



UNISUL

UNIVERSIDADE DO SUL DE SANTA CATARINA

BRUNO FINGER

GIÁCOMO PASA RIBEIRO

**CARDAPIO DIGITAL PARA TABLETS:
GERENCIANDO PEDIDOS COM A PLATAFORMA ANDROID**

Palhoça
2012

**BRUNO FINGER
GIÁCOMO PASA RIBEIRO**

**CARDÁPIO DIGITAL PARA TABLETS:
GERENCIANDO PEDIDOS COM A PLATAFORMA ANDROID**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade do Sul de Santa Catarina, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Saulo Popov Zambiasi, Dr.

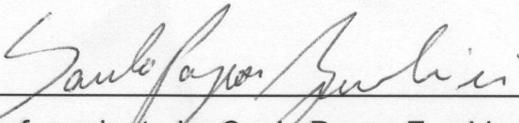
Palhoça
2012

**BRUNO FINGER
GIÁCOMO PASA RIBEIRO**

**CARDÁPIO DIGITAL PARA TABLETS:
GERENCIANDO PEDIDOS COM A PLATAFORMA ANDROID**

Este Trabalho de Conclusão de Curso foi julgado adequado à obtenção do título de Bacharel em Ciência da Computação e aprovado em sua forma final pelo curso de Ciência da Computação, da Universidade do Sul de Santa Catarina.

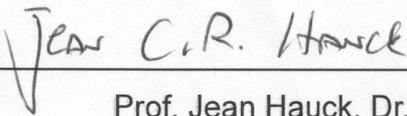
Palhoça, 19 de novembro de 2012.



Prof. e orientador Saulo Popov Zambiasi, Dr.
Universidade do Sul de Santa Catarina



Prof. Márcio Ghisi Guimarães, Msc.
Universidade do Sul de Santa Catarina



Prof. Jean Hauck, Dr.
Universidade do Sul de Santa Catarina

RESUMO

Este trabalho consiste no desenvolvimento de um protótipo de um aplicativo para a plataforma Android, com o objetivo de utilizar dispositivos móveis, porém com foco em *tablets*, para substituir os cardápios impressos. Durante este documento, são discutidos tópicos como: por que esta ideia, o desenvolvimento da informática até a chegada dos dispositivos móveis, o que é a plataforma Android, objetivos desta proposta, modelagem do protótipo. São apresentados sistemas similares ao que os autores propõem, e as diferenças entre eles. Há uma apresentação da modelagem do protótipo e banco de dados utilizando-se inúmeros diagramas, explicações sobre o padrão UML, o ICONIX e os requisitos do sistema. Ao entrar no desenvolvimento do protótipo, os autores explicam as tecnologias envolvidas para se alcançar o objetivo, entre elas o Android SDK, fundamental para o desenvolvimento de aplicativos para esta plataforma, softwares para criar diagramas utilizados na modelagem, como os softwares Dropbox e Subversion foram utilizados para garantir sincronia entre os autores, o ambiente de desenvolvimento utilizado, linguagem de programação e sistema gerenciador de banco de dados. Mais a frente é explicado com mais detalhes como o sistema funciona e as vantagens que o uso de um sistema digital pode trazer sobre um cardápio de papel, mostrando capturas de tela do sistema e explicando o que ocorre. Finalmente, é discutida a conclusão dos autores sobre o funcionamento real deste sistema e o planejamento do futuro deste projeto.

Palavras-chave: Android. *Tablet*. Java. Cardápio. Dispositivos móveis. Pedidos.

ABSTRACT

This paper consists on the development of a prototype of an application for the Android platform, with the objective of use mobile devices, focusing on tablets, to replace paper menus. Through this document, we discuss topics such: why this idea, the development of the informatics until the creation of the mobile devices, what is the Android platform, the objectives of this propose, prototype modeling. It's presented similar systems to the author's proposition and the differences between them. There is an explanation of the modeling of the database and the prototype using numerous diagrams, explanation about the ICONIX methodology and the UML modeling language, and the system requisites. Going into the prototype's development, the authors explain the involved technologies used to reach the objectives, such as the Android SDK, essential for the development of applications in the platform, softwares used to create the diagrams used in the modeling, how the authors used softwares such Dropbox and Subversion to keep synchronization between each other, the development environment used, programming language and database management system. Further more, it's explained with more details how the system works and the advantages of using a digital system over a paper menu, showing screen shots of the system and explaining what occurs. Finally, it is discussed the conclusion of the authors of the real functioning of this system and the planning of the future of this project.

Keywords: Android. Tablet. Java. Menu. Mobile devices. Order.

PROLOG

Niniejsza praca zawiera opracowanie prototypu aplikacji dla platformy Android, do wykorzystania na telefonach komórkowych jak również na tabletach, w celu zminimalizowania papierowych zapisów. W pracy poruszone zostały takie kwestie jak: w jakim celu aplikacja została stworzona, rozwój dziedziny informatyki, od początku aż po stworzenie mobilnych urządzeń, czym jest platforma Android, cel aplikacji, prototyp aplikacji. W dalszej części opracowania przedstawione zostały systemy które istnieją już na rynku, jednakże różnią się od prototypu zaprezentowanego przez autorów niniejszej pracy. Praca zawiera również wyjaśnienie modelu bazy danych oraz prototypu który oparty jest na użyciu diagramów numerycznych. Zostały wyjaśnione również zagadnienia takie jak metodologia ICONIX, praca w języku UML jak również przedstawione zostały wymagania systemowe. Dążąc do rozwoju prototypu, autorzy przedstawiają wpływ postępu technologii (Android SDK) na osiągnięcie celu, którym jest rozwój aplikacji w platformie. W pracy zostaną zaprezentowane także programy które zostały wykorzystywane przez autorów do tworzenia diagramów (Dropbox i Subversion). Aby programy mogły zachować między sobą synchronizację, wykorzystano środowisko programistyczne, język programowania i system zarządzania bazą danych. Co więcej, w pracy zostało wyjaśnione ze szczegółami, jak działa system, jakie są zalety korzystania z cyfrowego systemu nad danymi zachowanymi na papierze. Poprzez zaprezentowanie zdjęć w prosty sposób przedstawione są kolejne etapy użytkownika aplikacji. W podsumowaniu, zaprezentowane jest wdrożenie w życie codzienne jak również planowana wizja przyszłości aplikacji.

Słowa kluczowe: Android. Tablet. Java. Menu. Mobilne urządzenia. Zamówienie.

LISTA DE SIGLAS

ADB – Android Debug Bridge
ADT – Android Development Tools
DAO – Data Access Object
ER – Entidade-relacionamento
HTML – Hypertext Markup Language
IDE – Integrated Development Environment
IP – Internet Protocol
SDK – Software Development Kit
SVN – Subversion
UML – Unified Modeling Language
URL – Uniform Resource Locator

LISTA DE ILUSTRAÇÕES

Figura 1: Camadas do Software Android.....	20
Figura 2: Máquina virtual Dalvik.....	21
Figura 3: Captura de tela do menu principal do cardápio Easymenu.....	23
Figura 4: Etapas Metodológicas do projeto.....	31
Figura 5: Diagrama do esquema tecnológico.....	32
Figura 6: Diagrama de Implantação.....	33
Figura 7: Módulo do Tablet.....	34
Figura 8: Módulo do Servidor.....	35
Figura 10: Requisitos Funcionais.....	39
Figura 11: Requisitos não funcionais.....	42
Figura 12: Diagrama entidade-relacionamento.....	43
Figura 13: entidade Item e seus atributos.....	44
Figura 14: entidade Categoria e seus atributos.....	44
Figura 15: entidade Foto e seus atributos.....	44
Figura 16: entidade Ingrediente e seus atributos.....	45
Figura 17: entidade Tipo e seus atributos.....	45
Figura 18: entidade Comentário e seus atributos.....	46
Figura 19: entidade Pedido e seus atributos.....	46
Figura 20: entidade Promoção e seus atributos.....	46
Figura 21: entidade Cliente e seus atributos.....	47
Figura 22: entidade Cozinheiro e seus atributos.....	47
Figura 23: Pacote “bean” e respectivas classes.....	49
Figura 24: Pacotes “util”, “gui”, “main” e respectivas classes.....	50
Figura 25: Pacote “dao”, subpacotes e respectivas classes.....	51
Figura 26: Pacote “servidor”, subpacotes e respectivas classes.....	52
Figura 27: Diagrama de Casos de Uso.....	53
Figura 28: Diagrama de atividades dos casos de uso UC001 – Listar Ingrediente, UC002 – Cadastrar Ingrediente, UC003 – Editar Ingrediente e UC004 – Excluir Ingrediente.....	61
Figura 29: Diagrama de Robustez dos casos de uso UC001 – Listar Ingrediente, UC002 – Cadastrar Ingrediente, UC003 – Editar Ingrediente e UC004 – Excluir Ingrediente.....	62
Figura 30: Diagrama de Sequência do UC001 – Listar Ingrediente.....	63
Figura 31: Diagrama de Sequência do caso de uso UC002 – Cadastrar Ingrediente	64
Figura 32: Diagrama de Sequência do caso de uso UC003 – Editar ingrediente....	65
Figura 33: Diagrama de Sequência do UC004 – Excluir Ingrediente.....	66
Figura 34: Modelo de Domínio.....	67
Figura 35: Emulador de dispositivos virtuais do Android SDK.....	71
Figura 36: Captura de tela do software Dia.....	72
Figura 37: Captura de tela da estrutura de pastas do aplicativo Dropbox para Android.....	73
Figura 38: Captura de tela do editor de layout do ADT no Eclipse.....	75
Figura 39: Captura de tela do software Enterprise Architect.....	76
Figura 40: Captura de tela do software Pencil.....	79
Figura 41: Subversion em ação no ambiente de desenvolvimento Eclipse.....	80
Figura 42: Menu principal do Cardápio Digital.....	82

Figura 43: Menu de pratos.....	83
Figura 44: Detalhamento de um item.....	84
Figura 45: tela “Minha Conta” com os itens pedidos pelo usuário.....	85
Figura 46: tela de Cadastro de Cliente.....	86
Figura 47: Captura da tela inicial do servidor.....	87
Figura 48: tela de configuração do servidor.....	88
Figura 49: Tela para manipulação de itens.....	89
Figura 50: Tela de cadastro de itens do sistema.....	90
Figura 51: Tela para manipulação de categorias.....	91
Figura 52: Tela de cadastro de categorias.....	91
Figura 53: Tela para manipulação de tipos.....	92
Figura 54: Tela de cadastro de tipo.....	92
Figura 55: Tela para a manipulação de ingredientes.....	93
Figura 56: Tela de cadastro de ingredientes.....	94
Figura 57: Tela para a manipulação de cozinheiros.....	95
Figura 58: Tela de cadastro de cozinheiros.....	96
Figura 59: Tela para a manipulação de promoções.....	97
Figura 60: Tela de cadastro de promoção.....	98
Figura 61: Tela para seleção de itens de uma promoção.....	99

SUMÁRIO

1 INTRODUÇÃO	11
1.1 PROBLEMÁTICA.....	12
1.2 OBJETIVOS.....	13
1.2.1 Objetivo Geral	13
1.2.2 Objetivos Específicos	14
1.3 JUSTIFICATIVA.....	14
1.4 ESTRUTURA DA MONOGRAFIA.....	15
2 REVISÃO BIBLIOGRÁFICA	17
2.1 GERENCIAMENTO DE PEDIDOS EM RESTAURANTES.....	17
2.2 OS TABLETS.....	18
2.3 O SISTEMA OPERACIONAL ANDROID.....	19
2.4 APLICATIVOS DISPONÍVEIS NO MERCADO.....	22
2.4.1 Easymenu	22
2.4.2 SOPHIA - Cardápio Digital	24
2.4.3 Considerações sobre os sistemas pesquisados	25
2.5 O PADRÃO UML.....	25
2.6 O ICONIX.....	26
2.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO.....	27
3 MÉTODO	28
3.1 CARACTERIZAÇÃO DA PESQUISA.....	29
3.2 ETAPAS METODOLÓGICAS.....	30
3.3 PROPOSTA DA SOLUÇÃO.....	32
3.4 DELIMITAÇÕES.....	35
4 MODELAGEM	37
4.1 REQUISITOS.....	37
4.1.1 Regras de Negócio	37
4.1.2 Requisitos Funcionais	38
4.1.3 Requisitos Não Funcionais	42
4.2 DIAGRAMA ENTIDADE-RELACIONAMENTO.....	43
4.3 DIAGRAMAS DE CLASSE E PACOTE.....	48
4.4 CASOS DE USO.....	53
4.4.1 Especificação dos Atores	54
4.4.2 Especificações dos Casos de Uso	55
4.5 DIAGRAMA DE ATIVIDADES.....	60
4.6 DIAGRAMA DE ROBUSTEZ.....	61
4.7 DIAGRAMA DE SEQUÊNCIA.....	63
4.8 MODELO DE DOMÍNIO.....	67
5 DESENVOLVIMENTO	69
5.1 TECNOLOGIAS ENVOLVIDAS.....	69
5.1.1 Android Software Development Kit	70
5.1.2 Dia	71
5.1.3 Dropbox	72
5.1.4 Eclipse	74
5.1.5 Enterprise Architect	75
5.1.6 Java	76
5.1.7 MySQL	77
5.1.8 Pencil	78

5.1.9 Subversion	79
5.2 APRESENTAÇÃO DO SISTEMA.....	81
5.2.1 Cardápio	81
5.2.2 Servidor	86
5.3 CONCLUSÃO DO DESENVOLVIMENTO.....	99
6 CONSIDERAÇÕES FINAIS	100
6.1 SUGESTÕES PARA TRABALHOS FUTUROS	101
REFERÊNCIAS	102

1 INTRODUÇÃO

A sociedade está cada dia mais informatizada, criando mais e mais dispositivos com grandes potenciais. Nesse mesmo sentido, surgiram atualmente os *tablets*. Estes estão criando uma nova oportunidade tecnológica e é de grande importância avaliar todas as suas possibilidades a fim de trazer benefícios quanto à sua utilização, principalmente devido ao seu potencial e sua mobilidade. Segundo Lopes (2011):

O surgimento dos *tablets*, classificados por Steve Jobs, presidente da Apple, como produtos de uma era pós-PC, em conjunto com a popularização dos *smartphones*, está mudando nosso comportamento em relação ao consumo de conteúdo. A portabilidade, somada à autonomia de bateria e à incrível experiência de uso podem colocar em xeque o futuro dos outros computadores pessoais portáteis (LOPES, 2011).

Sobre os *tablets*, Ableson et al. (2012) têm a opinião de que existe oferta de inúmeras marcas conhecidas e desconhecidas e com isto em mente a equipe de desenvolvimento do Android avançou muito ao adicionar funcionalidades específicas de *tablets* à plataforma.

Ableson et al. (2012) também explicam sobre dispositivos móveis:

A maioria dos celulares no mercado continuam a ser telefones do tipo *flip* e *feature phones* – telefones que não são *smartphones*. Esses são aqueles que os consumidores recebem quando entram na loja e perguntam o que podem receber de graça. Esses consumidores são aqueles que dizem “eu só quero um telefone”. [...] Uma das grandes lacunas nas funcionalidades desses telefones básicos é a experiência web do usuário. Parte do problema é o tamanho da tela, mas igualmente desafiador, é a própria tecnologia do navegador, que muitas vezes luta para se igualar à rica experiência dos computadores *desktop* (ABLESON et al., 2012).

Ableson et al. (2012) também falam sobre as preocupações do mercado de *smartphones*, que necessitam de uma plataforma que possa acessar sistemas de informações corporativos e possuir sincronia de dados. A experiência de navegação em *smartphones* e *tablets* é considerada superior devido à sua tecnologia melhorada e melhores métodos de interação com o usuário, como telas sensíveis ao toque.

1.1 PROBLEMÁTICA

Em ambientes que precisam cada vez mais de agilidade, como restaurantes, dispositivos móveis como *tablets* podem trazer uma grande vantagem. Benefícios como: agilização na hora de efetuar os pedidos, apresentação de diversas informações sobre os produtos e dinamicidade no momento de efetuar atualizações no cardápio, são algumas das vantagens que estes dispositivos podem proporcionar devido à sua mobilidade e suporte à tecnologias convenientes de conectividade, como Wi-Fi e 3G, assim citado por Ableson (2009).

Além das vantagens já citadas, outra possibilidade que os *tablets* trazem é a da curiosidade que muitos consumidores têm de utilizar um dispositivo destes, o que pode unir o útil ao agradável, tanto para o usuário quanto para os funcionários e proprietários do estabelecimento.

Outra possibilidade interessante é a opção de os clientes poderem se cadastrar no estabelecimento. Com esse cadastro podem ser feitas propagandas com promoções exclusivas para cadastrados de acordo com seu perfil de consumo, e o estabelecimento tem também a vantagem de ser promovido, de forma gratuita, através das redes sociais por seus clientes se assim desejarem. Segundo a E-Commerce News (2010), os consumidores estão mais exigentes e cada vez com mais voz ativa, e promoções interativas feitas em redes sociais são as estratégias de marketing que estão fazendo a diferença.

A modernização e agilização do estabelecimento é algo que grande parte dos proprietários de restaurantes procuram fazer na medida do possível. Implantar algum sistema, ou dispositivo tecnológico, pode ser uma das soluções que os auxiliem nisso. Segundo Jucá (2008):

Um ciclo vicioso é aumentar o número de atendentes para suprir a demanda de pedidos. Porém, o alto número de garçons também acaba atrapalhando. E é aí que a tecnologia faz sua parte.

Alguns dos principais produtos que conquistaram seu espaço no mercado são as comandas eletrônicas e os terminais de toque.

Luis explica que com esses aparelhos o garçom pode dedicar a maior parte de seu tempo em dar atenção aos seus clientes, sugerir-lhes opções de pratos e bebidas e tudo mais. “Dessa forma, todo mundo sai ganhando, o cliente pela agilidade, e a casa pela boa prestação do serviço” (JUCÁ, 2008).

Outra característica que também pode trazer bastante benefício é um *software* que visa ter preços mais acessíveis como algum de uma plataforma livre.

Problemas, tais como: erros no fechamento da conta, divergência entre o que foi pedido e o que foi realmente recebido pelo cliente, reimpressão de todo o cardápio em papel em grande quantidade para a sua atualização, entre outros, são coisas que podem ser evitadas com a utilização de um *software* capaz de gerenciar estes dados.

Sistemas similares já existem para o iOS, plataforma móvel da Apple, e inclusive, para o Android, porém, a proposta deste trabalho é de um sistema mais dinâmico, uma vez que o conteúdo do cardápio se adapta, caso o cliente opte por logar-se com sua conta do restaurante no *tablet*, como por exemplo, mostrando promoções de acordo com o gosto deste cliente.

No decorrer deste documento, são apresentados diversos conceitos utilizados para a confecção do trabalho. É mostrado um pouco da evolução da computação no decorrer dos últimos anos até os dias atuais. Também é mostrado um breve histórico do sistema Android, e explicada a sua estrutura.

A ideia do trabalho é criar um sistema de gestão de pedidos de restaurante e estabelecimentos relacionados, e informatizar o cardápio, substituindo-o por *tablets*, mais especificamente *tablets* da plataforma Android, por se tratar de um sistema novo e livre. Faz parte deste trabalho também a implementação de um servidor, que é destinado a receber todos os pedidos efetuados pelos clientes, assim como os dados de cadastro, entre outras informações. Para o sistema também é criada uma base de dados que contém todos os pratos, fotos, dados de cozinheiros e de clientes do estabelecimento.

1.2 OBJETIVOS

A seguir são apresentados os objetivos deste trabalho.

1.2.1 Objetivo Geral

O presente trabalho objetiva apresentar uma proposta de modelagem e protótipo de implementação de um sistema de gerenciamento de pedidos em restaurantes utilizando *tablets*.

1.2.2 Objetivos Específicos

Como objetivos específicos têm-se:

- Estudo do estado-da-arte de aplicações já existentes que trabalham no mesmo sentido da proposta.
- Criação da documentação, modelagem e requisitos necessários para o sistema.
- Desenvolvimento de uma instância de implementação (protótipo) baseado na modelagem apresentada.

1.3 JUSTIFICATIVA

Este trabalho propõe um sistema de cardápio digital e informatização de pedidos, disponibilizando em cada mesa do estabelecimento um *tablet*, contendo todos os pratos e bebidas disponíveis. Com o uso deste aplicativo não existe a necessidade de reimpressão de todos os cardápios sempre que houver alguma mudança, o mesmo se atualiza automaticamente de acordo com elas. Grego (2009) comenta o seguinte:

A regra óbvia para economizar com impressão é só mandar para o papel o que for realmente imprescindível. Boletos, recibos, páginas da web e outros documentos podem ser armazenados no próprio micro em formato PDF ou XPS. Se, em algum momento, você precisar de uma cópia em papel, bastará imprimir o arquivo. Como essas ocasiões serão raras, você vai economizar papel, tinta e energia. [...] Economia anual: 80 reais (deixando de imprimir 3 página por dia) (GREGO, 2009)

Com o uso deste tipo de sistema, o cliente do estabelecimento pode atender a si mesmo, deixando os garçons com mais tempo livre. Neste caso, os garçons podem ser alocados para outras atividades, precisando apenas levar os pedidos até a mesa.

O aplicativo proposto aproveita o potencial dos *tablets* com o sistema operacional Android por ser totalmente livre e *open-source*. Estes dispositivos estão cada vez mais populares devido a preços mais acessíveis, e não só por isso, mas também pelo seu tamanho, mobilidade e desempenho. A respeito disso, Lopes (2011) ainda fala que uma “Estimativa da IDC aponta que as vendas de *tablets* por aqui ultrapassem 300 mil unidades neste ano. 'Cerca de 80% devem ser de iPad', diz Luciano Crippa, coordenador de pesquisas.”

Alguns sistemas semelhantes já foram criados para o iPad, *tablet* da Apple, que é uma plataforma com sistema fechado, um deles é o MenuPad. Existem também sistemas para o Android, como o Easymenu.

A motivação da utilização de *tablets*, além do que foi citado anteriormente, é pelo fato de que o mesmo é um dispositivo ideal e prático para o consumo de conteúdos como o cardápio digital em questão, no qual o usuário, se assim desejar, não precisa entrar com nenhum dado na aplicação, ou seja, não precisa digitar nada.

A razão dos autores de escolher o Android como plataforma é a sua natureza livre. Todo o aplicativo desenvolvido para a comunidade livre gera um avanço na tecnologia atual.

Vale lembrar que o objetivo do sistema não é substituir os garçons do estabelecimento, mas desenvolvido tendo em mente para o mesmo ser pelo menos uma forma de auxiliá-los.

1.4 ESTRUTURA DA MONOGRAFIA

Este trabalho está organizado como segue a descrição:

- **Capítulo 1:** Introdução, apresenta qual o assunto, o tema, a justificativa e os objetivos.

- **Capítulo 2:** Apresenta a revisão bibliográfica. É explanado o motivo da utilização de *tablets*, assim como um breve histórico da evolução dos computadores até a chegada dos dispositivos móveis; apresenta um breve resumo de aplicações similares já existentes no mercado e como elas auxiliaram na criação do sistema; também, como foi feita a pesquisa do funcionamento e gerenciamento de pedidos em um restaurante.
- **Capítulo 3:** Descreve como a monografia foi planejada; apresenta também as delimitações, cronograma e recursos necessários.
- **Capítulo 4:** Mostra a modelagem do sistema com todos os diagramas elaborados e explicações sobre o mesmo.
- **Capítulo 5:** Apresenta tecnologias envolvidas no desenvolvimento do sistema, o funcionamento do protótipo implementado e explicações de suas telas.
- **Capítulo 6:** Discute conclusões dos autores sobre a funcionalidade do sistema e considerações sobre o futuro do mesmo.

2 REVISÃO BIBLIOGRÁFICA

Neste capítulo são apresentados os motivos da escolha dos *tablets* como ponto focal do tema deste trabalho, a evolução dos dispositivos eletrônicos até o ponto de se tornarem móveis e o desenvolvimento dos *tablets*; a história do sistema operacional Android e as diferenças entre a proposta apresentada e os aplicativos já disponíveis no mercado.

2.1 GERENCIAMENTO DE PEDIDOS EM RESTAURANTES

No mercado atual estão presentes diversos tipos de restaurantes e cada um possui sua própria forma de gerenciar os pedidos.

Existem aqueles estabelecimentos que disponibilizam todos os seus pratos e bebidas em um *buffet* livre ou por peso, onde os próprios clientes servem-se do que desejam. Outros disponibilizam somente os pratos no *buffet* e as bebidas são pedidas aos garçons que atendem nas mesas. Outros, em que os pratos ficam em estruturas, onde o cliente escolhe o que quer e funcionários vão servindo essas escolhas no prato, fazendo com que o cliente não tenha contato direto com os alimentos. Existem, também, os restaurantes a *la carte*, em que todos os pratos e bebidas ficam disponíveis em cardápios físicos ou virtuais, e o cliente escolhe o que quer e faz os pedidos a garçons. Segundo Rebelato (1997):

Um estudo de caso foi realizado em um restaurante comercial localizado na cidade de São Carlos (SP) com o objetivo de avaliar questões estratégicas e operacionais do negócio. [...] A empresa em questão foi criada em 1992 como um bar. Evoluiu rapidamente e mudou seu foco de negócio, trabalhando atualmente com serviço de *buffet* e restaurante *self-service* “por quilo”. No início das atividades não havia nenhum funcionário contratado. [...] Com o tempo a empresa foi se expandindo e conquistando novos clientes. Opera, atualmente, com 45 pessoas contratadas e serve cerca de 18.000 refeições mensais. [...] O estudo de campo revelou as dificuldades de se gerenciar a capacidade de fornecimento. A grande variação da demanda no tempo é problemática para a empresa. Os restaurantes *self-service*, na média, operam com capacidade ociosa pois não trabalham no horário do jantar e, mesmo que ofereçam serviço de *buffet*, é improvável que este aconteça todos os dias (REBELATO, 1997).

Rebelato (1997) também explica que nos restaurantes *self-service* a produção não trabalha por encomenda. O cardápio, na verdade, tem de ter flexibilidade para acomodar a oferta das matérias-primas.

O registro de tudo o que foi consumido pelos clientes é feito de diversas formas, que vai desde simples pedaços de papel, os quais ficam nas mesas dos clientes ou com os funcionários, até sistemas com *pocket PCs* para garçons e, ultimamente, com *tablets*.

2.2 OS TABLETS

A informática percorreu um longo caminho até chegar onde está hoje. Meneses (2011) dá uma breve explicação sobre a sua evolução. Segundo ele, antes de falar de *tablets*, devemos lembrar dos imensos *mainframes* do final da década de 70, estes que ocupavam até prédios inteiros.

Não se pode esquecer também do trabalho de Steve Jobs e Steve Wozniak para trazer ao mercado o primeiro computador pessoal, o Apple II, algo que na época foi uma revolução que mudou a história (WOZNIAK, 2012). Após a criação do computador pessoal, vieram finalmente os *notebooks*, computadores pessoais verdadeiramente portáteis.

Mas foi a Compaq que lançou em 82 o primeiro computador portátil compatível com o IBM PC (referência em *desktop* até então). Abrigavam seus 12,5 kg um CPU de 4,77 MHz, memória RAM de 128 KB, disco de 320 KB e um monitor CRT de 9 polegadas, o que permitia que o computador fosse dotado de um sistema operacional DOS. Apesar do seu alto custo, U\$ 3.500,00, sua compatibilidade fez do portátil da Compaq um sucesso e promoveu a marca. (PRADA, 2009).

Porém, a mobilidade está se tornando não só cada vez mais comum, como também uma exigência dos consumidores. Segundo Lobato (2011) em uma reportagem para a Folha, “O uso do PC ficará restrito a nichos profissionais, que somarão no máximo 6% do total de usuários de internet.”. Lobato ainda explica que essa é uma consequência das conexões móveis já prevista pelo pesquisador do Centro para o Futuro Digital da University of Southern California, Jeffrey Cole.

Ainda a respeito de mobilidade, conforme Conexão Digital (2012):

Muito se fala de mobilidade e praticidade, além da interatividade, quando o assunto envolve esses dispositivos móveis. Ambos os *gadgets* oferecem recursos e possibilidades parecidas: acesso à internet, redes sociais, e-mail, aplicativos, registro e exibição de fotos e vídeos, registro de anotações e outras infinitas funcionalidades. (REBELATO, 1997)

Por fim, Pereira (2011) considera que estamos vivendo uma época semelhante à transição dos imensos *mainframes* para os computadores de mesa conhecidos hoje. *Desktops* já viraram *laptops* e agora estamos diante de computadores menores ainda, mais poderosos que computadores de 10 anos atrás e que podemos carregar para todos os lugares conosco.

2.3 O SISTEMA OPERACIONAL ANDROID

O sistema operacional Android é um sistema para dispositivos móveis da empresa Google, baseado na *kernel* Linux. Aplicativos desta plataforma são escritos na linguagem de programação Java. Ableson (2009) conta uma breve história do Android. Segundo ele, é o produto do *Open Handset Alliance*, uma colaboração liderada pelo Google de inúmeras empresas dos mais variados tipos, entre elas telefonia móvel, fabricantes de peças e *marketing*.

