

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

APLICATIVO ANDROID PARA COMPARTILHAMENTO DE
INFORMAÇÕES SOBRE SAÚDE ALIMENTAR

GABRIEL GUSTAVO SPENGLER

BLUMENAU
2016

GABRIEL GUSTAVO SPENGLER

**APLICATIVO ANDROID PARA COMPARTILHAMENTO DE
INFORMAÇÕES SOBRE SAÚDE ALIMENTAR**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Prof. Alexander Roberto Valdameri, Mestre - Orientador

**BLUMENAU
2016**

APLICATIVO ANDROID PARA COMPARTILHAMENTO DE INFORMAÇÕES SOBRE SAÚDE ALIMENTAR

Por

GABRIEL GUSTAVO SPENGLER

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

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

Membro: _____
Prof. Gabriele Jennrich Bambineti, Especialista – FURB

Membro: _____
Prof. Mauro Marcelo Mattos, Doutor – FURB

Blumenau, 07 de dezembro de 2016

Dedico este trabalho a toda minha família e aos entes falecidos que sempre me apoiaram na minha carreira.

AGRADECIMENTOS

Primeiramente a Deus por ter me dado saúde e força permitindo superar as dificuldades.

À minha família, que sempre acreditou em mim e me levantou todas as vezes que precisei, em especial ao meu irmão que me auxiliou e me incentivou a explorar o meu potencial durante todo o curso.

À minha namorada, por ser minha inspiração no tema deste trabalho, pela força e apoio em todos os momentos.

Ao meu orientador, Mestre Alexander Roberto Valdameri por também ter acreditado no meu projeto, por todo o empenho dedicado a colaboração e conclusão deste trabalho.

A todos os professores que pude conhecer no curso de Sistemas de Informação da FURB, que direta e indiretamente me fizeram crescer muito pessoal e profissionalmente.

As professoras do curso de Nutrição da FURB Camila Leandra Bueno de Almeida e Joseane Freygang que se dispuseram a tirar dúvidas e me incentivaram neste projeto.

Pouco conhecimento faz com que as pessoas se sintam orgulhosas. Muito conhecimento, que se sintam humildes. É assim que as espigas sem grãos erguem desdenhosamente a cabeça para o Céu, enquanto que as cheias as baixam para a terra, sua mãe.

Leonardo da Vinci

RESUMO

A estimativa de vida do ser humano tem aumentado cada vez mais e um dos fatores relacionados a este aumento é a melhora na qualidade de vida através da saúde alimentar. Os nutricionistas através do acompanhamento do paciente podem reeduca-lo na alimentação. Esta é uma tarefa complexa que exige além da excelência do profissional, o esforço do paciente para manter o tratamento com base em refeições recomendadas pelo nutricionista, e o nutricionista depende do relato do paciente para saber se a sua rotina de alimentação vem sendo seguida ou não. Devido a este tipo de problema enfrentado no processo de atendimento nutricional, o presente trabalho apresenta uma aplicação para smartphones Android para o compartilhamento de informações sobre saúde alimentar entre o paciente e o profissional de nutrição. Ele permite que o profissional cadastre refeições no seu smartphone e o paciente receba a lista de alimentos a serem consumidos por refeição, podendo então informar seu consumo em cada respectivo horário, tornando a relação e o feedback entre eles mais natural pois utiliza recursos tecnológicos acessíveis e presentes no dia-a-dia. Como inspiração para o trabalho, foi decidido além do desenvolvimento da aplicação, a extensão do trabalho de Kleinschmidt (2013). No desenvolvimento do banco de dados e do servidor de comunicação utilizou-se a tecnologia *webservice*, sendo desenvolvido através da ferramenta Eclipse na linguagem de programação Java em conjunto com o banco de dados MySQL e o servidor Apache TomCat. No desenvolvimento da interface utilizou-se a linguagem de programação Java e a ferramenta Android Studio. Como resultado obteve-se uma aplicação que torna mais concisa a relação e a análise do acompanhamento nutricional para o profissional e o paciente.

Palavras-chave: Plataforma Android. Android Studio. Webservice. Java. JAX-RS. REST. Compartilhamento de informações sobre saúde alimentar. Mercado de smartphones.

ABSTRACT

The human life estimate has been increasing and one of the factors related to this increase is the improvement in the quality of life through food health. Nutritionists through patient follow-up can re-educate them. This is a complex task that requires in addition to the professional excellence, the patient's effort to maintain the meal-based treatment recommended by the nutritionist, and the nutritionist depends on the patient's report on their eating routine, and if it has been followed or not. Due to this type of problem faced in the nutritional care process, the present work presents an application for Android smartphones for the sharing of information on food health between the patient and the nutrition professional. It allows the professional to register meals on his or her smartphone and the patient receives the list of foods to be consumed per meal, and it can then inform their consumption in each respective schedule, making the relationship and feedback between them more natural because it uses accessible technological resources which is present in our day to day routine. As an inspiration for the work, the extension of the work of Kleinschmidt (2013). In the development of the database and the communication webservice technology server was used, being developed through the Eclipse tool in the Java programming language in conjunction with the MySQL database and the Apache TomCat server. In the development of the interface, the Java programming language and the Android Studio tool were used. As a result, an application was obtained that makes the relationship and analysis of the nutritional monitoring for the professional and the patient more concise.

Key-words: Android Platform. Android Studio. Webservice. Java. JAX-RS. REST. Sharing information on food health. Smartphone market.

LISTA DE FIGURAS

Figura 1 - Pesquisa do uso de dispositivos móveis	20
Figura 2 - Tela de cadastro de paciente	22
Figura 3 - Tela de cadastro de alimento	23
Figura 4 - Tela de registro de consulta	23
Figura 5 - Tela de elaboração de cardápio.....	24
Figura 6 – Tela de registro de anamnese (parte 1)	25
Figura 7 – Tela de registro de anamnese (parte 2)	25
Figura 8 - Notificação na tela do usuário paciente	26
Figura 9 - Tela de notificação do usuário paciente.....	26
Figura 10 - Tela de análise do módulo nutricionista	27
Figura 11 - Diagrama de Caso de Uso - Módulo do nutricionista.....	31
Figura 12 - Diagrama de Caso de Uso - Módulo do paciente	32
Figura 13 - Diagrama de classe de entidade do <i>webservice</i>	33
Figura 14 - Diagrama de classe DAO e <i>resource</i> do <i>webservice</i>	36
Figura 15 - Diagrama de classe <i>entity</i> do aplicativo Nutrisys	37
Figura 16 - Diagrama de classe <i>resource</i> do aplicativo Nutrisys	38
Figura 17 - Android Studio Deployment Target	40
Figura 18 - Android Virtual Device Manager	41
Figura 19 - Android Monitor LogCat	42
Figura 20 - Camadas da aplicação <i>webservice</i>	43
Figura 21 - Tela de acesso do sistema	55
Figura 22 – Tela menu do nutricionista.....	56
Figura 23 - Submenu das telas de cadastro do nutricionista	57
Figura 24 – Tela de cadastro de alimentos	58
Figura 25 – Tela de cadastro do paciente	59
Figura 26 – Tela do menu do paciente	60
Figura 27 - Tela de cadastro de anamnese.....	61
Figura 28 - Tela de cadastro de dieta.....	62
Figura 29 - Tela de cadastro de cardápio.....	63
Figura 30 - Tela de cadastro de refeição.....	64
Figura 31 - Notificações do módulo do paciente.....	65

Figura 32 – Tela de cadastro de consumo de refeição do módulo do paciente	66
Figura 33 - Tela de consulta de uma refeição cadastrada.....	67
Figura 34 - Tela de consulta de consumo de refeição específica	68
Figura 35 - Tela menu de acompanhamento	69
Figura 36 - Tela de acompanhamento do peso do paciente por data de anamnese	70
Figura 37 - Tela de consulta de refeição.....	71
Figura 38 - Tela de acompanhamento de refeição específica.....	72
Figura 39 - Tela de acompanhamento geral por data de refeição específica.....	73
Figura 40 - Tela de acompanhamento individual por data de refeição específica	74
Figura 41 - Gráfico de avaliação da populariedade de sistemas móveis com o mesmo propósito	75
Figura 42 - Gráfico de avaliação da usabilidade do sistema	75
Figura 43 - Gráfico de avaliação da contribuição para o acompanhamento nutricional do paciente	76
Figura 44 - Gráfico de avaliação para adotar o aplicativo no dia-a-dia	76
Figura 45 - Questionário opcional de sugestão ou crítica ao aplicativo.....	77
Figura 46 - Diagrama do modelo de entidade e relacionamento.....	83
Figura 47 - Questionário para avaliação da usabilidade do aplicativo Nutrisys	101

LISTA DE QUADROS

Quadro 1 - Requisitos funcionais do módulo do nutricionista	29
Quadro 2 - Requisitos funcionais do módulo do paciente.....	30
Quadro 3 - Requisitos não funcionais	30
Quadro 4 - Classe Anamnese	44
Quadro 5 - AnamneseDao create.....	45
Quadro 6 - AnamneseDao update.....	45
Quadro 7 - AnamneseDao findById.....	46
Quadro 8 - AnamneseDao remove.....	46
Quadro 9 - AnamneseDao findAllByIdPaciente.....	47
Quadro 10 - DBUtils getConnection.....	48
Quadro 11 - Classe AnamneseResource	49
Quadro 12 - Classe AnamneseResource	50
Quadro 13 - Método cadastrarAlimento da classe CadastroAlimentoActivity.....	51
Quadro 14 - Método cadastrarAlimento da classe CadastroAlimentoActivity.....	51
Quadro 15 - Método cadastrarAlimento da classe CadastroAlimentoActivity.....	52
Quadro 16 - Método cadastrarAlimento da classe CadastroAlimentoActivity.....	52
Quadro 17 - Método carregarGrafico da classe AcompanharAnamneseActivity.....	53
Quadro 18 - Método carregarGrafico da classe AcompanharAnamneseActivity.....	54
Quadro 19 - Método carregarGrafico da classe AcompanharAnamneseActivity.....	55
Quadro 20 - Comparativo entre o sistema desenvolvido e os trabalhos correlatos.....	78
Quadro 21 – UC01 - Manter o paciente	84
Quadro 22 - UC02 - Manter os alimentos	85
Quadro 23 - UC06 - Manter as anamneses.....	86
Quadro 24 - UC010 - Manter as dietas.....	88
Quadro 25 - UC011 - Manter os cardápios	89
Quadro 26 - UC012 - Manter as refeições.....	90
Quadro 27 - UC014 - Analisar a quantidade de alimentos consumidos informada pelo paciente	92
Quadro 28 - UC015 - Gerar gráficos com informações das anamneses do paciente	93
Quadro 29 - UC016 - Gerar gráficos com informações das refeições consumidas pelo paciente	94

Quadro 30 - UC022 - Notificar o paciente com as refeições cadastradas nos respectivos horários	97
Quadro 31 - UC023 - Informar a quantidade de alimentos consumidos	99

LISTA DE ABREVIATURAS E SIGLAS

API – Application Programming Interface

ART - Android Runtime

CFN - Conselho Federal de Nutricionistas

CRN - Conselho Regional de Nutricionistas

DAO - Data Access Object

GPS - Global Positioning System

IBGE - Instituto Brasileiro de Geografia e Estatística

IDE - Integrated Development Environment

JAX-RS - Java API for RESTful Web Services

JSON - JavaScript Object Notation

MER - Modelo de Entidade e Relacionamento

NCP - Nutrition Care Process

OMS - Organização Mundial da Saúde

PAN - Processo de Assistência Nutricional

PHP - Personal Home Page

REST - Representation State Transfer

SDK - Software Development Kit

SQL - Structured Query Language

XML - Extensible Markup Language

WHO - World Health Organization

WSDL - Web Services Description Language

SUMÁRIO

1 INTRODUÇÃO.....	14
1.1 OBJETIVOS.....	15
1.2 ESTRUTURA.....	15
2 FUNDAMENTAÇÃO TEÓRICA	17
2.1 NUTRIÇÃO E REEDUCAÇÃO ALIMENTAR	17
2.2 AVALIAÇÃO ANAMNESE DO PACIENTE	18
2.3 MERCADO DE DISPOSITIVOS MÓVEIS.....	19
2.4 TRABALHOS CORRELATOS	21
2.4.1 Estudo de Cruz	21
2.4.2 Aplicação de Zimmermann	22
2.4.3 Protótipo de Kleinschmidt	24
3 DESENVOLVIMENTO DO SOFTWARE	28
3.1 LEVANTAMENTO DE INFORMAÇÕES	28
3.2 ESPECIFICAÇÃO	29
3.2.1 Requisitos do aplicativo	29
3.2.2 Diagrama de Casos de Uso	30
3.2.3 Diagrama de Classes	32
3.3 IMPLEMENTAÇÃO	39
3.3.1 Técnicas e ferramentas utilizadas.....	39
3.3.2 Operacionalidade da implementação	55
3.4 RESULTADOS E DISCUSSÕES.....	74
4 CONCLUSÕES.....	79
4.1 EXTENSÕES	80
APÊNDICE A – DIAGRAMA DO MODELO DE ENTIDADE E RELACIONAMENTO	83
APÊNDICE B – DESCRIÇÃO DOS PRINCIPAIS CASOS DE USO.....	84
APÊNDICE C – QUESTIONÁRIO APLICADO	101

1 INTRODUÇÃO

A história da preocupação com a alimentação é contada por Fisberg et al (2005, p. 1). Segundo os autores, ela vem sendo estudada desde 1785, onde médicos e cientistas observavam que a falta de alimentos decorrentes de guerras, viagens longas e indisponibilidade estava relacionada com doenças. Desde então grandes avanços científicos foram realizados na identificação da relação entre ingestão de energia e nutrientes e o desenvolvimento ou a prevenção de doenças. Com o passar dos anos o número de alimentos industrializados vem aumentando cada vez mais, as pessoas durante seu dia-a-dia tendem a ingerir alimentos prontos, enlatados, práticos e *fast-foods*.

Os alimentos fabricados com diversos conservantes acabam alterando certas propriedades do alimento. Polônio e Peres (2009) afirmam “é pertinente a constante preocupação com as substâncias químicas sintéticas, pois estas podem desencadear o câncer, estando presentes nos alimentos como aditivos ou contaminantes ambientais de uso na agricultura como, por exemplo, os agrotóxicos”. Estas substâncias possuem o objetivo de aumentar a preservação dos alimentos e também criar sabores e aromas artificiais para torná-los produtos mais atraentes.

Caldeira (2007) em seu blog de medicina menciona que entre as causas prováveis de obesidade estão o padrão de sedentarismo devido ao pouco esforço físico no dia-a-dia (automóveis, elevadores, controles remotos) que diminui o gasto de energia na forma de calorias, a industrialização dos alimentos (ricos em carboidratos e gorduras poliinsaturadas) e o hábito da alimentação rápida (*fast-food*) aumentaram a oferta de alimentos extremamente calóricos na dieta. Com isto, o excesso de gordura repercute-se de forma negativa em todos os sistemas do organismo com doenças tais como diabetes, problemas respiratórios, problemas com estrutura óssea, músculos entre outras doenças.

Segundo Varella (2009, p. 7-8), a Organização Mundial da Saúde (OMS) considera do ponto de vista técnico a obesidade como uma doença epidêmica. Ele também afirma que “Cerca de 10% dos brasileiros adultos são obesos, e outros 30% estão acima do peso saudável”. Nesse grupo, as possíveis complicações como pressão alta, diabetes, ataque cardíaco, derrame cerebral, artrite, elevação dos níveis de colesterol e de triglicérides, dificuldades respiratórias e certos tipos de câncer estão mais sujeitas a acontecerem e precisam o quanto antes perder peso.

Devido a isso, a preocupação com hábitos saudáveis e qualidade de vida vem aumentando e com ela a busca por profissionais nutricionistas também. Brunet (2014) informa

que uma pesquisa feita com 31 planos e seguros de saúde associados à FenaSaúde indicam que em 2012 e 2013 a procura por um nutricionista particular aumentou 32,5%, totalizando 478.900 consultas.

O trabalho de Kleinschmidt (2013) teve como objetivo desenvolver um protótipo de aplicativo na plataforma Android com *web service* para compartilhamento de informações nutricionais entre o profissional e o paciente demonstrando que se pode, de uma forma simplificada, possibilitar a um nutricionista cadastrar uma dieta e gerir um cardápio nutricional para adultos baseado em macronutrientes, possibilitando ao paciente realizar o seu acompanhamento e visualizar relatórios através do smartphone do mesmo, além de um maior feedback do paciente para o nutricionista em relação ao tratamento.

Diante deste contexto, este trabalho se propôs a estender o trabalho de Kleinschmidt (2013), com vistas a possibilitar a elaboração de cardápios baseados em micronutrientes, calcular o valor total de calorias e energias dos cardápios consumidos no dia, produzir gráficos com o desempenho do paciente referentes aos cardápios executados, possibilitar ao usuário pré-agendar os horários das notificações dos cardápios, e implementar a tabela de conversões e medidas.

1.1 OBJETIVOS

O objetivo geral do trabalho proposto é a extensão do protótipo Dieta Evoluída proposto por Kleinschmidt (2013).

Os objetivos específicos do trabalho proposto são:

- a) desenvolver um modulo para permitir ao nutricionista elaborar cardápios com base nos micronutrientes e unidades de medidas diversificadas de alimentos no aplicativo;
- b) desenvolver um controle de notificações das refeições executadas pelos pacientes através de cenários;
- c) permitir ao nutricionista um acompanhamento gráfico baseado na análise das práticas alimentares dos pacientes através do cálculo de quilocalorias ingeridas por alimentos em cenários executados no aplicativo.

1.2 ESTRUTURA

A estrutura deste trabalho está organizada em quatro capítulos, iniciando-se pela introdução ao tema principal e apresentando os objetivos.

No segundo capítulo é apresentada a fundamentação teórica baseada nos estudos sobre nutrição e reeducação alimentar, avaliação anamnese do paciente, mercado de dispositivos móveis e trabalhos correlatos.

No terceiro capítulo é descrito o desenvolvimento do sistema proposto, a especificação dos requisitos, as técnicas e ferramentas utilizadas dando sequência na implementação, e encerrando o capítulo, os resultados e discussões.

Por fim, no quarto capítulo tem-se as conclusões deste trabalho e extensões.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo aborda assuntos a serem apresentados nas seções a seguir, tais como nutrição e reeducação alimentar, avaliação anamnese do paciente, dispositivos móveis, além de trabalhos correlatos.

2.1 NUTRIÇÃO E REEDUCAÇÃO ALIMENTAR

O ano de 2007 constituiu um marco na história da nutrição no Brasil, pois foi o ano em que se completou 40 anos de regulamentação da profissão de nutricionista. Atualmente, o País está com quase 50 mil profissionais atuando em diversas áreas. A área de maior concentração é a Nutrição Clínica (41,7%), seguida da área de Alimentação Coletiva, na qual atuam 32,2% dos nutricionistas (BITTENCOURT et al, 2009).

Define-se como nutrição segundo a WHO (World Health Organization):

A nutrição é a ingestão de alimentos, considerado em relação às necessidades nutricionais do corpo. Uma boa nutrição - uma dieta adequada, bem equilibrada combinada com atividade física regular - é uma peça fundamental da boa saúde. A má nutrição pode levar a imunidade reduzida, aumento da susceptibilidade à doença, desenvolvimento físico e mental prejudicada, e redução da produtividade (WHO, 2016, tradução nossa).

