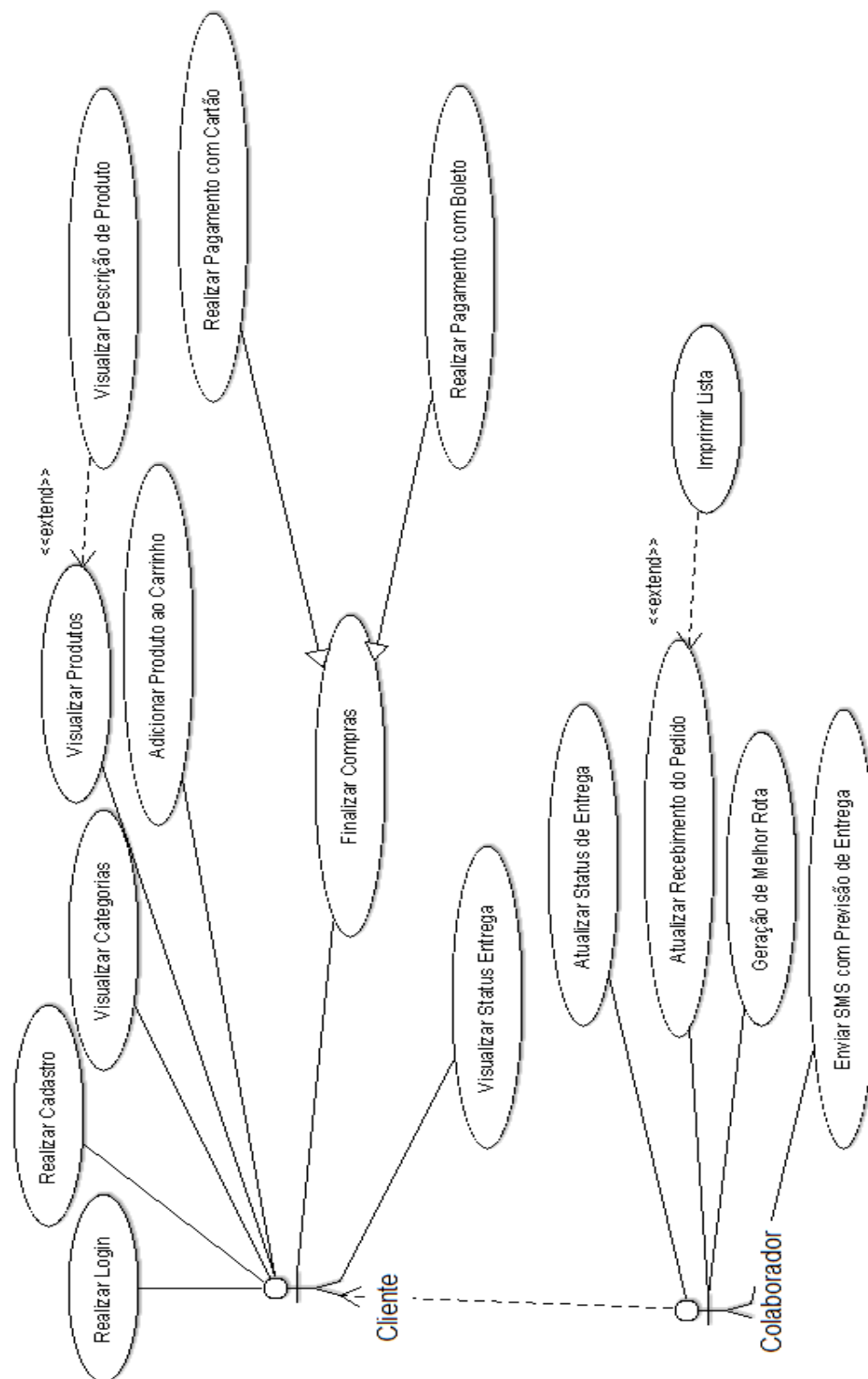


Figura 14 – Diagrama de U.C

3.3.1.4.1 Descrição de Caso de Uso Realizar *Login*

Aplicativo:		Market System
Caso de uso:		UC01 – Realizar <i>Login</i>
Descrição:		Através da página de <i>login</i> do sistema, o usuário realiza a sua autenticação para obter acesso às funcionalidades.
Atores		Cliente e Colaborador.
Pré-condições:		<ul style="list-style-type: none">Caso o usuário não seja administrador, deverá possuir o seu login e senha previamente cadastrados no sistema por um usuário que seja.
Pós-condições:		O Cliente e Colaborador deverão ter seus dados cadastrados no sistema
Fluxo Básico de Eventos		
ID	Ações do Ator	Ações do Sistema
01	Acessa o sistema via url.	Exibir campos para digitação necessários para a autenticação. Campos exibidos: - Login (preenchimento obrigatório) - Senha (preenchimento obrigatório)
02	Preenche os campos e clica no botão OK.	Verifica que o login e senha são válidos de acordo com os dados cadastrados.
03		Exibe a página inicial do sistema com as opções de menu disponíveis para o nível de acesso correspondente ao usuário corrente.
Primeiro Fluxo Alternativo de Eventos – Login e/ou Senha Inválidos		
ID	Ações do Ator	Ações do Sistema
01	No passo 02 do fluxo principal, preenche os campos e clica no botão OK.	Verifica que o login e senha são inválidos, ou seja, não existem no sistema.
02		Exibe a mensagem: “Login e/ou Senha Inválido(s)”
03		Aguarda uma nova ação do usuário na mesma página.
Fluxo de Exceção de Eventos – Campos obrigatórios		

ID	Ações do Ator	Ações alternativas
01		No passo 02 do fluxo principal, o sistema verifica que algum ou todos os campos não foram preenchidos.
02		Exibe a mensagem de erro: “Campo obrigatório” para cada campo não preenchido.
03		Aguarda uma nova ação do ator na mesma página.