O primeiro dispositivo capaz de executar o Android foi o HTC G1, distribuído pela operadora americana *T-Mobile*. Na época, havia apenas rumores de que o Google estava desenvolvendo um sistema operacional.

Durante o período de um ano, o dispositivo seguiu se aprimorando, contando com apenas algumas versões básicas do *Software Development Kit* (*SDK*) e foi somente após este período que foi disponibilizada a primeira versão oficial do *SDK*, tornando finalmente possível o desenvolvimento de aplicativos verdadeiramente úteis para a plataforma.

Com o intuito de estimular o desenvolvimento de aplicativos para o Android, o Google criou o “*Android Developer Challenge*”, uma espécie de competição na qual os desenvolvedores do melhor aplicativo do ano de uma série de categorias, selecionados pela própria empresa, recebem um prêmio de cem mil dólares.

O *Android Market* (atualmente Google Play), aplicativo que serve como uma espécie de índice para outros aplicativos, permite o *download* e instalação dos mesmos diretamente no dispositivo, foi lançado alguns meses depois do G1.

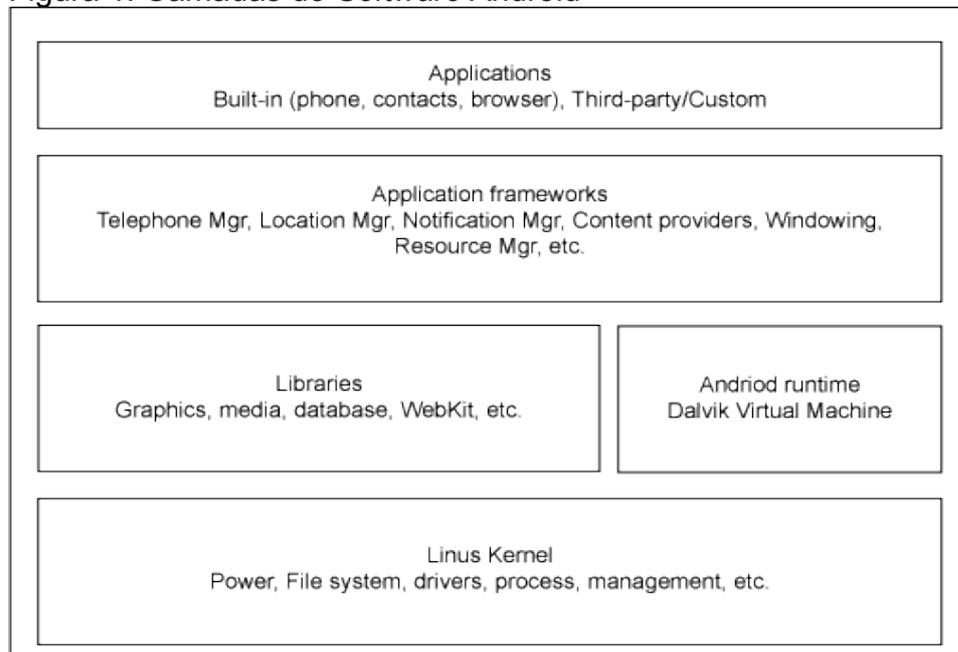
Ableson cita um vasto número de funcionalidades do sistema. Segundo ele, o sistema Android é um sistema em camadas baseado em Linux, com grande suporte à conectividade, incluindo *Wi-Fi*, *Bluetooth*, conexão celular como EDGE, GPRS, 3G, suporte à tecnologias de localização, como GPS e acelerômetro. O sistema também conta com suporte à aceleração 2D e 3D, usando a biblioteca de código livre OpenGL. Ele oferece o SQLite, *software* também de código livre, como banco de dados nativo.

Para navegação na *web*, o Android possui a biblioteca WebKit, desenvolvida pela Google em parceria com outras empresas e também de código livre.

Ableson também explica brevemente como funciona a arquitetura do Android.

Na Figura 1, pode-se visualizar as camadas do sistema Android.

Figura 1: Camadas do *Software* Android



Fonte: Ableson (2009)

Em sua camada de mais baixo nível, encontra-se a *kernel* Linux, responsável pela energia, sistema de arquivos, drivers, processos, gerenciamento do sistema etc.

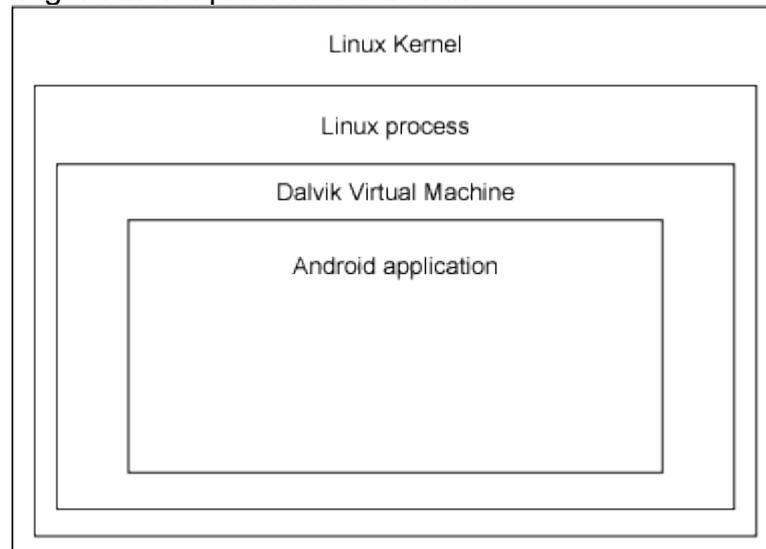
Em um nível mais elevado, é possível observar a camada de bibliotecas e a máquina virtual Dalvik. Nessa camada encontra-se a biblioteca gráfica (OpenGL), de mídia, banco de dados e navegação da *web*. A máquina virtual Dalvik é uma implementação da Google de uma máquina virtual para a linguagem Java, porém otimizada para o sistema Android.

Acima dessa camada está o framework de aplicativos. É nessa camada que ocorre o gerenciamento das principais funções do Android, como telefone, localização, notificação, provedores de conteúdo, janelas e recursos.

Por último está a camada de aplicações. É nessa camada que encontram-se aplicativos de terceiros e utilitários do sistema e faz o uso de todos os recursos oferecidos pelas camadas inferiores.

Finalizando sua explicação, Ableson mostra a arquitetura da máquina virtual Dalvik (Figura 2).

Figura 2: Máquina virtual Dalvik



Fonte: Ableson (2009)

Segundo Ableson, os aplicativos para Android são escritos na linguagem de programação Java, como já explicado anteriormente. Porém, é importante observar que esse código não é interpretado por uma *Java Virtual Machine (JVM)*, e

sim pela *Dalvik Virtual Machine*. A diferença é que esta é uma implementação de *software* livre da outra, e também otimizada para a execução na arquitetura dos dispositivos móveis.

Cada aplicativo do Android é executado em uma instância separada da *Dalvik Virtual Machine*, e cada instância reside em um processo gerenciado pela *kernel* Linux.

2.4 APLICATIVOS DISPONÍVEIS NO MERCADO

A seguir são descritos dois aplicativos de cardápios digitais feitos para dispositivos móveis disponíveis no mercado. Os dois são similares ao sistema desenvolvido. Um deles, Easymenu, utiliza a plataforma Android e outro, Sophia, feito para a plataforma iOS.

Os dois aplicativos eram os únicos existentes no mercado que tinham objetivos semelhantes aos deste trabalho quando a pesquisa foi elaborada, por isso a escolha dos mesmos. Atualmente existem mais aplicativos.

2.4.1 Easymenu

Segundo Easymenu (2011), o Easymenu é um aplicativo desenvolvido para *tablets* com o sistema operacional Android. Ele tem o intuito de informatizar o cardápio e a parte de pedidos de um restaurante. Os pedidos de pratos feitos pelos clientes vão diretamente para o servidor na cozinha e os pedidos de bebidas vão para um servidor ao qual os garçons têm acesso.

O sistema busca ser bastante customizável. Nele é possível acrescentar ou remover opções do menu em tempo real. Oferece também a possibilidade de multi-idiomas. Um dos principais objetivos da empresa desenvolvedora é o de proporcionar aos clientes uma experiência diferente e eficiente na forma de fazer os pedidos. O aplicativo fornece imagens dos produtos,

sugestões de pratos similares e a opção de fechamento de conta pelo dispositivo. Também dá suporte à criação de uma carta de vinhos.

Ele possui uma interface bastante intuitiva e disponibiliza uma ajuda *online*, para facilitar o entendimento dos clientes, como mostrado na Figura 3.

Figura 3: Captura de tela do menu principal do cardápio Easymenu



Fonte: Easymenu (2011)

Antes de utilizar os *tablets* é necessário configurar o servidor do restaurante. São configurações como a escolha do endereço IP do servidor, entre outras.

Nos *tablets* é preciso cadastrar qual é o IP do servidor, a moeda e a cor de fundo (preta ou branca) que será exibida.

Cada mesa do estabelecimento deve possuir um *login* e senha. Quando se inicializa o aplicativo no *tablet*, um funcionário precisa colocar o *login* e senha da respectiva mesa onde o dispositivo ficará.

O aplicativo para Android encontra-se atualmente na versão 2.01. Seu último *update* foi no dia 5 de julho de 2011. No *site* do produto existem duas versões *demo* disponíveis para *download*, uma para as versões 1.5 - 2.1 do Android, e outra, para a versão 2.2 desse sistema operacional.

O servidor Easymenu Server está atualmente na versão 2.0. Ele e um *patch* de atualização se encontram também disponíveis no *site* da empresa para

download. Ele é baseado em *Vfront* que é *open-source*. Foi desenvolvido para rodar em ambientes com o sistema operacional Linux ou no Microsoft Windows com uma máquina virtual Linux. Ele precisa de um computador ou servidor dedicado.

O cardápio possui dois módulos. Um deles é o passivo em que o dispositivo móvel funciona apenas como um cardápio digital, sem a possibilidade de efetuar pedidos pelo mesmo. O outro módulo é o *self-order* onde o *tablet*, além de ser o cardápio, serve para fazer os pedidos e mantém a lista do que já foi escolhido pelo cliente.

A empresa sugere a utilização de um ou mais *tablets* por mesa, sendo que, preferencialmente, só um esteja no módulo *self-order*.

No *website* dos desenvolvedores não foi encontrado nenhuma informação sobre o preço para o aplicativo.

2.4.2 SOPHIA - Cardápio Digital

O SOPHIA é um sistema de cardápio digital feito para os *tablets* iPad e iPad 2 da Apple com o sistema operacional iOS 4.2 ou superior, e recentemente, também Android 2.2 ou superior. Ele foi desenvolvido pela CVW - Soluções Tecnológicas. O aplicativo tem por objetivo ser um cardápio interativo e prático que traz a possibilidade de fechamento de conta entre outras funções (SOPHIA...2011).

Segundo a empresa, a aparência do aplicativo é totalmente personalizável, para seguir as características do estabelecimento.

Uma vantagem, que pode ser obtida com o uso do Sophia, é a diminuição da mão de obra, pois não serão necessários tantos funcionários para o atendimento de clientes. Outra vantagem que um estabelecimento tem usando este aplicativo, segundo a CVW - Soluções Tecnológicas (2011) no website do aplicativo SOPHIA, é a seguinte: "Como se trata de um produto da Apple, que investe cada vez mais em marketing e tecnologia, além de agregar a imagem de uma marca de status, o estabelecimento será bem conceituado por deter destes quesitos."

A empresa também defende que esse programa torna fácil a tarefa de atualização de produtos e preços contidos no cardápio. O aplicativo possui suporte para três idiomas: Português, Inglês e Espanhol.

Após todos os produtos estarem cadastrados no sistema, é necessário incluir o número da mesa no aplicativo do *tablet*. Isso faz com que o mesmo se comunique diretamente com o servidor principal. A partir daí todos os pedidos ficam registrados vinculados a esse número da mesa. O número da mesa só pode ser alterado no aplicativo quando o mesmo for fechado. Vale observar que cada mesa fica com apenas um *tablet*.

2.4.3 Considerações sobre os sistemas pesquisados

A pesquisa e a descoberta dos dois sistemas citados anteriormente, que já se encontravam disponíveis no mercado, foi de grande ajuda para a concretização deste trabalho. Os mesmos serviram como exemplos em muitas das suas funcionalidades. Proporcionaram também inspiração para novas funcionalidades que os mesmos não possuem, ou pelo menos não possuíam nas versões testadas.

2.5 O PADRÃO UML

UML significa em inglês *Unified Modeling Language*, e segundo a Object Management Group (2011, p. 9), ela é uma linguagem visual criada para especificar, construir e documentar os artefatos de sistemas. Ela é uma linguagem de modelagem de propósitos gerais que pode ser utilizada com a grande maioria dos métodos de objetos e componentes, e pode ser aplicada a todos os domínios de aplicação (por exemplo, saúde, finança, telecomunicação, aeroespacia) e plataformas de implementação (por exemplo, J2EE, .NET). De acordo com a Globalcode (2004, p.33), “A UML consiste de um número de elementos gráficos combinados para formar diagramas. Pelo fato de ser uma linguagem, a UML tem regras para a combinação desses elementos.”

A Globalcode (2004, p.33) ainda explica que nos anos 80 e começo dos anos 90, Grady Booch, James Rumbaugh e Ivar Jacobson desenvolviam em

separado suas próprias metodologias para análise e projeto de *softwares* orientados a objeto em suas empresas. Foi então que na metade dos anos 90, os três resolveram juntar o que havia de bom em cada uma de suas metodologias para criar a UML.

Em seguida, a UML foi padronizada pelo *Object Management Group* (OMG). Nas palavras da Globalcode:

Em 1997 o consórcio criado pelas grandes empresas do mercado, tais como, Oracle, HP, Texas Instruments, Rational, produziram a versão 1.0 da UML e submeteram para aprovação do OMG [...]. Mais tarde uma outra versão, a 1.1, foi submetida para OMG, que adotou a UML como linguagem de modelagem de software padrão. A partir de então o OMG passou a manter e produzir novas versões da UML. (GLOBALCODE, 2004)

No decorrer do capítulo 4 (Modelagem) é feita uma explicação de alguns dos diagramas da UML que foram utilizados para o desenvolvimento do sistema proposto.

2.6 O ICONIX

O ICONIX é um processo de desenvolvimento de *software* que aplica a UML. O mesmo foi o escolhido para a elaboração da aplicação.

Segundo Maia (2005 apud. SILVA et. al 2007):

O processo ICONIX trabalha a partir de um protótipo de interface onde se desenvolvem os diagramas de caso de uso baseados nos requisitos levantados. A partir dos diagramas de caso de uso, se faz a análise robusta para cada caso de uso. Com os resultados obtidos é possível desenvolver o diagrama de seqüência e, posteriormente, complementar o modelo de domínio com novos métodos e atributos descobertos.

Os principais modelos e diagramas do ICONIX são: o modelo de domínio, diagrama de casos de uso, diagrama de robustez, diagrama de seqüência e diagrama de classes.

Também no capítulo 4 (Modelagem) há uma explicação dos principais tipos de diagramas do ICONIX e de alguns outros pertencentes a UML que foram utilizados no desenvolvimento do projeto.

2.7 CONSIDERAÇÕES FINAIS DO CAPÍTULO

Durante a elaboração da revisão bibliográfica conseguiu-se expandir ainda mais o conhecimento sobre o funcionamento dos restaurantes, principalmente com a ajuda da pesquisa de campo feita com o auxílio de alguns estabelecimentos.

Foi explicada quando surgiu e como é a estrutura do Android e suas camadas.

A revisão ajudou a ter acesso a alguns aplicativos que possuem funções interessantes e serviram de base para o desenvolvimento do projeto.

3 MÉTODO

Este capítulo aborda os meios usados para se implementar o *software* como um todo. Conforme já discutido nos capítulos anteriores, o *software* é implementado em duas partes: cliente (os *tablets*) e servidor.

O projeto de ambas as partes é implementado na linguagem de programação Java, utilizando o ambiente de desenvolvimento Eclipse.

Como meio de controle de versionamento do código-fonte, é utilizada a tecnologia *Subversion* (SVN), utilizando o *website* SourceForge.net como meio de hospedagem do mesmo.

Durante o processo de modelagem, fez-se uso de dois *softwares* para a elaboração dos diagramas, o Dia (live.gnome.org/Dia/), desenvolvido pelo *GNOME Project*, e o *Enterprise Architect* em outras ocasiões, da *Sparx Systems* (www.sparxsystems.com/products/ea/index.html). Tais diagramas e outros elementos fundamentais de modelagem foram mantidos em sincronia entre os autores utilizando-se os serviços na nuvem do *Dropbox*, que também foi utilizado para manter o documento deste trabalho, assim como todas as suas versões com as correções e adaptações sugeridas pelo orientador.

Para fim de validação do sistema, foi utilizado um *tablet* genérico da marca Dropad, modelo A8. O *tablet* possui Android versão 2.3.4. As máquinas utilizadas para testes do servidor possuem as seguintes características de *hardware*:

Máquina 1

- Sistema operacional: Linux Mint 11, linux kernel 2.6.38
- Memória: 2GB
- Processador: Intel t5800

Máquina 2

- Sistema operacional: Ubuntu 12.10, linux kernel 3.5.0
- Memória: 4GB
- Processador: AMD Phenom II X4

Ambas as máquinas possuem o *Java Runtime Environment* versão 7.

3.1 CARACTERIZAÇÃO DA PESQUISA

A pesquisa, quanto à natureza do trabalho desenvolvido, pode ser considerada como uma pesquisa aplicada, e segundo Demo (2000), é o tipo de pesquisa em que a meta é contribuir com fins práticos, buscar soluções para problemas concretos e transformar em ações concretas os resultados do trabalho.

Isso deve-se ao fato de que tanto a parte do servidor quanto a parte do cliente estarem ligadas à realidade de um restaurante. O sistema serve para ajudar, auxiliar e dinamizar os pedidos e atualizações dos cardápios do estabelecimento.

O objetivo da pesquisa é ser uma pesquisa exploratória. Assis (2012 apud. GIL, 2006, p. 43) discute mais sobre este objetivo:

Tem como finalidade proporcionar maiores informações sobre determinado assunto, facilitar a delimitação de um tema de trabalho. Normalmente constitui a primeira etapa de uma investigação mais ampla. Desenvolve-se com o objetivo de proporcionar uma visão geral, de tipo aproximativo, acerca de determinado fato. (ASSIS, 2012 apud. GIL, 2006, p.43)

Parte da pesquisa foi feita com base na vivência. Ao frequentar qualquer estabelecimento que utiliza cardápios impressos e possua garçons, pode-se observar como é a estrutura de um menu, o funcionamento de um pedido, o fechamento da conta, a chamada de um garçom, entre outras; todas essas são funcionalidades presentes no sistema desenvolvido.

Do ponto de vista dos procedimentos técnicos a pesquisa pode ser considerada como uma pesquisa bibliográfica por que foi feita a partir de materiais publicados, como livros, artigos disponíveis na internet, sites, entre outros.

Quanto à forma de abordagem da pesquisa, ela pode ser definida como qualitativa, ou seja, as informações obtidas aqui não são quantificáveis e os dados obtidos são analisados de forma indutiva (ASSIS, 2012).

Além de proporcionar um cardápio dinâmico, o objetivo do sistema é fazer todo o gerenciamento de pedidos pelo *tablet*, o que auxilia o cliente e os funcionários do estabelecimento, além de trazer outros benefícios. Como foi

relatado por Smaal (2010) sobre o que o proprietário de um estabelecimento que utiliza iPads com o sistema SOPHIA disse a respeito do sistema e a aceitação do mesmo por sua clientela:

Carrasquillo comenta que mesmo aqueles clientes mais tradicionais e fiéis gostaram da novidade e não apenas se adaptaram rapidamente ao novo sistema, como também ficam interessados no iPad e navegam pelas opções ali mostradas. Diferente de tirar o garçom da mesa, ele melhora o atendimento, uma vez que o atendente pode trazer bebidas e circular pelo restaurante, oferecendo um serviço ainda melhor. Outra boa vantagem do iPad é na hora de opções não tão usuais. Ao brincar e navegar pelo aparelho, Carrasquillo afirma que os clientes prestam ainda mais atenção no que está exposto no cardápio e acabam consumindo sobremesas e afins com frequência. (SMAAL, 2010).

3.2 ETAPAS METODOLÓGICAS

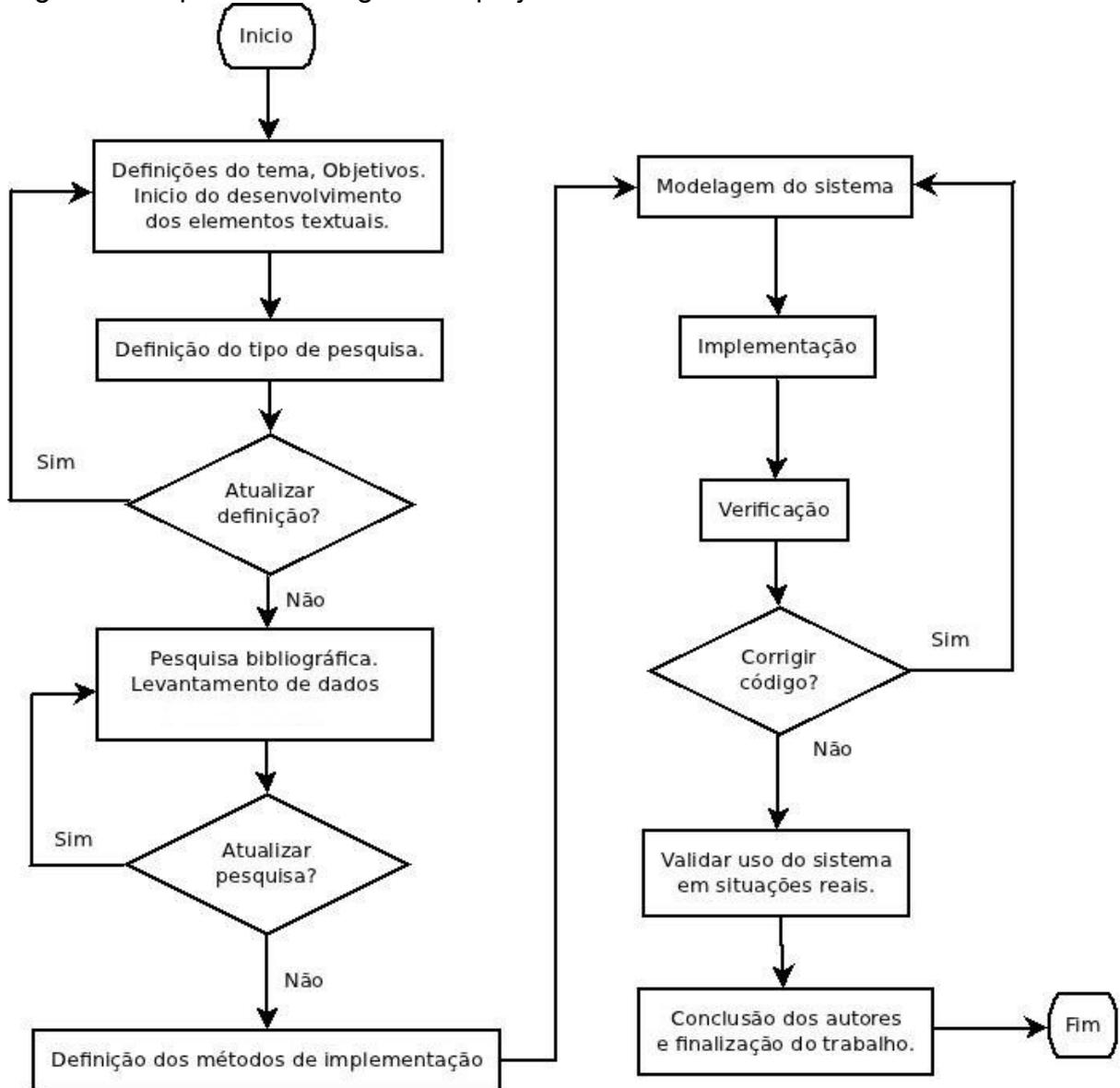
As etapas metodológicas do desenvolvimento do projeto podem ser conferidas em forma de um diagrama que explica os passos feitos para a elaboração da mesma, visto na figura 4.

Primeiro foram definidos o tema e os objetivos, depois foi dado início ao desenvolvimento escrito. Na segunda etapa foi definido o tipo de pesquisa.

Na terceira etapa foram feitas as pesquisas bibliográficas e o levantamento de dados.

Depois foram definidos os métodos de implementação, que se seguiram da criação dos diagramas e da modelagem do sistema.

Figura 4: Etapas Metodológicas do projeto



Fonte: O próprio autor (2011)

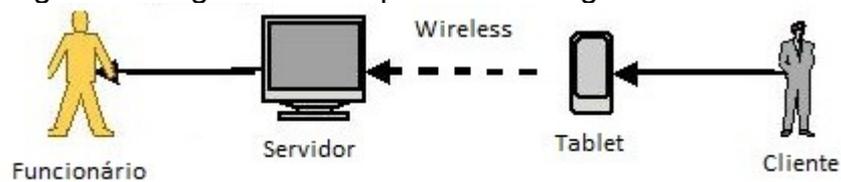
Após isso foi feita a implementação, depois uma verificação e correção de códigos.

As duas últimas etapas foram a validação do sistema e a criação das conclusões finais do trabalho.

3.3 PROPOSTA DA SOLUÇÃO

Para poder chegar no objetivo do projeto foram utilizados vários conhecimentos tanto na parte da implementação quanto na parte da engenharia de *software*. Seguem alguns diagramas do esquema tecnológico, de implantação e de módulos que ilustram o funcionamento do sistema, apresentados na figura 5.

Figura 5: Diagrama do esquema tecnológico

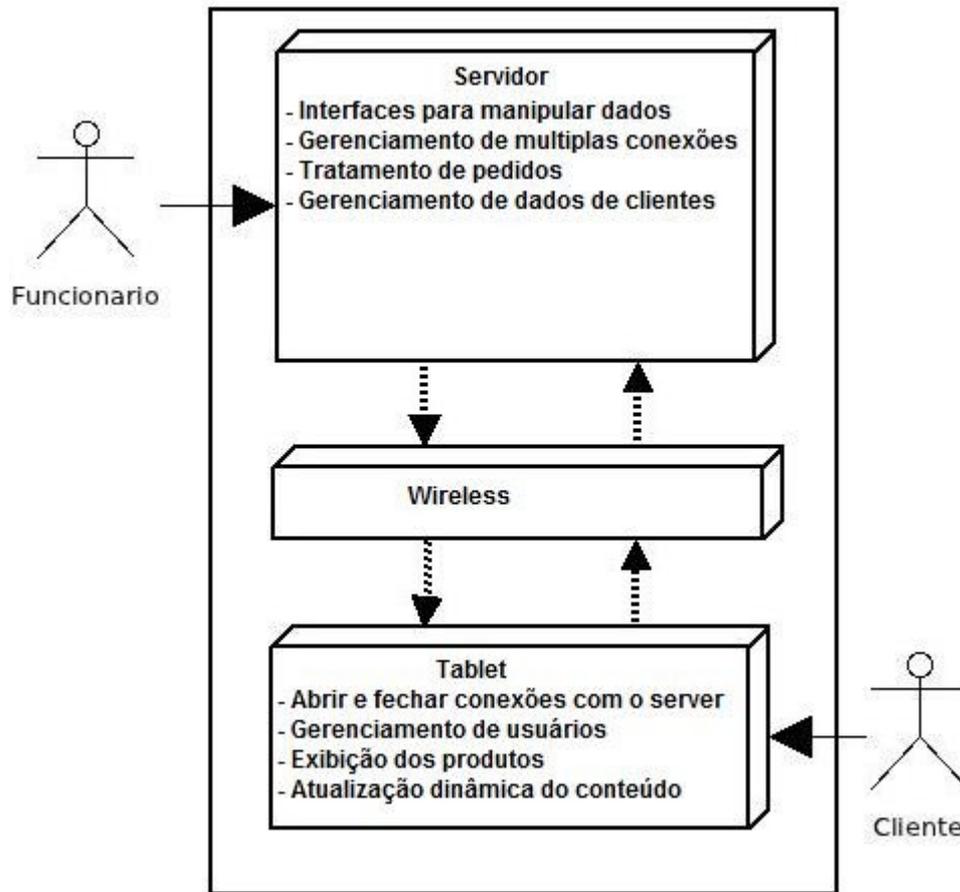


Fonte: O próprio autor (2011)

A Figura 5 mostra um diagrama do esquema tecnológico do funcionamento do sistema. Nele constam os dois principais atores (Funcionário e Cliente), o *tablet*, que é o cardápio, o servidor, e a conexão *wireless* utilizada para a comunicação entre os dois.

Para ilustrar o funcionamento do sistema e mostrar o que é necessário para que o mesmo possa ser executado, foi elaborado um diagrama de implantação (Figura 6).