Segundo Bon et al (2013, p. 85) as etapas diagnóstico e intervenção nutricional foram padronizadas, o Processo de Assistência Nutricional (PAN) foi definido com base em um programa americano, o Nutrition Care Process (NCP). Este programa é constituído de quatro etapas, cujos objetivos são a resolução de problemas e o cuidado nutricional, que são: a avaliação nutricional, o diagnóstico nutricional, a intervenção nutricional e o monitoramento e reavaliação nutricional. Cada etapa mencionada é composta da coleta de informações do paciente, informações pessoais, medidas, cálculos e acompanhamentos, todos necessários para garantir o funcionamento do processo de assistência nutricional.

De acordo comSizer e Whitney (2003, p. 17-18) uma dieta nutritiva é composta por cinco características:

- a) adequação: os alimentos fornecem o suficiente de cada nutriente essencial, fibra e energia;
- b) equilíbrio: as escolhas não dão ênfase excessiva a um nutriente ou tipo de alimento à custa de outro;
- c) controle calórico: os alimentos fornecem a quantidade de energia de que você necessita para manter o peso apropriado – nem mais nem menos;
- d) moderação: os alimentos não fornecem excesso de gordura, sal, açúcar ou outros

constituintes indesejados;

e) variedade: os alimentos escolhidos diferem dia a dia.

Através do processo de mudança e aprendizagem exercido pelas orientações nutricionais específicas, o indivíduo começa então a conhecer e incorporar a chamada reeducação alimentar. Papini (2007) explica que “a reeducação alimentar pode ter vários objetivos: perda ou ganho de peso, controle de colesterol, da glicose ou até de inúmeras outras doenças e situações que exigem uma mudança na alimentação”. A autora também deixa claro que, mesmo a pessoa estando totalmente saudável e queira manter-se de tal forma, essa reeducação é fundamental, pois a nutrição influencia diretamente no bem-estar e na qualidade de vida das pessoas.

Percebe-se a importância do profissional para mudança nos hábitos alimentares conforme Medeiros (2010, p. 32) ressalta:

Torna-se muito difícil o cumprimento de uma agenda de modificação alimentar realizada pelo próprio paciente. Percebe-se que ela sofre oscilações frequentes, muitas vezes induzidas por pressões sociais e familiares. O estabelecimento de metas com “auditoria” externa por um profissional da área alcança resultados muito mais expressivos.

Destaca-se ainda que todo e qualquer tratamento exercido pelo profissional de nutrição é plausível de leis que garantem a qualidade de seus serviços. Conforme o Conselho Federal de Nutricionistas (CFN, 2008. p. 2) “A responsabilidade técnica exercida pelo Nutricionista é o compromisso profissional e legal na execução de suas atividades, compatível com a formação e os princípios éticos da profissão, visando a qualidade dos serviços prestados à sociedade”.

2.2 AVALIAÇÃO ANAMNESE DO PACIENTE

Do grego “ana”, trazer de novo e “mnesis”, memória, segundo Conde (2012), a anamnese é uma entrevista realizada pelo profissional de saúde ao seu paciente, que tem a intenção de ser um ponto inicial do diagnóstico de uma doença. Em outras palavras, é uma entrevista que busca lembrar todos os fatos que se relacionam com a doença e à pessoa doente. Assim como qualquer outro tipo de entrevista, ela possui formas ou técnicas corretas de serem aplicadas da forma que se seguidas podem aproveitar melhor o tempo de atendimento e produzir um diagnóstico seguro e um tratamento correto.

Durão (2003, p. 12-15) ressalta que esta entrevista constitui uma das mais importantes fontes de informação no processo de avaliação e intervenção nutricional e que a subestimação

dessa importância pode prejudicar todo o processo de cuidados nutricionais assim como prejudica a qualidade do serviço prestado.

De acordo com a autora, as etapas dessa entrevista são:

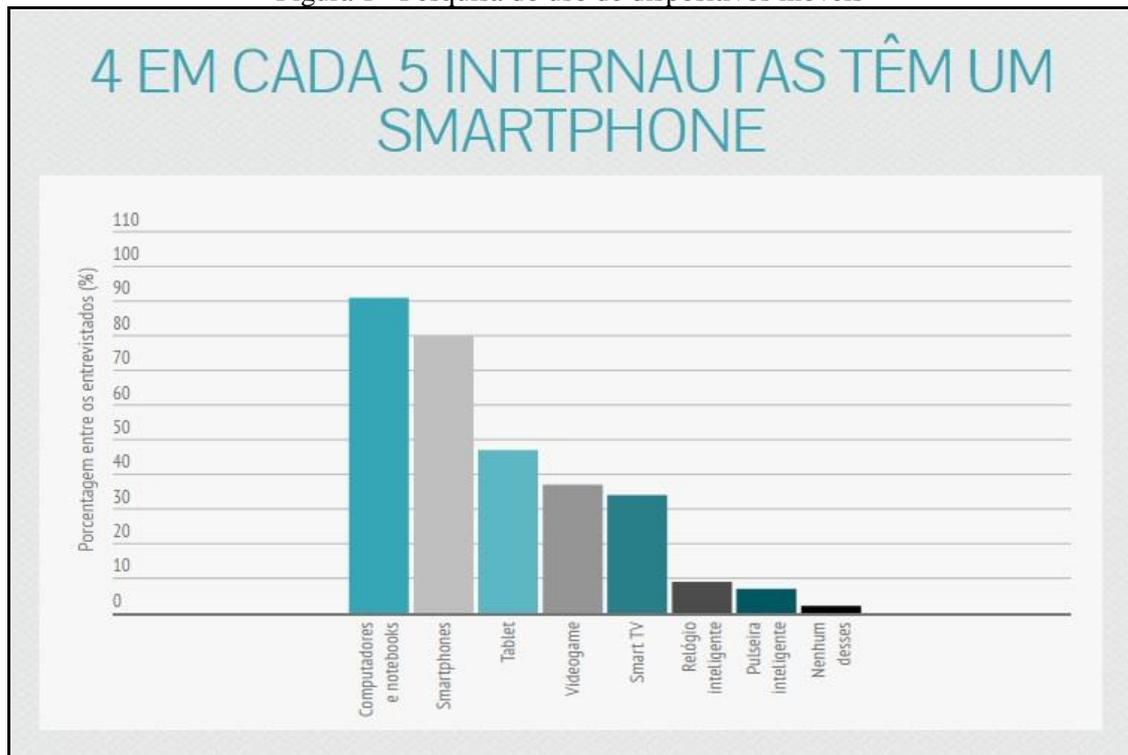
- a) identificação;
- b) história de saúde e medicamentos;
- c) história socioeconômica;
- d) avaliação da ingestão alimentar;
- e) análise de ingestão de alimentos e nutrientes.

As informações corretas recolhidas pela anamnese permitem posteriormente uma avaliação mais adequada dos dados antropométricos, laboratoriais e do exame físico. As etapas do PAN (Processo de Assistência Nutricional) possuem diversos recursos a serem coletados, eles envolvem entrevistas, anotação de medidas, cálculos energéticos, objetivos e acompanhamentos. Em uma consulta, todas essas etapas são feitas em um computador e as recomendações na grande maioria são impressas para os pacientes. As informações ficam somente armazenadas no computador e a mudança de cardápio, o acompanhamento e os relatórios só podem ser feitos através dos retornos dos pacientes, o quais podem informar suas rotinas de alimentação de uma forma imprecisa por não terem a opção de lembrar e anotar suas refeições durante o dia.

2.3 MERCADO DE DISPOSITIVOS MÓVEIS

Segundo Amador (2015), a empresa GlobalWebIndex, mundialmente especializada em estudos relacionados a internet e tecnologia disponibilizou um estudo do crescimento do uso de dispositivos móveis no mundo em 2014. A pesquisa foi feita com 40 mil usuários de internet em 32 países com idades entre 16 e 64 anos. A figura 1 mostra o resultado da pesquisa e comprova que 80% dos entrevistados tem um smartphone e quase metade dos entrevistados possuem tablets.

Figura 1 - Pesquisa do uso de dispositivos móveis



Fonte: Amador (2015).

Essas informações confirmam que houve um aumento considerável sendo que a mesma entidade fez a pesquisa em 2012 e a porcentagem foi de que 60% possuíam smartphones e em 2011 uma pesquisa referente a quantidade de pessoas que possuíam tablets resultou em apenas 10%. No Brasil, Villela (2013) publicou na revista Exame, uma matéria sobre a pesquisa do IBGE que mostra um crescimento de 107% no número de brasileiros que possuem celulares, tornando assim uma grande oportunidade para o desenvolvimento de diversas aplicações.

Em vista deste mercado, Lecheta (2016, p. 17-18) explica que o Android que é o sistema operacional móvel do Google, atualmente é líder mundial do segmento. O sistema Android está disponível para diversas plataformas como smartphones e tablets, TV (Google TV), relógios (Android Wear), óculos (Google Glass) e carros (Android Auto).

Ableson (2012, p. 4) vai além, de um ponto de vista mais técnico e explica que o Android é composto de um sistema operacional baseado em um kernel Linux, uma boa interface de usuário, aplicativos de usuário escritos em Java, bibliotecas de código, *framework* de aplicativo, suporte a multimídia e muito mais. Por ser um software de código aberto, o uso de suas funções permite então a criação de poderosos aplicativos que acessam recursos disponíveis nos dispositivos como telefone, contatos, navegador de internet, Global Positioning System (GPS) entre outros. Toda a disponibilidade do sistema, ferramentas e

mercado torna ele um atrativo para diversos desenvolvedores, desde aplicações acadêmicas até profissionais.

Carbonieri (2015, apud Topol 2015) na tradução do artigo do Dr. Eric Topol, publicado no *The Wall Street Journal*, mostra a mudança na relação médico paciente causada pela revolução da informação e da distribuição em massa de smartphones:

Agora, a qualquer momento do dia ou da noite, você pode procurar e agendar uma consulta online pelo seu *smartphone* (proibido ainda no Brasil) pelo mesmo preço de uma consulta compartilhada (aquela que divide o valor com o empregador). Pode parecer exótico neste momento, mas alguns planos de saúde americanos já preveem que isso será normal em breve. Uma delas diz que pelo menos uma em 6 consultas feitas em 2014 já foram virtuais. Em muitas cidades americanas você já pode solicitar uma consulta médica em seu domicílio, na qual ele pode, além de obviamente consultá-lo, realizar procedimentos como uma sutura simples, o que normalmente necessitaria a uma cara visita à sala de emergência.

No artigo traduzido, Carbonieri (2015, apud Topol 2015) também afirma que muitos aspectos diários da sua vida, como bancos, compras e entretenimento mudaram radicalmente devido aos smartphones. Ainda no artigo traduzido, Carbonieri (2015, apud Topol 2015) afirma “A medicina é a próxima. Com tecnologias digitais inovativas, nuvem de dados e inteligência artificial, o smartphone medicalizado irá mudar todos os aspectos da atenção à saúde. O resultado será que, você – o paciente – tomará o as decisões pela primeira vez”.

2.4 TRABALHOS CORRELATOS

Pode-se citar como trabalhos correlatos os artigos de Cruz (2008), Zimmermann (2011) e Kleinschmidt (2013), os quais serão detalhados a seguir.

2.4.1 Estudo de Cruz

O trabalho de Cruz (2008) estudou como utilizar as tecnologias disponíveis para a criação de uma aplicação para pacientes e nutricionistas. O autor sugeriu a utilização da linguagem Java para criação de *applets* que seriam então utilizados nos dispositivos da época. No seu trabalho o autor realizou o estudo necessário para o entendimento de uma consulta nutricional, onde ele menciona que as informações obtidas para realização da mesma demanda além dos dados pessoais do paciente, o peso atual, a taxa de metabolismo basal e a quantidade total de calorias no organismo. O sistema possui dois módulos: um para o nutricionista utilizar no seu computador para armazenar os dados do paciente, cadastrar alimentos e cardápios e outro no celular do paciente de forma que ambos se comunicam através de um servidor para gerenciamento e análise dos dados. Nesse trabalho o autor fez um levantamento de informações através de estudo e entrevistas com pacientes e profissionais de

nutrição para exemplificar a interação da área de Ciência da Computação junto a Nutrição através de smartphones.

2.4.2 Aplicação de Zimmermann

Zimmermann (2011) propôs a aplicação do conhecimento em tecnologia com o estudo da área de Nutrição e foi além com o desenvolvimento de uma aplicação final. A ideia do seu projeto foi reunir as informações de uma profissional nutricionista que anteriormente trabalhava com planilhas para fazer o acompanhamento dos clientes, e utiliza-las em um sistema em PHP que permitiria conduzir um atendimento nutricional, com cadastro de paciente, alimentos, porções, refeições, consultas, cardápios, e relatórios podendo então acompanhar a evolução de seus pacientes em um único local. Na figura 2 é exibido o cadastro de paciente.

Figura 2 - Tela de cadastro de paciente

The screenshot displays the 'Cadastro de Paciente' (Patient Registration) screen. At the top, it identifies the user as 'Nutricionista Susan Tribess' with CRN 1738. A navigation bar includes 'Home', 'Cadastros', 'Movimentos', 'Relatórios', and 'Sair'. Below this is a table listing existing patients with columns for 'cpf', 'nome', 'idade', and 'email'. The main area contains a form for entering new patient data, divided into 'Cadastro de Paciente' and 'Trabalho' sections. The 'Cadastro de Paciente' section includes fields for CPF (009.515.749-22), Idade (21), Nome (Natasha Zimmermann), Data Nascimento (01/01/2011), Endereço (aasdasd), Bairro, Número (8787), Complemento (ap 2077), Cidade (Blumenau), CEP (89238-788), UF (RN), E-mail (natasha.zimmermann@hotmail.com), Telefone (9989-9999), and Celular (9865-5444). The 'Trabalho' section includes fields for Função (Administradora) and E-mail (natasha.zimmermann@hotmail.com), with a Telefone field (7654-4332). At the bottom, there are buttons for 'Inserir/Alterar', 'Excluir', 'Novo', and 'Cancelar'.

cpf	nome	idade	email
9/0/53.247	Rafael Scolari Maciel	26	rafa.macinha@gmail.com
2/0/9.515	Natasha Zimmermann	21	natasha.zimmermann@hotmail.com
9/0/9.515	Zenaide Zimmermann Zendron	57	zenaide@terra.com.br
0/0/90.909	Augustinho Torres Marques	35	augustinho.marques@bol.com.br
3/0/878.979	Nicassia Kuhnen	29	nicassia.zimmermann@hotmail.com
6/0/988.786	Rodrigo Kobusewski	32	rodrigo.kobu@terra.com.br

Fonte: Zimmermann (2011).

A figura 3 mostra o cadastro de alimento permitindo adicionar os nutrientes específicos e suas respectivas quantidades.

Figura 3 - Tela de cadastro de alimento

The screenshot shows the 'Cadastro de Alimento' interface. At the top, there's a header for 'Nutricionista Susan Tribess CRN 1738' and a navigation menu with 'Home', 'Cadastros', 'Movimentos', 'Relatórios', and 'Sair'. Below this is a table listing foods and their energy values:

Descrição	Valor Energético
Arroz	10
Água	5
Refrigerante	20
Leite	58
Feijão	120

The main form is titled 'Cadastro de Alimento'. It includes fields for 'Porção:' (set to 'Colher'), 'Descrição:' (set to 'Feijão'), and 'Quantidade:' (set to '21'). There's also a 'Valor Energético:' field (set to '120') and a checkbox for 'Alimento com baixo IG'. Below this is a sub-form for 'Informação Nutricional' with tabs for 'Grupo de Alimento' and 'Refeição'. The 'Refeição' tab is active, showing a search bar for 'Nutriente' and a 'Quantidade' field (set to '21'). A table below lists nutrients and their quantities:

Nutriente	Quantidade	Excluir
Ferro	50	Excluir
Vitamina A	21	Excluir

At the bottom of the form are buttons for 'Inserir/Alterar', 'Excluir', 'Novo', and 'Cancelar'.

Fonte: Zimmermann (2011).

A figura 4 mostra o registro de consulta, onde nela são informados os dados da anamnese como medidas, avaliação física e objetivo, para posteriormente utilizar a consulta como referência para elaboração do cardápio.

Figura 4 - Tela de registro de consulta

The screenshot shows the 'Registro de consultas' interface. At the top, there's a header for 'Nutricionista Susan Tribess CRN 1738' and a navigation menu with 'Home', 'Cadastros', 'Movimentos', 'Relatórios', and 'Sair'. Below this is a table listing consultation dates and patient names:

Data	Paciente
29/11/2011	Natasha Zimmermann
29/11/2011	Auqustinho Torres Marques

The main form is titled 'Registro de consultas'. It includes fields for 'Data:' (set to '29/11/2011') and 'Paciente:' (set to 'Natasha Zimmermann'). Below this are various physical assessment fields:

Altura (cm):	2	Peso (gramas):	67	Peso Ósseo (gramas):	24
Gordura (%):	34	Gordura (gramas):	23		
CC (cm):	60	CA (cm):	60		
Flancos (cm):	60	Quadril (cm):	60		
Água (%):	34	Massa magra (%):	35		

Below the physical assessment fields is a field for 'Proposta de KCal da dieta: 1800'. At the bottom is a sub-form for 'Refeição' with a search bar for 'Refeição' and a 'KCal (%)' field. A table below lists meals and their energy values:

Refeição	KCal	Observações	Excluir
Café da Manhã	30		Excluir
Almoço	40		Excluir
Jantar	30		Excluir

At the bottom of the form are buttons for 'Inserir/Alterar', 'Excluir', 'Novo', and 'Cancelar'.

Fonte: Zimmermann (2011).

A figura 5 mostra a tela de elaboração de cardápio, já com o paciente cadastrado, os alimentos cadastrados, a consulta cadastrada, então são adicionados quais alimentos devem ser consumidos e suas respectivas porções.

Figura 5 - Tela de elaboração de cardápio

Nome: Nataasha Zimmermann
 Data: 29/11/2011
 Dieta: 1800 kcal

Refeições

- Café da Manhã 540
- Almoço 720
- Arroz 10 kcal 2 unidade(s)
- Feijão 120 kcal 5 unidade(s)
- Refrigerante 20 kcal 5 unidade(s)
- Jantar 540

Alimento	KCal	Quantidade	Usar?
Água	5	copo 500ml	<input type="checkbox"/>
Arroz	10	colher	<input checked="" type="checkbox"/>
Feijão	120	4 colher	<input checked="" type="checkbox"/>
Leite	58	copo 500ml	<input type="checkbox"/>
Refrigerante	20	copo 500ml	<input checked="" type="checkbox"/>

Fonte: Zimmermann (2011).

2.4.3 Protótipo de Kleinschmidt

Kleinschmidt (2013) desenvolveu um protótipo em Java para a plataforma Android acessando servidor de retaguarda através de *webservice*, para monitoramento das informações que permitisse a interação e acompanhamento de um PAN (Processo de Assistência Nutricional) de forma mais precisa entre nutricionista e paciente.

A solução possui dois módulos: no módulo para o profissional ela permite gerir cardápios nutricionais para adultos com base nos macronutrientes, estes compostos por proteína, lipídeo e carboidrato conforme cadastrado na tabela de alimentos do banco. Já no módulo do paciente, a aplicação emite alertas com as sugestões de cardápio através das notificações da plataforma Android, e também permite ao paciente interagir com as notificações, anotar a quantidade ingerida do alimento sugerido para então gerar um relatório ao nutricionista para feedback do tratamento.

Após gerado o cadastro do paciente, é realizado então o cadastro da anamnese pelo nutricionista. As figuras 6 e 7 exibem a tela de Cadastro de Anamnese do protótipo de Kleinschmidt (2013).