3.3.1.4.2 Descrição de Caso de Uso Realizar Cadastro Colaborador

Aplicativo:	Market System
Caso de uso:	UC02 – Realizar Cadastro de Colaborador
Descrição:	Através da página de Cadastro de Colaborador é feito o cadastro dos Colaboradores que terão acesso ao sistema.
Atores	Colaborador.
Pré-condições:	<ul style="list-style-type: none"> O usuário deverá estar logado no sistema para acessar esta funcionalidade.
Pós-condições:	O Colaborador deverá ter seus dados cadastrados no sistema

Fluxo Básico de Eventos

ID	Ações do Ator	Ações do Sistema
01	Acessa o item de menu: Cadastrar → Cadastrar Cliente	Exibir campos para digitação necessários para o Cadastro. Campos exibidos: <ul style="list-style-type: none"> - Nome - Logradouro - Bairro - CPF - RG - Telefone - Cidade - Estado - Pais - Cargo
02	Preenche os campos e clica no botão OK.	Verifica se todos os itens preenchidos corretamente
03		Exibe a mensagem que o cadastrado foi efetuado com sucesso

Fluxo de Exceção de Eventos – Campos obrigatórios

ID	Ações do Ator	Ações alternativas
----	---------------	--------------------

01		No passo 02 do fluxo principal, o sistema verifica que algum ou todos os campos não foram preenchidos.
02		Exibe a mensagem de erro: “Campo obrigatório” para cada campo não preenchido.
03		Aguarda uma nova ação do ator na mesma página.
Fluxo de Exceção de Eventos – Campos preenchidos incorretamente		
ID	Ações do Ator	Ações alternativas
01		No passo 02 do fluxo principal, o sistema verifica que algum ou todos os campos não foram preenchidos corretamente.
02		Exibe a mensagem de erro: “Campo preenchido incorretamente.” para cada campo que foi preenchido errado.
03	O Usuário clica em Ok	O sistema volta para a página de preenchimento de dados novamente.
04		Aguarda uma nova ação do ator na mesma página.

3.3.1.1.1 Descrição de Caso de Uso Realizar Cadastro Cliente

Aplicativo:		Market System
Caso de uso:		UC02.1 – Realizar Cadastro Cliente
Descrição:		Através da página de Cadastro de Cliente.
Atores		Cliente.
Pré-condições:		<ul style="list-style-type: none">O usuário deverá estar logado no sistema para acessar esta funcionalidade.
Pós-condições:		O Cliente e Colaborador deverá ter seus dados cadastrados no sistema
Fluxo Básico de Eventos		
ID	Ações do Ator	Ações do Sistema
01	Acessa o item de menu: Cadastrar → Cadastrar Cliente	Exibir campos para digitação necessários para o Cadastro. Campos exibidos: <ul style="list-style-type: none">- Nome- Logradouro- Bairro- CPF- RG- Telefone- Cidade- Estado- Pais
02	Preenche os campos e clica no botão OK.	Verifica se todos os itens preenchidos corretamente
03		Exibe a mensagem que o cadastrado foi efetuado com sucesso
Fluxo de Exceção de Eventos – Campos obrigatórios		
ID	Ações do Ator	Ações alternativas

01		No passo 02 do fluxo principal, o sistema verifica que algum ou todos os campos não foram preenchidos.
02		Exibe a mensagem de erro: “Campo obrigatório” para cada campo não preenchido.
03		Aguarda uma nova ação do ator na mesma página.

3.3.1.2 Caso de Uso Visualizar Categoria

Aplicativo:		Market System
Caso de uso:		UC03 – Visualizar Categoria
Descrição:		Através da página principal do sistema, visualiza todas as categorias existentes dos produtos do supermercado.
Atores		Cliente e Colaborador.
Pré-condições:		<ul style="list-style-type: none">O usuário deverá estar logado no sistema para acessar esta funcionalidade.
Pós-condições:		O Cliente ou Colaborador deverá ter acesso de visualização de todas as categorias dos Produtos
Fluxo Básico de Eventos		
ID	Ações do Ator	Ações do Sistema
01	Acessa o item de menu: Visualizar → Visualizar Categoria	Exibi todos as categorias existentes no supermercado
02	Clica na Categoria desejada	Abre a página com os Produtos de suas respectivas Categorias.
03		Aguarda uma nova ação do usuário na mesma página.