Figura 6: Diagrama de Implantação

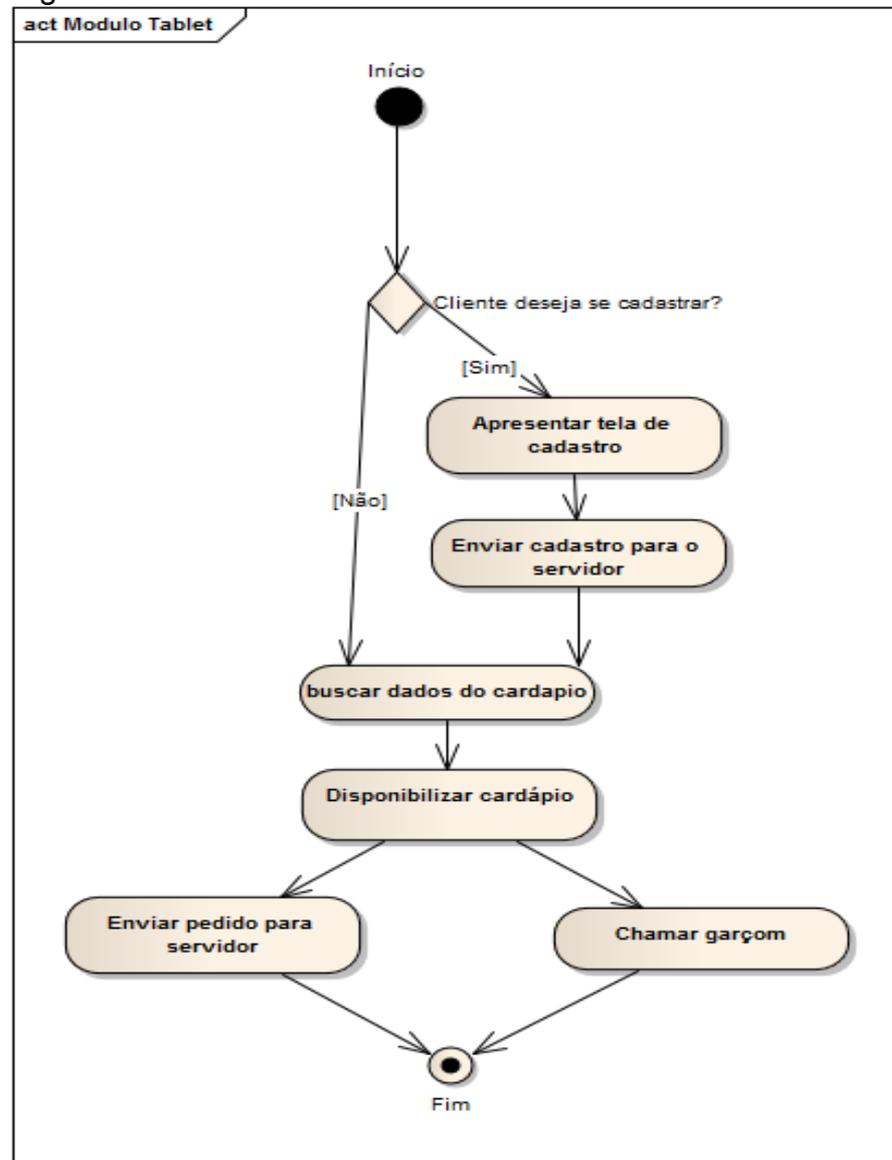


Fonte: O próprio autor (2012)

Como é mostrado no Diagrama de Implantação é necessário para o funcionamento do sistema um servidor, uma rede *wireless* e ao menos um *tablet*. Um funcionário é encarregado de utilizar o servidor e o cliente do estabelecimento é quem utiliza o aplicativo do cardápio contido no dispositivo móvel. A rede *wireless* é necessário para manter a comunicação entre o cardápio e o servidor.

Para um melhor entendimento do projeto, o mesmo foi dividido em 2 módulos, são eles: Módulo do Tablet e Módulo Servidor.

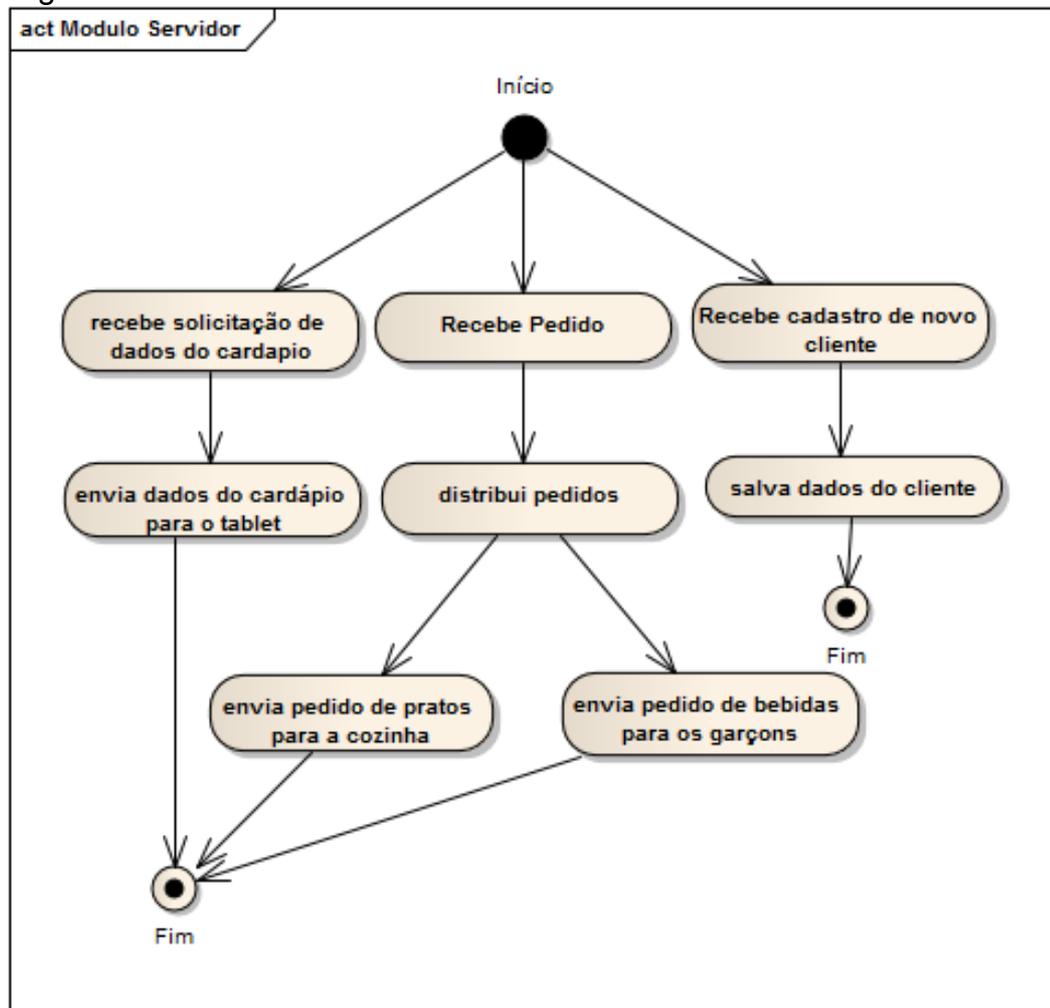
Figura 7: Módulo do Tablet



Fonte: O próprio autor (2012)

A figura 7 representa o Módulo do Tablet, onde primeiramente o cardápio apresenta as opções de *login*, criação de conta e acesso ao menu. Após o cliente ter feito a escolha, o menu busca as informações de preferências, pratos e promoções para apresentar ao cliente. Depois pode ser chamado um garçom ou ser efetuado um pedido.

Figura 8: Módulo do Servidor



Fonte: O próprio autor (2011)

Na figura 8 é apresentado o Módulo do Servidor, que tem o papel de gerenciar as requisições de informações de clientes, pratos e pedidos dos *tablets*. O usuário tem função de distribuir os pedidos para a cozinha ou para os garçons. O servidor também recebe os dados de cadastro de um novo cliente.

3.4 DELIMITAÇÕES

Devido à natureza generalizada e modular desse sistema, inúmeras ideias podem ser implementadas. Por este fato, os autores decidiram implementar os seguintes recursos apenas:

Lado do servidor:

- Gerenciamento de múltiplas conexões de clientes.
- Customização de conteúdo exibido nos clientes.
- Tratamento dos pedidos realizados pelos clientes.
- Gerenciamento dos dados de cadastros de clientes.
- Fechamento de conta.

Lado do cliente:

- Abrir e fechar conexões com o servidor.
- Gerenciamento de usuário (*registro/login*).
- Exibição dos produtos servidos pelo estabelecimento.
- Atualização dinâmica do conteúdo disponível.

4 MODELAGEM

Neste capítulo é apresentada a modelagem do sistema proposto. A modelagem é feita seguindo o padrão UML e o desenvolvimento do *software* seguiu o padrão ICONIX, os dois são explicados brevemente antes de ser apresentada a modelagem em si, para um melhor entendimento do leitor.

4.1 REQUISITOS

Os requisitos do sistema podem ser classificados em regras de negócio, requisitos funcionais e não funcionais. Os três tipos são discutidos nos próximos subtópicos e como eles são utilizados no desenvolvimento do sistema proposto.

4.1.1 Regras de Negócio

Segundo Grandi (2008 apud. DATE 2000), regras de negócios, dentro do contexto de Sistemas de Informação, é a parte do software responsável pela definição do comportamento do software, ou como ele deve operar.

O quadro 1 exhibe as regras de negócio do sistema proposto.

Quadro 1: Regras de negócio

Regra de negócio	Descrição
RN01 Dados de Tipo	Deve possuir um nome.
RN02 Dados de Categoria	Dever possuir um nome. Pode possuir descrição. Deve pertencer a um Tipo.
RN03 Dados de Item	Deve possuir nome, descrição, preço, no mínimo um (1) Ingrediente e no mínimo uma (1) Foto. Deve pertencer a uma Categoria.
RN04 Dados de Cliente	Deve possuir nome, sobrenome, endereço de e-mail, data de nascimento no formato UNIX e

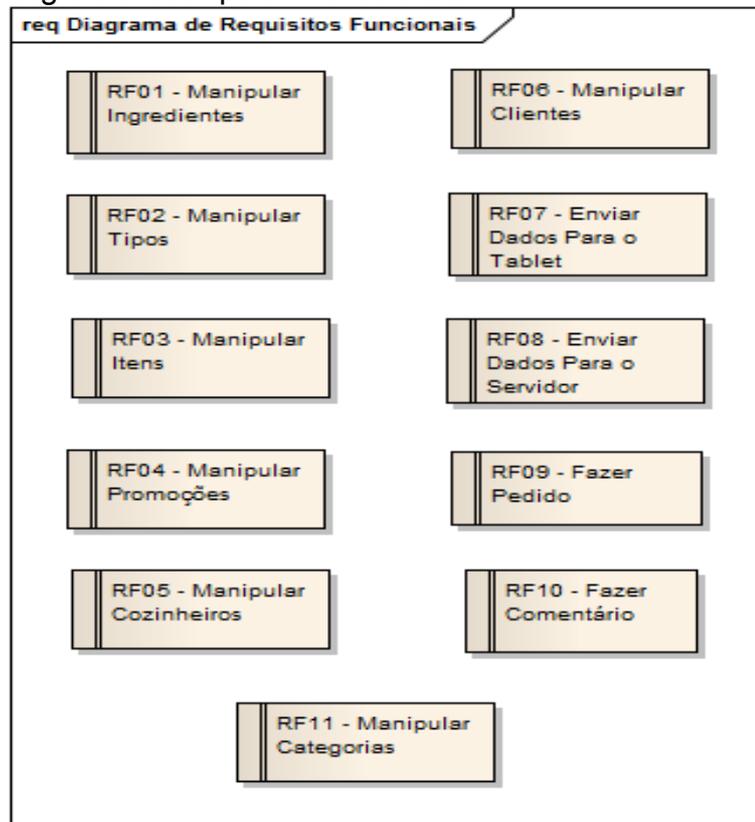
	Endereço. Pode possuir RG e CPF.
RN05 Dados de Cozinheiro	Deve possuir nome, sobrenome, endereço de e-mail, data de nascimento no formato UNIX, RG, CPF, Endereço, histórico de trabalhos anteriores, especialidade culinária, telefone residencial, telefone celular, e uma Foto.
RN06 Dados de Comentário	Deve possuir um texto como comentário, uma data no formato UNIX, um (1) Item no qual o comentário será atribuído, e um (1) Cliente no qual o comentário será atribuído.
RN07 Dados de Endereço	Deve possuir nome da rua, número (deverá também aceitar caracteres no número), CEP, bairro, cidade e sigla do estado.
RN08 Dados de Foto	Deve ser preenchido o caminho absoluto que o arquivo de imagem se encontra no sistema.
RN09 Dados de Ingrediente	Deve possuir nome e preço unitário. Pode possuir descrição.
RN10 Dados de Pedido	Deve possuir data no formato UNIX, o Cliente que fez o pedido, e no mínimo um (1) Item, com a quantidade pedida e uma observação (opcional) a respeito do item pedido.
RN11 Dados de Promoção	Deve possuir nome, descrição, data de início e validade, ambas no formato UNIX, uma (1) Foto e no mínimo um (1) Item.

Fonte: O próprio autor (2012)

4.1.2 Requisitos Funcionais

Requisitos funcionais podem ser definidos basicamente como aquilo que o cliente deseja que o sistema faça, ou, os problemas que o sistema, depois de pronto, deverá resolver. Segundo Deitel e Deitel (2008), “Ao projetar qualquer sistema deve-se solucionar corretamente o problema, mas - igualmente importante - solucionar o problema correto.”

Figura 10: Requisitos Funcionais



Fonte: O próprio autor (2011)

A figura 10 apresenta os requisitos funcionais do sistema desenvolvido e, logo após, no quadro 2, o título e a descrição de cada um deles.

Quadro 2: Tabela de Requisitos Funcionais

#	Nome	Descrição	Requisitos Não Funcionais Relacionados	Regras de Negócio Vinculadas
RF01	Manipular Ingredientes	O sistema deve prover uma interface para o cadastro, alteração, exclusão, e listagem de todos os ingredientes dos itens do cardápio.	RNF03, RNF05	RN09 Dados de Ingrediente
RF02	Manipular Tipos	O sistema deve prover uma interface para o cadastro, alteração, exclusão e listagem de todos os tipos de	RNF03, RNF05	RN01 Dados de Tipo

		itens que o estabelecimento possui.		
RF03	Manipular Itens	O sistema deve prover uma interface para o cadastro, alteração, exclusão e listagem de todos os itens do cardápio.	RNF03, RNF05	RN03 Dados de Item
RF04	Manipular Promoções	O sistema deve prover uma interface para o cadastro, alteração, exclusão e listagem de promoções. As promoções validas devem ser exibidas diretamente no cardápio.	RNF03, RNF04 e RNF05	RN11 Dados de Promoção
RF05	Manipular Cozinheiros	O sistema deve prover uma interface para o cadastro, alteração, exclusão e listagem de Cozinheiros, sua foto e alguns de seus dados profissionais devem estar disponíveis nos cardápios, para que os clientes saibam quem são os responsáveis pelas refeições do estabelecimento.	RNF03, RNF04 e RNF05	RN05 Dados de Cozinheiro
RF06	Manipular Clientes	O sistema deve prover uma interface no cardápio para o cadastro de clientes, para que os mesmos façam o seu próprio cadastro se desejarem. O sistema, na parte do servidor, deve possuir interface para listagem, edição, alteração e cadastro de clientes.	RNF02, RNF04, RNF05 e RNF06	RN04 Dados de Cliente
RF07	Enviar Dados Para o Tablet	Assim que um cardápio for ligado e o sistema for aberto, o servidor deve enviar todos os dados do cardápio	RNF01, RNF03, RNF05, RNF06	

		para o mesmo. O cardápio deve apresentar a lista de tipos (pratos, bebidas etc) dos itens que o restaurante possui, deve mostrar os itens separados por categorias, deve apresentar uma tela com detalhes dos itens. Deve também permitir que o usuário do sistema adicione itens em seu pedido		
RF08	Enviar Dados Para o Servidor	Dados como: pedidos dos clientes, e dados de cadastro de clientes devem ser enviados para o servidor e depois persistidos nas devidas bases de dados.	RNF01, RNF02 RNF03, RNF05, RNF06	
RF09	Fazer Pedido	O cliente do estabelecimento deve poder efetuar pedido de itens pelo cardápio digital. Deve poder inserir uma observação por item e escolher a quantidade desejada de cada item.	RNF04, RNF05, RNF06	RN10 Dados do Pedido
RF10	Fazer Comentário	O cliente do estabelecimento deve poder fazer comentários, que avaliam os itens que ele pedir, se o mesmo possuir cadastro e estiver logado.	RNF02, RNF04, RNF05, RNF06	RN06 Dados de Comentário
RF11	Manipular Categorias	O sistema deve prover uma interface para o cadastro, alteração, exclusão, e listagem de categorias para os tipos que possui cadastrados.	RNF03, RNF06	RN02 Dados de Categoria

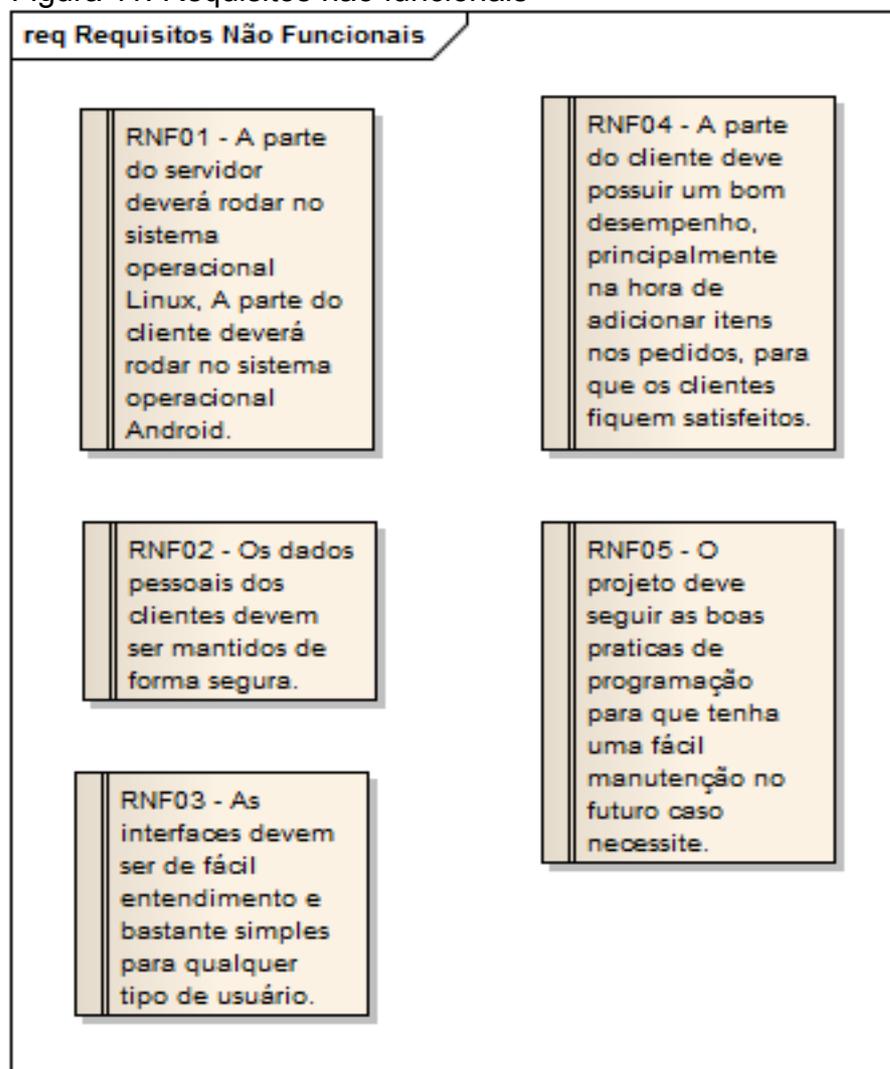
Fonte: O próprio autor (2011)

O quadro 2 apresenta a descrição os requisitos funcionais 01 ao 11.

4.1.3 Requisitos Não Funcionais

Requisitos não funcionais podem ser restrições dos requisitos funcionais, eles representam qualidades gerais para os softwares. Essas qualidades podem ser: portabilidade, segurança, manutenibilidade, desempenho, usabilidade, entre outras. Seguem na figura 11, os requisitos não funcionais do sistema desenvolvido.

Figura 11: Requisitos não funcionais



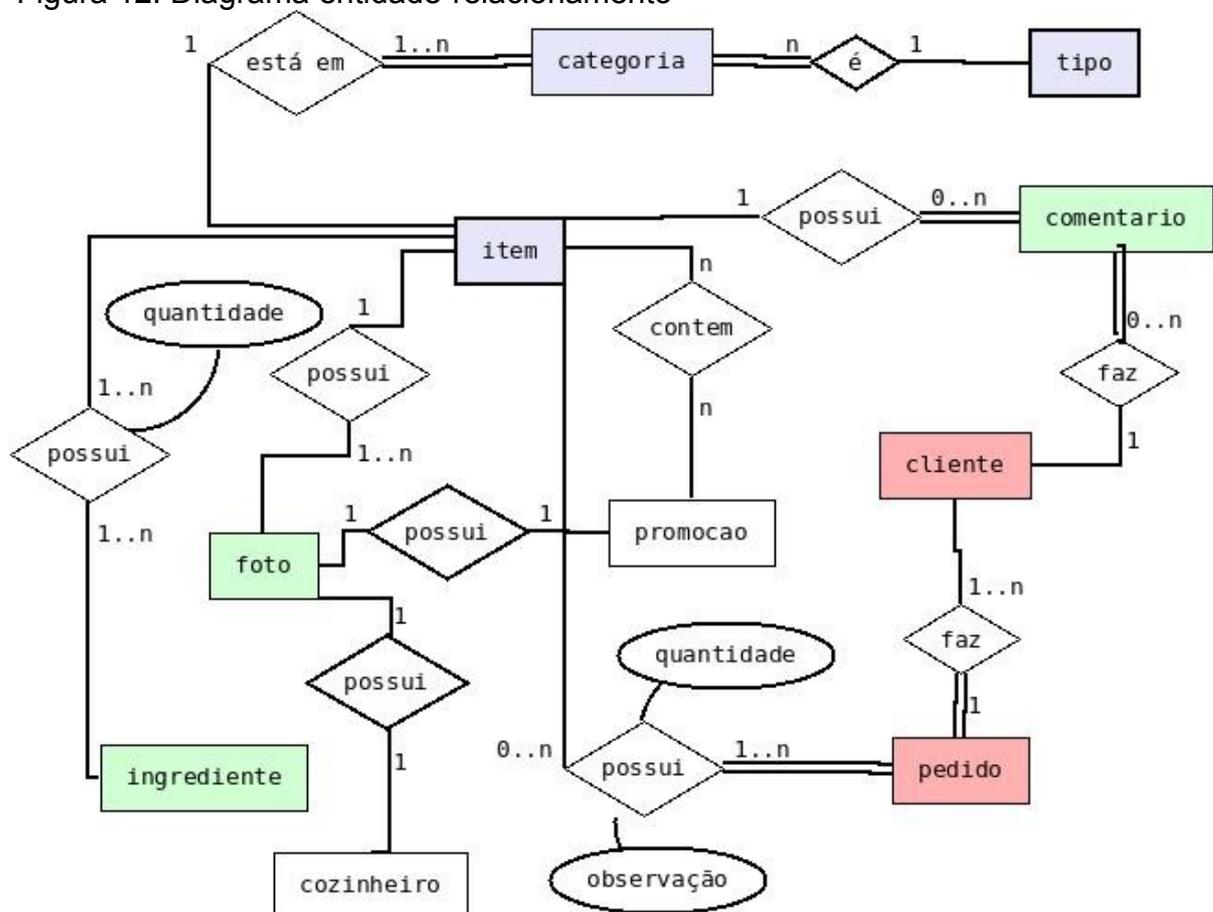
Fonte: O próprio autor (2011)

Segundo Braga (2009) os requisitos não funcionais descrevem como o sistema é, ou seja, as qualidades do sistema, isso difere dos requisitos funcionais que descrevem o que o sistema deve fazer.

4.2 DIAGRAMA ENTIDADE-RELACIONAMENTO

Neste tópicó é apresentado o diagrama entidade-relacionamento (ER) do projeto. A figura 12 mostra a forma de entidade sem atributos e com seus devidos relacionamentos, e as figuras 13 a 22 as representações das entidades com seus atributos.

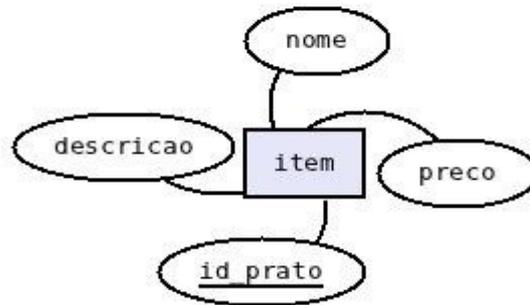
Figura 12: Diagrama entidade-relacionamento



Fonte: O próprio autor (2012)

A figura 2 mostra o diagrama ER. Este diagrama possui dez entidades e onze relacionamentos, dos quais dois possuem atributos próprios. Por motivos de economia de espaço, os atributos das entidades foram removidos e são apresentados individualmente em seguida.

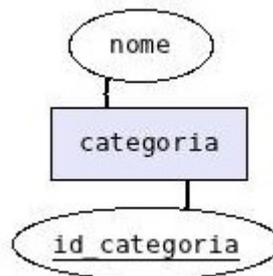
Figura 13: entidade Item e seus atributos



Fonte: O próprio autor (2012)

A figura 13 mostra a entidade “item”, com seus atributos “id_prato”, “nome”, “descrição” e “preço”. Esta entidade possui relacionamento com as entidades “categoria”, “comentário”, “ingrediente” com um atributo especificando a quantidade, “foto”, “promoção” e “pedido” com dois atributos especificando a quantidade e observações.

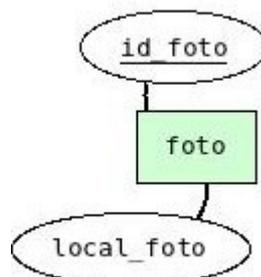
Figura 14: entidade Categoria e seus atributos



Fonte: O próprio autor (2012)

A figura 14 mostra a entidade “categoria”, com seus atributos “id_categoria” e “nome”. Esta entidade possui relacionamento com as entidades “item” e “tipo”.

Figura 15: entidade Foto e seus atributos



Fonte: O próprio autor (2012)

A figura 15 mostra a entidade “foto”, com seus atributos “id_foto” e “local_foto”, este especifica o caminho do arquivo de imagem no sistema de arquivos. Esta entidade possui relacionamento com as entidades “promoção”, “cozinheiro” e “item”.

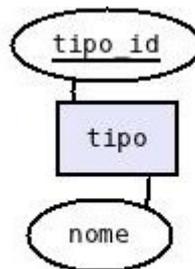
Figura 16: entidade Ingrediente e seus atributos



Fonte: O próprio autor (2012)

A figura 16 mostra a entidade “ingrediente” e seus atributos “id_ingrediente”, “nome”, “descrição” e “preço_unidade”. Esta entidade possui relacionamento com a entidade “item” com o atributo “quantidade”.

Figura 17: entidade Tipo e seus atributos



Fonte: O próprio autor (2012)

A figura 17 mostra a entidade “tipo” e seus atributos “tipo_id” e “nome”. Esta entidade possui relacionamento com a entidade “categoria”.

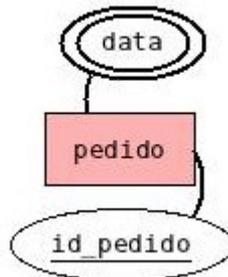
Figura 18: entidade Comentário e seus atributos



Fonte: O próprio autor (2012)

A figura 18 mostra a entidade “comentário” e seus atributos “id_comentario”, “comentário”, “dia”, “mês”, “ano”, “hora” e “minuto”. Esta entidade possui relacionamento com as entidades “item” e “cliente”.

Figura 19: entidade Pedido e seus atributos



Fonte: O próprio autor (2012)

A figura 19 mostra a entidade “pedido” e seus atributos “id_pedido” e “data”. Esta entidade possui relacionamento com as entidades “cliente” e “item” com os atributos “quantidade” e “observações”.