Figura 6 – Tela de registro de anamnese (parte 1)

CadastroAnamneseActivity

Dieta Evoluída

Cadastro de Anamnese

Descricao Anamnese
Anamnese Paciente Teste

Paciente
Teste

Data inicio Vigência: 04/02/2014 Data final Vigência: 08/03/2014

Peso Atual: 53 Peso Ideal: 50 Altura: 1.70

Motivo da Consulta:

Fonte: Kleinschmidt (2013).

Figura 7 – Tela de registro de anamnese (parte 2)

CadastroAnamneseActivity

Peso Atual: 53 Peso Ideal: 50 Altura: 1.70

Motivo da Consulta:
teste

Objetivo do Paciente:
teste

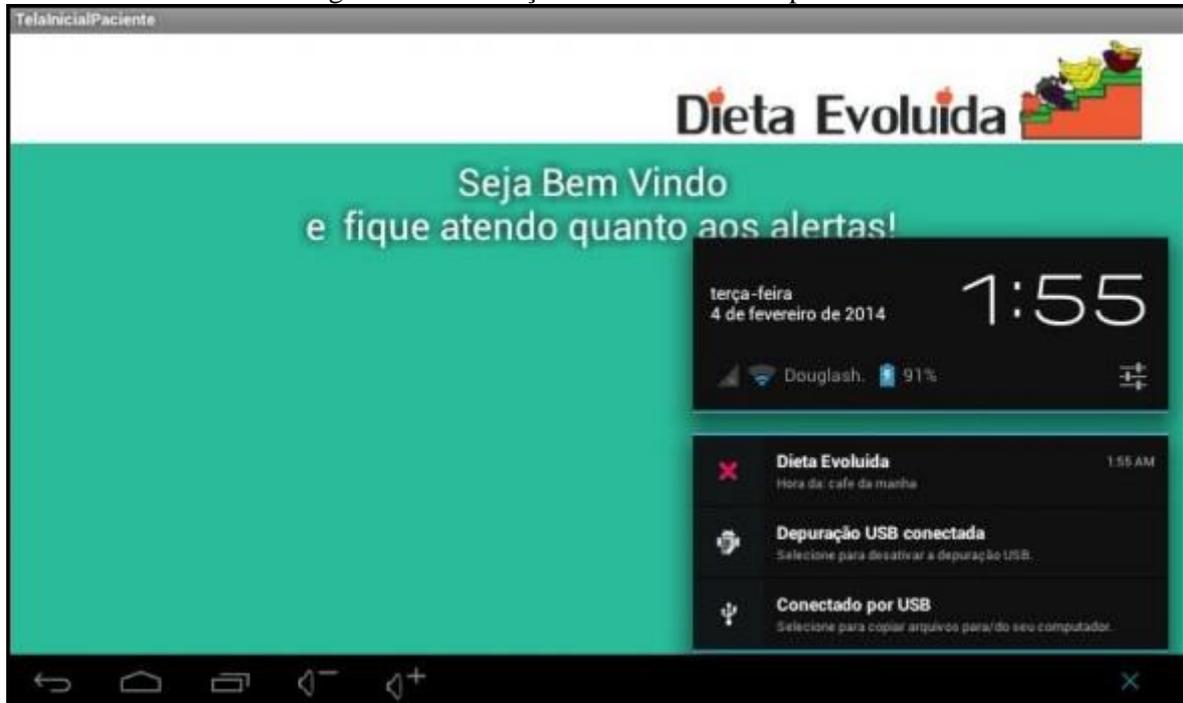
Alterar Cancelar Salvar Enviar

Fonte: Kleinschmidt (2013).

Com as informações da anamnese, o nutricionista pode cadastrar um cardápio para o paciente em questão com base no seu perfil nutricional. Feito isto o paciente, através de recursos inclusos na plataforma Android (como o *AlarmManager*, que permite interagir com o agendamento de tarefas no dispositivo, o *Service* que faz com que a aplicação funcione em segundo plano e as *Notifications* que permite que sejam exibidas mensagens na área de notificação do celular) permitem que o paciente receba durante os horários especificados as notificações da aplicação permitindo então que o paciente, ao interagir com ela, possa relatar ao nutricionista a quantidade ingerida do alimento sugerido. A figura 8 exibe a notificação do

aplicativo Dieta Evoluída na tela do dispositivo do usuário e a figura 9 a tela resultante ao selecionar a notificação para informar a quantidade ingerida do alimento:

Figura 8 - Notificação na tela do usuário paciente



Fonte: Kleinschmidt (2013).

Figura 9 - Tela de notificação do usuário paciente



Fonte: Kleinschmidt (2013).

A figura 10 exibe a tela de análise do módulo nutricionista que contém os horários das refeições com suas identificações, qual a anamnese utilizada e os alimentos consumidos e suas quantidades conforme informados pelas notificações de cada paciente.

Figura 10 - Tela de análise do módulo nutricionista



Fonte: Kleinschmidt (2013).

3 DESENVOLVIMENTO DO SOFTWARE

Esta seção tem como objetivo apresentar as fases para concepção, análise e desenvolvimento do sistema seguindo a seguinte ordem: análise de requisitos funcionais, não funcionais e regras de negócio, a especificação através de diagramas de casos de uso e diagramas de classes das principais funções, seguidos pela implementação e os resultados obtidos.

3.1 LEVANTAMENTO DE INFORMAÇÕES

O protótipo Dieta Evoluída de Kleinschmidt (2013) foi utilizado como inspiração para o desenvolvimento deste projeto, contudo, devido a descontinuação do ambiente de desenvolvimento utilizado naquele trabalho (o Eclipse com *plugin* para desenvolvimento Android) optou-se pela utilização do ambiente Android Studio.

Após definido o ambiente, decidiu-se então seguir as etapas de diagnóstico e intervenção apresentados em Bon et al (2013, p.85) que foram padronizadas e denominadas PAN (Processo de Assistência Nutricional), que foi definido como base em um programa americana, o Nutrition Care Process (NCP). Este programa é constituído de quatro etapas cujos objetivos são a resolução de problemas e o cuidado nutricional que são a avaliação nutricional, o diagnóstico nutricional, a intervenção nutricional e o monitoramento e reavaliação nutricional.

Com a metodologia definida, foi definido que este trabalho utilize dois módulos, cada um para o respectivo perfil de usuário que pode ser nutricionista ou paciente. O módulo do nutricionista permitirá realizar os cadastros de pacientes, anamneses, dietas, cardápios e refeições, transmitindo ao paciente as informações do seu respectivo atendimento e como resultado gerar gráficos com o acompanhamento do paciente. O módulo paciente, terá acesso às informações do seu atendimento para visualização, podendo somente alterar o horário das refeições que foram cadastradas pelo nutricionista, recebendo então as notificações visuais das refeições cadastradas no seu respectivo horário, podendo então informar a quantidade ingerida de cada alimento e também fazer o acompanhamento gráfico do seu atendimento.

3.2 ESPECIFICAÇÃO

Esta seção apresenta os requisitos do aplicativo, diagramas de casos de uso, diagramas de classe, sendo que o MER (Modelo de Entidade e Relacionamento) está descrito no Apêndice A, e a descrição dos principais casos de uso está descrita no Apêndice B.

3.2.1 Requisitos do aplicativo

O quadro 1 apresenta os requisitos funcionais para o módulo do nutricionista e sua rastreabilidade com o(s) caso(s) de uso associado(s).

Quadro 1 - Requisitos funcionais do módulo do nutricionista

Requisitos Funcionais do Módulo do Nutricionista	Caso de Uso
RF01: O sistema deverá permitir ao usuário nutricionista manter o paciente.	UC01
RF02: O sistema deverá permitir ao usuário nutricionista manter os alimentos.	UC02
RF03: O sistema deverá permitir ao usuário nutricionista manter as vitaminas e nutrientes.	UC03
RF04: O sistema deverá permitir ao usuário nutricionista manter os grupos de alimento.	UC04
RF05: O sistema deverá permitir ao usuário nutricionista manter as porções de alimento.	UC05
RF06: O sistema deverá permitir ao usuário nutricionista manter a anamnese.	UC06
RF07: O sistema deverá permitir ao usuário nutricionista manter as doenças.	UC07
RF08: O sistema deverá permitir ao usuário nutricionista manter os objetivos.	UC08
RF09: O sistema deverá permitir ao usuário nutricionista manter os motivos.	UC09
RF10: O sistema deverá permitir ao usuário nutricionista manter as dietas.	UC010
RF11: O sistema deverá permitir ao usuário nutricionista manter os cardápios.	UC011
RF12: O sistema deverá permitir ao usuário nutricionista manter as refeições.	UC012
RF13: O sistema deverá permitir ao usuário nutricionista manter os alimentos a consumir e consumidos.	UC013
RF14: O sistema deverá permitir ao usuário nutricionista analisar a quantidade de alimentos consumidos informada pelo paciente.	UC014
RF15: O sistema deverá permitir ao usuário nutricionista gerar gráficos com informações das anamneses do paciente.	UC015
RF16: O sistema deverá permitir ao usuário nutricionista gerar gráficos com informações das refeições consumidas pelo paciente.	UC016

Fonte: elaborado pelo autor.

O Quadro 2 apresenta os requisitos funcionais do módulo do paciente e sua rastreabilidade com o(s) caso(s) de uso associado(s).

Quadro 2 - Requisitos funcionais do módulo do paciente

Requisitos Funcionais do Módulo do Paciente	Caso de Uso
RF17: O sistema deverá permitir ao usuário paciente visualizar as informações do seu cadastro realizado pelo nutricionista.	UC017
RF18: O sistema deverá permitir ao usuário paciente visualizar as informações da sua anamnese cadastrada pelo nutricionista.	UC018
RF19: O sistema deverá permitir ao usuário paciente visualizar as informações da sua dieta cadastrada pelo nutricionista.	UC019
RF20: O sistema deverá permitir ao usuário paciente visualizar as informações do seu cardápio cadastrado pelo nutricionista.	UC020
RF21: O sistema deverá permitir ao usuário paciente visualizar as informações da sua refeição cadastrada pelo nutricionista e somente alterar o seu horário.	UC021
RF22: O sistema deverá permitir notificar o usuário paciente com as refeições cadastradas nos respectivos horários.	UC022
RF23: O sistema deverá permitir ao usuário paciente informar a quantidade de alimentos consumidos através das notificações e também da tela de consulta de suas refeições.	UC023
RF24: O sistema deverá permitir ao usuário paciente gerar gráficos com informações das suas anamneses cadastradas pelo nutricionista.	UC024
RF25: O sistema deverá permitir ao usuário paciente gerar gráficos com informações das suas refeições consumidas.	UC025

Fonte: elaborado pelo autor.

O quadro 3 apresenta os requisitos não funcionais da aplicação.

Quadro 3 - Requisitos não funcionais

Requisitos Não Funcionais
RNF01: O sistema deverá ser desenvolvido para a plataforma Android.
RNF02: O sistema deverá ser desenvolvido para versão 5 (Lollipop) em diante do Android.
RNF03: O sistema deverá ser implementado na linguagem Java.
RNF04: O sistema deverá ser desenvolvido na ferramenta Android Studio.
RNF05: O sistema deverá utilizar o servidor Tom Cat implementando o <i>Web Service</i> na linguagem Java como forma de comunicação da aplicação.
RNF06: O sistema deverá ter um tempo limite de cinco segundos para conexão com o <i>Web Service</i> .

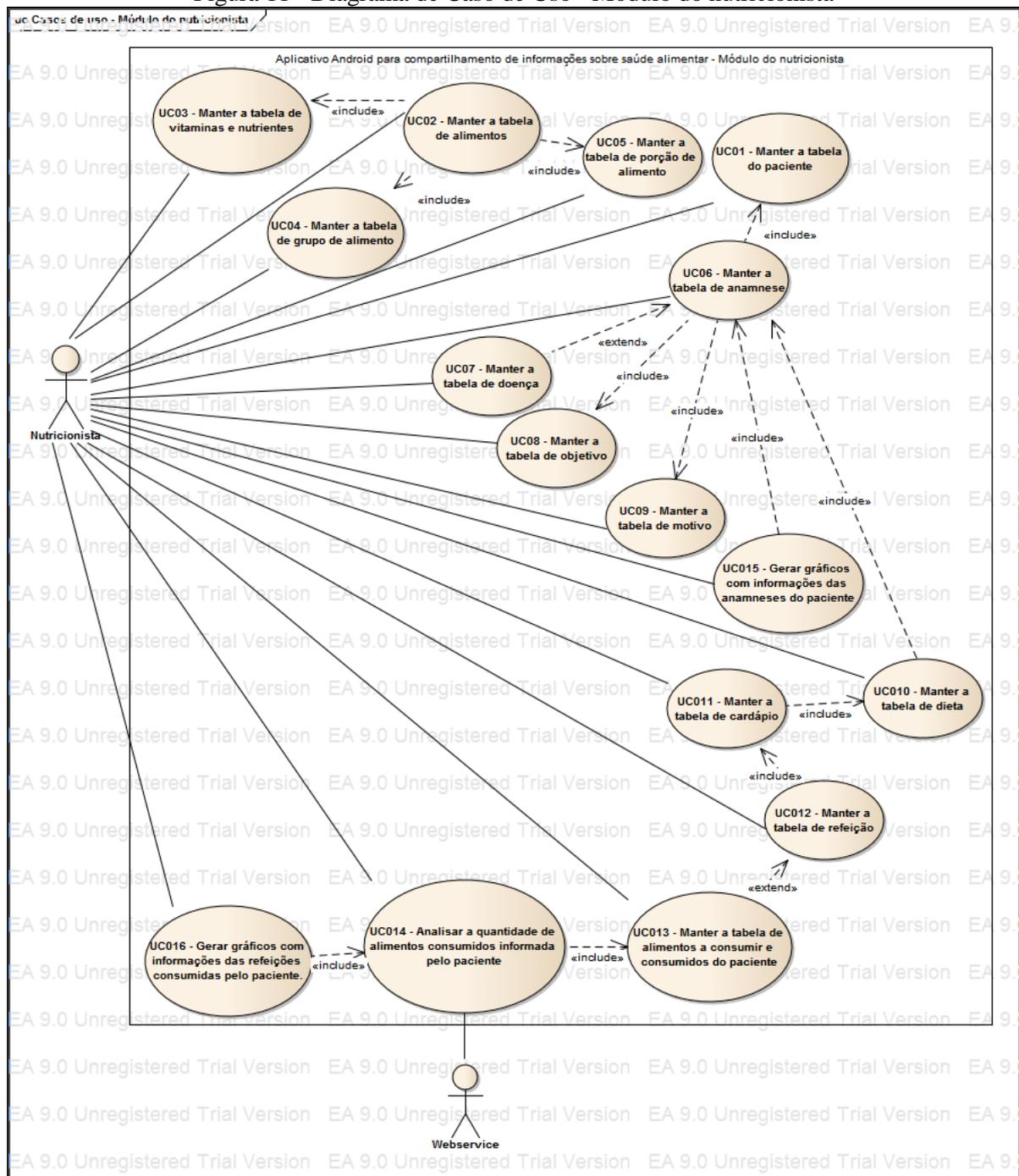
Fonte: elaborado pelo autor.

3.2.2 Diagrama de Casos de Uso

Esta subseção apresenta os diagramas de caso de uso necessários para o entendimento do funcionamento do aplicativo. As descrições dos principais casos de uso encontram-se no Apêndice B.

O aplicativo foi dividido em dois módulos, o módulo do nutricionista e o módulo do paciente. Inicialmente na figura 11 será exibido o diagrama de caso de uso do módulo do nutricionista.

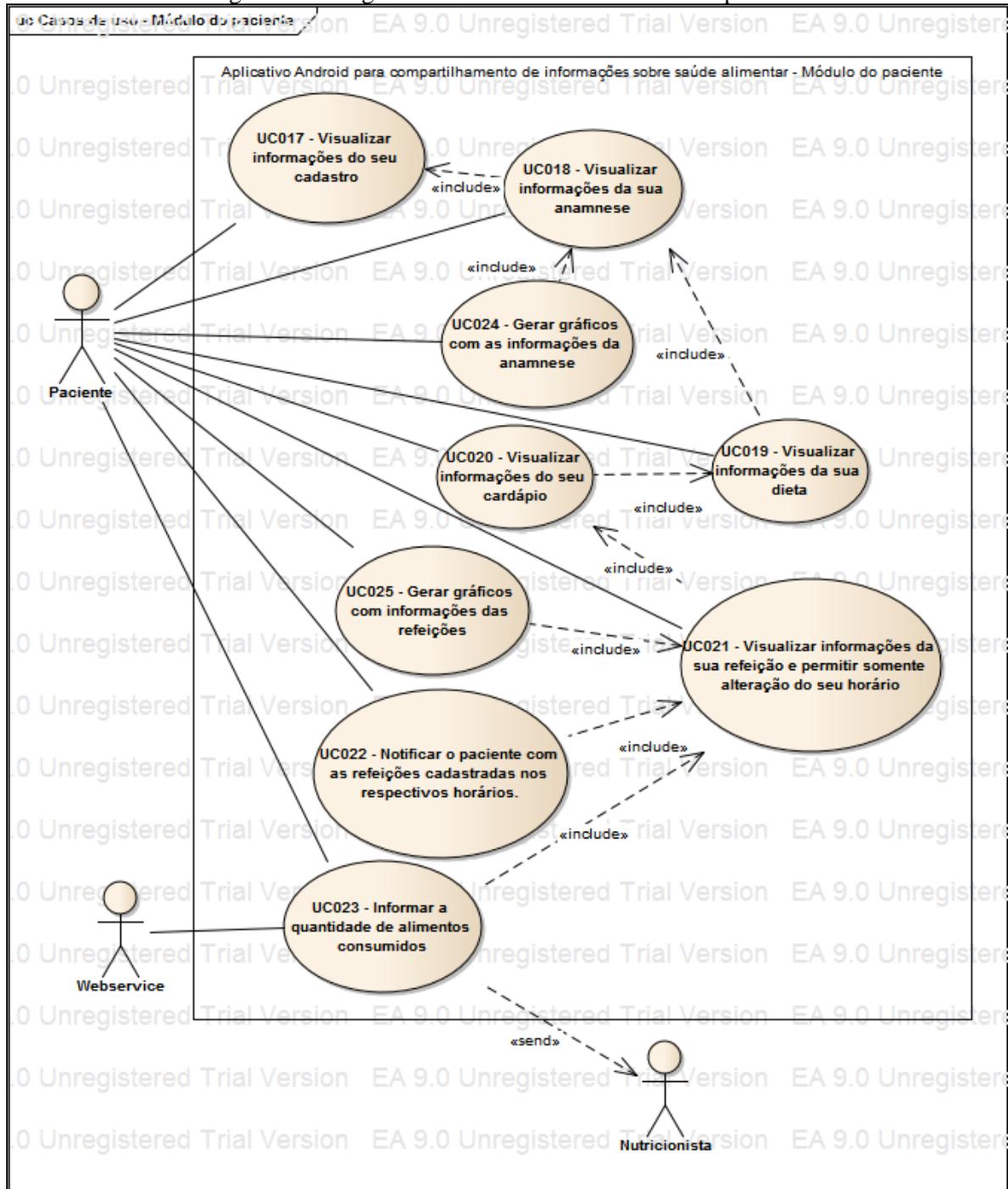
Figura 11 - Diagrama de Caso de Uso - Módulo do nutricionista



Fonte: elaborado pelo autor.

Na figura 12 é demonstrado o diagrama de casos de uso do módulo do paciente.