3.3.1.3 Descrição de Caso de Uso Visualizar Produto

Aplicativo:		Market System
Caso de uso:		UC04 – Visualizar Produto
Descrição:		Através da página principal do sistema, visualiza todas as categorias existentes dos produtos do supermercado.
Atores		Cliente e Colaborador.
Pré-condições:		<ul style="list-style-type: none">• O usuário deverá estar logado no sistema para acessar esta funcionalidade.• O Usuário deve ter acessado ao “Visualizar Categoria”.
Pós-condições:		O Cliente ou Colaborador deverá ter acesso de visualização de todos os Produtos que estarão alocados nas suas devidas Categorias.
Fluxo Básico de Eventos		
ID	Ações do Ator	Ações do Sistema
01	Acessa o item de menu: Visualizar → Visualizar Categoria → Produtos	Exibi todos as Produtos existentes no supermercado, em suas devidas Categorias.
02	Clica no Produto desejado.	Exibi a opção de enviar os Produtos para o Carrinho de Compra, podendo assim, escolher a quantidade do Produto que o Cliente deseja .
03	O Usuário digita a quantidade desejada do Produto.	O Produto é adicionado ao Carrinho de Compra .
04		Retorna para a página onde o Usuário pode escolher mais Produtos. Aguarda uma nova ação do Usuário na mesma página.

3.1.2.4.1 Descrição de Caso de Uso Visualizar Descrição de Produto

Aplicativo:		Market System
Caso de uso:		UC04.1 – Visualizar Descrição de Produto
Descrição:		Através da página principal do sistema, visualiza todas as categorias existentes dos produtos do supermercado.
Atores		Cliente e Colaborador.
Pré-condições:		<ul style="list-style-type: none">O Usuário deve estar na página de lista de produtos para poder visualizar as descrições (vide descrição do caso de uso Listar Produtos).
Pós-condições:		O Cliente ou Colaborador deverá ter acesso de Visualicão de toda descrição e detalhe do Produto selecionado.
Fluxo Básico de Eventos		
ID	Ações do Ator	Ações do Sistema
01	O Usuário clica sobre o nome de um determinado Produto	O Sistema exibe a descrição do Produto, contendo os dados com detalhes do Produto selecionado.
02	O Usuário seleciona o produto para compra-lo.	O Sistema envia o Produto selecionado ao Carrinho de Compra
03		Aguarda uma nova ação do Usuário na mesma página.
Fluxo de Exceção de Eventos – Voltar a Visualização de Produtos		
ID	Ações do Ator	Ações alternativas
01	No passo 02 do fluxo principal, o Usuário não deseja comprar esse Produto, e então clica no botão “Voltar”	O Sistema retorna para a página anterior, da qual apresenta a lista de Produtos.
02		Aguarda uma nova ação do ator na mesma página.

3.3.1.4 Descrição de Caso de Uso Adicionar ao Carrinho de Compra

Aplicativo:		Market System
Caso de uso:		UC05 – Adicionar ao Carrinho de Compra
Descrição:		Através deste, é possível a soma de todos produtos e o fechando da compra para sua finalização.
Atores		Cliente e Colaborador.
Pré-condições:		<ul style="list-style-type: none">O Usuário deve estar na página de lista de produtos para poder escolher o produto e sua devida quantidade para poder adicioná-lo ao Carrinho de Compra.
Pós-condições:		Todos os produtos selecionados, deverão estar no Carrinho de Compra, sendo assim, o Carrinho deve estar pronto para ser enviado para o: Finalizar Compras, com os produtos somados e listados.
Fluxo Básico de Eventos		
ID	Ações do Ator	Ações do Sistema
01	O Usuário escolhe a quantidade de produto desejado, clicando sob uma caixa onde é permitido a escrita, e é digitado a quantidade escolhida.	O Sistema exibe a quantidade de produtos na caixa de quantidade de produto, logo ao lado do nome do produto.
02	O Usuário clica no botão “adicionar ao carrinho”.	O Sistema envia o Produto selecionado ao Carrinho de Compra
03		Volta para a página de listagem de produto. Aguarda uma nova ação do Usuário na mesma página.

3.3.1.5 Descrição de Caso de Uso Finalizar Compra

Aplicativo:		Market System
Caso de uso:		UC06 – Finalizar Compra
Descrição:		Através deste, é possível ser finalizado a Compra..
Atores		Cliente e Colaborador.
Pré-condições:		<ul style="list-style-type: none">O Usuário deve ter informado quais produtos foram escolhidos, e suas respectivas quantidades. Estes devem ter sido adicionados ao carrinho de Compra
Pós-condições:		O Cliente ou Colaborador poderá escolher a forma que ele deseja fazer o pagamento pelas suas compras.
Fluxo Básico de Eventos		
ID	Ações do Ator	Ações do Sistema
01	O Usuário clica sobre o botão “Finalizar Compra”, que é mostrado na página que representa o carrinho de compra.	O Sistema então Calcula a quantidade de produtos, e o preço total deles. Assim, o sistema abre uma nova página com a lista dos produtos selecionados, e o valor total da compra completa. O Sistema mostra uma janela de aviso: “Deseja realmente Finalizar sua compra? S/N”.
02	O Usuário seleciona o botão “SIM”, da janela de confirmação.	O Sistema abre a página para o Usuário ter como escolher a forma de pagamento desejada.
Fluxo de Exceção de Eventos – Abandonar Compras		
ID	Ações do Ator	Ações alternativas
01	No passo 01 do fluxo principal, o Usuário não deseja finalizar essa compra, e então clica no botão “Abandonar Compras”.	O Sistema descarta os produtos deste carrinho de compra, e retorna a página anterior, da qual apresenta a lista de Produtos.
02		Aguarda uma nova ação do ator na mesma página.
Fluxo de Exceção de Eventos – Botão “NÃO” selecionado		

ID	Ações do Ator	Ações alternativas
01	No passo 02 do fluxo principal, o Usuário não deseja finalizar compra, e então clica no botão “NÃO”	O Sistema retorna para a página anterior, da qual apresenta a lista de Produtos no Carrinho de compra.
02		Aguarda uma nova ação do ator na mesma página.