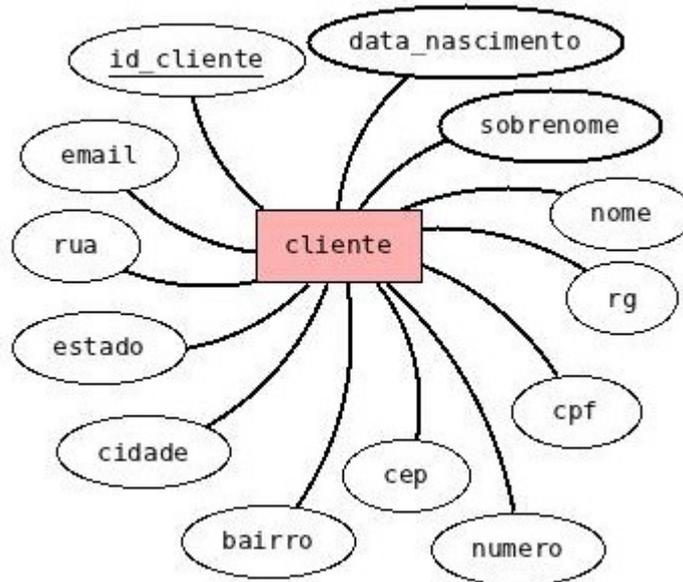
Figura 20: entidade Promoção e seus atributos



Fonte: O próprio autor (2012)

A figura 20 mostra a entidade “promoção” e seus atributos “id_promocao”, “nome”, “descrição”, “data_inicio” e “validade”. Esta entidade possui relacionamento com as entidades “item” e “foto”.

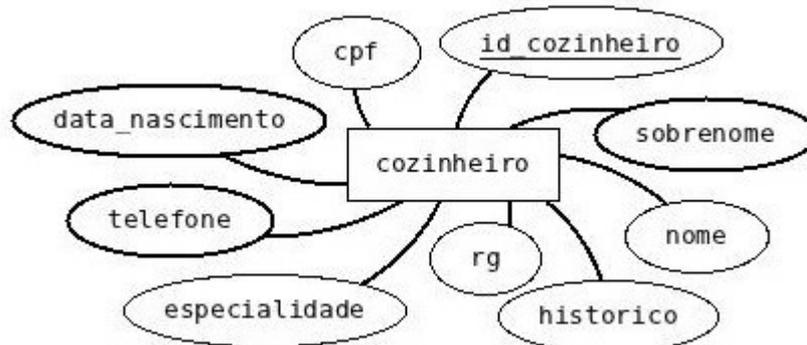
Figura 21: entidade Cliente e seus atributos



Fonte: O próprio autor (2012)

A figura 21 mostra a entidade “cliente” e seus atributos “id_cliente”, “nome”, “sobrenome”, “rg”, “cpf”, “email”, e atributos relacionados ao endereço. Esta entidade possui relacionamento com as entidades “comentário” e “pedido”.

Figura 22: entidade Cozinheiro e seus atributos



Fonte: O próprio autor (2012)

A figura 22 mostra a entidade “cozinheiro” e seus atributos “id_cozinheiro”, “nome”, “sobrenome”, “data_nascimento”, “telefone”, “especialidade”, “histórico”, “rg” e “cpf”. Esta entidade possui relacionamento com a entidade “foto”.

4.3 DIAGRAMAS DE CLASSE E PACOTE

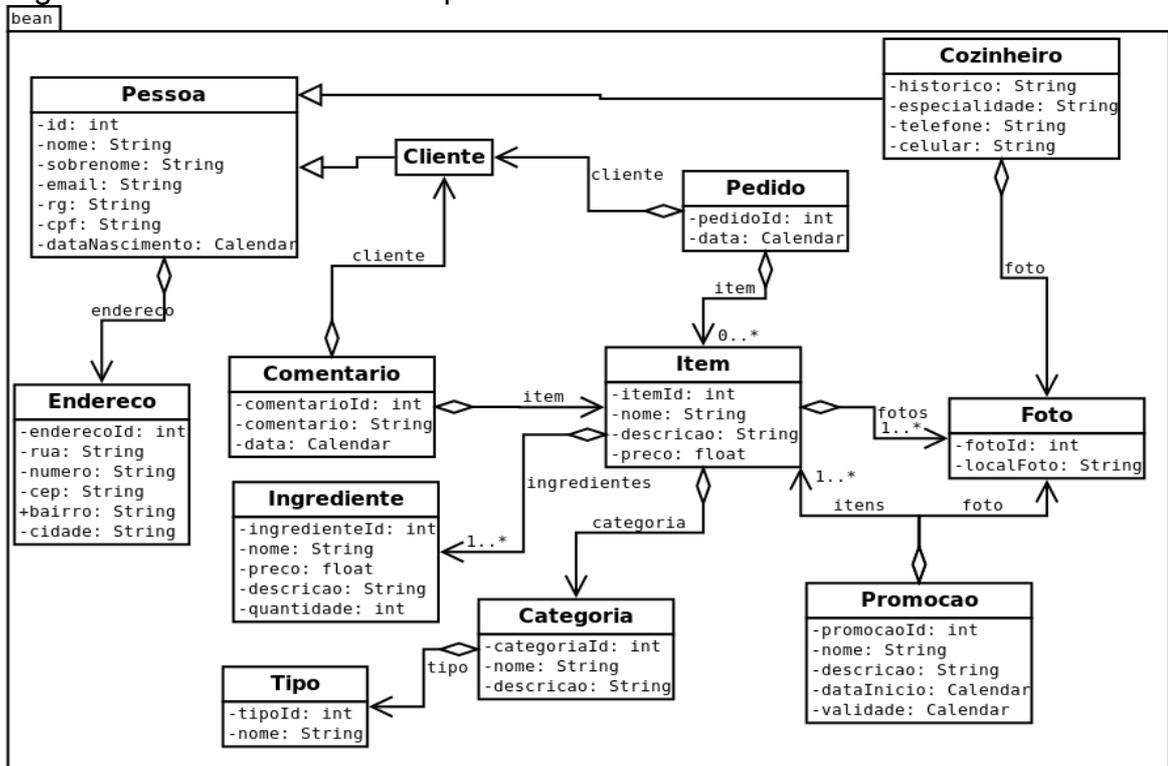
Neste tópico, os diagramas de classe e pacotes são apresentados. As imagens 23 a 26 compõem o diagrama de pacote e classes.

Segundo a Globalcode (2004), o diagrama de pacotes “mostra a organização das classes em um sistema, ou seja, a divisão em diretórios das classes”. Ele pode ser utilizado em qualquer fase do processo de modelagem e visa organizar os modelos.

Segundo a Object Management Group (2011, p. 21), o pacote de classes contem subpacotes que lidam com os conceitos básicos de modelagem com UML, em classes específicas e seus relacionamentos.

A Globalcode (2004) ainda explica que o diagrama de classes é utilizado para representar a estrutura do sistema. Ele pode ser utilizado na fase de análise de requisitos para fazer a modelagem conceitual do sistema, representando o relacionamento entre entidades mais evidentes e, então, é feito o seu refinamento durante a fase de implementação.

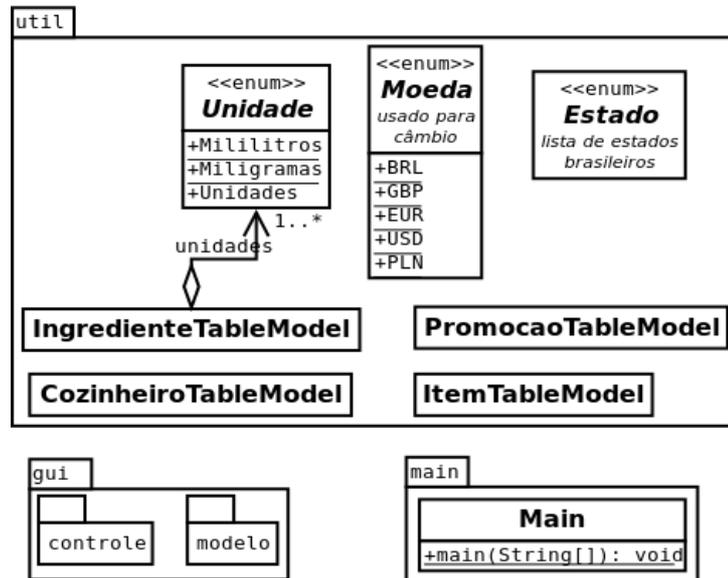
Figura 23: Pacote “bean” e respectivas classes.



Fonte: O próprio autor (2012)

A figura 23 mostra o pacote “bean” e suas respectivas classes. Estas classes são mapeadas diretamente a partir do diagrama ER e constituem a base dos dados a serem manipulados e persistidos pela linguagem de programação. É interessante notar a semelhança entre os relacionamentos dos *beans* e os relacionamentos das entidades do diagrama ER.

Figura 24: Pacotes “util”, “gui”, “main” e respectivas classes.



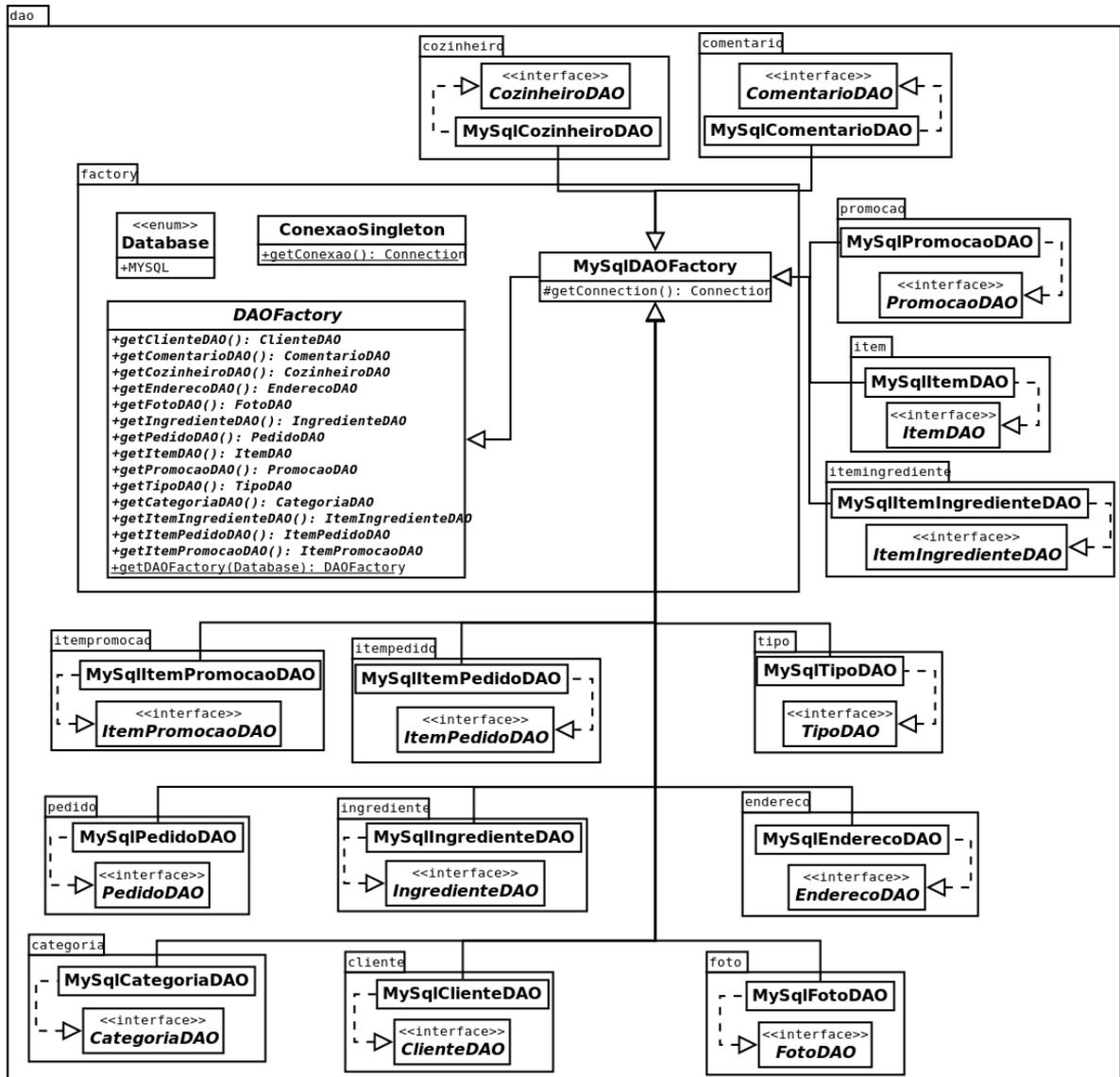
Fonte: O próprio autor (2012)

A figura 24 exibe os pacotes “util”, “gui” e “main” do sistema. No pacote “util” existem classes utilitárias do sistema, que são utilizadas nos mais diferentes e variados propósitos, como por exemplo a classe “Unidade” possui uma enumeração de diferentes unidades de medida utilizadas, a classe “Moeda” também é uma enumeração de diferentes moedas utilizadas para definir os preços dos pratos. Já as classes que terminam o nome com “TableModel” são utilizadas para a formatação dos dados exibidos em certas tabelas na interface gráfica em determinadas situações.

O pacote “gui” é o responsável por todos os modelos e respectivos controles da interface gráfica. Por motivos de alta complexidade, a modelagem dos diagramas de classes para este pacote foram omitidos.

Já o pacote “main” possui a única classe “Main”, que é a classe responsável pela montagem e execução inicial dos componentes do sistema.

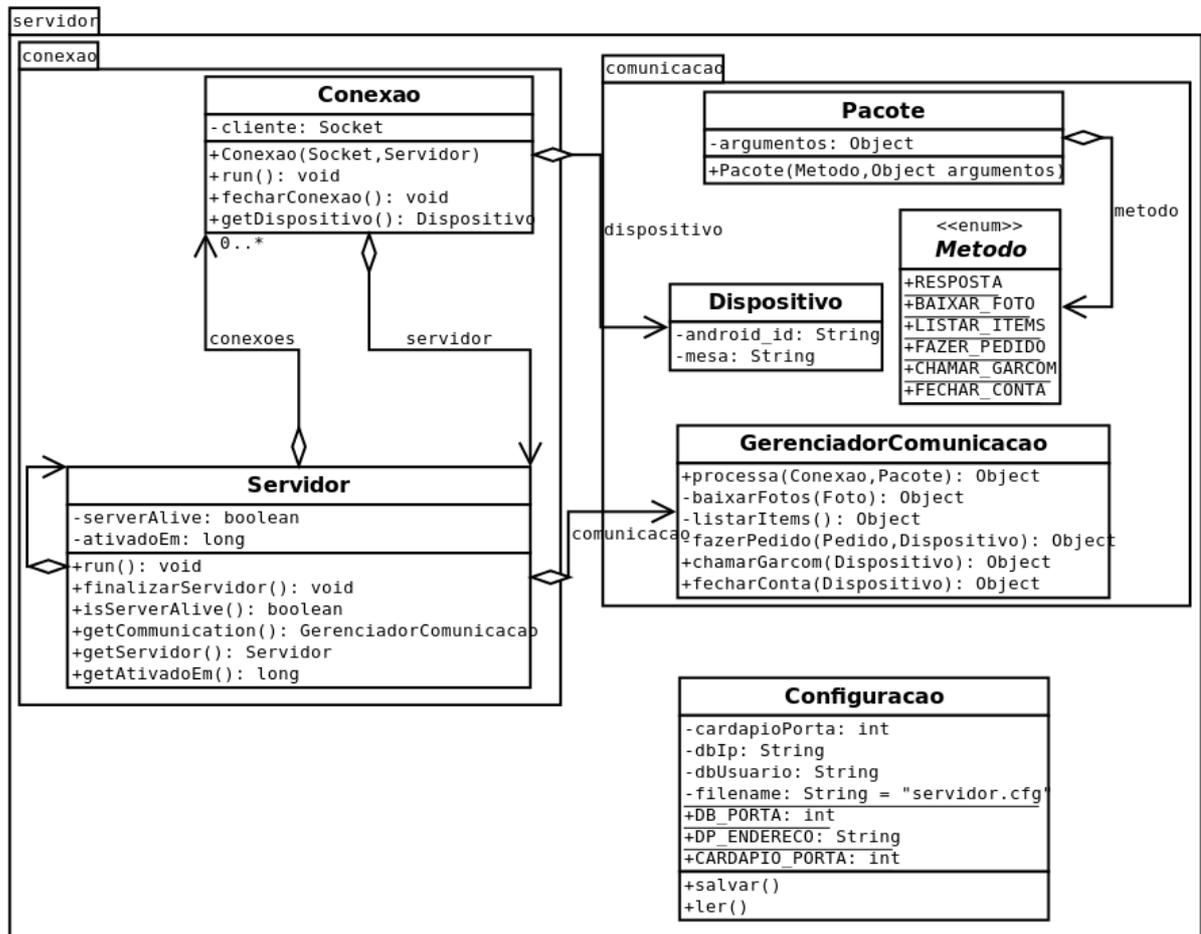
Figura 25: Pacote “dao”, subpacotes e respectivas classes.



Fonte: O próprio autor (2012)

A figura 25 exhibe o pacote “dao”. DAO vem do inglês *data access object*, e em programação, é um padrão utilizado para persistir dados de *beans* e outros objetos de forma transparente e maleável. Em conjunto com o padrão DAO, foi-se utilizado o padrão *Abstract Factory*, que cria de forma transparente determinados tipos de objetos, e neste caso foi utilizado para poder-se introduzir ao sistema de forma transparente novas formas de persistência, como por exemplo arquivos, ou outros sistemas gerenciadores de banco de dados. Pode-se contemplar a utilização em conjunto destes dois padrões neste diagrama de classe e pacotes.

Figura 26: Pacote “servidor”, subpacotes e respectivas classes



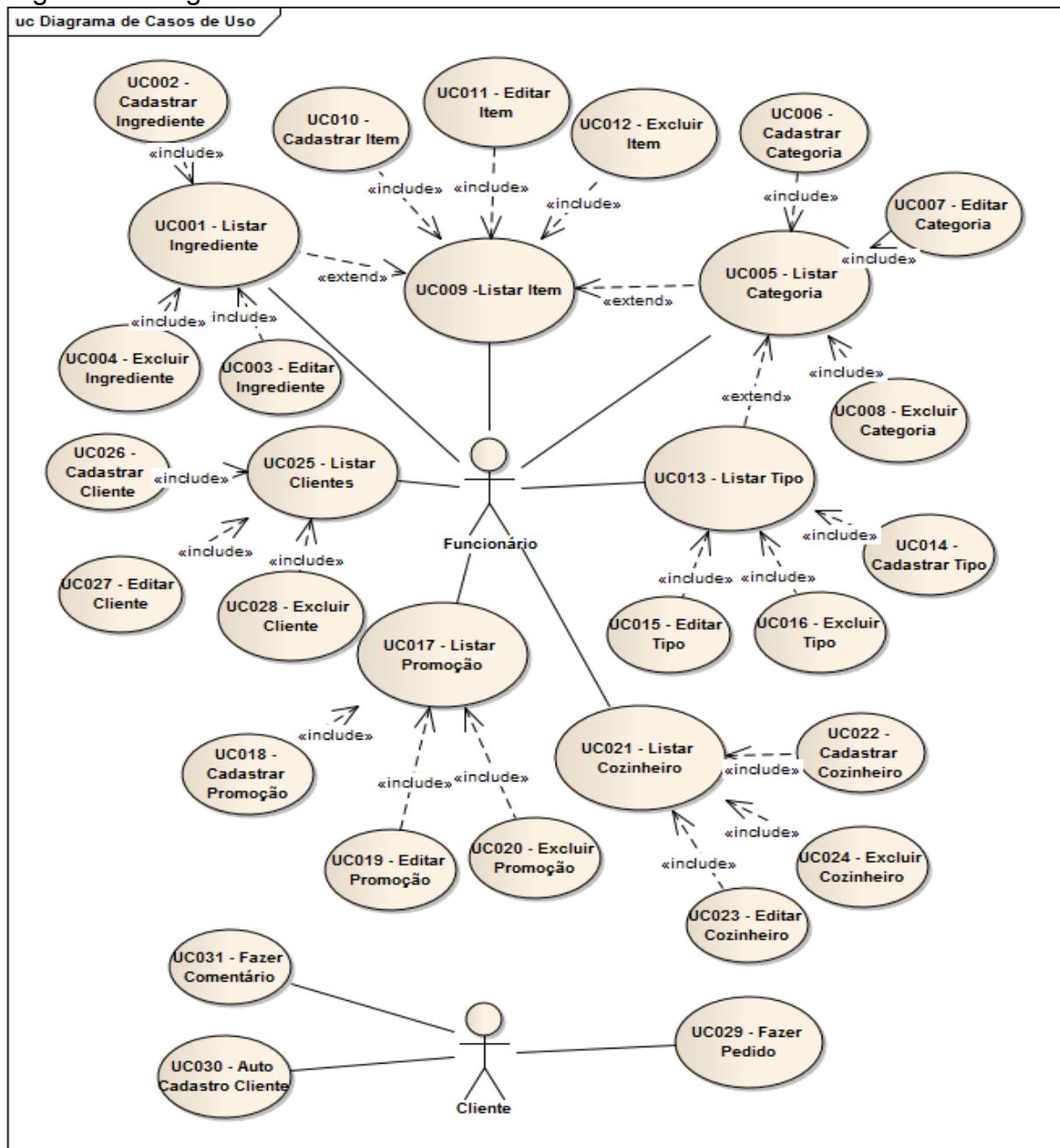
Fonte: O próprio autor (2012)

A figura 26 apresenta o pacote “servidor” e seus subpacotes. Este pacote é responsável pela criação, configuração, gerenciamento e finalização de múltiplas conexões de *tablets* com o servidor. Toda comunicação é empacotada pela classe “Pacote”, e ao receber um destes pacotes, o servidor o redireciona para um “GerenciadorComunicacao”, responsável por decodificar e abrir este pacote, que trará informações sobre o que se fazer, e dados para auxiliar esta ação, quando necessário. Todo pacote recebido pelo servidor espera obrigatoriamente uma resposta (que também será empacotada em um “Pacote”), mesmo que esta resposta seja nula.

4.4 CASOS DE USO

Segundo a Object Management Group (2011, p. 597), os casos de uso são um meio para especificar usos necessários de um sistema. De um meio geral, são utilizados para capturar os requisitos de um sistema, ou seja, o que o sistema deve fazer.

Figura 27: Diagrama de Casos de Uso



Fonte: O próprio autor (2012)

O diagrama de casos de uso mostra a interação dos atores com as funcionalidades do sistema. O cliente pode fazer pedido (UC029), fazer comentários em pratos (UC031) e fazer seu cadastro no sistema (UC030). O ator funcionário pode fazer as seguintes operações: manipular (cadastrar, excluir, listar, e editar) ingredientes a partir do UC001, manipular categorias a partir do UC005, manipular itens a partir do UC009, manipular tipos a partir do UC013, manipular promoções a partir do UC017, manipular cozinheiros a partir do UC021 e manipular clientes a partir do UC025.

A figura 27 apresenta o Diagrama de Casos de Uso do sistema. Os sub tópicos seguintes apresentam uma explicação mais detalhada sobre cada característica contida neste diagrama.

4.4.1 Especificação dos Atores

A seguir, no quadro 3, é apresentada uma tabela contendo as informações dos atores do sistema.

Quadro 3: Atores do sistema

Ator	Descrição
Cliente	<p>*Definição: Pessoas que frequentam o restaurante e utilizam o cardápio digital.</p> <p>*Frequência de Uso: Todos os dias que o estabelecimento estiver aberto.</p> <p>*Conhecimento em Informática: Varia de cliente para cliente, uns possuem, outros não. Alguns clientes talvez precisem da bastante auxílio de algum funcionário do estabelecimento.</p> <p>*Permissões de Acesso: Tem acesso ao cardápio digital com lista de itens, promoções, pode efetuar pedidos, se cadastrar e fazer comentários se possuir cadastro.</p>
Funcionário	<p>*Definição: Funcionário que tem acesso ao sistema. Pode efetuar cadastros consultas e modificações de itens, tipos, categorias, ingredientes, clientes, cozinheiros e promoções. Também é responsável por fazer o controle dos pedidos que chegam ao servidor.</p> <p>*Frequência de Uso: Todos os dias.</p> <p>*Conhecimento em Informática: Saber utilizar um computador e um <i>tablet</i>.</p> <p>*Grau de Escolaridade: Saber ler e escrever.</p>

	*Permissões de Acesso: Acesso a todas as funcionalidades do sistema.
--	---

Fonte: O próprio autor (2012)

Segundo a Object Management Group (2011, p. 597), os usuários e qualquer outro sistema que possa interagir com o sujeito são representados como atores. Atores sempre modelam entidades que estão fora do sistema, e o comportamento requerido pelo sujeito é especificado por um ou mais casos de uso, que são definidos de acordo com as necessidades dos atores.

4.4.2 Especificações dos Casos de Uso

Como muitos processos do sistema são semelhantes, por exemplo, as manipulações (cadastros, alterações, exclusões e listagens), os autores preferiram focar no detalhamento do processo de manipular ingrediente que envolve os casos de uso, UC001 – Listar Ingrediente, UC002 – Cadastrar Ingrediente, UC003 – Editar Ingrediente e UC004 – Excluir Ingrediente.

Para estes casos de uso foram feitos um diagrama de atividades, um diagrama de robustez, diagramas de sequência e a especificação de cada caso de uso.

Seguem abaixo, nos quadros 4 ao 7, as especificações dos casos de uso responsáveis pela manipulação de ingredientes no sistema.

Quadro 4: Especificação do caso de uso UC001 – Listar Ingrediente

Nome	UC_001 - Listar Ingrediente
Descrição	Este caso de uso descreve como é feita a listagem de ingredientes no sistema.
Ator	Funcionário
Fluxo Principal	1 - O usuário entra na aplicação (servidor); 2 - O usuário clica em “Manipular” e seleciona a opção “Ingrediente...”; 3 - O sistema apresenta uma tela (Lista de Ingredientes) com uma lista contendo todos os ingredientes cadastrados no sistema. A lista pode estar vazia caso seja a primeira vez que o usuário cadastra um ingrediente;

	4 - O caso de uso se encerra.
Pós-condições	Lista de ingredientes apresentada para o usuário.
Regra de Negócio	RN08 – Dados de Ingrediente
Requisito Funcional Relacionado	RF01 – Manipular Ingredientes

Fonte: O próprio autor (2012)

O quadro 4 mostra as especificações do caso de uso “UC001”, referente à listar ingredientes.

Quadro 5: Especificação do caso de uso UC002 – Cadastrar Ingrediente

Nome	UC002 – Cadastrar Ingrediente
Descrição	Este caso de uso descreve como é feito o cadastro de um novo ingrediente no sistema.
Ator	Funcionário
Pré-condições	Executar o caso de uso UC001 – Listar Ingrediente
Fluxo Principal	1 - O usuário executa o caso de uso UC001 – Listar Ingrediente e clica no botão “Cadastrar Novo” (FA_01, FA_02); 2 - O sistema apresenta a tela para cadastro de um novo Ingrediente; 3 - O usuário preenche o campo nome, descrição, preço de compra e clica em “OK” (FA_03, FA_04, FE_01); 4 - O sistema volta para a tela de Lista de Ingredientes que apresenta a lista contendo o Ingrediente recém inserido; 5 - O caso de uso se encerra.
Pós-condições	Novo ingrediente cadastrado no sistema.
FA_01 Editar Ingrediente Selecionado	1 - Este fluxo alternativo se inicia caso o usuário selecione um dos ingredientes apresentados na tela Lista de Ingredientes e clique no botão “Editar selecionado...” (no passo 1 do fluxo principal), o sistema executa o caso de uso UC003 – Editar Ingrediente.
FA_02 Excluir Selecionado	1 - Este fluxo alternativo se inicia caso o usuário selecione um dos Ingredientes apresentado na tela Lista de Ingredientes e clique no botão “Excluir Selecionado” (no passo 1 do fluxo principal), o sistema executa o caso de uso UC004 – Excluir Ingrediente.
FA_03 Limpar	1 - Este fluxo alternativo se inicia caso o usuário clique no

Campos	botão “Limpar”; 2 - O sistema apaga o que está escrito nos campos nome, descrição e preço de compra e vai para o passo 3 do fluxo principal.
FA_04 Cancelar	1 - Este fluxo alternativo se inicia caso o usuário clique no botão “Cancel”; 2 - O sistema volta para o caso de uso UC001 – Listar Ingrediente.
FE_01 Campos Inválidos	1 - Este fluxo de exceção se inicia caso o sistema verifique que o usuário digitou valores inválidos ou em branco nos campos nome, descrição e preço de compra (no passo 3 do fluxo principal); 2 - O sistema apresenta uma tela com a mensagem: “Erros nos campos”, o usuário clica em “OK”, o sistema volta para o passo 3 do fluxo principal.
Regra de Negócio	RN08 – Dados de Ingrediente
Requisito Funcional Relacionado	RF01 – Manipular Ingredientes

Fonte: O próprio autor (2012)

O quadro 5 mostra as especificações do caso de uso “UC002”, referente à cadastrar ingredientes.