Figura 12 - Diagrama de Caso de Uso - Módulo do paciente

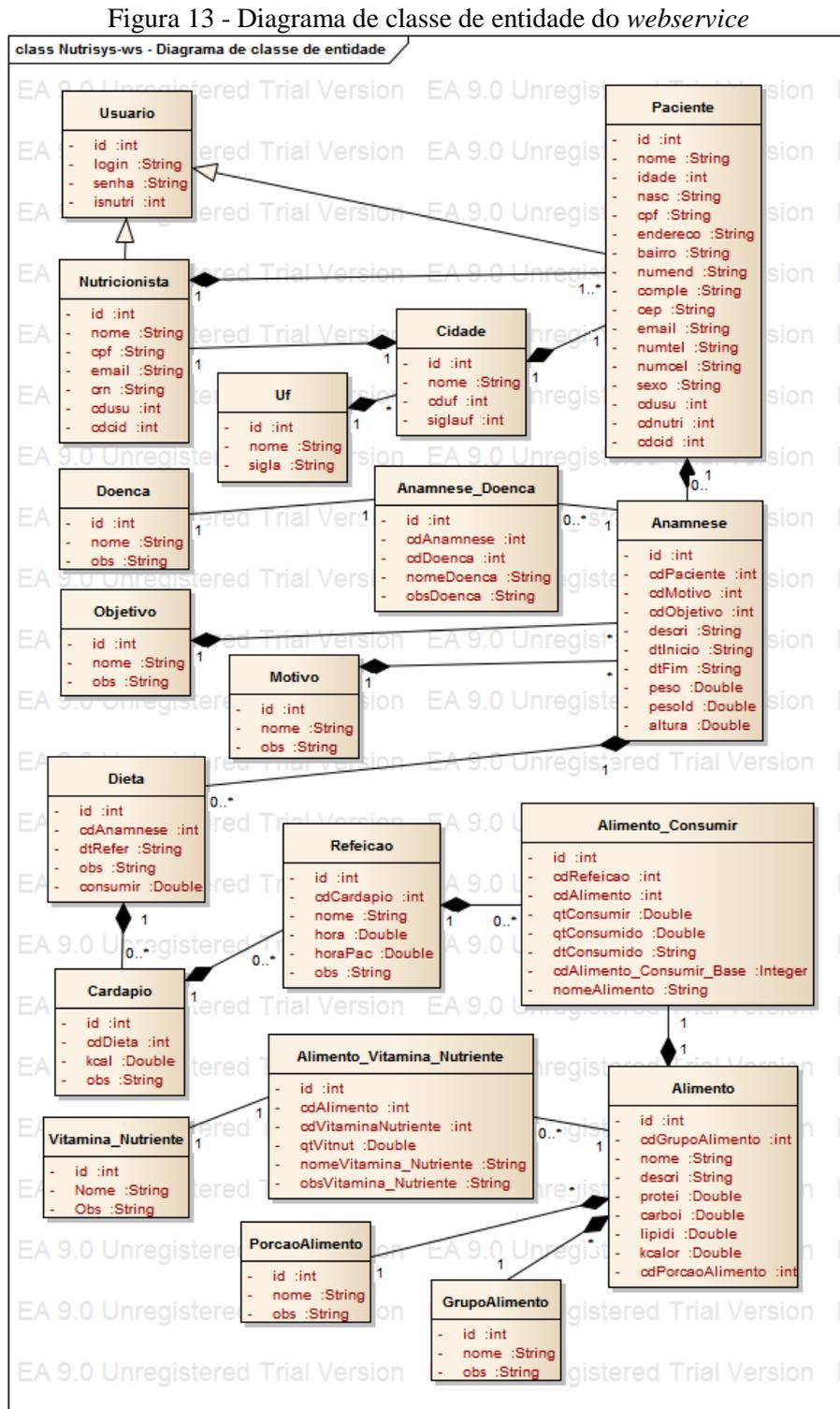


Fonte: elaborado pelo autor.

3.2.3 Diagrama de Classes

Esta subseção exhibe os diagramas de classes das camadas necessárias para o entendimento do funcionamento do aplicativo. Estas são divididas na aplicação do projeto Nutrisys-ws que representa o *webservice* para comunicação com o banco, que possui a camada de entidade, camada DAO (Data Access Object) e camada *resource*. Para aplicação da interface, as camadas pertinentes no projeto Nutrisys são: camada *entity* e a camada

resource. Na figura 13 é apresentado o diagrama de classe da camada de entidade do webservice.



Fonte: elaborado pelo autor.

A seguir, apresenta-se de forma resumida a descrição das classes que compõem o diagrama de classe anterior:

- a) classe Usuario: classe que compõe os dados de acesso ao sistema como *login*,

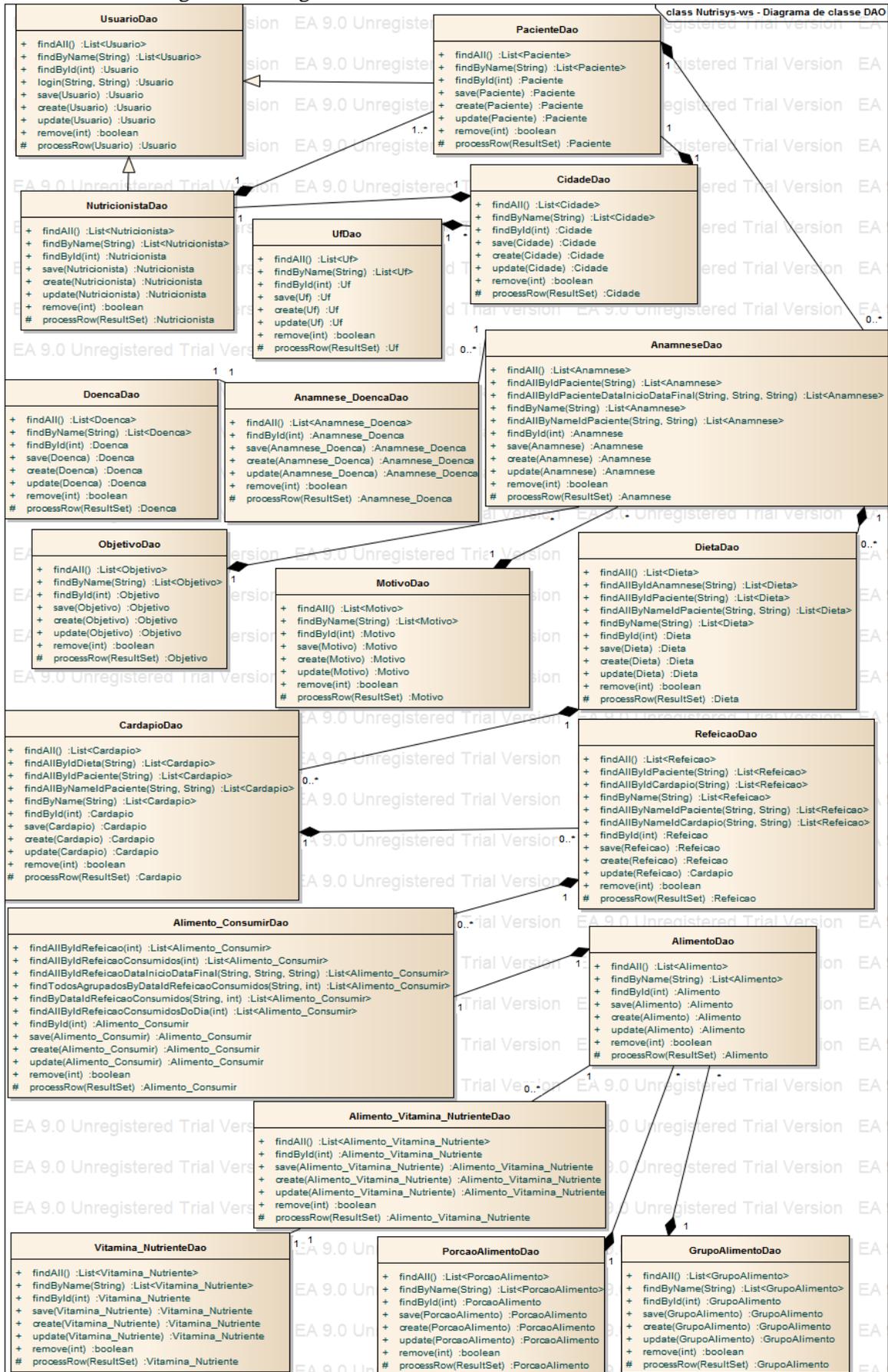
- senha e `isnutri` que representa se é um usuário de nutricionista ou paciente;
- b) classe `Uf`: classe que compõe as informações do objeto UF, como nome e sigla;
 - c) classe `Cidade`: classe que compõe as informações do objeto Cidade, como nome e também é composta pela ligação com a classe `Uf`;
 - d) classe `Nutricionista`: classe que compõe as informações do objeto nutricionista, como nome, CPF, email, código CRN (Conselho Regional de Nutricionistas) e também é composta pela ligação com as classe `Cidade` e `Usuario`;
 - e) classe `Paciente`: classe que compõe as informações credenciais do objeto paciente, que é também composta pela ligação com as classes `Cidade`, `Usuario` e `Nutricionista`;
 - f) classe `Doenca`: classe que compõe as informações do objeto de doença, como nome e observação;
 - g) classe `Objetivo`: classe que compõe as informações do objeto de objetivo, como nome e observação;
 - h) classe `Motivo`: classe que compõe as informações do objeto de motivo, como nome e observação;
 - i) classe `Anamnese`: classe que compõe as informações do objeto de anamnese, como descrição, data de início, data de fim, peso do paciente, peso ideal do paciente, altura do paciente e também é composta pela ligação com as classes `Paciente`, `Motivo` e `Objetivo`;
 - j) classe `Anamnese_Doenca`: classe que compõe as informações de ligação entre as classes `Anamnese` e `Doenca`, utilizada para indicar se existem doenças vinculadas a anamnese do paciente;
 - k) classe `Dieta`: classe que compõe as informações do objeto de dieta, como data de referência, observação, quantidade a consumir e também é composta pela ligação com a classe `Anamnese`;
 - l) classe `Cardapio`: classe que compõe as informações do objeto de cardápio, como quantidade de quilocalorias, observação e também é composta pela ligação com a classe `Dieta`;
 - m) classe `Refeicao`: classe que compõe as informações do objeto de refeição, como o nome da refeição, a hora de execução definida pelo nutricionista, a hora de execução definida pelo paciente, observação e também é composta pela ligação

com a classe `Cardapio`;

- n) classe `Vitamina_Nutriente`: classe que compõe as informações do objeto de vitaminas e nutrientes, como nome e observação;
- o) classe `PorcaoAlimento`: classe que compõe as informações do objeto de porção do alimento, como nome e observação;
- p) classe `GrupoAlimento`: classe que compõe as informações do objeto de grupo do alimento, como nome e observação;
- q) classe `Alimento`: classe que compõe as informações do objeto de alimento, como nome, descrição, os macro nutrientes representados pela quantidade de proteínas, quantidade de carboidratos, quantidade de lipídios, quilocalorias e também é composto pela ligação com as classes `PorcaoAlimento` e `GrupoAlimento`;
- r) classe `Alimento_Vitamina_Nutriente`: classe que compõe as informações de ligação entre a classe `Alimento` e `Vitamina_Nutriente`, utilizada para indicar quais vitaminas e nutrientes compõe um alimento;
- s) classe `Alimento_Consumir`: classe que compõe as informações do objeto de alimentos a consumir, como quantidade a consumir, quantidade consumida, data de consumo, código de alimento a consumir base e também é composta pela ligação com as classes `Refeicao` e `Alimento`.

A camada `DAO` possui as *queries* necessárias para comunicação com o banco. Baseado nos mesmos métodos que a camada `DAO`, a camada *resource* é utilizada para interpretar o acesso ao *webservice* via REST (Representational State Transfer). Na figura 14 é apresentado o diagrama de classe das camadas `DAO` e *resource* do *webservice* `Nutrisys-ws`.

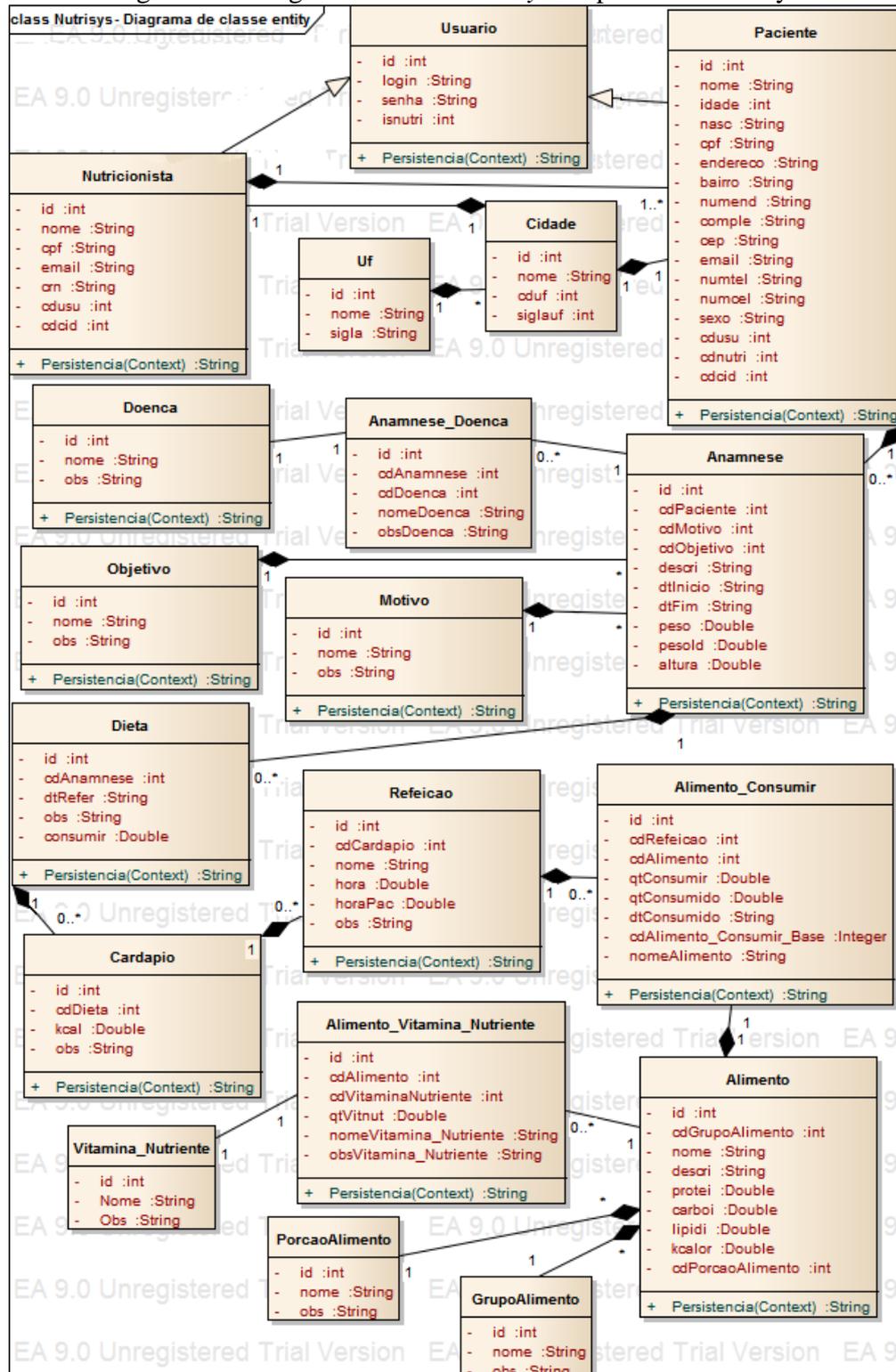
Figura 14 - Diagrama de classe DAO e resource do webservice



Fonte: elaborado pelo autor.

A figura 15 apresenta o diagrama de classe da camada *entity* da aplicação Nutrisys, que apresenta as mesmas funções da camada de entidade do *webservice* mas possui como diferencial um método de persistência em algumas classes para validar as informações antes de tentar inserir ou alterar no banco.

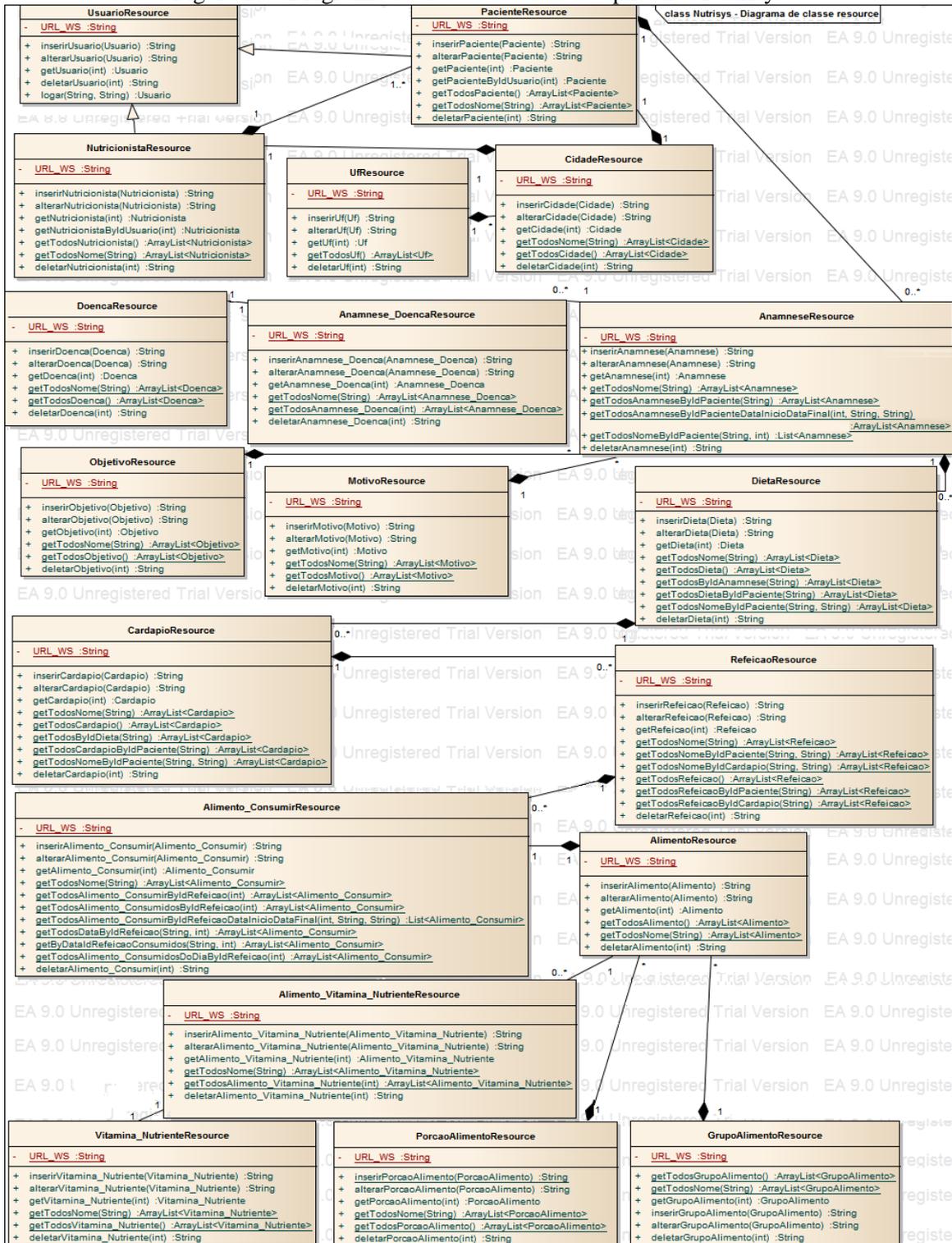
Figura 15 - Diagrama de classe *entity* do aplicativo Nutrisys



Fonte: elaborado pelo autor.