3.2.3 Diagrama de Classes

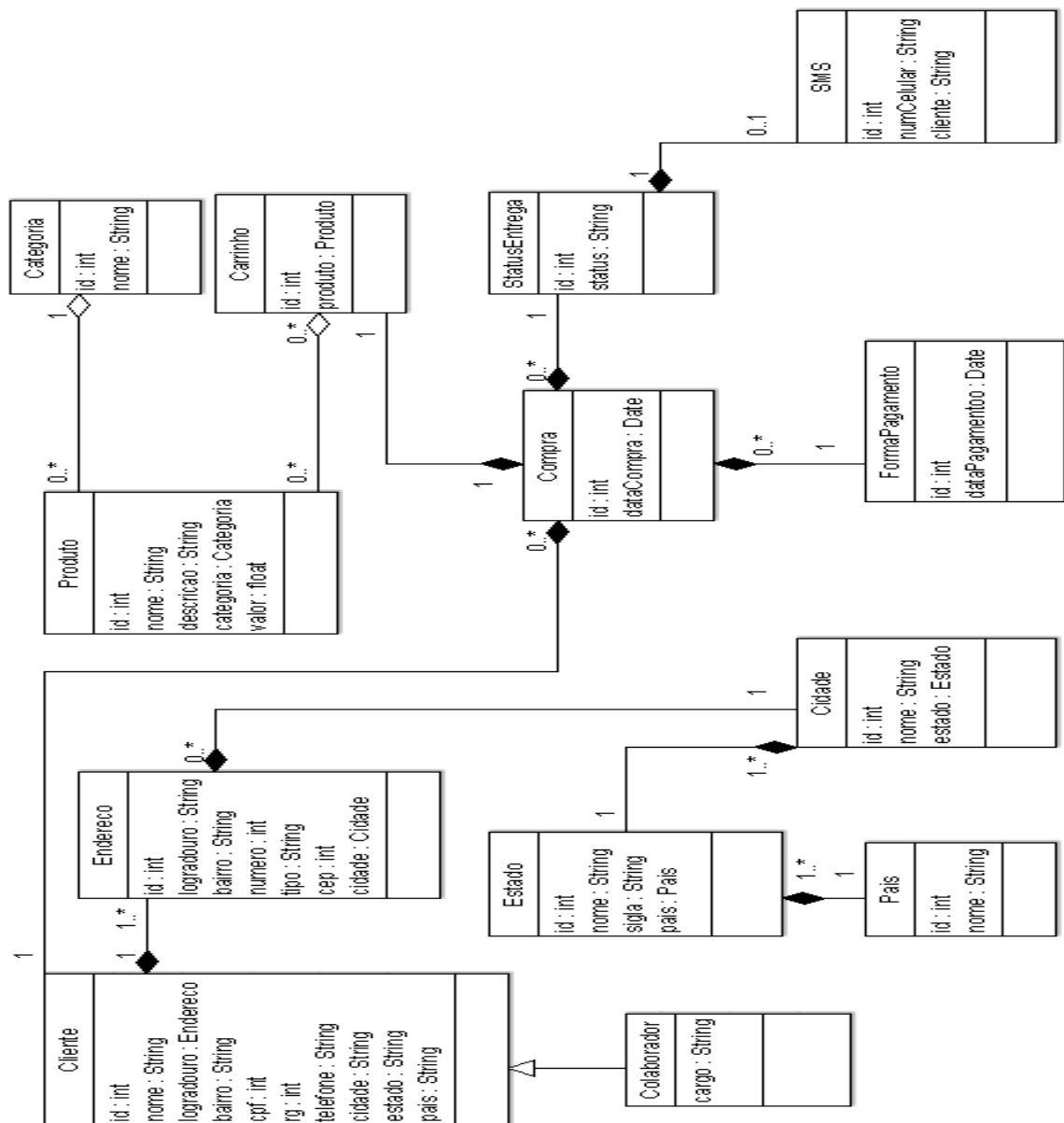
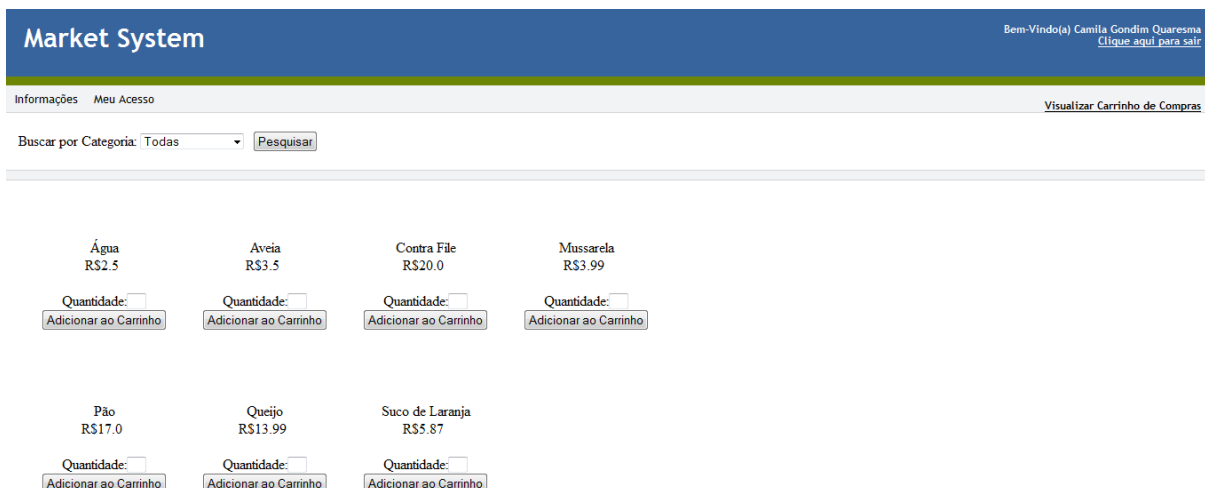