Quadro 6: Especificação do caso de uso UC003 – Editar Ingrediente

Nome	UC003 – Editar Ingrediente
Descrição	Este caso de uso descreve como é feita a alteração de um ingrediente cadastrado no sistema
Ator	Funcionário
Pré-condições	1 - Executar o caso de uso UC001 – Listar Ingrediente; 2 - Existir um Ingrediente cadastrado.
Fluxo Principal	1 - O usuário executa o caso de uso UC001 – Listar Ingrediente. Seleciona um dos ingredientes da lista de ingredientes e clica no botão “Editar selecionado...” (FA_01, FA_02, FE_01); 2 - O sistema apresenta uma tela com os campos nome, descrição e preço de compra preenchidos relativos ao Ingrediente que o usuário escolheu na tela de Lista de Ingredientes;

	<p>3 - O usuário faz a modificação que deseja nos campos nome, descrição e preço de compra e clica em “Editar” (FA_03, FE_02);</p> <p>04 - O sistema apresenta uma tela com a mensagem: “Ingrediente editado com sucesso” e o usuário clica em “OK”;</p> <p>05 - O sistema volta para a tela de Lista de Ingredientes que apresenta a lista com o Ingredientes recém atualizado;</p> <p>06 - O caso de uso se encerra.</p>
Pós-condições	Dados do ingrediente alterados.
FA_01 Cadastrar Ingrediente	1 – Este fluxo alternativo se inicia caso o usuário clique no botão “Cadastrar Novo”. O sistema executa o caso de uso UC002 – Cadastrar Ingrediente.
FA_02 Excluir Selecionado	1 - Este fluxo alternativo se inicia caso o usuário selecione um dos Ingredientes apresentado na tela Lista de Ingredientes e clique no botão “Excluir Selecionado”. O sistema executa o caso de uso UC004 – Excluir Ingrediente;
FA_03 Cancelar	<p>1 - Este fluxo alternativo se inicia caso o usuário clique no botão “Cancel”;</p> <p>2 - O sistema volta para o caso de uso UC001 – Listar Ingrediente.</p>
FE_01 Ingrediente Não Selecionado	<p>1 - Este fluxo de exceção se inicia caso o usuário clique no botão “Editar selecionado...” sem ter selecionado um Ingrediente da lista de Ingredientes;</p> <p>2 - O sistema apresenta uma tela com a mensagem: “Selecione um ingrediente da lista para poder editar”, o usuário clica em “OK” e o sistema volta para o caso de uso UC001 – Listar Ingrediente.</p>
FE_02 Campos Inválidos	<p>1 - Este fluxo de exceção se inicia caso o sistema verifique que o usuário digitou valores inválidos ou em branco nos campos nome, descrição e preço de compra;</p> <p>2 - O sistema apresenta uma tela com a mensagem: “Erros nos campos”, o usuário clica em “OK”, o sistema volta para o passo 3 do fluxo principal.</p>
Regra de Negócio	RN08 – Dados de Ingrediente
Requisito Funcional Relacionado	RF01 – Manipular Ingredientes

Fonte: O próprio autor (2012)

O quadro 6 mostra as especificações do caso de uso “UC003”, referente à editar ingredientes.

Quadro 7: Especificação do caso de uso UC004 – Excluir Ingrediente

Nome	UC004 – Excluir Ingrediente
Descrição	Este caso de uso descreve como é feita a exclusão de um ingrediente cadastrado no sistema.
Ator	Funcionário
Pré-condições	1 - Executar o caso de uso UC001 – Listar Ingrediente; 2 - Existir um Ingrediente para poder ser excluído.
Fluxo Principal	1 - O usuário executa o caso de uso UC001 – Listar Ingrediente. Seleciona um dos ingredientes da lista de ingrediente e clica no botão “Excluir selecionado...” (FA_01, FA_02, FA_03, FE_01); 2 - O sistema verifica se o ingrediente selecionado é utilizado por algum Item cadastrado no sistema. (FE_02); 3 - O sistema verifica que não existem Itens que utilizem o Ingrediente selecionado para exclusão e exclui o ingrediente; 4 - O sistema apresenta uma tela com a mensagem: “Ingrediente excluído com sucesso”; 5 - O sistema volta para a tela Lista de Ingredientes que não mostra mais o ingrediente recém excluído. 6 - O caso de uso se encerra.
Pós-condições	Ingrediente excluído.
FA_01 Cadastrar Ingrediente	1 – Este fluxo alternativo se inicia caso o usuário clique no botão “Cadastrar Novo”. O sistema executa o caso de uso UC002 – Cadastrar Ingrediente.
FA_02 Editar Ingrediente Selecionado	1 - Este fluxo alternativo se inicia caso o usuário selecione um dos ingredientes apresentados na tela Lista de Ingredientes e clique no botão “Editar selecionado...”, o sistema executa o caso de uso UC003 – Editar Ingrediente.
FA_03 Cancelar	1 - Este fluxo alternativo se inicia caso o usuário clique no botão “ <i>Cancel</i> ”; 2 - O sistema volta para o caso de uso UC001 – Listar Ingrediente.
FE_01 Ingrediente Não Selecionado	1 - Este fluxo de exceção se inicia caso o usuário clique no botão “Excluir selecionado...” sem ter selecionado um

	<p>Ingrediente da lista de Ingredientes;</p> <p>2 - O sistema apresenta uma tela com a mensagem: “Selecione um ingrediente da lista para poder excluir”, o usuário clica em “OK” e o sistema volta para o caso de uso UC001 – Listar Ingrediente.</p>
FE_02 Ingrediente é utilizado em um Item	<p>01 - Este fluxo de exceção se inicia caso o sistema verifique que o usuário está tentando excluir um ingrediente que é utilizado em um item;</p> <p>02 - O sistema apresenta uma tela com a mensagem: “Não foi possível deletar o ingrediente selecionado. O mesmo é utilizado por um ou mais itens, favor deletar o(s) item(s) primeiro.”, o usuário clica em OK;</p> <p>03 - O sistema volta para o caso de uso UC001 – Listar Ingrediente.</p>
Regra de Negócio	RN08 – Dados de Ingrediente
Requisito Funcional Relacionado	RF01 – Manipular Ingredientes

Fonte: O próprio autor (2012)

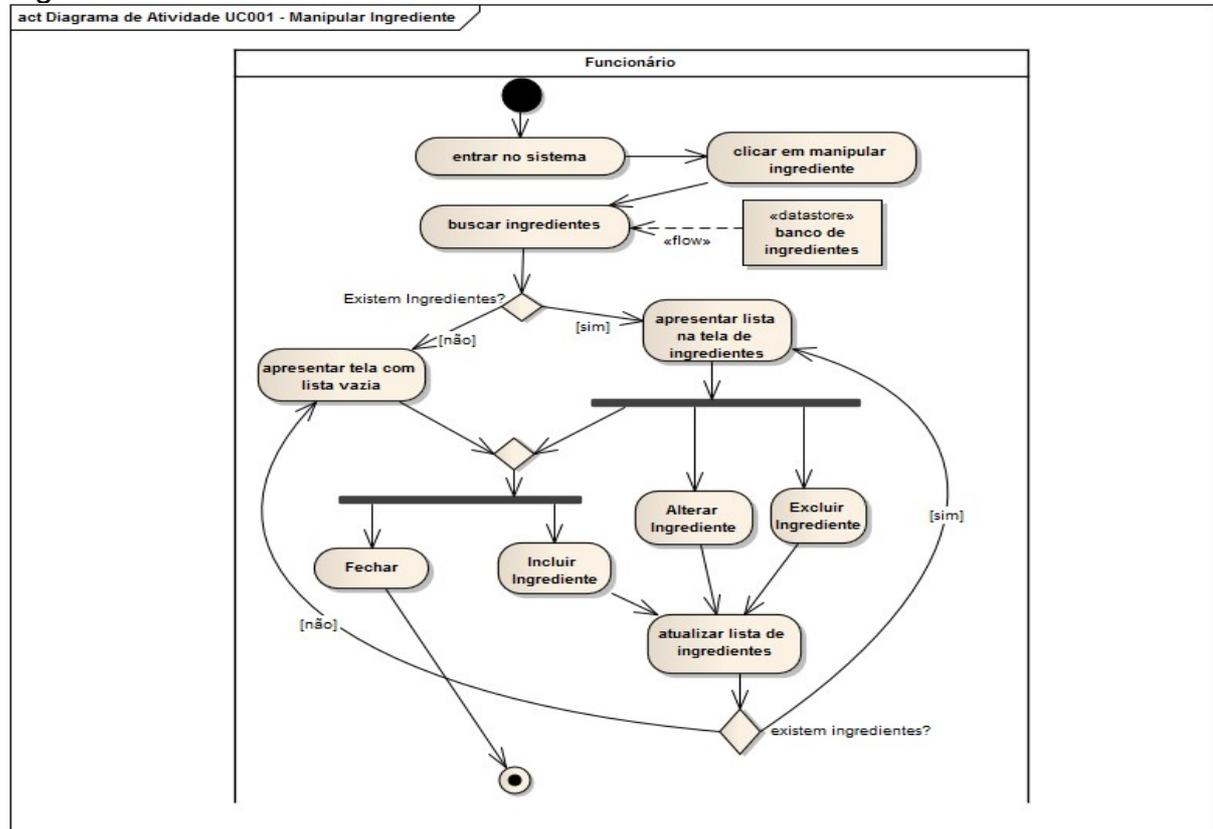
O quadro 7 mostra as especificações do caso de uso “UC004”, referente à excluir ingredientes.

4.5 DIAGRAMA DE ATIVIDADES

A seguir, representado pela figura 28, o diagrama de atividades dos casos de uso especificados.

O diagrama da figura 28 mostra como é feita a manipulação de ingredientes no sistema. A manipulação envolve as operações de listar, cadastrar, editar e excluir um ingrediente (UC001, UC002, UC003, UC004 respectivamente). O diagrama mostra alguns dos passos ou atividades que o ator faz até executar alguma das operações.

Figura 28: Diagrama de atividades dos casos de uso UC001 – Listar Ingrediente, UC002 – Cadastrar Ingrediente, UC003 – Editar Ingrediente e UC004 – Excluir Ingrediente



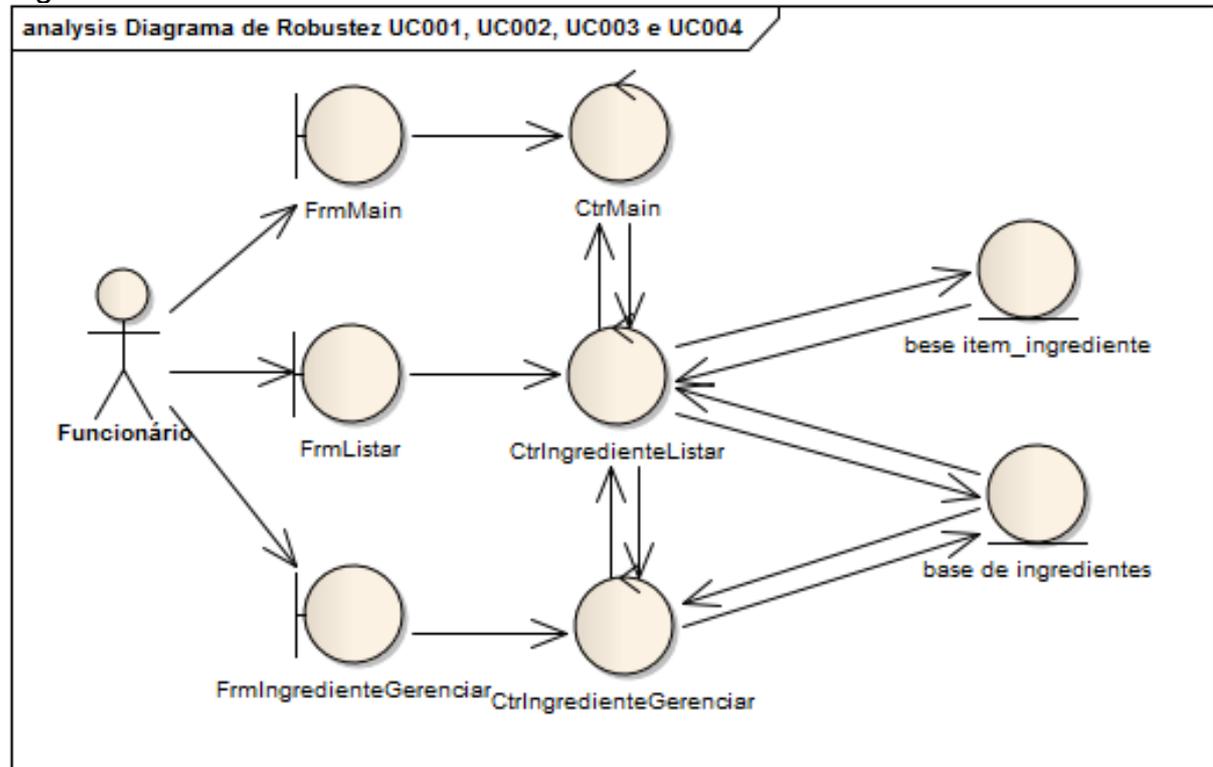
Fonte: O próprio autor (2012)

Segundo a Object Management Group (2011, p. 303), a modelagem de atividades enfatiza as sequencias e condições para coordenar comportamentos de baixo nível. As ações coordenadas por modelos de atividades podem ser inicializadas pelo término de outras ações, por um objeto ou dado que se tornou disponível, ou por eventos que podem ocorrer fora do fluxo.

4.6 DIAGRAMA DE ROBUSTEZ

A seguir, é apresentado na figura 29, o diagrama de robustez para o caso de uso UC001 – Listar Ingrediente, UC002 – Cadastrar Ingrediente, UC003 – Editar Ingrediente e UC004 – Excluir Ingrediente.

Figura 29: Diagrama de Robustez dos casos de uso UC001 – Listar Ingrediente, UC002 – Cadastrar Ingrediente, UC003 – Editar Ingrediente e UC004 – Excluir Ingrediente



Fonte: O próprio autor (2012)

Como pode ser observado, o ator funcionário pode interagir com o sistema a partir das três interfaces (FrmMain, FrmListar e FrmIngredienteGerenciar) essas são respectivamente a tela inicial do sistema, a tela de listagem de ingredientes que também é utilizada para excluir um ingrediente e por ela pode-se chegar a tela para cadastro e edição de um ingrediente. As interfaces são controladas pelos objetos (CtrMain, CtrlIngredienteListar e CtrlIngredienteGerenciar). Esses objetos são responsáveis por todo o controle das interfaces e faz sua ligação com o banco de dados ou seja as classes DAO, que foram mostradas em diagramas anteriormente.

Os dados manipulados pelas classes de controle são persistidos e utilizados em duas tabelas do banco de dados a tabela “item_ingrediente” e “ingrediente” que foram apresentadas no diagrama entidade relacionamento.

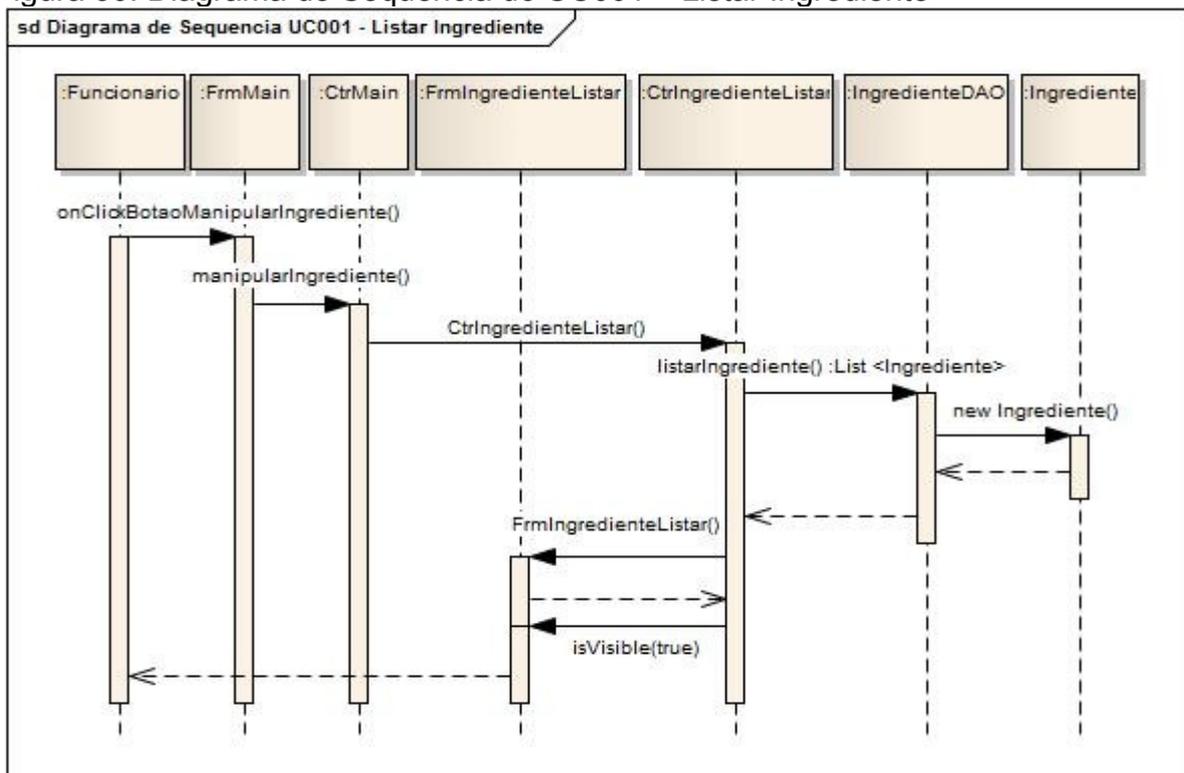
Segundo Silva et. al (2007) o modelo ou diagrama de robustez serve para se assegurar de que a especificação dos casos de uso está correta. É possível também descobrir classes novas que não foram colocadas no modelo de domínio.

O diagrama de robustez é feito com três objetos: o de interface, o de controle e o de entidade. As interfaces representam o objeto que o usuário interage, o controle faz a comunicação das interfaces com as entidades que, por sua vez, representam a manipulação com o banco de dados.

4.7 DIAGRAMA DE SEQUÊNCIA

A figura 30 apresenta o diagrama de sequência para o caso de uso UC001 – Listar Ingrediente.

Figura 30: Diagrama de Sequência do UC001 – Listar Ingrediente

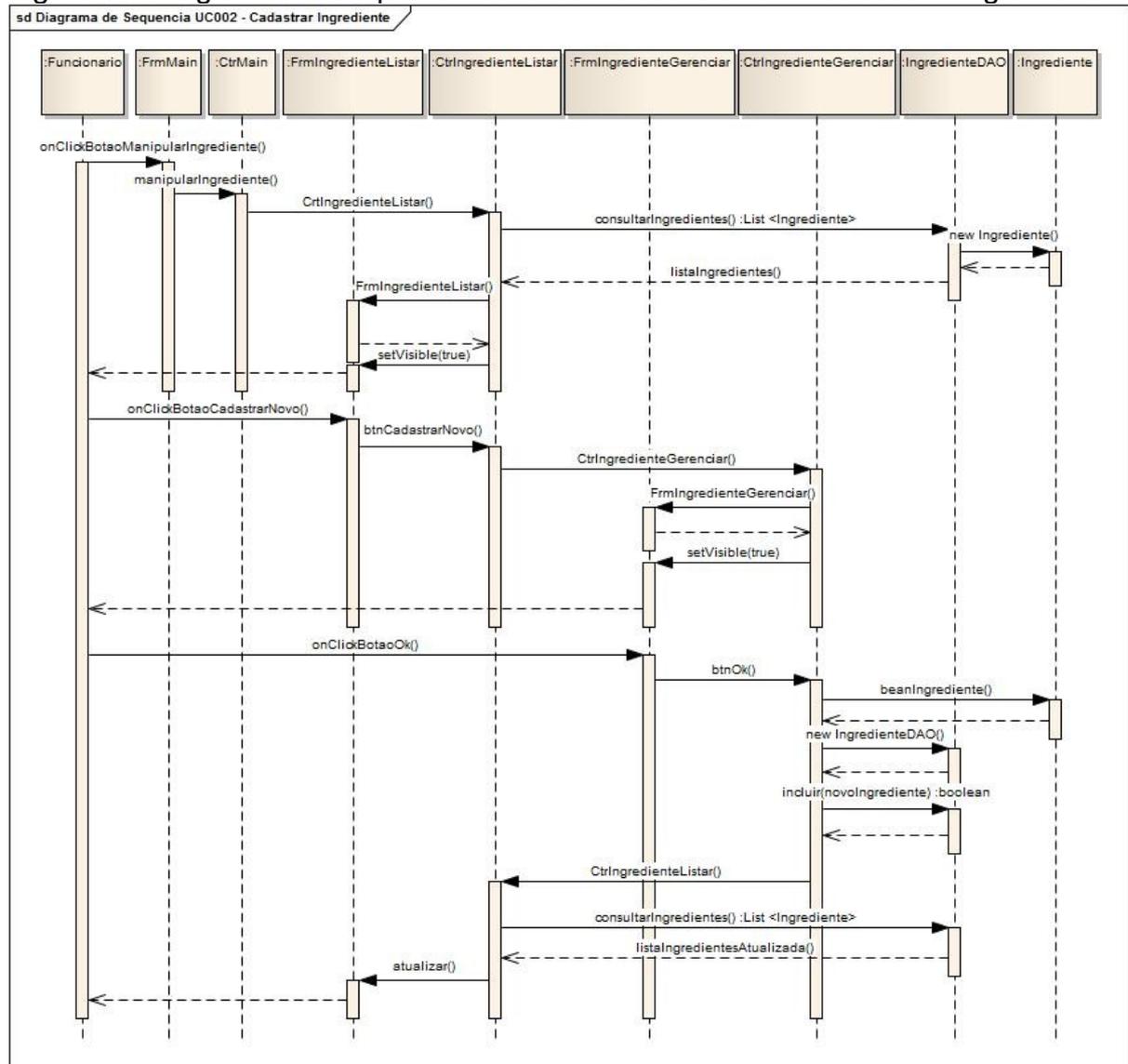


Fonte: O próprio autor (2012)

O diagrama de sequência do UC001 apresenta todas as classes do sistema envolvidas para que uma listagem de ingrediente possa ocorrer. Primeiramente o ator funcionário clica em manipular ingrediente na tela principal (FrmMain), depois é feita uma consulta na base de dados para que os ingredientes sejam listados na tela "FrmIngredienteListar".

Abaixo, é exibido o diagrama de sequência do caso de uso UC002 – Cadastrar Ingrediente (figura 31).

Figura 31: Diagrama de Sequência do caso de uso UC002 – Cadastrar Ingrediente



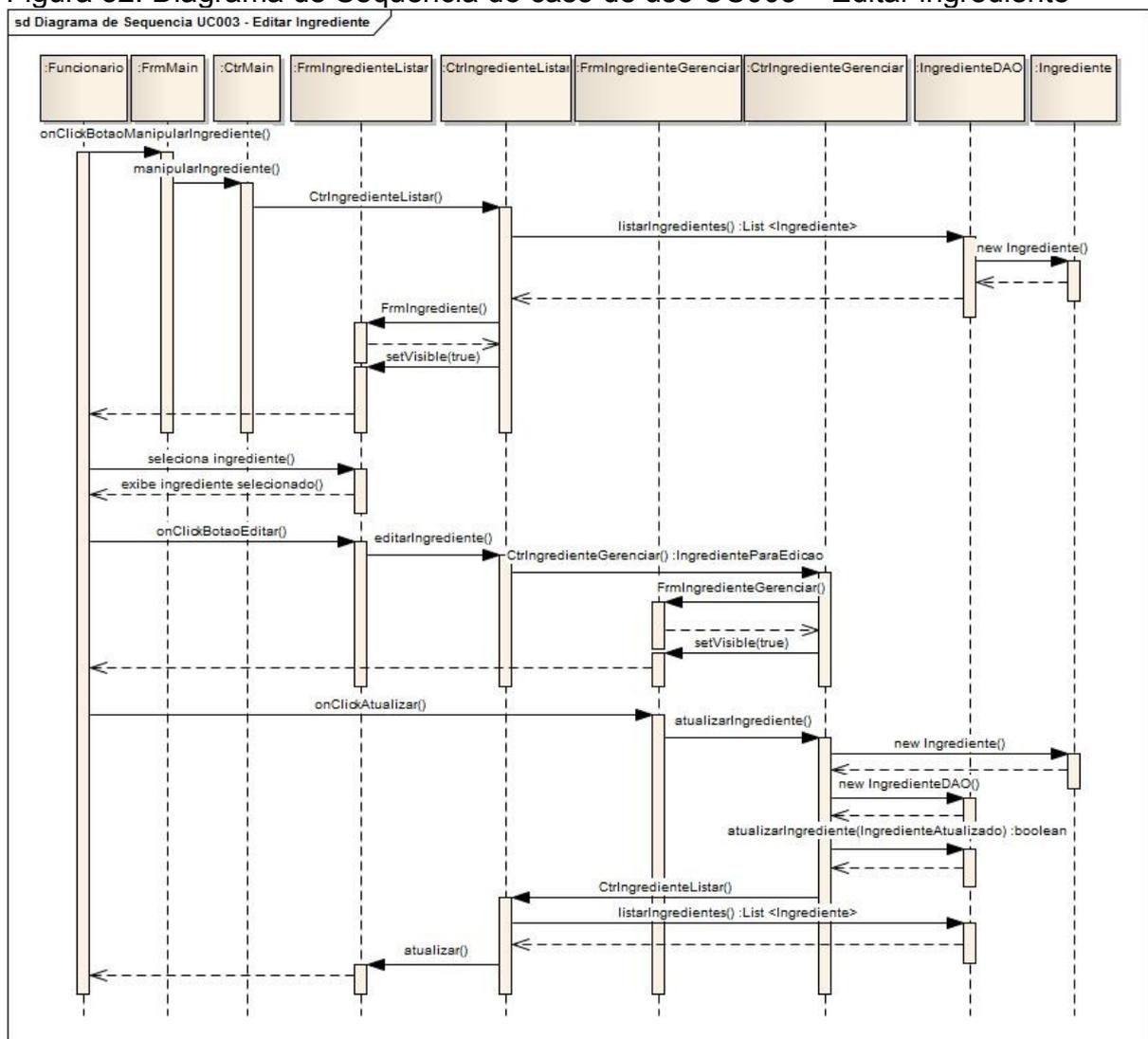
Fonte: O próprio autor (2012)

Como pode ser observado no diagrama a cima são mostradas todas as classes do sistema envolvidas para executar a operação de um cadastro de ingrediente. Primeiramente é feita a listagem de ingredientes que é uma operação obrigatória para que depois possa ser feita uma inclusão.

O próximo diagrama de sequência (figura 32) representa o caso de uso UC003 – Editar Ingrediente.

O diagrama da figura 32 também mostra todas as classes envolvidas para que possa ser executada uma edição de ingrediente. Como poderá ser visto é obrigatória a execução da listagem de ingredientes para que o usuário (funcionário) possa selecionar um dos ingredientes cadastrados e clicar em “editar” para que todas as informações sobre o ingrediente escolhido sejam exibidas em outra tela (FrmIngredienteGerenciar) e possam ser alteradas.

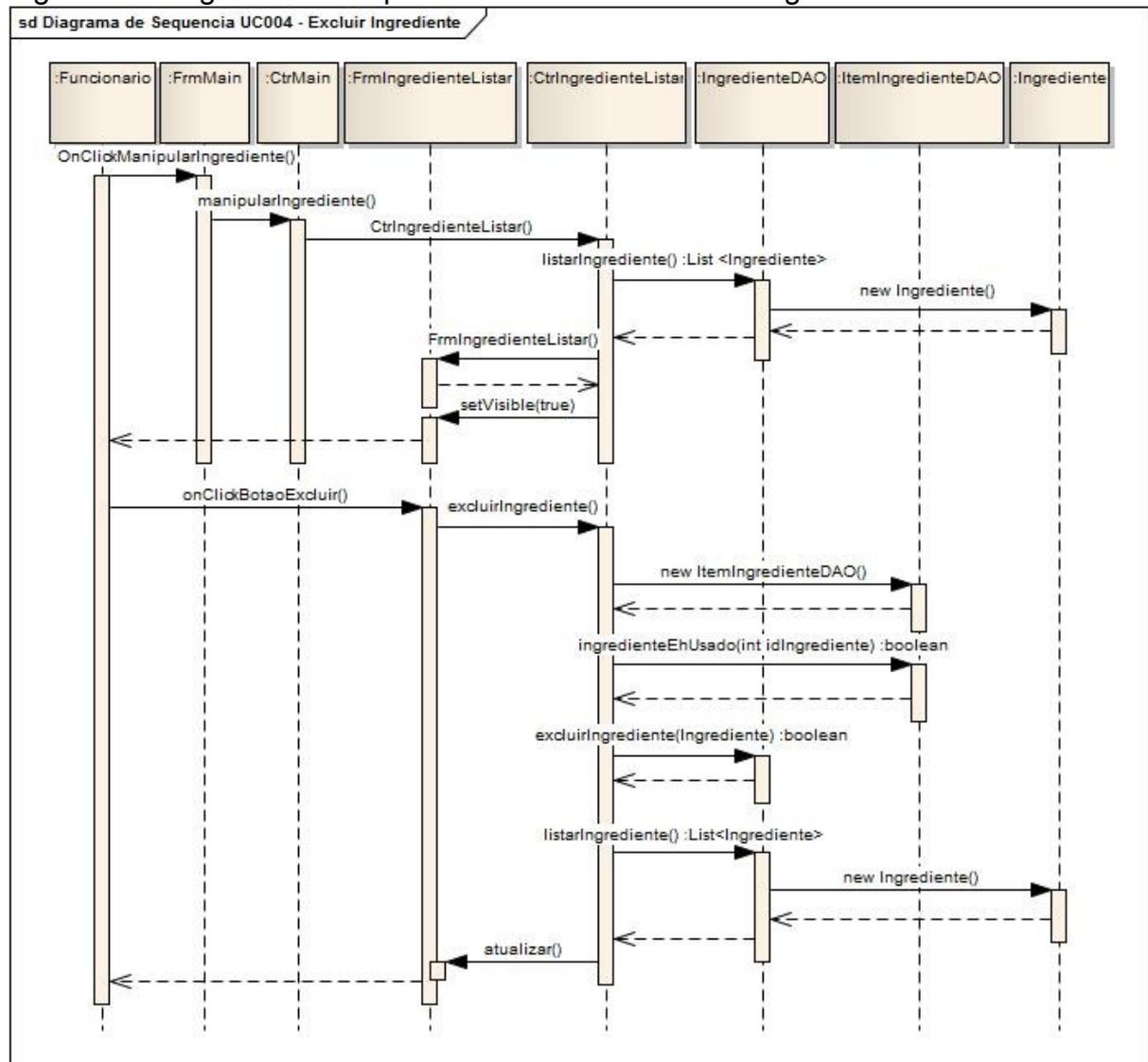
Figura 32: Diagrama de Sequência do caso de uso UC003 – Editar ingrediente



Fonte: O próprio autor (2012)

A figura 33 representa o diagrama de sequência da operação de excluir um ingrediente (UC004 – Excluir Ingrediente).