Finalizando, na figura 16 é apresentado o diagrama de classe da camada *resource* da aplicação Nutrisys, que apresenta as mesmas funções da camada de *resource* do *webservice* Nutrisys-ws, mas possui como diferencial as assinaturas dos métodos e uma variável estática que armazena qual endereço o aplicativo deve utilizar para acessar as funções do *webservice* via REST.

Figura 16 - Diagrama de classe *resource* do aplicativo Nutrisys



Fonte: elaborado pelo autor.

3.3 IMPLEMENTAÇÃO

Nesta seção, serão exibidas as técnicas e ferramentas utilizadas e a operacionalidade da implementação.

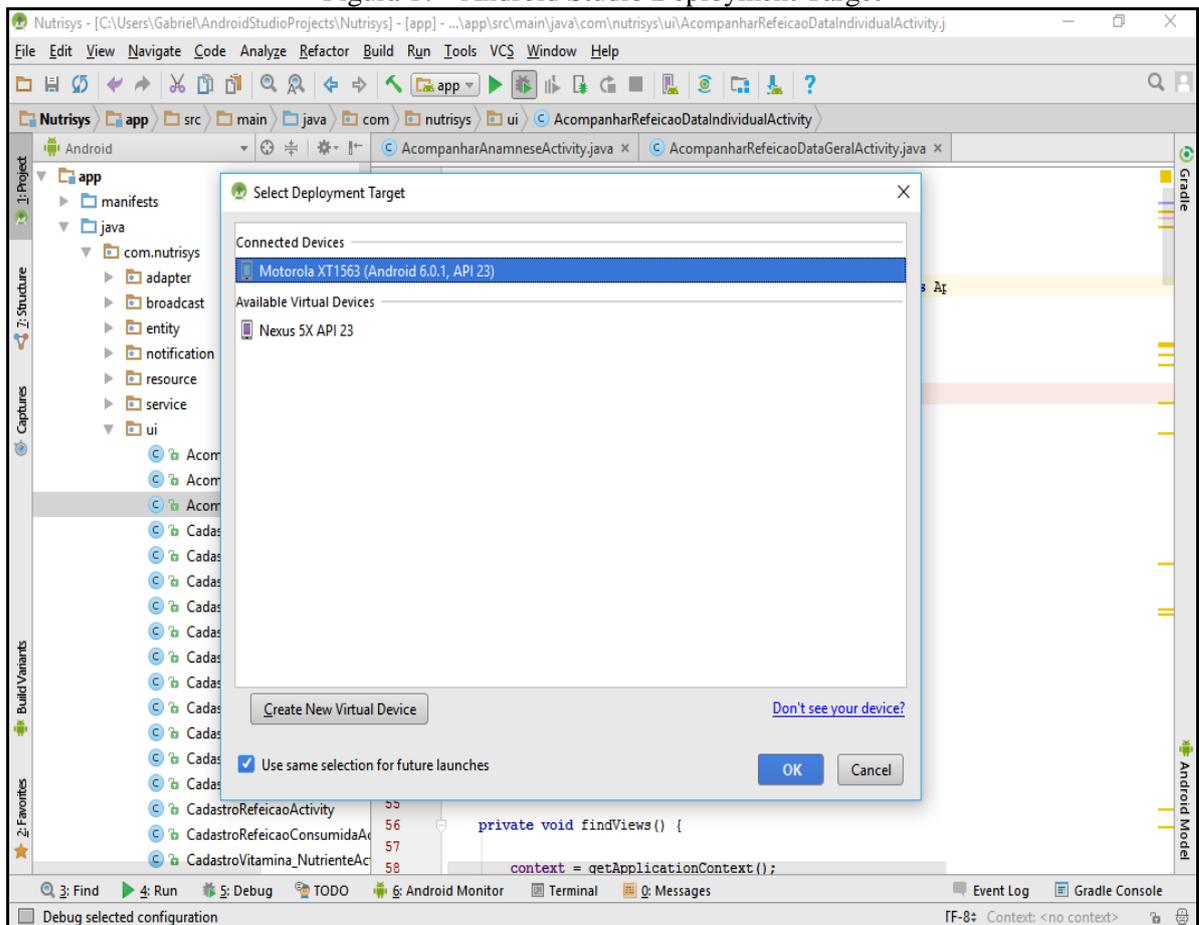
3.3.1 Técnicas e ferramentas utilizadas

Para o desenvolvimento da aplicação foi utilizada a ferramenta Android Studio que é o ambiente oficial criado pela Google para trabalhar com Android. Esta IDE (Integrated Development Environment) foi anunciada em 2013 e é baseado no IntelliJ IDEA da JetBrains. Segundo Lecheta (2016, p. 44-45), ela apresenta alguns diferenciais importantes se comparado a antecessora Eclipse, que até o ano de anúncio do Android Studio, era a ferramenta oficial. Entre os diferenciais da nova IDE estão:

- a) editor visual mais fluído;
- b) sistema de *build* mais moderno baseado em Gradle (gradle.org);
- c) diversas utilidades e facilidades ao desenvolver para Android, sendo muito integrado ao kit de desenvolvimento Software Development Kit (SDK) Android;
- d) *templates* de projetos para smartphones, tablets, relógios etc;
- e) atualização e melhorias frequentes.

A figura 17 apresenta a interface da IDE Android Studio.

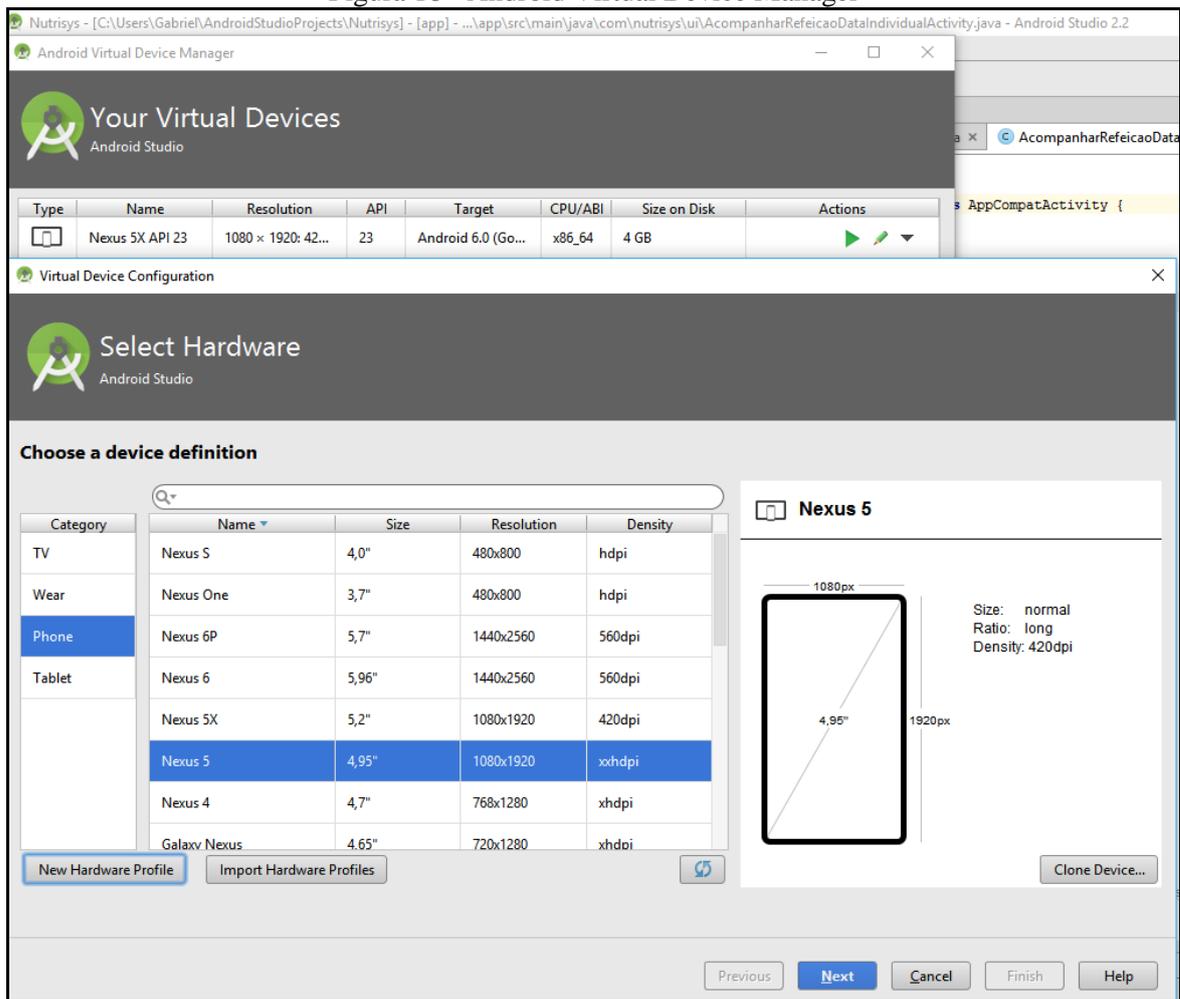
Figura 17 - Android Studio Deployment Target



Fonte: elaborado pelo autor.

Lecheta (2016, p. 42) explica que o kit de desenvolvimento do Android disponibiliza além das ferramentas utilitárias e uma API (Application Programming Interface) completa para a linguagem Java, ele é contemplado também com o Android Virtual Device Manager. Ele é um gerenciador de dispositivos virtuais, estes baseados na máquina virtual ART (Android Runtime) que é responsável pela compilação e otimização do código segundo Lecheta (2016, p. 29), permitindo um emulador ser utilizado para simular os dispositivos e suas configurações conforme apresentado na figura 18.

Figura 18 - Android Virtual Device Manager



Fonte: elaborado pelo autor.

Completando o conjunto de ferramentas a ser utilizado do Android Studio, Lecheta (2016, p. 61) menciona como umas das janelas importantes do programa a conhecer é a de depuração chamada LogCat. Esta janela mostra todos os *logs* do sistema operacional ou apenas do processo ou *tag* que for especificado conforme pode ser visualizado na figura 19.

Figura 19 - Android Monitor LogCat

```

11-06 22:20:44.315 4159-4159/com.nutrisys I/System.out: waiting for debugger to settle...
11-06 22:20:44.517 4159-4159/com.nutrisys I/System.out: waiting for debugger to settle...
11-06 22:20:44.717 4159-4159/com.nutrisys I/System.out: waiting for debugger to settle...
11-06 22:20:44.918 4159-4159/com.nutrisys I/System.out: waiting for debugger to settle...
11-06 22:20:45.119 4159-4159/com.nutrisys I/System.out: debugger has settled (1342)
11-06 22:20:45.123 4159-4159/com.nutrisys W/System: ClassLoader referenced unknown path: /data/app/com.nutrisys-1/lib/x86_64
11-06 22:20:45.132 4159-4159/com.nutrisys I/InstantRun: Instant Run Runtime started. Android package is com.nutrisys, real app
11-06 22:20:47.703 4159-4159/com.nutrisys W/System: ClassLoader referenced unknown path: /data/app/com.nutrisys-1/lib/x86_64
11-06 22:20:48.042 4159-4159/com.nutrisys W/art: Before Android 4.1, method android.graphics.PorterDuffColorFilter android.su
11-06 22:20:48.337 4159-4285/com.nutrisys D/OpenGLESRenderer: Use EGL_SWAP_BEHAVIOR_PRESERVED: true

[ 11-06 22:20:48.343 4159: 4159 D/
HostConnection::get() New Host Connection established 0x7f363e7141

[ 11-06 22:20:48.390 4159: 4285 D/
HostConnection::get() New Host Connection established 0x7f363e7146

11-06 22:20:48.405 4159-4285/com.nutrisys I/OpenGLESRenderer: Initialized EGL, version 1.4
11-06 22:21:10.474 4159-4578/com.nutrisys I/get: Result from post JsonPost : 200 : {"id":1,"login":"Gabriel","isnutri":1,"senh
11-06 22:21:12.534 4159-4165/com.nutrisys W/art: Suspending all threads took: 1.846s
11-06 22:21:12.536 4159-4159/com.nutrisys I/Choreographer: Skipped 111 frames! The application may be doing too much work on
11-06 22:21:12.963 4159-4165/com.nutrisys W/art: Suspending all threads took: 272.707ms
11-06 22:21:13.015 4159-4285/com.nutrisys E/Surface: getSlotFromBufferLocked: unknown buffer: 0x7f36467580a0
11-06 22:21:13.206 4159-4165/com.nutrisys W/art: Suspending all threads took: 6.124ms
11-06 22:21:13.227 4159-4169/com.nutrisys I/art: Background partial concurrent mark sweep GC freed 884(42KB) AllocSpace object
11-06 22:21:14.269 4159-4159/com.nutrisys I/Choreographer: Skipped 52 frames! The application may be doing too much work on :
11-06 22:21:14.344 4159-4285/com.nutrisys E/Surface: getSlotFromBufferLocked: unknown buffer: 0x7f3646757230

```

Fonte: elaborado pelo autor.

Para desenvolver a aplicação, devido as características de processamento, memória, e qualidade de sinal da rede sem fio dos dispositivos atuais, optou-se por utilizar o conceito de *webservice* com a arquitetura REST para comunicação com o banco de dados. Segundo Lecheta (2016, p. 501-502) a outra opção seria a arquitetura Web Services Description Language (WSDL) que é um formato de serviço escrito em Extensible Markup Language (XML), porém, este modelo vem sendo evitado devido ao seu protocolo ser muito extenso e pode causar lentidão, principalmente na conexão 3G.

Para o formato de transmissão dos dados, optou-se por utilizar o JavaScript Object Notation (JSON) baseado na especificação Java API for RESTful Web Services (JAX-RS) para definir a criação do webservice. O JSON conforme explica Smith (2015, p. 66) é derivado dos literais da linguagem de programação JavaScript, é um formato de troca de dados considerado leve e atende o requisito de limite de banda disponível.

Para organização do *webservice*, as classes foram separadas em camadas, sendo elas: a camada de entidade representada pela classe específica de cada pacote, a camada data access object (DAO) e a camada *resource*, conforme mostra a figura 20.

Figura 20 - Camadas da aplicação *webservice*

Fonte: elaborado pelo autor.

A camada de entidade é responsável pelas informações que serão manipuladas durante o processo de entrada e saída de informações do *webservice*. Estas camadas possuem características semelhantes as tabelas do banco conforme a classe *Anamnese* apresentada no quadro 4, elas servem para manipulação dos dados a serem inseridos e alterados. No projeto do aplicativo, esta camada foi adicionada ao pacote *entity*.

Quadro 4 - Classe Anamnese

```
1 package com.nutrisys.anamnese;
2
3
4 * @author Gabriel Gustavo Spengler
5
6
7
8
9
10 public class Anamnese {
11
12     private int id;
13     private int cdPaciente;
14     private int cdMotivo;
15     private int cdObjetivo;
16     private String descri;
17     private String dtInicio;
18     private String dtFim;
19     private Double peso;
20     private Double pesoId; //Peso Ideal
21     private Double altura;
22
23     public int getId() {
24         return id;
25     }
26     public void setId(int id) {
27         this.id = id;
28     }
29     public int getCdPaciente() {
30         return cdPaciente;
31     }
32     public void setCdPaciente(int cdPaciente) {
33         this.cdPaciente = cdPaciente;
34     }
35     public int getCdMotivo() {
36         return cdMotivo;
37     }
38     public void setCdMotivo(int cdMotivo) {
39         this.cdMotivo = cdMotivo;
40     }
41 }
```

Fonte: elaborado pelo autor.

A camada DAO de cada pacote contém as regras necessárias para o acesso ao banco de dados, executando os comandos Structured Query Language (SQL) básicos para manipulação de dados (*insert*, *update*, *delete*, *select*) processando assim o objeto específico de cada tabela do banco. O quadro 5 apresenta o método de inclusão de anamnese da classe `AnamneseDao`, responsável por inserir os dados na entidade `anamnese` do banco de dados a partir do seu respectivo objeto da camada de entidade.

Quadro 5 - AnamneseDao create

```

164 public Anamnese create(Anamnese anamnese) {
165     Connection c = null;
166     PreparedStatement ps = null;
167     try {
168         c = DBUtils.getConnection();
169         ps = c.prepareStatement("INSERT INTO anamnese "
170             + "(cd_paciente, cd_motivo, cd_objetivo,"
171             + " ds_descri, dt_inicio, dt_fim,"
172             + " nm_peso, nm_pesoid, nm_altura)"
173             + " VALUES (?, ?, ?, ?, str_to_date(?, '%d-%m-%Y'), str_to_date(?, '%d-%m-%Y'), ?, ?, ?)",
174             new String[] { "ID" });
175         ps.setInt(1, anamnese.getCdPaciente());
176         ps.setInt(2, anamnese.getCdMotivo());
177         ps.setInt(3, anamnese.getCdObjetivo());
178         ps.setString(4, anamnese.getDescri());
179         ps.setString(5, anamnese.getDtInicio());
180         ps.setString(6, anamnese.getDtFim());
181         ps.setDouble(7, anamnese.getPeso());
182         ps.setDouble(8, anamnese.getPesoId());
183         ps.setDouble(9, anamnese.getAltura());
184         ps.executeUpdate();
185         ResultSet rs = ps.getGeneratedKeys();
186         rs.next();
187         // Update the id in the returned object. This is important as this value must be returned to the client.
188         int id = rs.getInt(1);
189         anamnese.setId(id);
190     } catch (Exception e) {
191         e.printStackTrace();
192         throw new RuntimeException(e);
193     } finally {
194         DBUtils.close(c);
195     }
196     return anamnese;
197 }

```

Fonte: elaborado pelo autor.

O quadro 6 apresenta o método responsável por alterar as informações de um registro de anamnese específico existente na entidade Anamnese do banco de dados.

Quadro 6 - AnamneseDao update

```

199 public Anamnese update(Anamnese anamnese) {
200     Connection c = null;
201     try {
202         c = DBUtils.getConnection();
203         PreparedStatement ps = c.prepareStatement("UPDATE anamnese SET"
204             + " cd_paciente=?, cd_motivo=?, cd_objetivo=?"
205             + " ds_descri=?, dt_inicio=str_to_date(?, '%d-%m-%Y'), dt_fim=str_to_date(?, '%d-%m-%Y'),"
206             + " nm_peso=?, nm_pesoid=?, nm_altura=?"
207             + " WHERE id_anamnese=?");
208         ps.setInt(1, anamnese.getCdPaciente());
209         ps.setInt(2, anamnese.getCdMotivo());
210         ps.setInt(3, anamnese.getCdObjetivo());
211         ps.setString(4, anamnese.getDescri());
212         ps.setString(5, anamnese.getDtInicio());
213         ps.setString(6, anamnese.getDtFim());
214         ps.setDouble(7, anamnese.getPeso());
215         ps.setDouble(8, anamnese.getPesoId());
216         ps.setDouble(9, anamnese.getAltura());
217         ps.setInt(10, anamnese.getId());
218         ps.executeUpdate();
219     } catch (SQLException e) {
220         e.printStackTrace();
221         throw new RuntimeException(e);
222     } finally {
223         DBUtils.close(c);
224     }
225     return anamnese;
226 }

```

Fonte: elaborado pelo autor.