Figura 16 – Diagrama de Classes

3.3.2 Implementação

Ao acessar o sistema, é possível visualizar toda a lista de produtos disponível no mercado. Então, o cliente poderá escolher o produto, e a quantidade desejada. Após isso, pode ser adicionado ao carrinho de compra. Como é possível ver na figura 17.

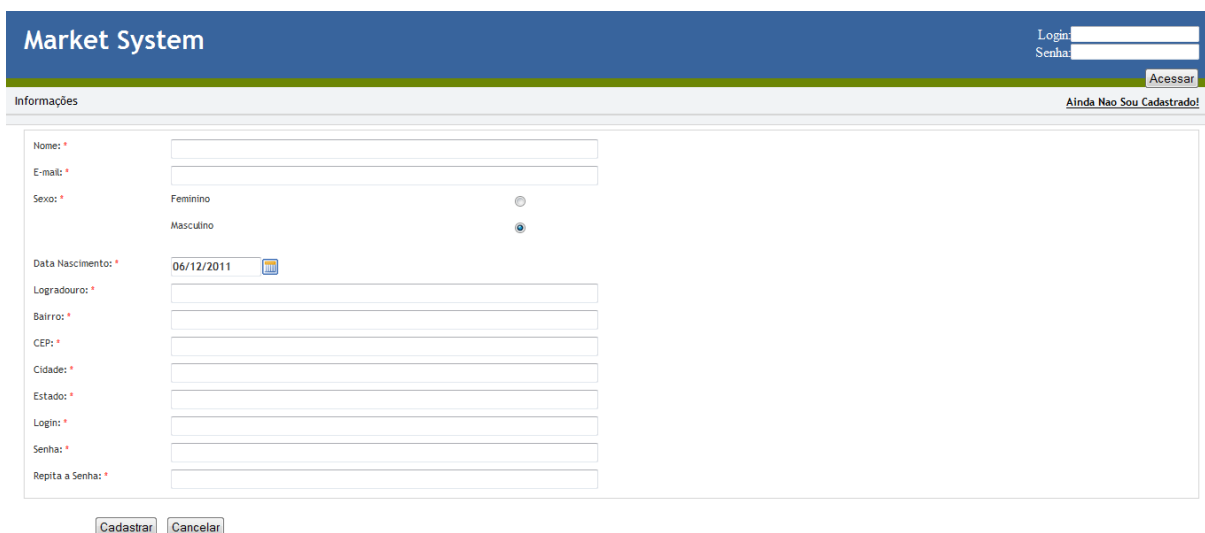


The screenshot shows the 'Market System' interface. At the top, there's a blue header with the system name and a user greeting: 'Bem-Vindo(a) Camila Gondim Quaresma' with a link 'Clique aqui para sair'. Below the header, there's a navigation bar with 'Informações' and 'Meu Acesso'. A search bar is present with a dropdown for 'Categoria: Todas' and a 'Pesquisar' button. The main content area displays a grid of products, each with its name, price, a quantity input field, and an 'Adicionar ao Carrinho' button.

Produto	Preço	Quantidade	Ação
Água	R\$2.5	<input type="text"/>	Adicionar ao Carrinho
Aveia	R\$3.5	<input type="text"/>	Adicionar ao Carrinho
Contra File	R\$20.0	<input type="text"/>	Adicionar ao Carrinho
Mussarela	R\$3.99	<input type="text"/>	Adicionar ao Carrinho
Pão	R\$17.0	<input type="text"/>	Adicionar ao Carrinho
Queijo	R\$13.99	<input type="text"/>	Adicionar ao Carrinho
Suco de Laranja	R\$5.87	<input type="text"/>	Adicionar ao Carrinho

Figura 17 – Interface de Compra

Caso o Cliente ainda não seja cadastrado, é possível fazer seu cadastramento. Como é apresentado na figura 18.



The screenshot shows the 'Market System' interface with the registration form for a new user. The header and navigation bar are the same as in Figure 17. The main content area is titled 'Informações' and contains a registration form with the following fields: Nome, E-mail, Sexo (Feminino/Masculino), Data Nascimento, Logradouro, Bairro, CEP, Cidade, Estado, Login, Senha, and Repita a Senha. There are 'Cadastrar' and 'Cancelar' buttons at the bottom.

Figura 18 – Interface de Cadastramento de Não-Cliente

A visualização do carrinho é como mostra na figura 19. Onde é apresentada a lista de todos os produtos que foram selecionados, com valor individual dos produtos, e logo abaixo fazendo a totalização da compra. Que ao ser clicado 'Finalizar Compra', é feito uma confirmação de cadastro do cliente.