Figura 33: Diagrama de Sequência do UC004 – Excluir Ingrediente



Fonte: O próprio autor (2012)

Como pode ser observado no diagrama apresentado a cima são exibidas todas as classes envolvidas para que o usuário possa efetuar a exclusão de um ingrediente. Mais uma vez é obrigatória a listagem de ingredientes para que o funcionário selecione um e o exclua. Somente efetuando a listagem pode-se fazer as operações de incluir, alterar e excluir. Nesse diagrama pode ser visto que é envolvida mais uma classe DAO além da “IngredienteDAO” que é a classe “ItemIngredienteDAO”, isso ocorre porque é feita uma verificação se o ingrediente é utilizado em algum item, se o ingrediente é utilizado o sistema apresenta uma mensagem orientando o usuário a excluir primeiro os itens que usam esse ingrediente.

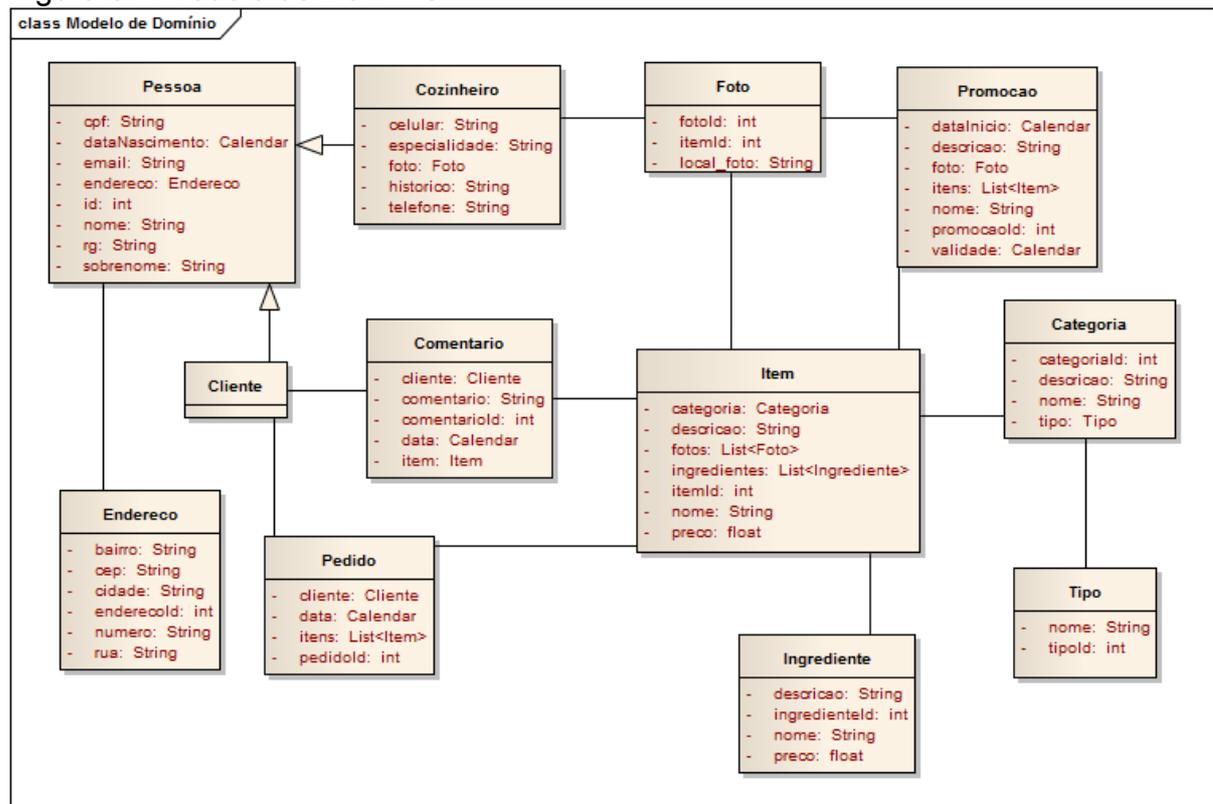
O diagrama de sequência envolve uma linha do tempo. A Globalcode (2004) explica que nesta linha do tempo são demonstrados os passos que o sistema irá percorrer para executar uma determinada tarefa. Na versão mais detalhada deste tipo de diagrama, os passos podem ser representados pelas chamadas de métodos.

Este tipo de diagrama pode ser utilizado para especificar a arquitetura do sistema e processos complicados, entre outros.

4.8 MODELO DE DOMÍNIO

Abaixo, na figura 34, é apresentado o modelo de domínio do projeto.

Figura 34: Modelo de Domínio



Fonte: O próprio autor (2012)

O modelo de domínio apresentado a cima mostra a refinação do levantamento das classes, seus atributos e suas relações feitos numa primeira

análise. Esse diagrama foi útil porque ajudou a planejar as classes e parte do banco de dados.

Segundo Silva et. al (2007) o modelo de domínio pode ser feito ao analisar os requisitos do sistema interpretando o que está escrito e procurando possíveis candidatos a classes, objetos, atributos e associações entre os mesmos.

5 DESENVOLVIMENTO

Neste capítulo é discutido todo o desenvolvimento do sistema, a apresentação de algumas partes do mesmo e mostradas as tecnologias utilizadas para auxiliar seu desenvolvimento.

5.1 TECNOLOGIAS ENVOLVIDAS

Para o desenvolvimento do sistema proposto, foram utilizadas as seguintes tecnologias, seguidas de suas descrições:

- **Android Software Development Kit** - Coleção de ferramentas para o desenvolvimento de *softwares* para a plataforma Android.
- **Dia** - *Software* para criação de variados tipos de diagramas em inúmeras áreas, inclusive desenvolvimento de *software* e modelagem de dados.
- **Dropbox** - Ferramenta para manter sincronia de arquivos entre um conjunto de computadores.
- **Eclipse** - Ambiente integrado de desenvolvimento de *software* com suporte a múltiplas tecnologias relacionadas e linguagens de programação.
- **Enterprise Architect** - Outro *software* para criação de diagramas utilizados para desenvolvimento de *software*.
- **Java** - Linguagem de programação utilizada para manipular e direcionar dados no sistema.
- **MySQL** - Sistema de gerenciamento de banco de dados utilizado para gerenciar e persistir dados de forma transparente para o sistema.

- **Pencil** – *Software* para desenvolvimento de protótipos, utilizado para elaborar os protótipos da interface do servidor e do cliente.
- **Subversion** – Sistema gerenciador de versões, amplamente utilizado para versionamento de código fonte.

Seguem, então, descrições mais detalhadas sobre cada tecnologia da lista e a forma que foram utilizadas para o desenvolvimento do sistema.

5.1.1 Android Software Development Kit

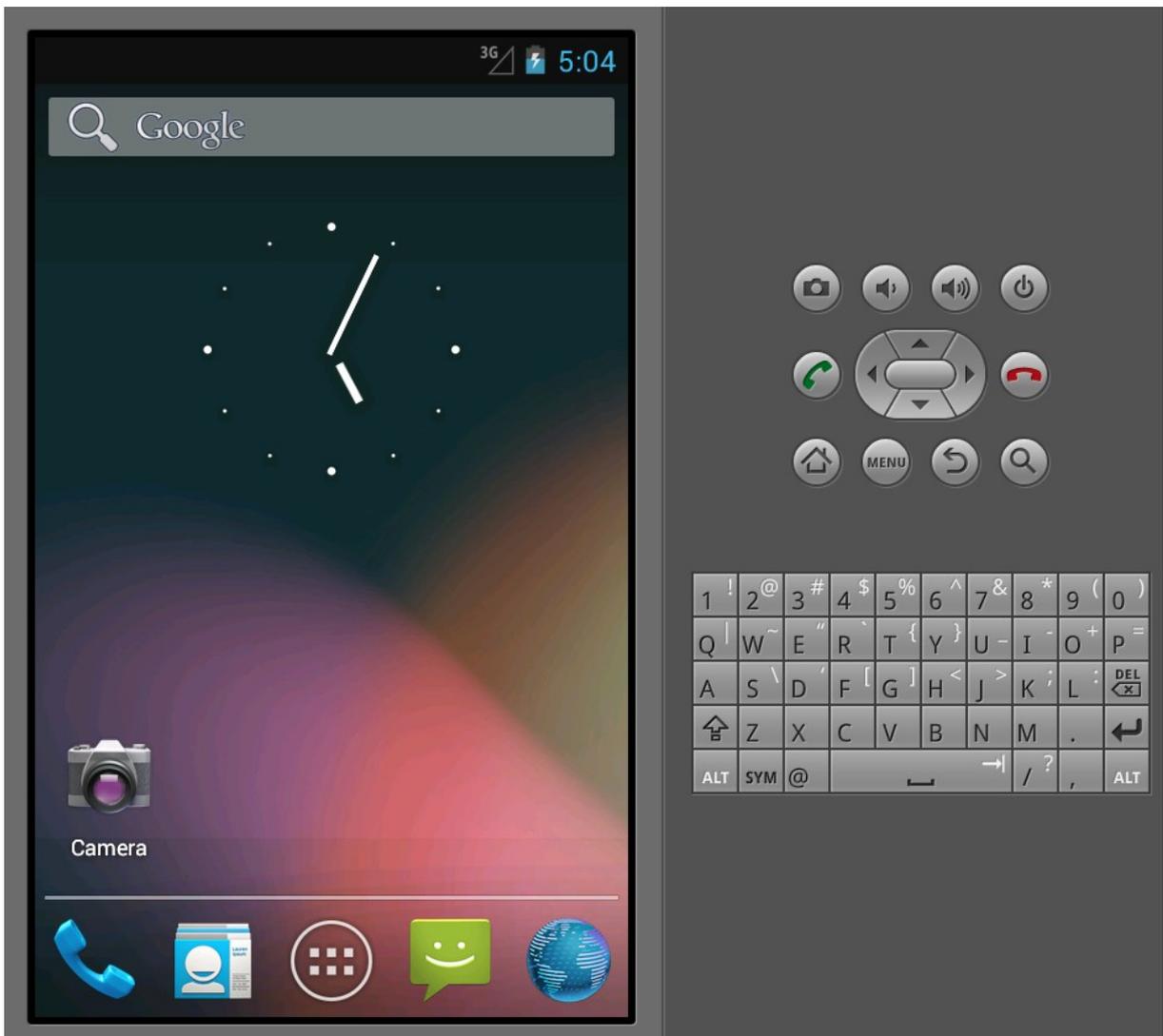
O *Android Software Development Kit* – ou SDK – é um conjunto de ferramentas essencial para o desenvolvimento na plataforma Android. Ele inclui ferramentas como emulador de dispositivos virtuais – exibido na figura 35 –, criador de interface gráfica, o *plug-in Android Development Tools (ADT)* para o ambiente de desenvolvimento integrado Eclipse, que é discutido além.

Segundo Ableson et al. (2012), quando abraçamos uma nova plataforma, a primeira tarefa de um desenvolvedor é compreender o SDK e seus componentes. Ableson et al. (2012) ainda falam sobre a documentação disponível no Android SDK:

Como em qualquer ambiente de desenvolvimento, é útil se familiarizar com a estrutura de classes, por isso, ter a documentação de referência à mão é uma boa ideia. O Android SDK inclui documentação em HTML, que consiste basicamente em páginas em formato Javadoc que descrevem os pacotes e classes disponíveis. A documentação do Android SDK está no diretório /doc da sua instalação do SDK. Por causa da natureza altamente mutável dessa plataforma, você pode querer ficar de olho em quaisquer mudanças no SDK. (ABLESON et al., 2012)

O Android SDK também conta com ferramentas de linha de comando para quando se precisa ter um maior controle sobre o ambiente de desenvolvimento, como o *Android Debug Bridge* – ou ADB. Segundo Ableson et al. (2012), com o ADB é possível interagir diretamente com o emulador Android a partir da linha de comando ou *script*.

Figura 35: Emulador de dispositivos virtuais do Android SDK.



Fonte: O próprio autor (2012).

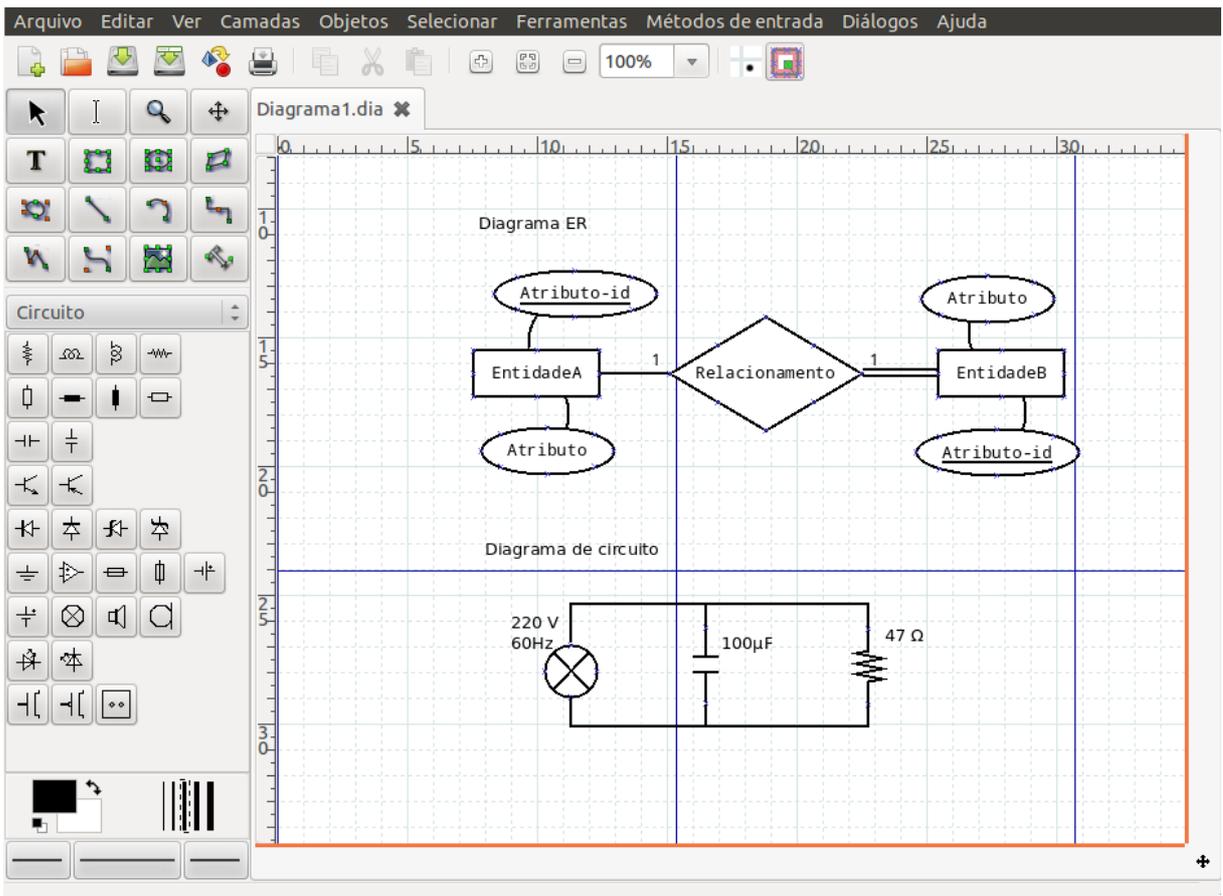
O Android SDK está disponível para *download* no endereço <<http://developer.android.com/sdk>>.

5.1.2 Dia

O Dia é uma ferramenta de código aberto, escrita na linguagem de programação Python, disponível para Windows e Linux, utilizada para criar inúmeros tipos de diagramas de diferentes áreas, entre eles mapas isométricos, pneumático/hidráulico, diagramas de rede no padrão Cisco, UML, ER, fluxogramas, e outros.

É possível criar novos tipos de padrões de diagramas, com tutoriais no *site*, e transferência de outros tipos existentes criados por outros usuários, como exibido na figura 36.

Figura 36: Captura de tela do *software* Dia



Fonte: O próprio autor (2012)

O Dia foi utilizado para a criação do modelo entidade-relacionamento do aplicativo.

O Dia está disponível para *download* no endereço <<https://live.gnome.org/Dia>>.

5.1.3 Dropbox

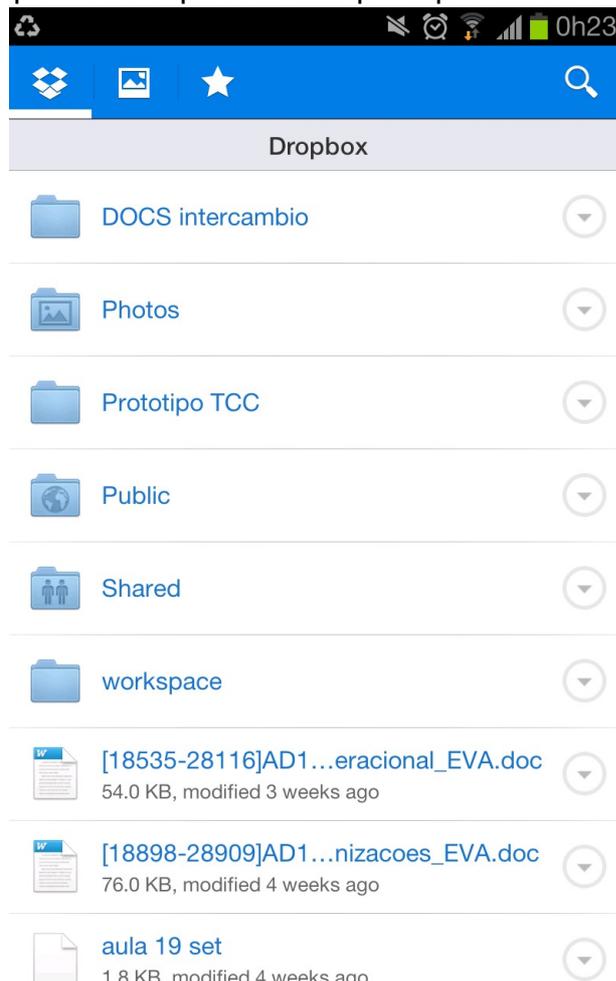
É um serviço que disponibiliza determinada quantidade de espaço para armazenamento na nuvem (em servidores *web*) onde é possível guardar os

arquivos, que depois de feito o *upload*, ficam disponíveis para serem acessados por seus proprietários e as pessoas que ele autoriza o acesso através da internet.

O serviço possui uma quantidade limitada para uso gratuito após o cadastro. É possível também efetuar uma assinatura mensal/anual para que o espaço seja aumentado, de acordo com os planos disponibilizados.

Para se ter acesso aos arquivos, pode-se instalar um *software*, que após instalado cria um diretório de pastas, exibido na figura 37, específico do Dropbox no computador, *smartphone* ou *tablet* do usuário e sincroniza aquilo que ele já fez o *upload* para o servidor. Outra forma de acesso é efetuando o *login* no site do Dropbox, onde também é possível fazer os *downloads* e *uploads* de arquivos.

Figura 37: Captura de tela da estrutura de pastas do aplicativo Dropbox para Android



Fonte: O próprio autor (2012)

O Dropbox também permite a criação de URLs que facilitam na distribuição rápida de arquivos.

O *software* possui versões para o Windows, Mac OS, Linux, Android, iOS, BlackBerry e Kindle Fire.

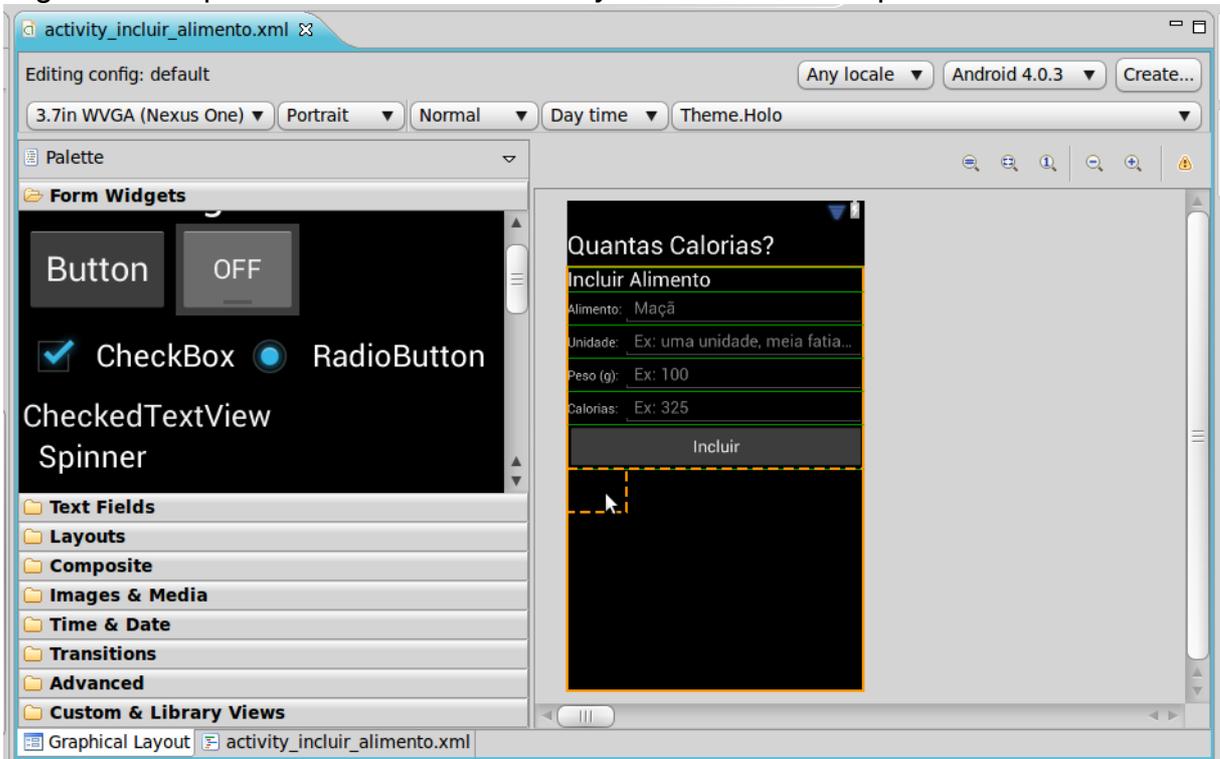
O Dropbox está disponível para *download* no endereço <<http://www.dropbox.com>>.

5.1.4 Eclipse

É uma IDE – ambiente de desenvolvimento integrado – de código aberto que foi criado pela IBM. Assim como outras IDEs ele é utilizado para o desenvolvimento de *software* e, atualmente, possui diversas versões disponíveis para download. Entre estas versões existem uma para desenvolvedores Java Enterprise Edition, C/C++, *testers*, desenvolvedores de aplicativos móveis, entre outras. Estas versões são disponibilizadas para Windows, Mac Os X e Linux.

Uma parte importante do Eclipse é a possibilidade da instalação de *plug-ins* que auxiliam no desenvolvimento em outras linguagens, além das já citadas. O *plug-in* para desenvolvimento de aplicações Android é o ADT – Android *Development Tools* – este *plug-in* torna mais fácil a criação de um projeto Android, construindo toda a estrutura de pacotes e classes básicas para uma aplicação. Ele fornece também editores de XML customizados e uma interface de fácil utilização e entendimento para a criação do *layout* – figura 38 – dos projetos onde é possível editar o mesmo “arrastando e soltando” (*drag-and-drop*) os componentes.

Figura 38: Captura de tela do editor de *layout* do ADT no Eclipse



Fonte: O próprio autor (2012)

O Eclipse está disponível para *download* no endereço <<http://www.eclipse.org>>.

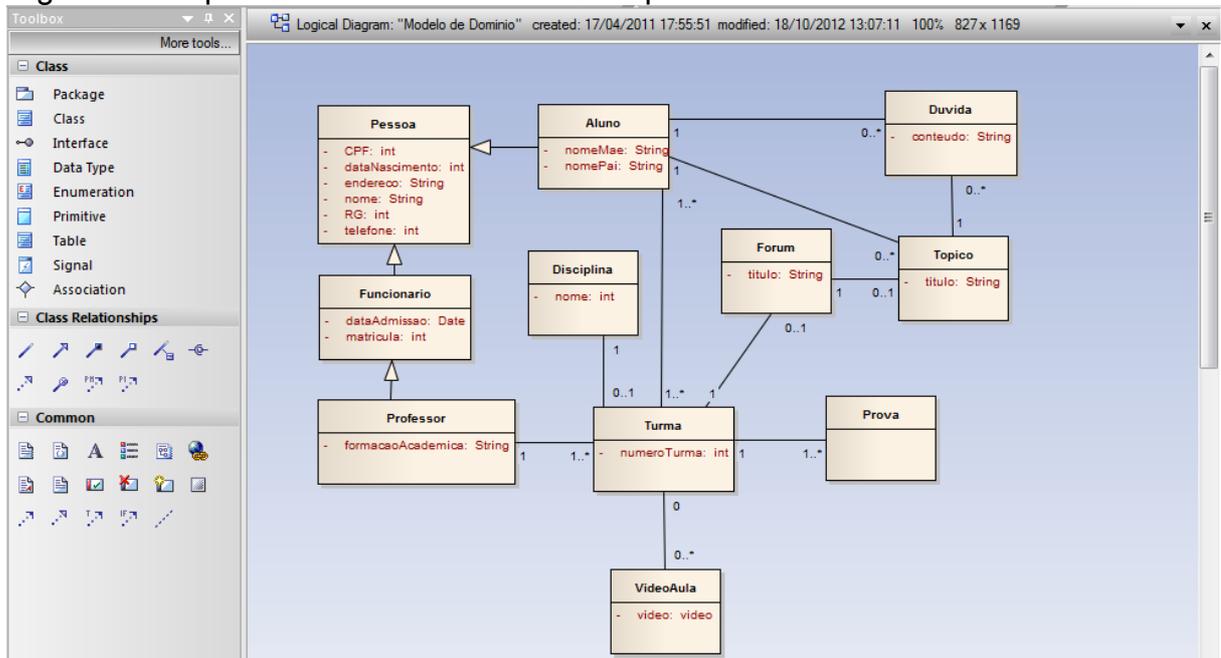
5.1.5 Enterprise Architect

Enterprise Architect, também conhecido como EA, é um software proprietário da Sparx Systems que é utilizado para o desenvolvimento de diversas etapas de um projeto, seja ele de *software* ou não. É compatível com UML (entre outros).

Com o EA é possível criar diversos tipos de diagramas e elaborar protótipos. Ele possui suporte a várias linguagens de programação e, inclusive, gera parte do código daquilo que o usuário modelar nele, como exibido na figura 39.

Segundo a Spark Systems (2012) o Enterprise Architect conta com cerca de trezentos mil usuários licenciados em todo o mundo e está presente em 130 países.

Figura 39: Captura de tela do *software* Enterprise Architect



Fonte: O próprio autor (2012)

O *Enterprise Architect* está disponível no endereço <http://www.sparxsystems.com/products/ea/index.html>.