O quadro 7 apresenta o método responsável por buscar as informações de um registro de anamnese inserido na entidade Anamnese do banco de dados.

Quadro 7 - AnamneseDao findById

```

138 public Anamnese findById(int id) {
139     String sql = "SELECT * FROM anamnese WHERE id_anamnese = ?";
140     Anamnese anamnese = null;
141     Connection c = null;
142     try {
143         c = DBUtils.getConnection();
144         PreparedStatement ps = c.prepareStatement(sql);
145         ps.setInt(1, id);
146         ResultSet rs = ps.executeQuery();
147         if (rs.next()) {
148             anamnese = processRow(rs);
149         }
150     } catch (Exception e) {
151         e.printStackTrace();
152         throw new RuntimeException(e);
153     } finally {
154         DBUtils.close(c);
155     }
156     return anamnese;
157 }

```

Fonte: elaborado pelo autor.

O quadro 8 apresenta o método responsável por excluir as informações de um registro de anamnese específico na entidade Anamnese do banco de dados.

Quadro 8 - AnamneseDao remove

```

228 public boolean remove(int id) {
229     Connection c = null;
230     try {
231         c = DBUtils.getConnection();
232         PreparedStatement ps = c.prepareStatement("DELETE FROM anamnese WHERE id_anamnese=?");
233         ps.setInt(1, id);
234         int count = ps.executeUpdate();
235         return count == 1;
236     } catch (Exception e) {
237         e.printStackTrace();
238         throw new RuntimeException(e);
239     } finally {
240         DBUtils.close(c);
241     }
242 }

```

Fonte: elaborado pelo autor.

O quadro 9 apresenta o método responsável por buscar e listar todos os registros existentes na entidade Anamnese utilizando como parâmetro um código da tabela Paciente do banco de dados.

Quadro 9 - AnamneseDao findAllByIdPaciente

```
42 public List<Anamnese> findAllByIdPaciente(String idPaciente) {
43     List<Anamnese> list = new ArrayList<Anamnese>();
44     Connection c = null;
45     String sql = "SELECT an.* FROM anamnese as an " +
46                 "WHERE cd_paciente = ? ";
47     try {
48         c = DBUtils.getConnection();
49         PreparedStatement ps = c.prepareStatement(sql);
50         ps.setString(1, idPaciente);
51         ResultSet rs = ps.executeQuery();
52         while (rs.next()) {
53             list.add(processRow(rs));
54         }
55     } catch (SQLException e) {
56         e.printStackTrace();
57         throw new RuntimeException(e);
58     } finally {
59         DBUtils.close(c);
60     }
61     return list;
62 }
```

Fonte: elaborado pelo autor.

Todos estes métodos de citados anteriormente se aplicam as outras classes da camada DAO em seu respectivo pacote no *webservice*. Para o funcionamento destas classes de conexão, foi criado um pacote de utilitários de conexão denominado *utils*, nele é encontrada a classe *DBUtils*, que possui o método *getConnection* responsável por fazer a conexão com o banco de dados conforme apresentado no quadro 10.

Quadro 10 - DBUtils getConnection

```

90 * @author Gabriel Gustavo Spengler
14 public class DBUtils {
15     private String url;
16     private static DBUtils instance;
17
18     private DBUtils()
19     {
20         String driver = null;
21         try {
22             Class.forName("com.mysql.jdbc.Driver");
23             url = "jdbc:mysql://localhost/directory?user=root";
24             ResourceBundle bundle = ResourceBundle.getBundle("nutrisys");
25             driver = bundle.getString("jdbc.driver");
26             Class.forName(driver);
27             url=bundle.getString("jdbc.url");
28         } catch (Exception e) {
29             e.printStackTrace();
30         }
31     }
32
33     public static Connection getConnection() throws SQLException {
34         if (instance == null) {
35             instance = new DBUtils();
36         }
37         try {
38             return DriverManager.getConnection(instance.url);
39         } catch (SQLException e) {
40             throw e;
41         }
42     }
43
44     public static void close(Connection connection)
45     {
46         try {
47             if (connection != null) {
48                 connection.close();
49             }
50         } catch (SQLException e) {
51             e.printStackTrace();
52         }
53     }
54 }

```

Fonte: elaborado pelo autor.

Por fim, para completar a aplicação do *webservice*, foi criada a camada *resource* em cada pacote, o qual é responsável por interpretar requisições através das anotações disponibilizadas pelo JAX-RS. Estas anotações são interpretadas pela biblioteca utilizada e servem para mapear as funcionalidades do *webservice* que possui seus métodos com as mesmas nomenclaturas da camada DAO e também faz a chamada dos mesmos para realizar suas ações.

As anotações resumidamente são divididas da seguinte forma conforme o RESTful Web Services Developer's Guide da Microsystems (2009, p. 15-16):

- a) @Path: indica o caminho url referente a cada classe e método a ser utilizado;

- b) @Consumes: indica qual tipo de mídia é consumida pelo servidor;
- c) @Produces: indica qual tipo de mídia o servidor produz e envia como retorno ao cliente;
- d) @POST: é um método de requisição HTTP, utilizado para criar registros através do servidor neste trabalho;
- e) @PUT: é um método de requisição HTTP, utilizado para alterar registros através do servidor neste trabalho;
- f) @DELETE: é um método de requisição HTTP, utilizado para deletar registros através do servidor, apesar de implementado, decidiu-se não ter esta opção no aplicativo;
- g) @GET: é um método de requisição HTTP, utilizado para buscar registros através do servidor neste trabalho.

O quadro 11 apresenta a classe `AnamneseResource` e seus respectivos métodos para buscar todos os registros, buscar todos os registros de um paciente específico, buscar todos os registros de um paciente específico e dentro de um período de data e o método para buscar uma anamnese pelo seu nome, todos retornando o tipo de mídia JSON.

Quadro 11 - Classe `AnamneseResource`

```

1 package com.nutrisys.anamnese;
2
3 import java.util.List;
13
15 * @author Gabriel Gustavo Spengler
20 @Path("/anamnese")
21 public class AnamneseResource {
22
23     AnamneseDao dao = new AnamneseDao();
24
25     @GET
26     @Path("/buscatodos")
27     @Produces({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
28     public List<Anamnese> findAll() {
29         System.out.println("Busca todas anamneses");
30         return dao.findAll();
31     }
32
33     @GET
34     @Path("/buscatodosidpaciente/{id}")
35     @Produces({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
36     public List<Anamnese> findAllByIdPaciente(@PathParam("id") String idPaciente) {
37         System.out.println("Busca todas anamneses do paciente de id "+idPaciente);
38         return dao.findAllByIdPaciente(idPaciente);
39     }
40
41     @GET
42     @Path("/buscatodosidpaciente/{id}/datainicio/{datainicio}/datafinal/{datafinal}")
43     @Produces({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
44     public List<Anamnese> findAllByIdPacienteDataInicioDataFinal(@PathParam("id") String idPaciente,
45         @PathParam("datainicio") String dataInicio, @PathParam("datafinal") String dataFinal) {
46         System.out.println("Busca todas anamneses do paciente de id "+idPaciente+" de "+dataInicio+" até "+dataFinal);
47         return dao.findAllByIdPacienteDataInicioDataFinal(idPaciente,dataInicio,dataFinal);
48     }
49
50     @GET
51     @Path("/buscanome/{query}")
52     @Produces({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
53     public List<Anamnese> findByName(@PathParam("query") String query) {
54         System.out.println("Busca anamnese de nome: " + query);
55         return dao.findByName(query);
56     }

```

Fonte: elaborado pelo autor.

O quadro 12, apresenta a continuação da classe `AnamneseResource` e respectivamente seus métodos para buscar uma anamnese por nome, de um paciente específico, uma busca de anamnese específica, um método para inserir um registro e outro para alterar um registro, estes consumindo o tipo de mídia JSON, e por último o método para deletar registros.

Quadro 12 - Classe `AnamneseResource`

```

58 @GET
59 @Path("/buscanome/{query}/idpaciente/{id}")
60 @Produces({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
61 public List<Anamnese> findByNameIdPaciente(@PathParam("query") String query, @PathParam("id") String idPaciente) {
62     System.out.println("Busca anamnese de nome: " + query + " do paciente id " + idPaciente);
63     return dao.findByNameIdPaciente(query, idPaciente);
64 }
65
66 @GET
67 @Path("/buscaid/{id}")
68 @Produces({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
69 public Anamnese findById(@PathParam("id") String id) {
70     System.out.println("Busca anamnese de id " + id);
71     return dao.findById(Integer.parseInt(id));
72 }
73
74 @POST
75 @Path("/add")
76 @Consumes({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
77 @Produces({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
78 public Anamnese create(Anamnese anamnese) {
79     System.out.println("Criando anamnese");
80     return dao.create(anamnese);
81 }
82
83 @PUT
84 @Path("/alterar/{id}")
85 @Consumes({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
86 @Produces({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
87 public Anamnese update(Anamnese anamnese) {
88     System.out.println("Atualizando anamnese: " + anamnese.getDescri());
89     dao.update(anamnese);
90     return anamnese;
91 }
92
93 @DELETE
94 @Path("/deletar/{id}")
95 @Produces({ MediaType.APPLICATION_JSON, MediaType.APPLICATION_XML })
96 public void remove(@PathParam("id") int id) {
97     dao.remove(id);
98 }
99
100 }

```

Fonte: elaborado pelo autor.

Concluindo a aplicação *webservice*, são apresentadas algumas das principais funções e suas respectivas classes da aplicação Nutrisys. Inicialmente mostra-se o método `cadastrarAlimento` da classe `CadastroAlimentoActivity`, responsável por controlar as ações da tela de cadastro de alimento. Este método recebe os dados informados em tela, faz a persistência dos dados tenta inserir/alterar o alimento e em seguida percorre a lista de vitaminas e nutrientes, faz a persistência de cada item individual da lista e tenta também inserir/alterar cada item. Conforme o quadro 13, o método recebe como parâmetros os seguintes valores: `idGrupoAlimento`, `idPorcaoAlimento`, `nome`, `descri`, `protei`, `carboi`, `lipidi`, `kcalor` e o *arraylist* dos itens exibidos na tela.

Quadro 13 - Método cadastrarAlimento da classe CadastroAlimentoActivity

```

231 private String cadastrarAlimento(Integer idGrupoAlimento, Integer idPorcaoAlimento, String nome,
232     String descri, String protei, String carboi,
233     String lipidi, String kcalor, ArrayList<ItensViewAlimento_Vitamina_Nutriente> itens) {
234
235     //Adicionar alimento, em seguida, adicionar Alimento_Vitamina_Nutriente.
236     AlimentoResource alimentoResource = new AlimentoResource();
237     Alimento alimento = new Alimento();
238     //Caso o parametro idGrupoAlimento informado seja inválido, atribui zero devido a persistência final.
239     if (String.valueOf(idGrupoAlimento).isEmpty() || idGrupoAlimento == null){
240         alimento.setCdGrupoAlimento(0);
241     }else{
242         alimento.setCdGrupoAlimento(idGrupoAlimento);
243     }
244     //Caso o parametro idPorcao informado seja inválido, atribui zero devido a persistência final.
245     if (String.valueOf(idPorcaoAlimento).isEmpty() || idPorcaoAlimento == null){
246         alimento.setCdPorcaoAlimento(0);
247     }else{
248         alimento.setCdPorcaoAlimento(idPorcaoAlimento);
249     }
250
251     alimento.setNome(nome);
252     alimento.setDescri(descri);
253     //Caso o parametro protei seja inválido, atribui zero devido a persistência final.
254     if (String.valueOf(protei).isEmpty() || protei == null){
255         alimento.setProtei(0.00);
256     }else{
257         alimento.setProtei(Double.parseDouble(protei));
258     }
259     //Caso o parametro carboi seja inválido, atribui zero devido a persistência final.
260     if (String.valueOf(carboi).isEmpty() || carboi == null){
261         alimento.setCarboi(0.00);
262     }else{
263         alimento.setCarboi(Double.parseDouble(carboi));

```

Fonte: elaborado pelo autor.

O quadro 14 apresenta a continuidade do método cadastrarAlimento.

Quadro 14 - Método cadastrarAlimento da classe CadastroAlimentoActivity

```

264 }
265 //Caso o parametro lipidi seja inválido, atribui zero devido a persistência final.
266 if (String.valueOf(lipidi).isEmpty() || lipidi == null){
267     alimento.setLipidi(0.00);
268 }else{
269     alimento.setLipidi(Double.parseDouble(lipidi));
270 }
271 //Caso o parametro kcalor seja inválido, atribui zero devido a persistência final.
272 if (String.valueOf(kcalor).isEmpty() || kcalor == null){
273     alimento.setKcalor(0.00);
274 }else{
275     alimento.setKcalor(Double.parseDouble(kcalor));
276 }
277 //Executa a persistência da classe que retorna se houver algum erro, a mensagem da persistência.
278 persistencia = alimento.Persistencia(context);
279 //Caso a persistência não for vazia, cancela o processo e retorna a mensagem da persistência.
280 if (!persistencia.isEmpty()){
281     return persistencia;
282 }
283 //Caso exista um código de alimento na variável inicial da tela idRetornoAli, atribui ela pois é uma alteração.
284 if (!idRetornoAli.equals("0")){
285     consultar = true;
286     alimento.setId(Integer.parseInt(idRetornoAli));
287 }
288
289 try {
290     //Caso for uma consulta, executa o método para alterar o alimento.
291     if (consultar){
292         idRetornoAli = alimentoResource.alterarAlimento(alimento);
293     }else{
294         //Caso não seja uma consulta, insere o alimento.
295         idRetornoAli = alimentoResource.inserirAlimento(alimento);
296     }

```

Fonte: elaborado pelo autor.

O quadro 15 apresenta a continuidade do método cadastrarAlimento.

Quadro 15 - Método cadastrarAlimento da classe CadastroAlimentoActivity

```

297     } catch (Exception e) {
298         if (e.getMessage().toString().equals("Falha de rede!")){
299             //Caso o retorno do webservice contenha uma mensagem de status diferente de 200 e conter a mensagem
300             //tratada "Falha de rede!" é retornado -1 e exibido a mensagem de falha de rede.
301             idErro = "-1";
302         }else{
303             //Caso o retorno do webservice contenha qualquer outra mensagem diferente de 200 e não tenha a mensagem
304             //tratada "Falha de rede!" é retornado 0 e exibido a mensagem de falha de cadastro.
305             idErro = "0";
306         }
307     }
308     //Caso haja erro, retorna o valor do erro.
309     if (idErro.equals("-1") || idErro.equals("0"))
310         return idErro;
311
312     //Caso hajam vitaminas ou nutrientes inseridos na lista do alimento, ela é percorrida até seu último item.
313     if (!itens.isEmpty()){
314         for (int i=0; i<itens.size();i++){
315             //Cria o objeto responsável pelos métodos de conexão com o vs.
316             Alimento_Vitamina_NutrienteResource alimento_vitamina_nutrienteResource = new Alimento_Vitamina_NutrienteResource();
317             //Cria o objeto de alimento vitamina nutriente.
318             Alimento_Vitamina_Nutriente alimento_vitamina_nutriente = new Alimento_Vitamina_Nutriente();
319             //Atribui o código do alimento inserido/atualizado.
320             alimento_vitamina_nutriente.setCdAlimento(Integer.parseInt(idRetornoAli));
321             //Cria um objeto chamado item, que permite buscar as informações de cada item da lista.
322             //Atribuindo os valores do item a ser percorrido.
323             AdapterViewAlimento_Vitamina_Nutriente item = adapterListView.getItem(i);
324             //Atribui no objeto a ser inserido pelo Resource o código da Vitamina/Nutriente.
325             alimento_vitamina_nutriente.setCdVitaminaNutriente(item.getCdVitamina_Nutriente());
326             //Atribui o nome da vitamina a ser inserida somente para ser utilizado na persistência se houver erro.
327             alimento_vitamina_nutriente.setNomeVitamina_Nutriente(item.getTvNomeVitamina_Nutriente());
328             //Cria um objeto de View, que permite percorrer todos os itens que compõe cada linha da lista.
329             //Isto é necessário para buscar o valor que representa a quantidade de cada vitamina.

```

Fonte: elaborado pelo autor.

O quadro 16 apresenta a continuidade do método cadastrarAlimento.

Quadro 16 - Método cadastrarAlimento da classe CadastroAlimentoActivity

```

330     View view = adapterListView.getViewByPosition(i,listView);
331
332     if (view != null) {
333         //Procura o campo de nome "etQtVitNut" responsável pela quantidade da vitamina.
334         EditText etQtVitNut = (EditText) view.findViewById(R.id.etQtVitNut);
335         //Se estiver vazio, atribui zero para a persistência tratar.
336         if (etQtVitNut.getText().toString().isEmpty()){
337             alimento_vitamina_nutriente.setQtVitnut(0.00);
338         }else{
339             //Se houver valor, atribui no objeto a ser inserido pelo Resource.
340             alimento_vitamina_nutriente.setQtVitnut(Double.parseDouble(etQtVitNut.getText().toString()));
341         }
342     }
343     //Executa a persistência e interrompe o processo, retornando o erro e qual vitamina apresentou.
344     persistencia = alimento_vitamina_nutriente.Persistencia();
345     if (!persistencia.isEmpty()){
346         return persistencia+" para "+item.getTvNomeVitamina_Nutriente();
347     }
348     //Verifica se já existe uma idAlimento_Vitamina_Nutriente, caso sim, tenta realizar uma alteração.
349     if (item.getIdAlimento_Vitamina_Nutriente() > 0){
350         alimento_vitamina_nutriente.setId(item.getIdAlimento_Vitamina_Nutriente());
351         try {
352             idRetornoAlimento_VitNut = alimento_vitamina_nutrienteResource.alterarAlimento_Vitamina_Nutriente(alimento_vitamina_nutriente);
353         } catch (Exception e) {
354             e.printStackTrace();
355         }
356     }else{
357         //Caso não tenta realizar uma inserção.
358         try {
359             idRetornoAlimento_VitNut = alimento_vitamina_nutrienteResource.inserirAlimento_Vitamina_Nutriente(alimento_vitamina_nutriente);
360         } catch (Exception e) {
361             e.printStackTrace();
362         }

```

Fonte: elaborado pelo autor.

Por fim, para realização do gráfico de acompanhamento de anamnese, que exibe o peso do paciente conforme as consultas realizadas, é apresentado abaixo o método carregarGrafico da classe AcompanharAnamneseActivity, responsável por controlar as regras da tela de acompanhamento de anamnese. Neste método a biblioteca de código aberto GraphView realiza o processo de gerar o gráfico recebendo as informações a

serem exibidas de uma lista de anamneses que são recebidas como parâmetro das consultas no período de filtro da tela.

O quadro 17 apresenta como as informações são tratadas com a biblioteca GraphView, inicialmente removendo qualquer outro gráfico que já possa ter sido gerado e então percorrendo todas as anamneses encontradas ordenadas pela data armazenadas na lista `listAnamnese`. Conforme a linha 129 é criado em cada laço de repetição o objeto de `DataPoint` denominado `peso`, este armazena a posição do eixo X, que é o *index* da lista, e Y recebe o peso cadastrado na anamnese percorrida, que respectivamente são utilizados como parâmetro em seu construtor.