The screenshot displays a web interface for a shopping cart. At the top, there are links for 'Informações' and 'Meu Acesso', and a button for 'Visualizar Carrinho de Compras'. Below this is a section titled 'Meu Carrinho de Compras' containing a table with the following data:

Produto	Valor	Quantidade	
Contra File	R\$ 20.0	5.0	[Atualizar] [Excluir]
Aveia	R\$ 3.5	1.0	[Atualizar] [Excluir]
Água	R\$ 2.5	3.0	[Atualizar] [Excluir]

Below the table, the total value is shown as 'Valor Total: R\$ 111'. There are two buttons: 'Continuar Comprando' and 'Finalizar Compra'. At the bottom, a section titled 'Dados da Entrega - Pedido de Carlos José da Silva - Data: 07/12/2011 12:48.' contains a form with the following fields:

- Nome: * Carlos José da Silva
- Logradouro: * Rua Dom José Lázaro Neves, 414
- Bairro: * Vila Adileta
- CEP: * 19814-391
- Cidade: * Assis
- Estado: * SP
- **Pagamento: No ato da entrega.

An 'Enviar Pedido' button is located at the bottom of the form.

Figura 19 – Interface do Carrinho de Compras

No módulo administrativo, é possível realizar cadastramento dos produtos, das categorias, e dos clientes onde é necessário informar, se este cliente terá acesso ao módulo de administradores, ou seja, se ele é cliente, ou se ele é colaborador.

Na figura 20, pode ser observada a lista de clientes, e também onde pode ser feito seu cadastramento, ao escolher a opção 'inserir' aparecerá então campos para ser preenchidos, como pode ser visto na figura 21.

Administração - Lista de Clientes

Bem Vindo(a) Camila Gondim Quaresma
Clique aqui para sair

Administração

Cadastros

Pedidos

Listar

Inserir

Salvar

Cancelar

Lista de Clientes

Codigo	Nome (voce pode utilizar o filtro abaixo)	E-mail	Sexo	Data Nascimento	Logradouro	Bairro	CEP	Cidade	Estado	Administrador?	Acoes	Status
5	Camila	camila@gmail.com	Feminino	01/11/2007	R. Padre Anchieta, 789	Vila Palhares	19800210	Assis	SP	Não	Alterar - Inativar	
2	Camila Gondim Quaresma	kamilagondim@gmail.com	Feminino	27/11/2011	Rua treze de maio, 40	Centro	19800240	ASSIS	SP	Sim	Alterar - Inativar	
4	Carlos José da Silva	Carlos@hotmail.com	Masculino	01/11/2001	Rua Dom José Lázaro Neves, 414	Vila Adileta	19814-391	Assis	SP	Não	Alterar - Inativar	
1	Fulano Da Silva	teste@teste.com	Feminino	27/11/2011	teste	teste	teste	teste	teste	Não	Alterar - Inativar	
3	João da Silva	joao@gmail.com	Masculino	29/11/2011	AV. Rui Barbosa	Centro	19814000	Assis	SP	Não	Alterar - Inativar	

Figura 20 – Interface da Lista de Clientes






Administração - Cadastro de Cliente												Bem Vindo(a) Clique aqui para sair
Administração Cadastros Pedidos												
												
Nome: *	<input type="text"/>											
E-mail: *	<input type="text"/>											
Sexo: *	Feminino <input type="radio"/> Masculino <input checked="" type="radio"/>											
Data Nascimento: *	<input type="text" value="07/12/2011"/> 											
Logradouro: *	<input type="text"/>											
Bairro: *	<input type="text"/>											
CEP: *	<input type="text"/>											
Cidade: *	<input type="text"/>											
Estado: *	<input type="text"/>											
Administrador:	<input type="checkbox"/>											

Figura 21 – Interface de Cadastro de Clientes

Na figura 21 é possível visualizar a lista de produtos, onde pode ser feito seu cadastramento, ao escolher a opção 'inserir'.








Administração - Lista de Produtos						
Bem Vindo(a) Camila Gondim Quaresma Clique aqui para sair						
Administração Cadastros Pedidos						
<div> <div>Listar</div> <div>Inserir</div> <div>Salvar</div> <div>Cancelar</div> </div>						
Lista de Produtos						
Código	Nome (você pode utilizar o filtro abaixo) ▾	Descrição	Valor	Categoria	Ações	Status
8	Água	Água com Gas SANTA BARBARA 1 litro	2.5	Bebida	[Alterar] - [Inativar]	
5	Aveia	Aveia dr oetker, banana mel e canela	3.5	Cereais	[Alterar] - [Ativar]	
4	Contra File	Contra File Bourbon	20.0	Açougue	[Alterar] - [Inativar]	
10	Mussarela	Mussarela	3.99	Frios e Laticínios	[Alterar] - [Inativar]	
6	Pão	Pão Francês	17.0	Padaria	[Alterar] - [Inativar]	
9	Queijo	Queijo de Coalho com Orégano BALKIS 400g	13.99	Frios e Laticínios	[Alterar] - [Inativar]	
7	Nectar de	Nectar de	2.00	Bebida	[Alterar] - [Inativar]	

Figura 22– Interface da lista de Produtos

Na figura 22 pode ser visualizada a lista de categorias, onde pode ser feito o seu cadastramento, ao escolher a opção ‘inserir’.