5.1.6 Java

Java é uma linguagem de programação híbrida de alto nível orientada a objeto, com sintaxe similar à linguagem de programação C++. Deitel e Deitel (2001) falam um pouco sobre linguagens de programação de alto nível:

O uso de computadores aumentou rapidamente com o advento das linguagens simbólicas, mas estas ainda exigiam muitas instruções para realizar até as tarefas mais simples. Para acelerar o processo de programação, foram desenvolvidas as linguagens de alto nível, nas quais uma única instrução realiza tarefas significativas. Programas tradutores chamados compiladores convertem os programas em linguagens de alto nível para linguagem de máquina. As linguagens de alto nível permitem que os programadores escrevam instruções que pareçam quase com o inglês comum e contêm notações matemáticas comumente usadas. (DEITEL E DEITEL, 2001).

Deitel e Deitel (2001) também explicam sobre o paradigma da orientação a objetos:

Objetos são, essencialmente, componentes de software reutilizáveis que modelam coisas do mundo real. Os desenvolvedores de software estão descobrindo que usar uma abordagem de implementação e projeto modulares, orientada a objetos, pode tornar os grupos de desenvolvimento de software muito mais produtivos do que é possível usando-se técnicas de programação anteriormente populares, como a programação estruturada. Os programas orientados a objetos são fáceis de entender, corrigir e modificar. (DEITEL E DEITEL, 2001)

A grande diferença entre o Java e outras linguagens de programação mais comuns é o conceito de ser uma plataforma híbrida, garantindo de forma segura sua execução em inúmeros *hardwares* e sistemas operacionais. Mattos (2007) dá uma breve explicação sobre este conceito:

As plataformas que implementam o conceito híbrido têm como objetivo a segurança das verificações existentes em um processo de compilação e a portabilidade dos ambientes interpretados. O processo adotado para implementação do modelo híbrido baseia-se na utilização de uma representação intermediária denominada bytecode, que é gerada pelo compilador e interpretada no momento da execução. (MATTOS, 2007)

De tal forma descrita, o Java torna-se uma linguagem independente de plataforma, porém, depende que seu interpretador, ou máquina virtual, esteja instalado para garantir execução.

O ambiente de execução do Java está disponível para *download* no endereço <<http://www.java.com/>>.

5.1.7 MySQL

O MySQL foi o Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido para ser utilizado neste projeto. A respeito de SGBDs, Novelli (2006) explica que: “Um SGBD – Sistema de Gerenciamento de Banco de Dados é uma coleção de programas que permitem ao usuário definir, construir e manipular Bases de Dados para as mais diversas finalidades.”

O MySQL utiliza a linguagem SQL (Linguagem de Consulta Estruturada), que é uma linguagem muito utilizada por diversos SGBDs. Sobre a linguagem SQL, Mendes (2011) comenta o seguinte:

O órgão American National Standards Institute (ANSI) ficou responsável pela padronização desta linguagem e de tempos em tempos realiza encontros entre fabricantes para discutir a linguagem SQL e propor melhorias; no entanto esta padronização não impede que cada fabricante personalize a linguagem SQL para atender suas necessidades, e é aí que surgem os dialetos. Por exemplo, o “dialeto” do gerenciador de banco de dados Oracle é o PL/SQL; o do SQL Server é o T-SQL (transact SQL) e etc. (MENDES, 2011)

O MySQL é bastante compatível com diversas linguagens de programação e possui alguns *drivers* para elas como o *driver* JDBC (Java *Database Connectivity*) que é o *driver* necessário para que o Java possa manipular o banco de dados utilizado no projeto.

Um dos motivos que levou os autores a escolher este SGBD foi o fato de ele ser um dos mais populares, o que faz com que seja possível encontrar com facilidade muito material, como exemplos e tutoriais, para sanar as dúvidas e resolver os problemas.

O MySQL está disponível para *download* no endereço <<http://www.mysql.com/downloads/>>.

5.1.8 Pencil

Pencil é um *software* de código aberto que serve para a criação de protótipos de interfaces gráficas.

Sua utilização é bastante intuitiva. Pode-se criar os seguintes tipos de protótipos: de aplicativos móveis, de programas para *desktop*, de sites, alguns tipos de diagramas, entre outras possibilidades.

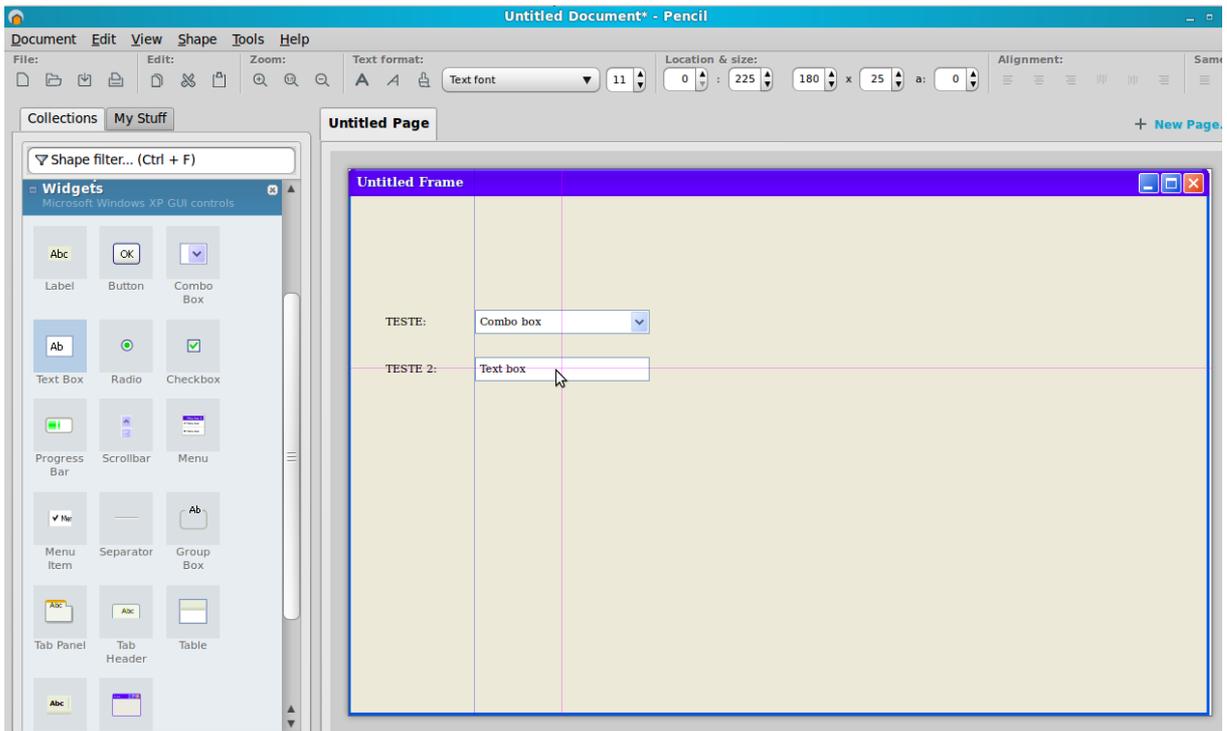
Ele possui diversos modelos de interfaces gráficas que simulam a aparência de alguns sistemas operacionais como: Windows XP, Android, iOS, entre outros. Ele também possibilita a instalação de novos modelos, que podem ser encontrados para *download* gratuitamente em seu *site*.

Com essa ferramenta é possível exportar imagens e até protótipos interativos parcialmente funcionais em HTML (onde ao clicar em um botão ou *link*, por exemplo, o usuário é levado a outra página do protótipo).

O Pencil possui versões para Linux, Windows e Mac OS X.

A figura 40 exibe uma captura de tela da interface do Pencil em sua versão 2.0.2.

Figura 40: Captura de tela do *software* Pencil



Fonte: O próprio autor (2012)

O Pencil está disponível para *download* no endereço <<http://pencil.evolus.vn/>>.

5.1.9 Subversion

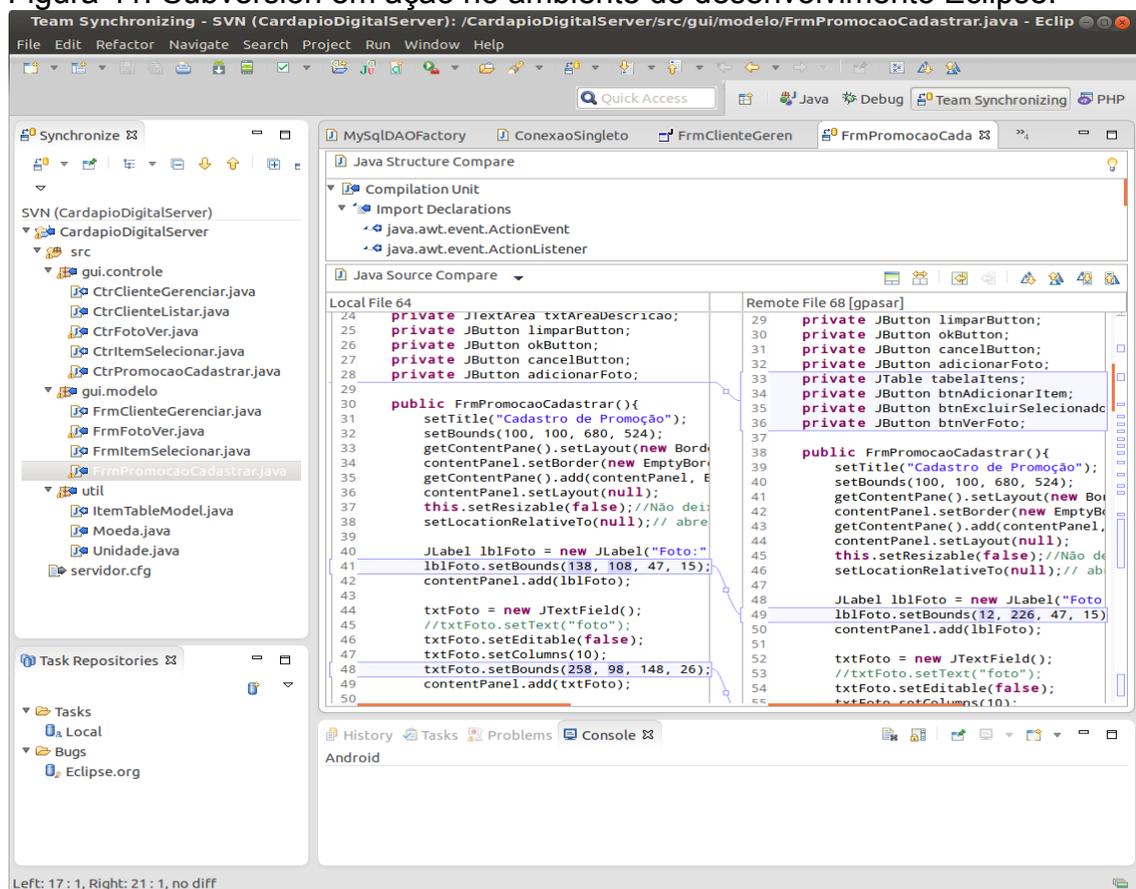
O Subversion – ou SVN – é um *software* de código aberto utilizado para o controle de versionamento de arquivos em um diretório. Sua maior utilização é na área de desenvolvimento, gerenciando linha a linha mudanças no código fonte para organizar o desenvolvimento em equipe. Segundo Collins-sussman, Fitzpatrick e Pilato (2008), no livro oficial publicado no *website*:

[...] isto permite a recuperação de versões antigas de seus dados ou examinar o histórico de mudanças. Em função disto, muitas pessoas

pensam deste sistema de controle de versão como uma "máquina do tempo". O Subversion pode operar através de redes, o que permite seu uso por diferentes pessoas em diferentes computadores. A certo nível, a habilidade de várias pessoas de modificarem e gerenciarem o mesmo conjunto de dados de seus respectivos locais promove a colaboração. (COLLINS-SUSSMAN, FITZPATRICK E PILATO, 2008)

A imagem 41 mostra o SVN em ação no ambiente de desenvolvimento Eclipse.

Figura 41: Subversion em ação no ambiente de desenvolvimento Eclipse.



Fonte: O próprio autor (2012)

O Subversion está disponível para *download* no endereço <<http://subversion.apache.org/packages.html>>.

5.2 APRESENTAÇÃO DO SISTEMA

O sistema constitui-se basicamente de duas partes: o servidor e os clientes (*tablets*).

O cliente possui um menu principal, telas que listam as bebidas e os pratos, agrupados por categorias, telas que apresentam um item com mais detalhes, tela com a conta da mesa e tela para cadastro do cliente.

O servidor possui interfaces para a sua configuração, a configuração do banco de dados, telas para manipulação (operações de cadastro, consulta, exclusão e edição) dos itens, tipos, categorias, ingredientes, cozinheiros, promoções e de pedidos.

5.2.1 Cardápio

Seguem as capturas de telas do protótipo do cardápio digital. A figura 42 é uma captura da tela principal do cardápio. Nela são apresentadas imagens “clicáveis” que são os tipos de itens que o restaurante disponibiliza. Itens no caso são pratos, entradas, bebidas, sobremesas, entre outros.

Figura 42: Menu principal do Cardápio Digital



Fonte: O próprio autor (2012)

Caso o usuário clique em uma dessas imagens, o sistema apresenta uma tela que lista todos os pratos agrupados por categoria.

A figura 43 mostra esta agrupação com todos os pratos cadastrados no sistema, os mesmos são agrupados por categoria (ex.: massas, saladas, carnes). Além do nome dos pratos é apresentada uma pequena foto e o preço do prato.

Figura 43: Menu de pratos



Fonte: O próprio autor (2012)

Se o usuário clicar em um dos pratos é exibida uma tela (figura 44) que mostra detalhes do mesmo.

A figura 44 mostra a estrutura da tela de detalhe de um item, no caso mostra os detalhes do prato “Lasanha de Frango”.

Figura 44: Detalhamento de um item



Fonte: O próprio autor (2012)

A figura 45, mostra a tela apresentada quando o usuário clicar no botão “Minha Conta”, que é basicamente uma lista do que o usuário já pediu. Esta lista é obtida fazendo a consulta no servidor de todos os pedidos desta seção deste *tablet* em específico, e então enviada pela rede sem fio para o *tablet*.

Figura 45: tela “Minha Conta” com os itens pedidos pelo usuário



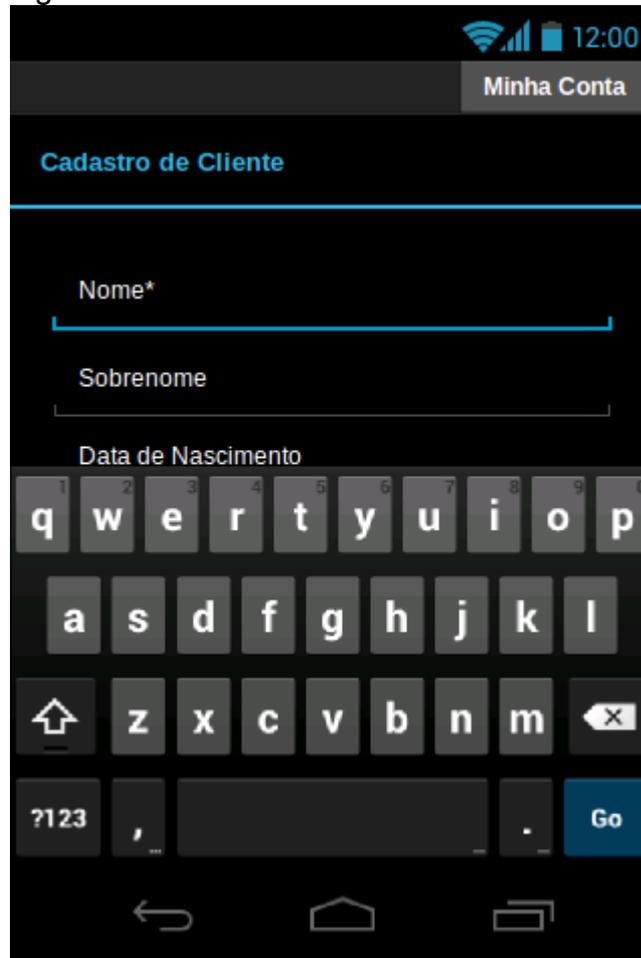
Prato	Quantidade	Valor
Lasanha de Frango	1	25,00
Coca Cola	2	7,00

Total: R\$ 32,00

Fonte: O próprio autor (2012)

A lista possui o nome do item, a quantidade pedida e o valor (quantidade pedida multiplicada pelo valor unitário de cada item). Ela mostra também o valor total da conta da mesa, para que o usuário tenha controle de quanto está gastando.

Figura 46: tela de Cadastro de Cliente



Fonte: O próprio autor (2012)

A figura 46 mostra uma captura da tela de cadastro de cliente, onde são apresentados alguns campos para que o cliente preencha seus dados (se desejar). Quando o usuário clica num campo o Android exibe um pequeno teclado virtual.

5.2.2 Servidor

Neste tópico são apresentadas telas do servidor, que gerencia os dados do sistema, pedidos, clientes cadastrados, contas em aberto, promoções que serão exibidas na tela do cardápio.

A figura 47 mostra a tela inicial do servidor. Nesta tela é apresentada uma lista dos novos pedidos do sistema, onde é possível exibir melhores detalhes sobre

um selecionado. Pedidos finalizados são escondidos desta lista - por exemplo, pedidos já entregues. Há também uma lista de cardápios conectados ao servidor, com também a possibilidade de exibir mais detalhes, como endereço IP, horário em que a conexão foi feita, perfil do cliente cadastrado, se usando um. Em status do servidor, são exibidas informações relativas à máquina local e serviços, como o horário atual, o gerenciador de banco de dados em uso, endereço IP e porta selecionadas, tempo em que o servidor está ativo, número de cardápios conectados, e quais destes possuem um perfil de cliente cadastrado no estabelecimento ou estão em modo anônimo. Também é exibido o lucro total da sessão.

Figura 47: Captura da tela inicial do servidor



Fonte: O próprio autor (2012)

A configuração do servidor é exibida conforme figura 48. Nesta tela, todos os campos são de preenchimento obrigatório e configuram a base do servidor, como a porta para as conexões de entrada dos cardápios, qual gerenciador de banco de dados será usado pelo sistema, o endereço IP que este

gerenciador de banco de dados encontra-se instalado - dando a oportunidade para se ter uma máquina dedicada apenas para isso-, ou mesmo salvar tais dados na rede de Internet.

Figura 48: tela de configuração do servidor

Configurar servidor

Cardápio

Porta: 4445

Banco de Dados

Banco: MYSQL

Endereço IP: localhost

Porta: 3306

Usuário: root

Senha: ****

As configurações serão salvas em:
/home/bruno/workspace/CardapioDigitalServer/s

Limpar OK Cancelar

Fonte: O próprio autor (2012)

A figura 49 mostra a tela de manipular itens. Nela é exibida uma tabela com todos os itens cadastrados no sistema. Existe um campo onde o usuário pode fazer a busca de um item por nome.

Por essa tela pode-se selecionar um item para poder ver mais informações do mesmo e editá-lo e também excluí-lo.

A tela também apresenta um botão que leva a uma tela de cadastro que será mostrada mais a frente.

Figura 49: Tela para manipulação de itens

Nome	Descrição	Preço
Pizza Gigante de Calabreza	Pizza de Calabreza tamanh...	45,5
Jonnie Walker Red Label	Famoso Whisky da mais fa...	12
Calzone de Calabresa	Calzone de Calabresa. Rec...	7,5
Salada Brasileira	Salada típica do brasil.	4,2

Fonte: O próprio autor (2012)

A figura 50 mostra como é feito o cadastro de novos itens do cardápio, itens estes assim que cadastrados geram uma notificação nos cardápios que baixarão imediatamente uma lista de itens atualizada, incluindo o novo item cadastrado, mostrando o poder e dinamicidade do sistema.

O preço de compra nesta tela é calculado a partir do cálculo feito pelo preço de compra unitário de cada ingrediente previamente cadastrado no sistema, multiplicado pela sua quantidade utilizada neste item, especificada na lista de ingredientes. É possível especificar determinada quantidade de fotos de tal item, que será exibida na tela dos cardápios assim que o cliente clicar neste item para exibir informações detalhadas e, possivelmente, fazer o pedido. O preço de venda pode ser especificado em diferentes moedas, e o cálculo será feito de forma transparente pelo sistema para a moeda selecionada no cardápio do cliente usando a taxa de conversão mais atualizada.

Figura 50: Tela de cadastro de itens do sistema.

Gerenciar Itens

Nome:

Descrição

Preço de venda: BRL

Categoria: Gerenciar...

Ingredientes selecionados:
0 Miligramas de Fatia de maçã, 0 Miligramas de Ketchup,

Preço de compra: BRL 0.0

Ingredientes:		
Fatia d...	0,08	0 Miligra...
Ketchup	2,75	0 Miligra...

Ver
Adicionar...
Remover

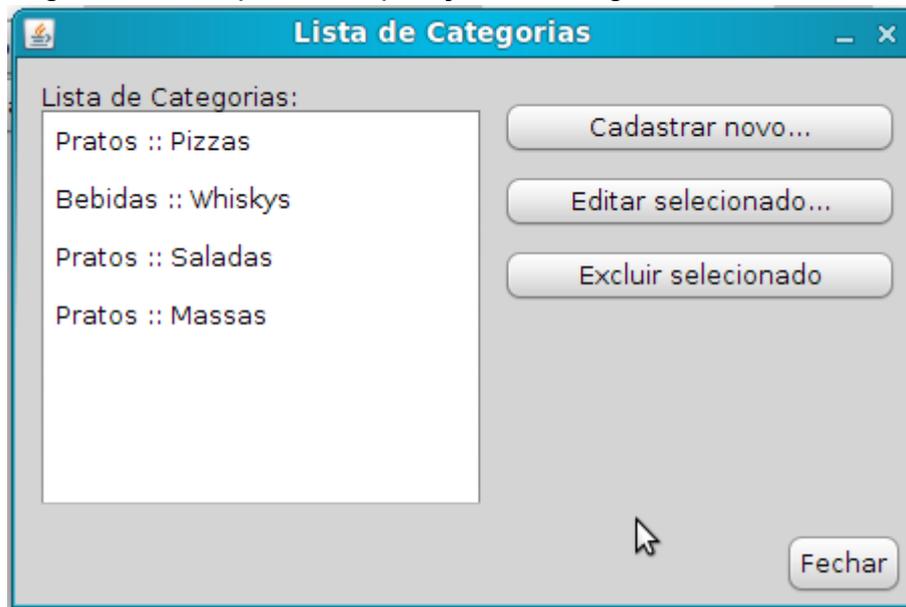
Limpar OK Cancel

Fonte: O próprio autor (2012)

A tela a seguir (Figura 51) tem a função de manipular categorias. Ela primeiramente mostra uma lista com todas as categorias cadastradas no sistema. Pode-se selecionar uma categoria da lista para fazer as operações de edição e exclusão.

Também é possível cadastrar uma nova categoria clicando no botão “Cadastrar novo...”.

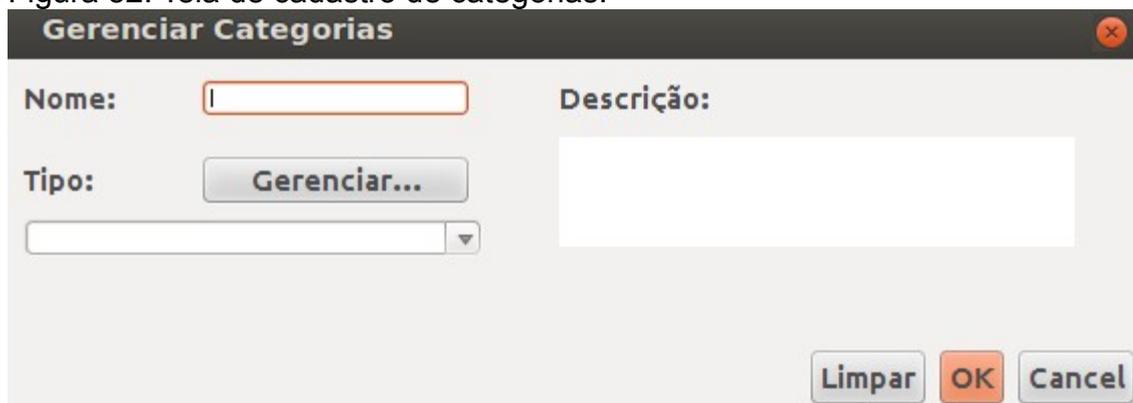
Figura 51: Tela para manipulação de categorias



Fonte: O próprio autor (2012)

A tela de cadastro de categorias é exibida conforme figura 52. Nesta tela, é possível especificar um nome e uma breve descrição para uma categoria, e selecionar qual o seu tipo. No sistema, uma categoria é um subtipo de tipo.

Figura 52: Tela de cadastro de categorias.

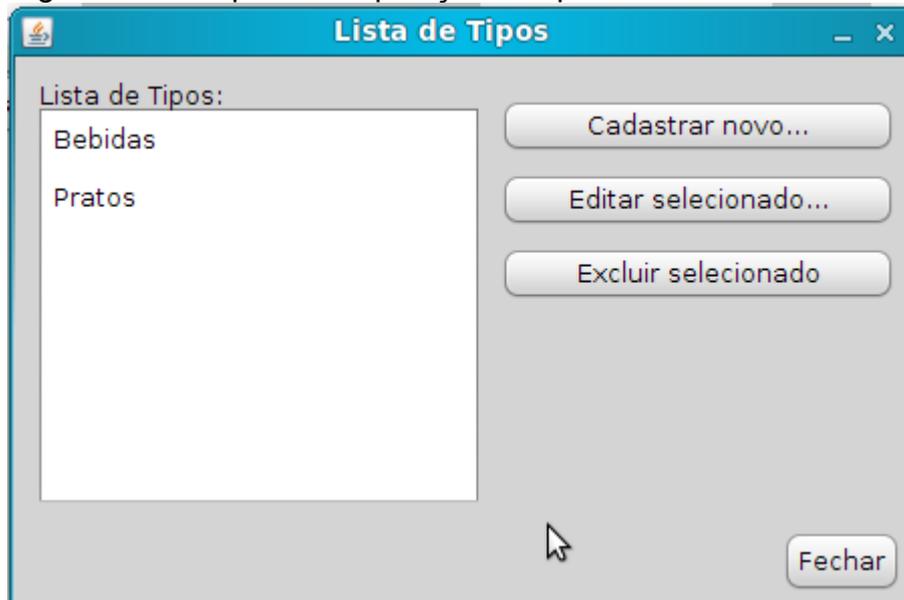


Fonte: O próprio autor (2012)

Na figura 53 é apresentada a tela para a manipulação de tipos. Ela exibe uma lista com todos os tipos cadastrados no sistema. Ao selecionar um tipo dessa lista é possível fazer as operações de editar e excluir.

Ao clicar em “Cadastrar Novo...” é apresentada uma tela para o cadastro de um novo tipo.

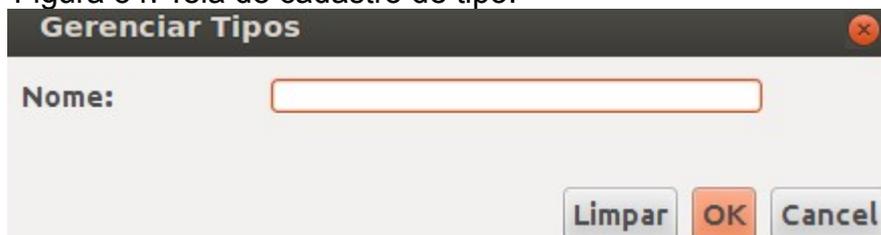
Figura 53: Tela para manipulação de tipos



Fonte: O próprio autor (2012)

Os tipos são usados como primeira tela a ser exibida no cardápio, e como exemplo, podem contar com Entrada, Prato principal, Sobremesa, Bebidas. Cada um destes tipos possui um subtipo especificado pelas categorias que é exibido em formato de lista nos cardápios. A figura 54 mostra a tela de cadastro de tipos.

Figura 54: Tela de cadastro de tipo.

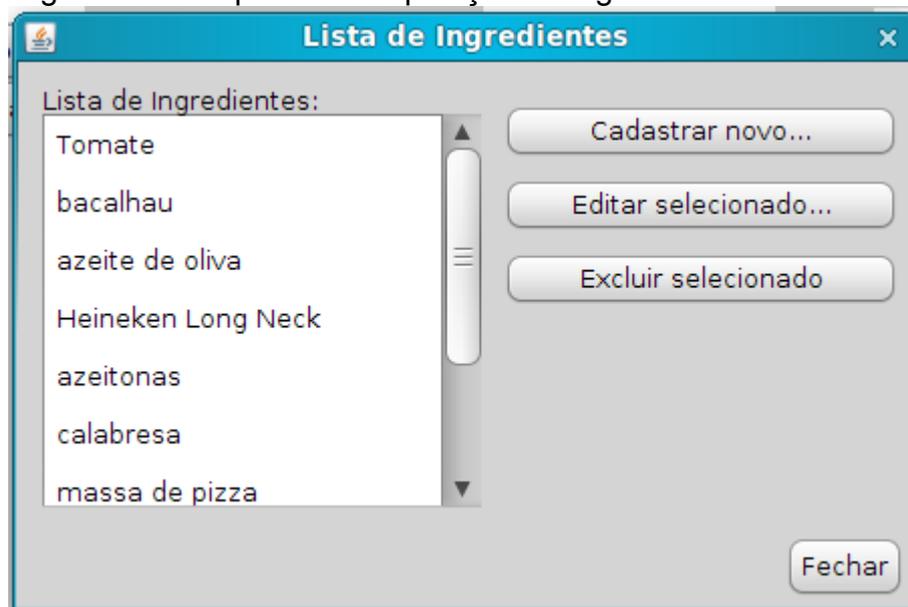


Fonte: O próprio autor (2012)

A tela a seguir (Figura 55) exhibe como é feita a manipulação de ingredientes no sistema. A tela mostra uma lista com todos os ingredientes cadastrados no sistema.