Em seguida o objeto `peso` é adicionado em um *array* de `DataPoint` o qual é previamente criado com o mesmo tamanho da quantidade de itens na lista de anamnese com o nome de `dataPointPeso`. Em seguida é utilizado o método `isChecked` do objeto `chkGraphBarras`, que representa o *checkbox* exibido na tela para definir se o gráfico a ser gerado será em linha ou em barras e então é adicionada a legenda do gráfico e o seu título.

Quadro 17 - Método `carregarGráfico` da classe `AcompanharAnamneseActivity`

```

122 private void carregarGráfico(final List<Anamnese> listAnamnese) {
123     //Remove qualquer gráfico já gerado da tela.
124     graph.removeAllSeries();
125     final ConversorData conversorData = new ConversorData();
126     //Cria o array de DataPoints, que são pontos utilizados para o gráfico com o tamanho da lista de anamnese encontrada.
127     final DataPoint[] dataPointPeso = new DataPoint[listAnamnese.size()];
128     //Percorre a lista de anamneses.
129     for (int index = 0; index < listAnamnese.size(); index++) {
130         //Cria um objeto de DataPoint, atribui uma posição no eixo X com o valor de index
131         // e o peso no eixo Y do gráfico.
132         DataPoint peso = new DataPoint(index, listAnamnese.get(index).getPeso());
133         //Atribui o objeto no array de DataPoints na sua respectiva posição.
134         dataPointPeso[index] = peso;
135     }
136     //Caso seja marcada a opção de gerar o gráfico em barras, chama o método que atribui o array,
137     //na criação de um objeto de BarGraphSeries<DataPoint> e adiciona ao objeto da view graph.
138     if (chkGraphBarras.isChecked()) {
139         gerarGráficoDeBarras(dataPointPeso);
140         graph.getLegendRenderer().setVisible(false);
141     } else {
142         //Caso não seja marcada a opção de gerar o gráfico em barras, chama o método que atribui o array,
143         //na criação de um objeto de LineGraphSeries<DataPoint> e adiciona ao objeto da view graph.
144         gerarGráficoDeLinha(dataPointPeso);
145         graph.getLegendRenderer().setVisible(true);
146         graph.getLegendRenderer().setAlign(LegendRenderer.LegendAlign.TOP);
147     }
148
149     // Define o título do gráfico.
150     graph.setTitle(listAnamnese.size() + " anamneses de " + etDataInicio.getText().toString() + " até "
151         + etDataFinal.getText().toString());
152
153     // Define as legendas de X e Y
154     //Obs: Este método é sempre interpretado ao gerar o gráfico, ele permite alterar os valores apresentados

```

Fonte: elaborado pelo autor.

O quadro 18 apresenta a continuação do método `carregarGráfico`. Neste trecho dentro do método `formatLabel` (linha 159) são definidos quais rótulos serão utilizados no eixo X e Y. Neste gráfico, são apresentados respectivamente pela data inicial de cada cadastro de anamnese e pelo peso do paciente. Os parâmetros deste método são o `value` do tipo

double que representa cada valor a ser exibido no gráfico, e o *isValueX* do tipo *boolean* que representa se o valor a ser lido pertence ao eixo X ou Y.

Como os valores armazenados nos parâmetros do objeto *peso* são do tipo *double*, foi definido para que os valores que representam o X, sejam substituídos pela data da anamnese conforme a sua posição no index durante o processamento do gráfico. Na linha 163, o método *isChecked* do objeto *chkPrimeiraEUltima* serve também como um parâmetro exibido na tela que foi criado para limitar a exibição do eixo X para o primeiro registro de anamnese encontrado e o último devido a limitação do tamanho das telas dos smartphones.

Ainda no eixo X, caso não seja utilizado o parâmetro que limita a visualização das datas, também foi adicionado uma lógica para tentar utilizar um espaço menor na tela do celular que é representado a partir da linha 173, caso seja o primeiro registro a ser lido, é exibida a data completa com dia, mês e ano, nos registros seguintes, caso sejam do mesmo ano, é exibido somente o dia e o mês.

Quadro 18 - Método *carregarGrafico* da classe *AcompanharAnamneseActivity*

```

155 //no eixo X e Y.
156 //Foi utilizado também para evitar que o ano apareça em todos os pontos devido ao espaço disponível,
157 //ele só é exibido no primeiro registro e quando a data a ser percorrida troca de ano.
158 graph.getGridLabelRenderer().setLabelFormatter(new DefaultLabelFormatter() {
159     public String formatLabel(double value, boolean isValueX) {
160         // Legendas de X = são substituídas por datas
161         if (isValueX) {
162             // Se for marcada a opção de exibir somente a primeira e última data no gráfico no eixo X.
163             if (chkPrimeiraEUltima.isChecked()) {
164                 //Ao gerar o gráfico, ele percorre este código e faz esta validação.
165                 //Se for o index zero ou último, insere a data da anamnese.
166                 if ((value == 0) || (value == dataPointPeso.length - 1)) {
167                     return conversorData.getDataBanco(listAnamnese.get(((int) value)).getDtInicio());
168                 } else {
169                     return "";
170                 }
171             } else {
172                 //Caso NÃO for marcada a opção de exibir somente a primeira e última data no gráfico no eixo X.
173                 if (value == 0) {
174                     //Se for o primeiro valor do index, salva em qual ano ele se encontra para não ser mais exibido.
175                     anoDoPonto = conversorData.getDataBanco(listAnamnese.get(0).getDtInicio()).substring(6, 10);
176                     //Exibe a data completa incluindo o ano
177                     return conversorData.getDataBanco(listAnamnese.get(0).getDtInicio()).substring(0, 5);
178                 } else {
179                     //Percorre cada index criado pelos dataPoints
180                     for (int index = 0; index < dataPointPeso.length; index++) {
181                         //Como o value é Double, ele verifica sempre se o valor é igual ao index, que representa um inteiro.
182                         if (value == index) {
183                             //Quando a data pertencer ao ano da data inicial da pesquisa, exibirá somente o dia e o mês.
184                             if (conversorData.getDataBanco(listAnamnese.get(index).getDtInicio()).substring(6, 10).equals(anoDoPonto)) {
185                                 return conversorData.getDataBanco(listAnamnese.get(index).getDtInicio()).substring(0, 5);
186                             } else {
187                                 //Quando a data não pertencer ao ano da data inicial da pesquisa, exibirá dia, mês e ano

```

Fonte: elaborado pelo autor.

O quadro 19 apresenta o fim do método *carregarGrafico* que conclui o método *formatLabel* na linha 200 que apresenta como são tratados os rótulos do eixo Y que apenas recebem o valor do peso armazenado da anamnese e o sufixo “Kg”. Por fim é definido o tamanho da visualização inicial do gráfico e habilitada a opção de ele poder ser redimensionado com gestos na tela.

Quadro 19 - Método carregarGrafico da classe AcompanharAnamneseActivity

```

188 // Então irá atualizar o anoDoPonto para exibir somente a dia e o mês nos pontos seguintes.
189 anoDoPonto = conversorData.getDataBanco(listAnamnese.get(index).getDtInicio()).substring(6, 10);
190 return conversorData.getDataBanco(listAnamnese.get(index).getDtInicio());
191
192
193
194
195     return "";
196
197
198 } else {
199     // Legenda de Y = retorna o valor original mais a palavra "Kg"
200     return super.formatLabel(value, isValueX) + "Kg";
201
202 }
203
204 });
205 //Seta escala de visualização para o tamanho
206 graph.getViewport().setMinX(0);
207 graph.getViewport().setMaxX(listAnamnese.size() - 1);
208 graph.getViewport().setScalable(true);

```

Fonte: elaborado pelo autor.

3.3.2 Operacionalidade da implementação

Nesta subseção serão apresentadas as principais telas do fluxo de uso do usuário nutricionista e paciente. Primeiramente é exibida a tela de acesso ao sistema com os respectivos campos de usuário e senha conforme a figura 21.

Figura 21 - Tela de acesso do sistema



Fonte: elaborado pelo autor.

Assim que confirmado o acesso, o usuário é redirecionado para o seu respectivo menu, o menu do nutricionista que apresenta as opções: cadastro de paciente, cadastro de objetivo, cadastro de motivo, cadastro de doença e cadastro de alimentos conforme exibido na figura

22. Estes cadastros são os básicos necessários para dar sequência no fluxo do sistema. O menu do paciente será detalhado mais adiante.

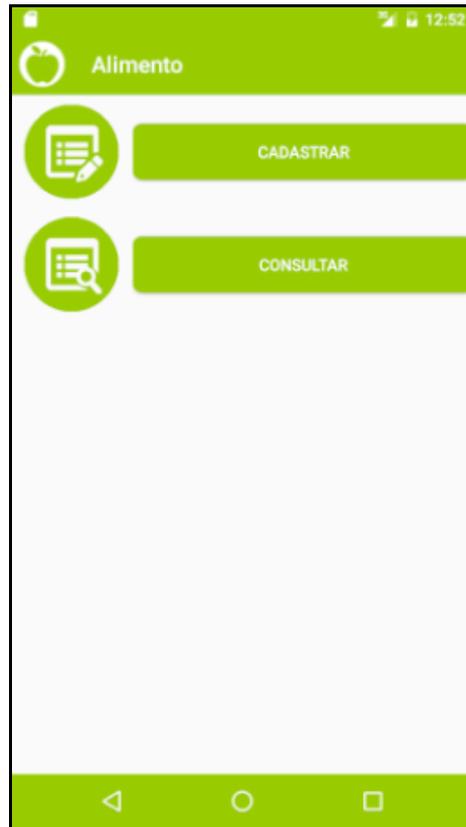
Figura 22 – Tela menu do nutricionista



Fonte: elaborado pelo autor.

A figura 23 apresenta a tela de submenu “Alimentos”, esta tela representa um submenu genérico do nutricionista para indicar qual ação ele deseja realizar. As ações disponíveis são cadastrar e consultar.

Figura 23 - Submenu das telas de cadastro do nutricionista



Fonte: elaborado pelo autor.

A figura 24 apresenta a tela de cadastro de alimentos, onde devem ser informados os seguintes dados conforme o lado A da figura 24: nome do alimento, descrição, grupo de alimento, porção do alimento, os macronutrientes através de quantidade de proteínas, carboidratos e lipídios. No lado B da figura 24, na continuação da tela estão as quilocalorias e por último a lista dos micronutrientes, permitindo adicionar quais vitaminas/nutrientes fazem parte do alimento e a quantidade de cada um deles.

Figura 24 – Tela de cadastro de alimentos

A

Alimentos

Nome: Batata Doce

Descrição: Crua

Grupo: Hortaliças

Porção: Unidade - Média

Proteínas (g): 1.82

Carboidratos (g): 40.04

Lipídios (g): 0.42

BUSCA

BUSCA

B

Kcal: 140.0

Vitamina/Nutriente

Cálcio (mg)	43.4
Ferro (mg)	1.4
Vitamina C (mg)	43.4
Vitamina A (mg)	420.0

Fonte: elaborado pelo autor.

A figura 25 apresenta a tela de cadastro de paciente, que possui as seguintes informações (lado A da figura 25): nome, idade, data de nascimento, cpf, cidade, endereço, bairro e número. No lado B da figura 25, estão na continuação da tela: complemento, CEP, e-mail, telefone, celular, sexo, usuário do paciente, senha do paciente e por último a confirmação da senha para eventuais erros de digitação.

Figura 25 – Tela de cadastro do paciente

Field	Value
Nome	A
Paciente A	
Idade	30
Data de nascimento	11-11-1986
CPF	03074216813
Cidade	Blumenau - SC
Endereço	Rua Antônio da Veiga
Bairro	Itoupava Seca
Número	141
Complemento	Próximo a Furb
Cep	89012900
E-mail	pacientea@gmail.com
Telefone	4712345678
Celular	47123456789
Sexo	Feminino
Usuário	Pacientea
Senha	123
Confirmar senha	

Fonte: elaborado pelo autor.

Ao confirmar o cadastro do paciente, o nutricionista é direcionado ao menu do paciente. Para o paciente, este menu serve para visualizar os cadastros relacionados ao seu atendimento, para o nutricionista ele serve para cadastrar todo o atendimento de cada paciente.

A figura 26 apresenta as opções do menu do paciente que são: o cadastro de anamnese, o cadastro de dieta, o cadastro de cardápio, o cadastro de refeição, a opção de ir para o menu de acompanhamento do atendimento e por último a opção de alterar o cadastro do paciente.

Figura 26 – Tela do menu do paciente



Fonte: elaborado pelo autor.

A figura 27 apresenta o cadastro de anamnese, onde devem ser informados os seguintes dados conforme o lado A da figura 27: descrição, motivo, objetivo, data inicial, data final, peso, peso ideal, altura. No lado B da figura 27, a lista de doenças relacionadas ao atendimento atual do paciente.

Figura 27 - Tela de cadastro de anamnese

The image displays two side-by-side screenshots of a mobile application interface for recording medical history (anamnese). Both screens feature a green header with the title 'Anamnese' and a status bar at the top showing the time as 00:46.

Left Screenshot (Form A):

- Descrição:** A
- Anamnese 1 - Paciente A**
- Motivo:** Cansaço excessivo (with a green 'BUSCA' button)
- Objetivo:** Bem estar (with a green 'BUSCA' button)
- Data inicial:** 01-11-2016
- Data final:** 18-11-2016
- Peso (kg):** 80
- Peso ideal (kg):** 72
- Altura (cm):** 170
- Doença:** (field partially visible)

Right Screenshot (Form B):

- Data final:** 18-11-2016
- Peso (kg):** 80
- Peso ideal (kg):** 72
- Altura (cm):** 170
- Doença:** (field with a list of diseases)
- Buttons:** ALTERAR (green), ADICIONAR (green), CANCELAR (white), SALVAR (green)

The disease list in the right screenshot includes: Ansiedade, Dor de Cabeça, and Gastrite/Úlcera/Refluxo.

Fonte: elaborado pelo autor.

A figura 28 apresenta o cadastro de dieta, onde devem ser informados os seguintes dados: anamnese, data de referência, observação e a quantidade a consumir por semana da dieta.

Figura 28 - Tela de cadastro de dieta

Dieta

Anamnese

Anamnese 1 - Paciente A

BUSCA

Data de referência

17-10-2016

Observação

Perda de Peso - Restrição Alimentar Intensa

Qtd. consumir p/ semana

6.0

CANCELAR

SALVAR

Fonte: elaborado pelo autor.

A figura 29 apresenta o cadastro de cardápio, onde devem ser informados os seguintes dados: dieta, a quantidade de quilocalorias do cardápio e por última o campo de observação.

Figura 29 - Tela de cadastro de cardápio



Fonte: elaborado pelo autor.

A figura 30 apresenta o cadastro de refeição. Este cadastro indica quando e quais alimentos devem ser consumidos em sua respectiva refeição. Ao cadastrar uma refeição, assim que o paciente acessar o sistema, o serviço de notificação é iniciado e então ele cria as notificações e seus alarmes no horário definido pelo nutricionista ou pelo paciente. O horário da refeição é a única informação que o paciente pode alterar.

Como um atendimento nutricional pode mudar conforme o acompanhamento do paciente, as notificações do paciente são baseadas sempre nos cadastros mais recentes de anamnese, dieta e cardápio, evitando que o paciente execute refeições de atendimentos anteriores.

Para o cadastro de refeição devem ser informados os seguintes dados: cardápio, nome da refeição, horário, observação e a lista de alimentos com a respectiva quantidade de consumo de cada alimento.

Figura 30 - Tela de cadastro de refeição

Refeição

Cardápio
Dieta para reeducação BUSCA

Nome
Café da manhã

Horário
08.00

Observação
Podem ser adicionados condimentos na om

Alimentos

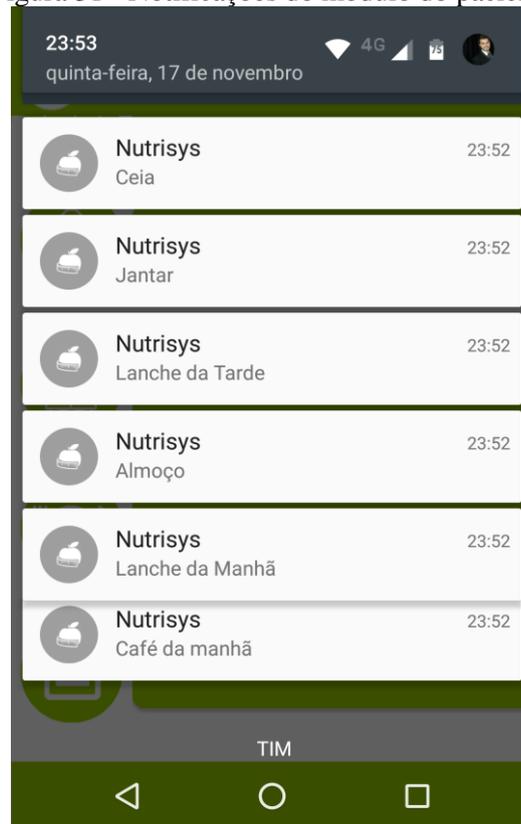
ALTERAR ADICIONAR

Suco de Laranja (Copo Pequeno Cheio)	1
Omelete (Unidade - Pequena)	2

Fonte: elaborado pelo autor.

A figura 31 exibe as notificações do módulo do paciente ao acessar o sistema. As notificações serão exibidas caso o paciente ainda não tenha informado ela no dia atual, caso ele não tenha informado o consumo de alguma refeição do dia e o seu horário tenha sido ultrapassado, também é reemitida a notificação. Além do alerta visual, é emitido o som padrão de notificação configurado no celular e também o alerta de vibração.

Figura 31 - Notificações do módulo do paciente



Fonte: elaborado pelo autor.

A figura 32 exibe a tela do módulo do paciente de cadastro de consumo da refeição que é exibida ao selecionar a notificação. Nesta tela são exibidas a data do consumo da refeição, a lista de alimentos da refeição notificada com a quantidade a consumir definida pelo nutricionista e a quantidade consumida que deve ser informada pelo paciente.

Figura 32 – Tela de cadastro de consumo de refeição do módulo do paciente

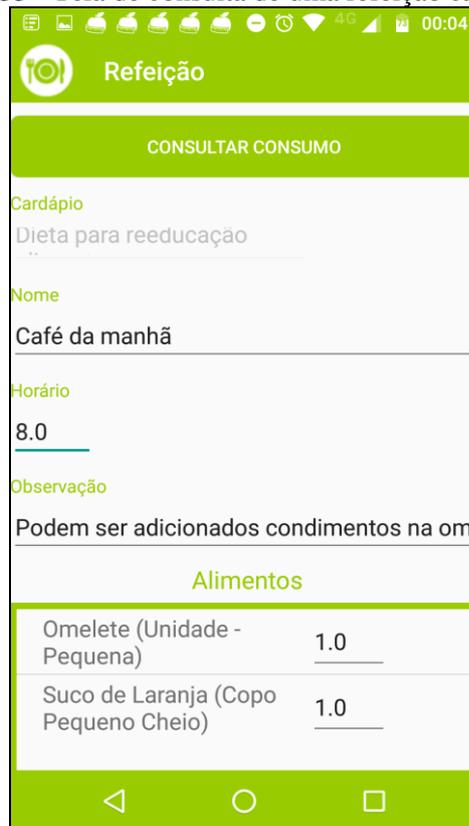
The screenshot displays a mobile application interface for recording meal consumption. The title bar is green and contains a camera icon and the text "Refeição - Café da manhã". Below the title bar, the date "17-11-2016" is displayed under the label "Data". The main content area is titled "Alimentos" and contains a table with two rows of food items. The first row is "Omelete (Unidade - Pequena)" with a quantity of "1.0" and a value of "0.0". The second row is "Suco de Laranja (Copo Pequeno Cheio)" with a quantity of "1.0" and a value of "0.0". At the bottom of the screen, there are two buttons: "CANCELAR" on the left and "SALVAR" on the right. The Android navigation bar is visible at the very bottom.