Administração - Lista de Categorias			
Bem Vindo(a) Camila Gondim Quaresma Clique aqui para sair			
Administração Cadastros Pedidos			
<div> <div>Listar</div> <div>Inserir</div> <div>Salvar</div> <div>Cancelar</div> </div>			
Lista de Categorias			
Código	Descrição (você pode utilizar o filtro abaixo) ▾	Ações	Status
1	Açougue	[Alterar] - [Inativar]	
6	Bebida	[Alterar] - [Inativar]	
3	Cereais	[Alterar] - [Inativar]	
5	Frios e Laticínios	[Alterar] - [Inativar]	
4	Limpeza	[Alterar] - [Inativar]	
7	Limpeza	[Alterar] - [Inativar]	
2	Padaria	[Alterar] - [Inativar]	

Figura 23 – Interface da lista de Categorias

Ainda na parte administrativa, mas no módulo de Pedidos, pode ser feito a visualização de todos os pedidos. Como mostra a figura 23. E ao clicar em 'Ver Detalhes', este pedido abrirá, como pode ser visto na figura 24. Nesta interface, existe a opção 'Visualizar Rotas', onde é possível visualizar a rota onde a compra deve ser entregue, pelo endereço que o cliente informou ao fazer o cadastro. Como pode ser visualizado na figura 25

Administração - Pedidos de Compra							
Bem Vindo(s) Camila Gondim Quaresma Clique aqui para sair							
Administração Cadastros Pedidos							
Lista de Pedidos							
Código	Descrição (você pode utilizar o filtro abaixo) »	Nome do Cliente	E-mail do Cliente	Endereço de Entrega	Data	Valor Total	Ações
10	Pedido de Camila- Data: 29/11/2011 02:19	Camila	camila@gmail.com	R. Padre Anchieta, 789 - Vila Palhares, Assis - SP, CEP: 19800210	29/11/2011	R\$ 65.0	Ver Detalhes
9	Pedido de Carlos José da Silva- Data: 29/11/2011 10:45	Carlos José da Silva	Carlos@hotmail.com	Rua Dom José Lázaro Neves, 414 - Vila Adileta, Assis - SP, CEP: 19814-391	29/11/2011	R\$ 150.0	Ver Detalhes
8	Pedido de Camila Gondim Quaresma- Data: 29/11/2011 03:33	Camila Gondim Quaresma	kamilagondim@gmail.com	Rua treze de maio, 40 - Centro, ASSIS - SP, CEP: 19800240	29/11/2011	R\$ 40.0	Ver Detalhes
7	Pedido de João da Silva- Data: 29/11/2011 02:47	João da Silva	joao@gmail.com	AV. Rui Barbosa - Centro, Assis - SP, CEP: 19814000	29/11/2011	R\$ 80.0	Ver Detalhes
6	Pedido de Camila Gondim Quaresma- Data: 29/11/2011 01:58	Camila Gondim Quaresma	kamilagondim@gmail.com	Rua treze de maio, 40 - Centro, ASSIS - SP, CEP: 19800240	29/11/2011	R\$ 60.0	Ver Detalhes
4	Pedido de Camila Gondim Quaresma- Data: 29/11/2011 01:36	Camila Gondim Quaresma	kamilagondim@gmail.com	Rua treze de maio, 40 - Centro, ASSIS - SP, CEP: 19800240	29/11/2011	R\$ 0.0	Ver Detalhes
5	Pedido de Camila Gondim Quaresma- Data: 29/11/2011 01:30	Camila Gondim Quaresma	kamilagondim@gmail.com	Rua treze de maio, 40 - Centro, ASSIS - SP, CEP: 19800240	29/11/2011	R\$ 80.0	Ver Detalhes

Figura 24 – Interface da lista de Pedidos de Compra

Administração - Pedido de Compra

Bem Vindo(a) Camila Gondim Quaresma
[Clique aqui para sair](#)

Administração

Cadastros

Pedidos

Descrição:

Pedido de Camila- Data: 29/11/2011 02:19

E-mail do Cliente:

camila@gmail.com

Endereço de Entrega:

R. Padre Anchieta, 789 - Vila Palhares, Assis - SP, CEP: 19800210
[Visualizar Rotas](#)

Lista de Itens

Descrição	Quantidade	Valor
Contra File Bourbon	3.0Kilo(s)	R\$20.0
Água com Gás SANTA BARBARA 1 litro	2.0Unidade(s)	R\$2.5

Valor Total: R\$ 65.0

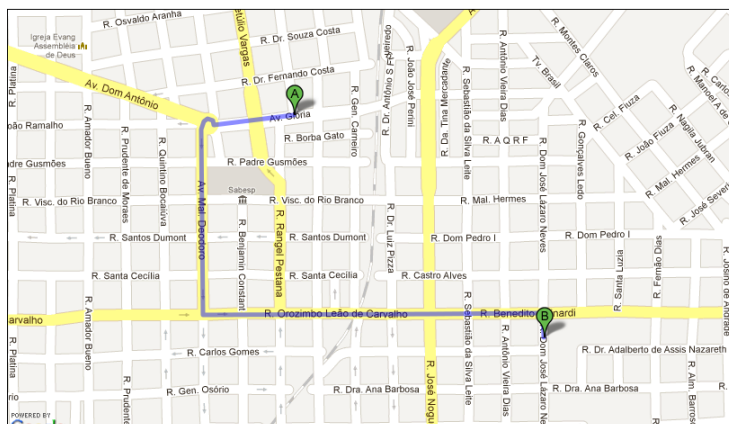
Figura 25 – Interface da lista de Pedidos de Compra individual

Administração - Pedido de Compra

Bem Vindo(a) Camilla Condin Quaresma
[Clique aqui para sair](#)

Administração Cadastros Pedidos

Rota Para a Entrega do Pedido: 9.