É possível selecionar um ingrediente para fazer sua edição ou excluí-lo. Ao clicar no botão “Cadastrar Novo...” é apresentada uma tela para o cadastro que é mostrada mais a frente.

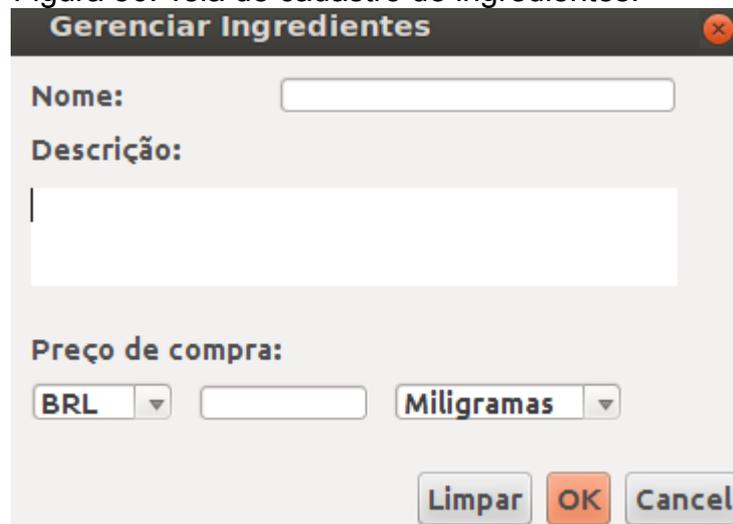
Figura 55: Tela para a manipulação de ingredientes



Fonte: O próprio autor (2012)

Na tela de cadastro de ingredientes, conforme figura 56, é possível criar um ingrediente que será usado para novos pratos no sistema. Nesta tela, cadastra-se seu nome, uma descrição, e o seu preço de compra, em uma moeda especificada, por uma unidade também especificada, tais como: miligramas, mililitros ou unidades. Estas informações são utilizadas para calcular o preço de compra de qualquer item do sistema.

Figura 56: Tela de cadastro de ingredientes.



Nome:

Descrição:

Preço de compra:
BRL Miligramas

Limpar OK Cancel

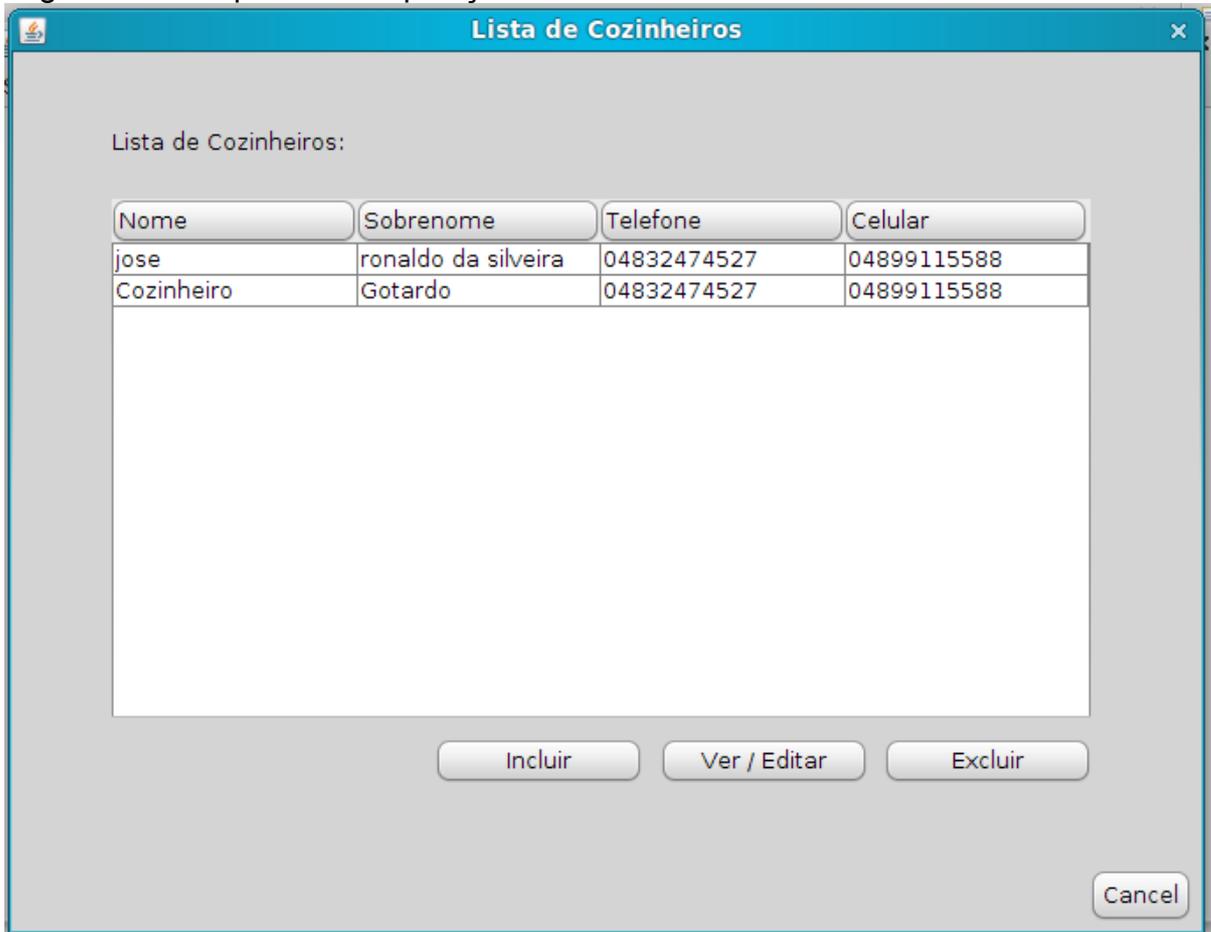
Fonte: O próprio autor (2012)

A figura 57 mostra a tela para manipulação de cozinheiros. Nela é exibida uma tabela com o nome, sobrenome, número de telefone e número do celular de todos os cozinheiros cadastrados no sistema.

Para ver e editar os dados de um cozinheiro é preciso selecioná-lo na tabela e clicar no botão “Ver / Editar”. Para excluir um cozinheiro é preciso selecionar o mesmo e clicar em “Excluir”.

Ao clicar em “Incluir” é exibida uma tela para o cadastro de um novo cozinheiro.

Figura 57: Tela para a manipulação de cozinheiros



Lista de Cozinheiros:

Nome	Sobrenome	Telefone	Celular
jose	ronaldo da silveira	04832474527	04899115588
Cozinheiro	Gotardo	04832474527	04899115588

Incluir Ver / Editar Excluir

Cancel

Fonte: O próprio autor (2012)

A figura 58 mostra a tela de cadastro de cozinheiros. Todos os seus campos são obrigatórios. Ao clicar no botão adicionar foto é apresentada uma tela para que o usuário consiga navegar pelo diretório de pastas de seu computador e escolha uma única foto para o Cozinheiro.

Figura 58: Tela de cadastro de cozinheiros

Gerenciar Cozinheiros

Nome:

Sobrenome:

Data de Nascimento:

CPF: ". . -"/>

RG: ". . -"/>

Telefone:

Celular:

CEP:

Número:

Rua:

Estado: ▾

Cidade:

Bairro:

Histórico:

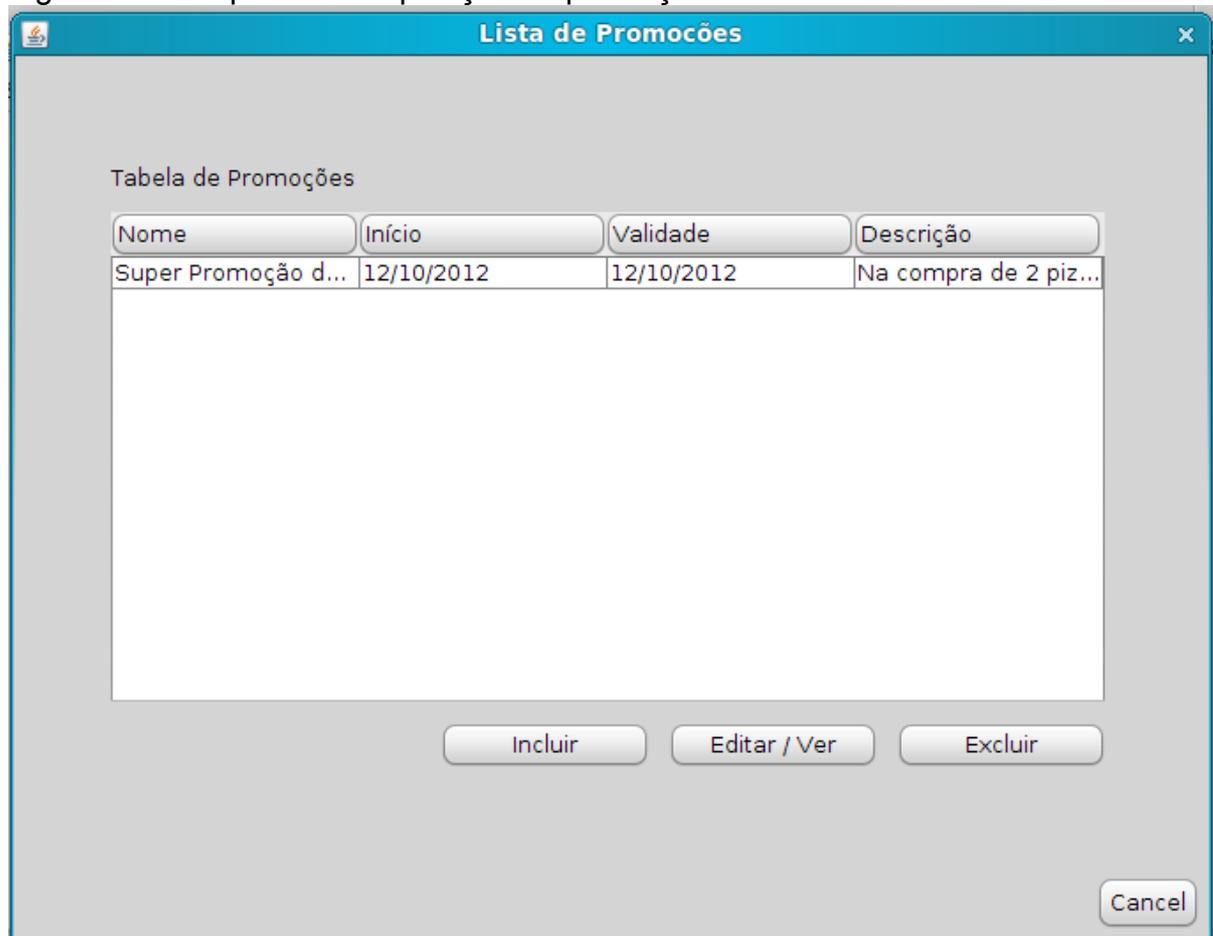
Especialidade:

Foto:

Fonte: O próprio autor (2012)

A figura 59 mostra a tela para manipular promoções. Ela exibe uma tabela com todas as promoções cadastradas no sistema. Por ela é possível abrir uma nova janela para o cadastro de novas promoções, editar e excluir as que já existem.

Figura 59: Tela para a manipulação de promoções



Fonte: O próprio autor (2012)

A figura 60 mostra uma captura da tela de cadastro de promoção. Os campos obrigatórios são: nome, data de início e descrição.

Ao clicar no botão adicionar foto, é apresentada ao usuário uma tela para que ele escolha o caminho em que a foto se encontra em seu computador.

Ao clicar no botão adicionar itens, é apresentado ao usuário uma tela (figura 61) com uma tabela com todos os itens cadastrados no sistema (pratos, bebidas, sobremesas etc) onde é possível fazer a seleção de vários ao mesmo tempo. Ao clicar em OK, o sistema volta para a tela de cadastro da promoção onde serão exibidos os dados que ele já tinha preenchido e uma tabela logo abaixo do texto "Itens da promoção", com todos os itens que ele selecionou para essa promoção.

Figura 60: Tela de cadastro de promoção

The image shows a software window titled "Cadastro de Promoção" with a close button in the top right corner. Inside the window, there is a legend in the top right corner that reads "*Campos Obrigatórios". The form contains the following elements:

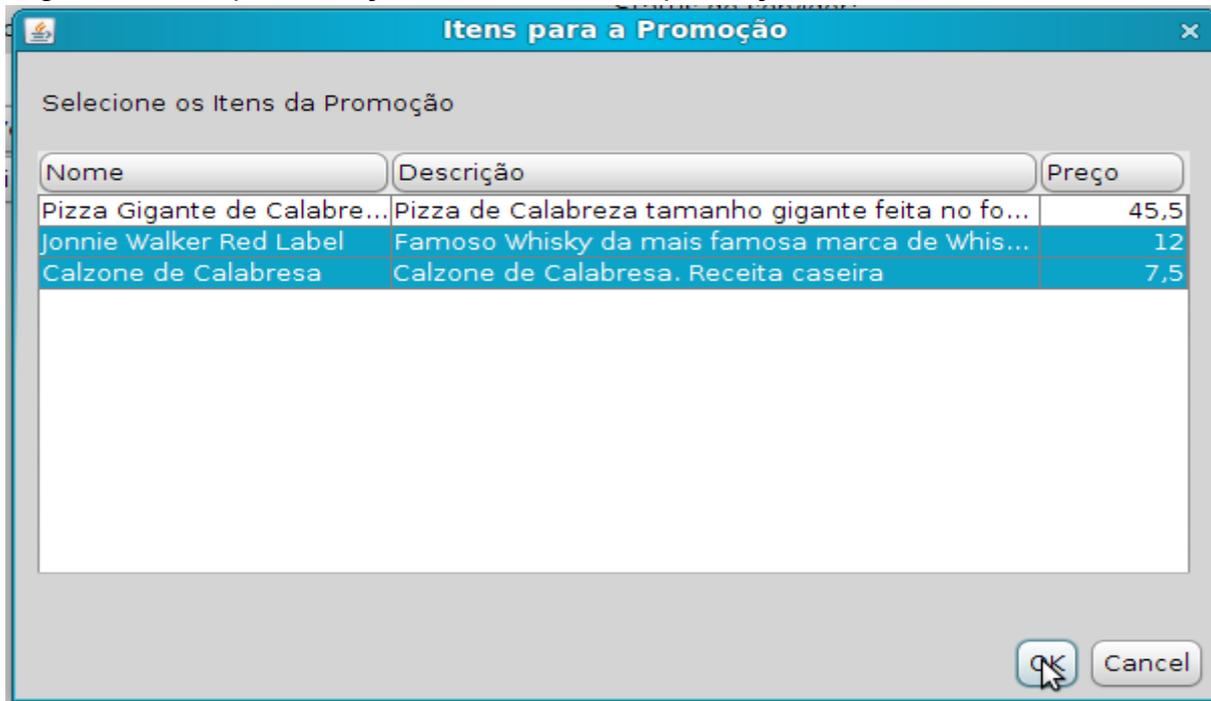
- Nome:***: A text input field with a red border, indicating it is required.
- Data de Início:***: A date input field with slashes as a placeholder, indicating it is required.
- Validade:**: A date input field with slashes as a placeholder.
- Foto:**: A text input field, followed by two buttons: "Adicionar" and "Ver Foto".
- Descrição:***: A large text area for entering the promotion description, indicating it is required.
- Itens da promoção:**: A section with a title and a large empty text area for listing items.
- Below the items section are two buttons: "Adicionar Itens" and "Excluir Selecionados".
- At the bottom right of the window are three buttons: "Limpar", "OK", and "Cancel".

Fonte: O próprio autor (2012)

Na figura 61 é apresentada a tela para seleção de itens que estarão presentes na promoção, que como explicada anteriormente, é exibida ao usuário do sistema ao se clicar no botão "Adicionar itens".

À partir do momento em que todos os campos estão preenchidos, inclusive os itens que participarão desta promoção, ao se clicar no botão "OK", será cadastrada uma nova promoção no sistema.

Figura 61: Tela para seleção de itens de uma promoção



Fonte: O próprio autor (2012)

Como explicado anteriormente, esta tela suporta múltipla seleção, pois uma promoção pode envolver mais de um item. Ao selecionar “OK”, o usuário volta para a tela de cadastro de promoção com o campo “Itens da promoção” preenchido com os itens selecionados nesta tela.

5.3 CONCLUSÃO DO DESENVOLVIMENTO

A implementação começou com o desenvolvimento paralelo dos métodos de comunicação entre o cliente (*tablet*) e servidor, a criação da estrutura de pacotes, os *beans*, DAOs, e a implementação do banco de dados. Com essas etapas prontas foi desenvolvida primeiramente a interface gráfica do servidor e por último a interface gráfica do cliente.

No decorrer da implementação foram surgindo determinados problemas, que foram sanados sem muita dificuldade através de pesquisas na internet, principalmente em fóruns.

6 CONSIDERAÇÕES FINAIS

Este trabalho propôs a modelagem e a criação de um protótipo de implementação, baseada na modelagem feita de um sistema de gerenciamento de pedidos em restaurantes usando *tablets*; a criação da documentação necessária para esse sistema e o estudo de aplicações semelhantes já existentes.

Para tal, foram apresentadas pesquisas em teorias sobre a evolução dos computadores até a criação dos dispositivos móveis; os *tablets*, o sistema operacional Android e sua história; como é o gerenciamento de pedidos em restaurantes; sobre o padrão UML e a explicação de alguns dos seus diagramas que foram desenvolvidos neste trabalho; o que é o ICONIX e seus diagramas específicos. Também foram pesquisadas aplicações relacionadas com o mesmo tópico, com o intuito de avaliar suas especificidades e para servirem de inspiração no desenvolvimento de certas funcionalidades do sistema proposto.

A partir da pesquisa em teoria relacionada, em aplicações existentes (estado-da-arte) que serviram de inspiração e, ainda, nos requisitos funcionais e não-funcionais do sistema, foi criada a modelagem para servir como documento norteador no desenvolvimento de um protótipo. A produção desse protótipo vai de encontro à validação do que se propõe na introdução dessa monografia.

Tendo a modelagem sido feita, foi possível partir para o desenvolvimento do protótipo. Para este foi implementada uma aplicação servidora com interface de alimentação do sistema utilizando a linguagem de programação Java, banco de dados MySQL, Ambiente de desenvolvimento (IDE) Eclipse, entre outras ferramentas que são mostradas neste documento e que auxiliaram na implementação. Também foi desenvolvido um aplicativo para dispositivos móveis, no caso para a plataforma Android. Este aplicativo foi desenvolvido com o propósito de servir como o cardápio digital para um restaurante, onde o mesmo apresenta tudo o que o restaurante tem disponível para seus clientes consumirem e, através do próprio cardápio, é possível fazer os pedidos.

A execução das aplicações, de forma integrada, foi testada da seguinte forma: foram primeiro feitas as inclusões dos conteúdos do cardápio através da aplicação servidora e depois, foi feito o teste da “leitura” deste conteúdo, através do aplicativo cliente, ou seja, o aplicativo para o *tablet*; foram efetuadas as simulações

de pedidos pelo cardápio e verificado se os mesmos eram apresentados no servidor. As aplicações se mostraram eficazes para o que foram propostas, por meio de testes qualitativos. Sendo assim, os objetivos do trabalho foram alcançados.

6.1 SUGESTÕES PARA TRABALHOS FUTUROS

Mesmo tendo o protótipo sido desenvolvido e tendo funcionado conforme os requisitos definidos neste trabalho, muito ainda há de se fazer para que tal aplicação possa apresentar uma boa aparência, robustez na execução e que possa ter uma gama de funcionalidades ao nível de se tornar um possível produto vendável, e não apenas um protótipo.

Dessa forma, são apresentadas a seguir algumas sugestões para dar continuidade a esse trabalho:

- Implementar funções de múltiplos idiomas;
- Fazer integração com redes sociais;
- Criação de uma área de perfil de clientes cadastrados como forma de “saber” suas preferências e utilizar tais dados para fornecer um melhor serviço a este ou efetuar tarefas de estatísticas para a melhora dos produtos do estabelecimento;
- Criação de uma nova versão do sistema, com as sugestões acima implementadas, como forma de validá-lo em um ambiente real, estabelecimento e clientes.

REFERÊNCIAS

ABLESON, Frank. **Introdução ao Desenvolvimento do Android: A Plataforma do Dispositivo de Software Livre**, 12 jun. 2009. Disponível em: <<http://www.ibm.com/developerworks/br/library/os-android-devel/>>. Acesso em: 9 out. 2011.

ABLESON, W. Frank et al. **Android em Ação**. 3. ed. Rio de Janeiro: Elsevier Editora Ltda., 2012. 622 p.

ASSIS, Maria Cristina de. **Metodologia do trabalho científico**. Disponível em: <http://portal.virtual.ufpb.br/biblioteca-virtual/files/pub_1291081139.pdf>. Acesso em: 27 out. 2012.

_____. **Metodologia do trabalho científico**. Disponível em: <http://portal.virtual.ufpb.br/biblioteca-virtual/files/pub_1291081139.pdf>. Acesso em: 27 out. 2012. In: GIL, Antonio Carlos. **Métodos e Técnicas de Pesquisa Social**. 5. ed. São Paulo:Atlas, 2006.

BRAGA, Bruno. **Requisitos não funcionais**. Disponível em: <<http://www.brunobraga.com.br/2009/02/12/requisitos-nao-funcionais/>>. Acesso em: 20 nov. 2012.

COLLINS-SUSSMAN, Ben; FITZPATRICK, Brian W.; PILATO, C. Michael. **Version Control with Subversion: For Subversion 1.7**. 2. ed. Sebastopol, CA, Estados Unidos: O'Reilly Media, 2008. Disponível em: <<http://svnbook.red-bean.com>>. Download gratuito. Licenciado sob Creative Commons. Tradução livre do autor.

CONEXÃO DIGITAL. **Smartphone, tablet ou os dois?**. 27 julho 2012. Disponível em: <<http://www.conexaomercado.com.br/wp/index.php/2012/07/smartphone-tablet-ou-os-dois/>>. Acesso em: 10 ago. 2012.

DEITEL, H. M.; DEITEL, P. J.. **C++ Como Programar**. 3. ed. Porto Alegre: Bookman, 2001. 1098 p. Reimpressão 2004.

_____. **JAVA Como Programar**. 6. ed. São Paulo: Pearson Prentice Hall, 2008. 1110 p. 45.

DEMO, Pedro. **Metodologia do conhecimento científico**. São Paulo: Atlas, 2000.

EASYMENU: Easymenu Home. Disponível em:
<<http://www.easymenu.biz/em/img/personalization.png>>. Acesso em: 22 out. 2011.

E-COMMERCE NEWS. **Como fazer promoções e sorteios nas redes sociais**. 25 out. 2010. Disponível em:
<<http://ecommercenews.com.br/artigos/tutoriais/como-fazer-promoco-es-e-sorteios-na-s-redes-sociais>>. Acesso em: 23 out. 2011.

GREGO, Mauricio. Quer economizar R\$ 1.600,00 por ano?: Sete dicas para poupar energia, papel e tinta em casa e no escritório. **Info Exame**, São Paulo, ago. 2009. Mensal. Disponível em:
<<http://planetasustentavel.abril.com.br/noticia/energia/dicas-economizar-energia-490512.shtml?func=2>>. Acesso em: 15 ago. 2011.

GLOBALCODE. **Academia do Java: AJ2 - Programação orientada a Objeto e UML com Java**. Florianópolis: Globalcode, 2004. Cap. 2, p. 31-40.

GRANDI, Marco Antonio de. **Uma abordagem de identificação e modelagem de regras de negócio e seus relacionamentos transversais**. 2008. 120 f. Dissertação (Mestrado) - Curso de Ciência da Computação, Centro Universitário Eurípedes de Marília, Marília, 2008. Cap. 1. Disponível em:
<http://www.dataplussistemas.com.br/wp-content/uploads/2011/01/Disserta%C3%A7%C3%A3o_Final.pdf>. Acesso em: 29 out. 2012. In: DATE, C. J. What Not How – The Business Rules Approach to Application Development. Boston: Addison-Wesley, 2000.

JUCÁ, Marcelo. **Restaurantes investem em tecnologia para garantir melhor qualidade de atendimento ao seu público - Gastronomia & Negócios**. 18 maio 2008 Disponível em:
<<http://gastronomiaenegocios.com/gn/home/tendencias/imprimir/44/restaurantes-investem-em-tecnologia-para-garantir-melhor-qualidade-de-atendimento-ao-seu-publico>>. Acesso em: 23 out. 2011.

LOBATO, Elvira. Computador pessoal vai ficar restrito a nichos. **Folha**, São Paulo. 17 set. 2011. Disponível em: <<http://folha.com/no976681>>. Acesso em: 23 set. 2011.

LOPES, Airton. A era dos tablets. **Info**, São Paulo, maio 2011. Mensal. Disponível em:
<<http://info.abril.com.br/noticias/tecnologia-pessoal/a-era-dos-tablets-23052011-9.shl>>. Acesso em: 18 ago. 2011.

MATTOS, Érico Casella Tavares de. **Programação de softwares em Java**. São Paulo: Digerati Books, 2007. 160 p.

MENDES, Silas. **O que é SQL?**, 26 jul. 2011. Disponível em:
<<http://silasmendes.com/dba/o-que-e-sql/>>. Acesso em: 16 out. 2012.

MENESES, Jefferson. **10 tentativas, na linha do tempo, a evolução dos Tablets!**, 31 ago. 2011. Disponível em:
<<http://www.webdig.com.br/7309/linha-tempo-evolucao-tablets/>>. Acesso em: 13 set. 2011.

NOVELLI, Marcio. **SGBD - Sistema Gerenciador de Banco de Dados**, 01 set. 2006. Disponível em:
<<http://www.plugmasters.com.br/sys/materias/108/1/SGBD---Sistema-Gerenciador-de-Banco-de-Dados>>. Acesso em: 10 out. 2012.

OBJECT MANAGEMENT GROUP. **OMG Unified Modeling Language (OMG UML): Infrastructure**. 2.4.1 Needham, 2011. 230 p. Tradução livre do autor. Disponível em:
<<http://www.omg.org/spec/UML/2.4.1/Infrastructure>>. Acesso em: 5 dez. 2012.

_____. **OMG Unified Modeling Language (OMG UML): Superstructure**. 2.4.1 Needham, 2011. 748 p. Tradução livre do autor. Disponível em:
<<http://www.omg.org/spec/UML/2.4.1/Superstructure>>. Acesso em: 5 dez. 2012.

PEREIRA, Halex. **O computador pessoal morreu, mas a Apple foi a única empresa a seguir a luz**, 26 jul. 2011. Disponível em:
<<http://macmagazine.com.br/2011/07/26/o-computador-pessoal-morreu-mas-a-apple-foi-a-unica-empresa-a-seguir-a-luz/>>. Acesso em: 23 set. 2011.

PRADA, Rodrigo. **A história dos Notebooks**, 10 jun. 2009, Disponível em:
<<http://www.tecmundo.com.br/2231-a-historia-dos-notebooks.htm>>. Acesso em: 13 set. 2011.

REBELATO, Marcelo Giroto. **Uma análise sobre a estratégia competitiva e operacional dos restaurantes self-service**. 1997. 14 f. Dissertação (Doutorado) - Curso de Engenharia de Produção, Escola de Engenharia de São Carlos - USP, São Carlos, 1997.

SILVA, George et al. **Utilizando ICONIX no desenvolvimento de aplicações Delphi**. Disponível em:
<http://www.redenet.edu.br/publicacoes/arquivos/20080212_080829_INFO-067.pdf>. Acesso em: 20 out. 2012.

SMAAL, Beatriz. **Restaurante curitibano usa iPad como cardápio**, 21 dez. 2010, Disponível em: <<http://www.tecmundo.com.br/7170-restaurante-curitibano-usa-ipad-como-cardapio.htm>>. Acesso em: 16 out. 2011.

SOPHIA: Cardápio digital para Tablets. Disponível em: <<http://www.cardapiosophia.com.br/>>. Acesso em: 27 out. 2011.

SPARX SYSTEMS: Enterprise Architect. Disponível em: <<http://www.sparxsystems.com.au/>>. Acesso em: 10 out. 2012.

WOZNIAK, Steve. **About | woz.org**. Website oficial de Steve Wozniak, cofundador da Apple. Disponível em: <<http://www.woz.org/about>>. Acesso em: 28 out. 2012.