[Voltar](#)

A Av. Glória - Assis - São Paulo, 19806-095, Brasil

1,6 km (aprox. 4 min)

1. Siga na direção oeste na **Av. Glória** em direção à **R. Rangel Pestana** 200 m
2. Na rotatória, pegue a 4ª saída para a **Av. Mal. Deodoro** 500 m
3. Vire à esquerda na **R. Orozimbo Leão de Carvalho** 550 m
4. Continue para **R. Benedito Spinardi** 280 m
5. Vire à direita na **R. Dom José Lázaro Neves** 59 m
O destino estará à esquerda

B R. Dom José Lázaro Neves, 414 - Assis - São Paulo, 19814-391, Brasil

Dados cartográficos ©2011 MapLink

Figura 26 – Interface da realização do calculo da Rota, e suas devidas coordenadas.

5. CONCLUSÃO

É possível, através do uso de tecnologias, criarmos sistemas de informação capazes de nos ajudar em nossas atividades diárias, com objetivo de melhorar nossa qualidade de vida, sobretudo nos poupando tempo e/ou nos trazendo comodidade e conforto em nosso dia/dia.

A partir dessa premissa, realizamos os levantamentos de requisitos e funcionalidades necessárias para a construção de um sistema que atenda ao objetivo proposto. Foi definida a arquitetura sistêmica, as ferramentas a serem utilizadas na construção e a modelagem do aplicativo, com seus diagramas e descrições de caso de uso. Lançamos mão de diferentes conhecimentos ao longo desse trabalho (orientação a objetos, banco de dados relacional, UML, mapeamento objeto-relacional) permitindo visualizar a importância dos conceitos obtidos ao longo da graduação bem como a elaboração de um projeto adequado ao propósito, conectando e utilizando esses diferentes conceitos.

Os conceitos de implementação de software e as tecnologias utilizadas neste trabalho, juntamente com o desenvolvimento do aplicativo, resultou em uma importante contribuição para o aprendizado e fixação desses conhecimentos na prática, obtendo assim grandes avanços do ponto de vista pessoal e profissional.

Do ponto de vista de contribuição à sociedade, através da construção do sistema em sua versão candidata, é possível demonstrar a viabilidade do uso da tecnologia a favor da sociedade, cumprindo seu papel mais nobre de estreitar os laços entre tecnologia e humanidade.

Sob os aspectos aqui citados, pode-se concluir que o aplicativo desenvolvido, além de atender aos requisitos propostos inicialmente, também pôde comprovar na prática, de maneira satisfatória, o uso de tecnologia em tarefas diárias.

REFERÊNCIAS

NILDERAUER JULIANO: Guia de Consulta Rápida. Editora Novatec, 2011.

Álvares, Miguel. Disponível em: <http://www.criarweb.com/artigos/230.php>.

Publicado em: 30/10/04. Acesso em: 18/04/2011.

Sylvia Campos, Francisco Leal, João Henrique e Paulo Borba; 2008. Disponível em :
<<http://www.cin.ufpe.br/~phmb/ip/MaterialDeEnsino/IntroducaoAoEclipse/IntroducaoAoEclipse.htm>>. Acesso em: 20/04/ 2011.

D'ÁVILA, M. **Tomcat**. Disponível em:
<<http://www.mhavila.com.br/topicos/java/tomcat.html#intro>>. Acesso em: 19/04/2011.

King, Gavin; Bauer, Christian; Andersen, Max Rydahl; Bernard, Emmanuel; Ebersole, Steve. ***Hibernate Reference Documentation***. 2009.

Macoratti, J. Carlos. **UML – Conceitos Básicos II**. Disponível em:
<http://www.macoratti.net/vb_uml2.htm>. Acesso em: 18/04/2011.

SUN J2EE. Java 2 Enterprise Edition Technology Center. Disponível em:
<<http://java.sun.com/developer/technicalArticles/J2EE>>. Acesso em
24/06/2011.

Chaure, Javier. O básico para começar a desenvolver com Google Maps I.
Disponível em: < <http://www.criarweb.com/artigos/basico-comecar-desenvolver-google-maps.html> >. Acesso em 25/06/2011

LECHETA, Ricardo. Google Android 2º Edição, Editora Novatec, 2010.

AZEVEDO, C.: Imasters: Por Uma Internet Mais Criativa e Dinâmica,
Disponível

em: <http://imasters.uol.com.br/artigo/7832/programacao/google_maps_api/>.
Acesso em Outubro de 2011.

GOOGLE DEVELOPER GUIDE.: Google Maps Java API Data, Disponível em:
<http://code.google.com/intl/pt-BR/apis/maps/documentation/mapsdata/developers_guide_java.html>. Acesso
em
Outubro de 2011.

GOOGLE MAPS. Disponível em: <<http://maps.google.com.br/maps?ct=reset>>.
Acesso em Outubro de 2011.