Alimentos	Quantidade	Valor
Omelete (Unidade - Pequena)	1.0	0.0
Suco de Laranja (Copo Pequeno Cheio)	1.0	0.0

Fonte: elaborado pelo autor.

Caso o paciente exclua a notificação ou informe algum consumo errado, ao consultar alguma refeição do seu cadastro, a opção de consultar consumo é exibida no topo da tela de cadastro de refeição depois de consultar o seu cadastro. Esta opção é apresentada na figura 33.

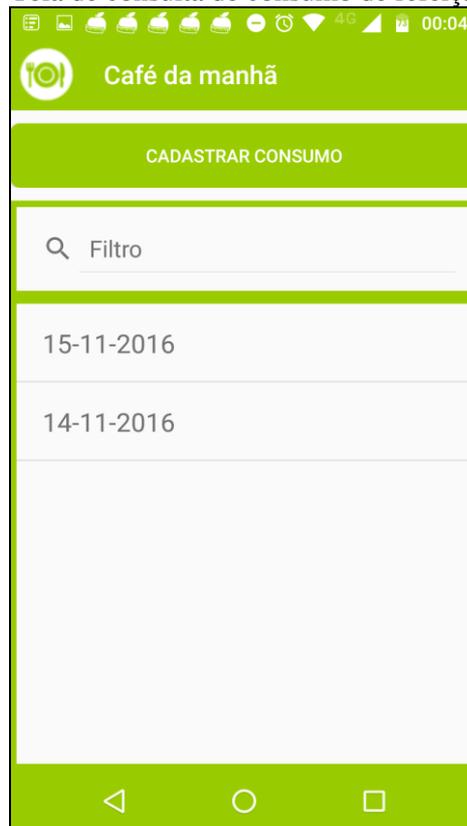
Figura 33 - Tela de consulta de uma refeição cadastrada



Fonte: elaborado pelo autor.

Ao consultar o consumo de uma refeição, é exibida a tela consulta de consumo de refeição específica, com a opção de cadastro de consumo, que redireciona para a mesma tela exibida ao selecionar a notificação da refeição, e abaixo a lista de refeições consumidas ordenadas pela data mais recente. A tela conta também com um campo de pesquisa semelhante as outras telas de consulta do sistema, ele realiza as buscas no banco de dados conforme o usuário insere os caracteres da pesquisa, filtrando a lista da tela de consulta de consumo. A figura 34 apresenta a tela de consulta de consumo de refeição específica.

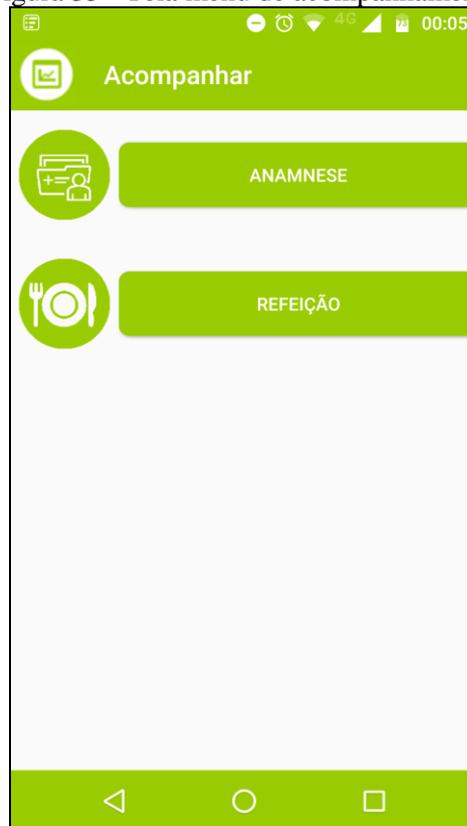
Figura 34 - Tela de consulta de consumo de refeição específica



Fonte: elaborado pelo autor.

Por fim, a última opção do fluxo de atendimento nutricional não mencionada do sistema, a tela do menu do paciente: acompanhar. Esta tela também pode ser acessada pelo paciente e pelo nutricionista e ela direciona para a tela de menu de acompanhamento. Neste menu podem ser selecionados o acompanhamento de anamnese e o acompanhamento de refeição conforme apresentado na figura 35.

Figura 35 - Tela menu de acompanhamento



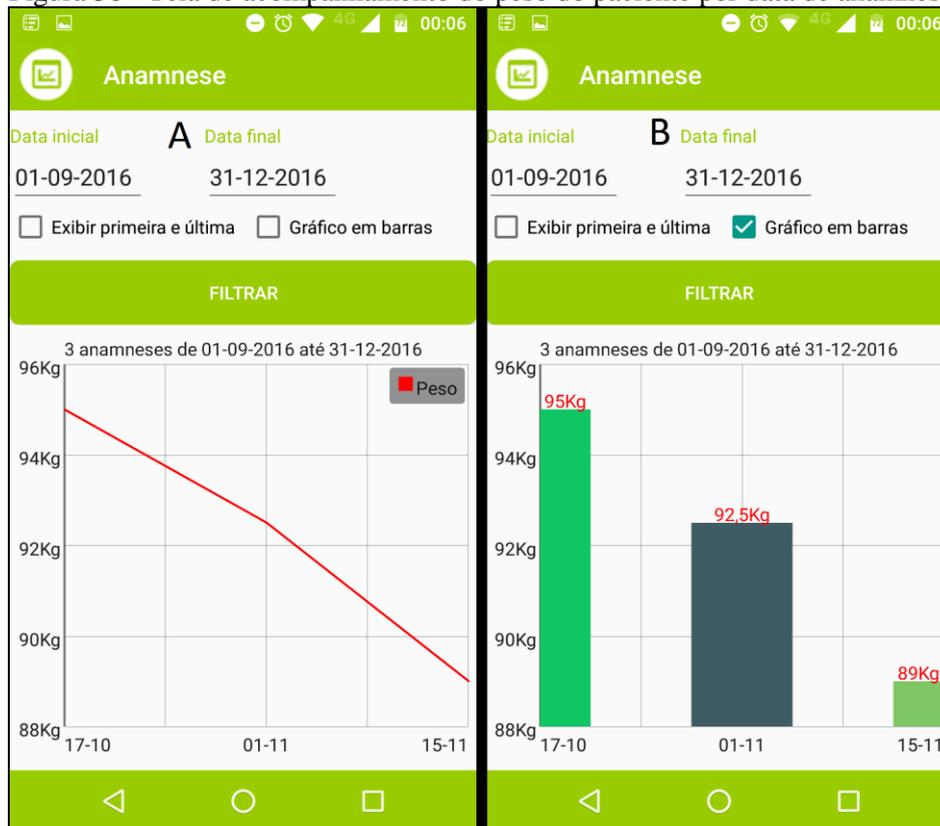
Fonte: elaborado pelo autor.

A figura 36 apresenta a tela de acompanhamento de anamnese, nesta tela podem ser gerados gráficos de linha e de coluna que comparam as anamneses realizadas pelo paciente, mostrando a diferença de peso em quilogramas que o paciente teve durante seu processo de acompanhamento nutricional.

Para funcionamento do gráfico são necessários no mínimo duas anamneses cadastradas pelo paciente. Para gerar o gráfico, é necessário informar um período através de uma data inicial e final. Abaixo possui a opção de exibir somente a primeira e a última anamnese caso existam mais de duas cadastradas, e a outra opção permite exibir um gráfico comparativo em barras (lado B da figura 36). Caso não seja selecionado, por padrão é exibido o gráfico de linha conforme o lado A da figura 36.

Para poder exibir todas as datas do gráfico quando existem várias a serem apresentadas, foi adicionado o recurso de gestos de aproximação e rolagem no gráfico padrões de smartphones através da aproximação ou afastamento de dois dedos ao mesmo tempo na tela. Todas as telas de acompanhamento gráfico possuem este recurso.

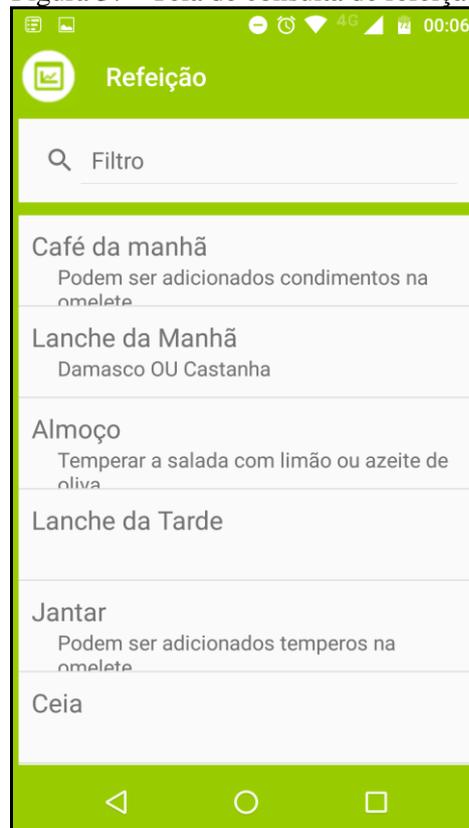
Figura 36 - Tela de acompanhamento do peso do paciente por data de anamnese



Fonte: elaborado pelo autor.

A outra opção existente no menu acompanhar (Figura 35) é o acompanhamento de refeição. Ao ser selecionada direciona para a consulta de refeições do paciente conforme apresentado na figura 37.

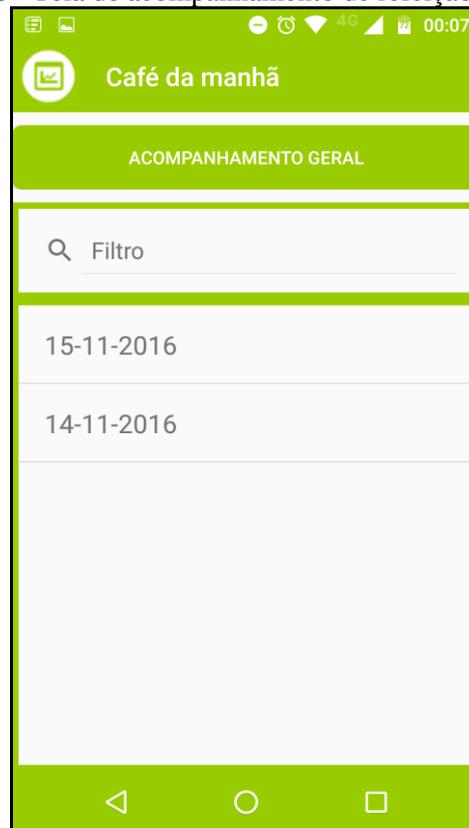
Figura 37 - Tela de consulta de refeição



Fonte: elaborado pelo autor.

Ao selecionar uma refeição a ser acompanhada, é exibida a tela de acompanhamento da refeição específica. Esta tela possui no topo uma opção de acompanhamento geral de todos os consumos realizados da refeição e abaixo a lista de datas em que foi cadastrado o consumo da refeição a ser acompanhada. A figura 38 apresenta a menu de acompanhamento de refeição específica. Apesar de ser semelhante à figura 34, ela apresenta a opção “Acompanhamento geral” no lugar da opção “Cadastrar consumo”.

Figura 38 - Tela de acompanhamento de refeição específica



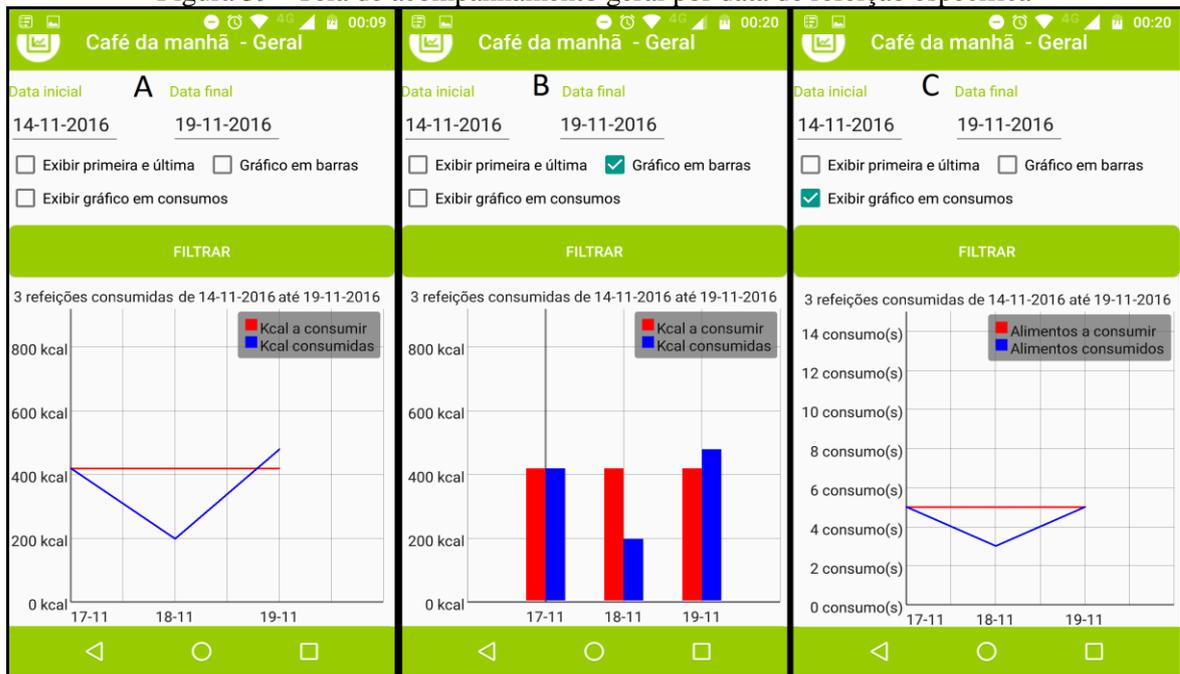
Fonte: elaborado pelo autor.

Ao selecionar a opção de acompanhamento geral, é apresentada a tela com as mesmas opções do acompanhamento de anamnese, tendo como diferencial as informações a serem exibidas.

Ao gerar o gráfico, na cor vermelha, é exibida a linha (ou colunas caso marcada opção) que representa a quantidade de quilocalorias a consumir da refeição no eixo Y, e no eixo X cada dia consumido da refeição. Em azul, seguindo a mesma atribuição de valores aos eixos da cor vermelha, é exibida a quantidade de quilocaloria consumida informada pelo paciente em cada dia com base na quantidade dos alimentos informados. Para gerar o gráfico são necessários ao menos dois dias de refeições consumidas serem encontradas no período definido pelo filtro. Abaixo do filtro existem os mesmos filtros do acompanhamento de anamnese, tendo como diferencial, a opção de exibir o gráfico em quantidade a consumir total dos alimentos da refeição e a quantidade total consumida informada pelo paciente.

A figura 39 apresenta a tela de acompanhamento geral de uma refeição e exibe no lado A o gráfico padrão em linhas, no lado B o gráfico em colunas e no lado C o gráfico em consumos.

Figura 39 - Tela de acompanhamento geral por data de refeição específica



Fonte: elaborado pelo autor.

Por fim, a outra opção de acompanhamento de refeição é através da lista de datas de consumo de uma refeição específica. Ao selecionar uma data, é apresentada a tela de acompanhamento de consumo individual por refeição. Esta tela possui também as opções para exibir o gráfico em barras e em consumos. Ao filtrar são exibidos da mesma forma que a tela de acompanhamento geral, a linha (ou colunas) vermelha representando a quantidade de quilocalorias a consumir de cada alimento, e em azul a quantidade de quilocalorias consumidas.

O diferencial é que esta tela apresenta o gráfico mostrando a quantidade de consumo por alimento e não por data como é feito no acompanhamento geral. A figura 40 apresenta a tela de acompanhamento individual por data de uma refeição específica. No lado A o gráfico em linhas do cálculo de quilocalorias por alimento a consumir e consumido, no lado B o gráfico em barras, e no lado C o gráfico em barras e em quantidade de alimentos a consumir e consumidos ao invés do cálculo de quilocalorias.

Figura 40 - Tela de acompanhamento individual por data de refeição específica



Fonte: elaborado pelo autor.

3.4 RESULTADOS E DISCUSSÕES

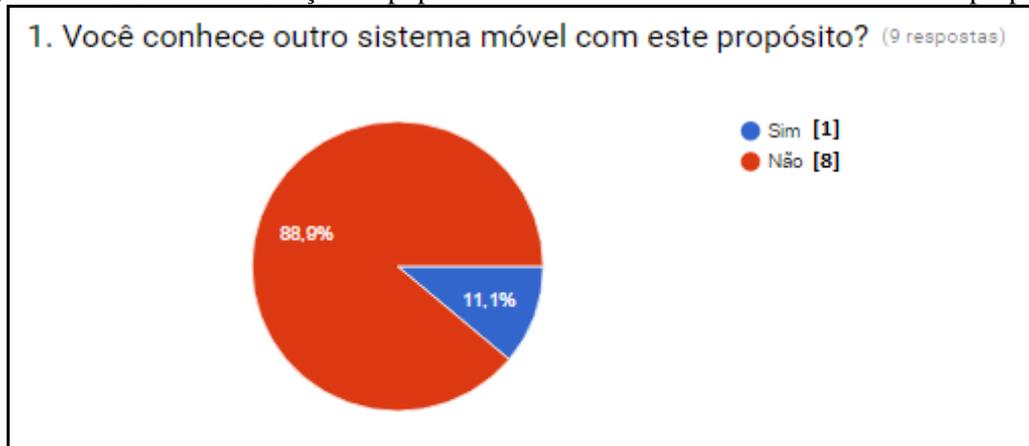
Este trabalho teve como propósito inicial desenvolver somente as extensões do protótipo de Kleinschmidt (2013), porém, devido as diferenças tecnológicas das ferramentas utilizadas foi decidido criar um novo sistema, tendo como inspiração o trabalho de Kleinschmidt (2013).

Os objetivos propostos foram alcançados, quais seja: de criar uma ferramenta de compartilhamento de informações sobre saúde alimentar entre nutricionista e paciente que permitisse além das características presentes no trabalho utilizado como inspiração, os diferenciais que eram elaborar cardápios com base em micronutrientes e unidades de medidas dos alimentos, permitir um controle de notificações das refeições executadas pelo paciente e permitir um acompanhamento gráfico baseado na análise das práticas alimentares dos pacientes através de cenários executados no aplicativo.

Para validação dos resultados alcançados, foi realizada uma apresentação da aplicação Nutrisys para uma professora do curso de Nutrição da FURB, uma aluna do terceiro semestre do mesmo curso e também outras sete pessoas. Onde foi demonstrado o conjunto de funcionalidades do aplicativo e explicados os seus módulos. Em seguida foi aplicado um questionário com quatro perguntas de múltipla escolha e uma questão descritiva opcional para críticas e sugestões com vista a avaliação da usabilidade da aplicação (Apêndice C).

Na figura 41 apresenta-se o gráfico que representa a popularidade deste tipo de aplicação nos dispositivos móveis, que apesar de todos os respondentes possuírem smartphones, apenas um conhece outro sistema móvel com o mesmo propósito deste trabalho e os outros oito não conhecem nenhum.

Figura 41 - Gráfico de avaliação da popularidade de sistemas móveis com o mesmo propósito



Fonte: elaborado pelo autor.

A figura 42 apresenta que apesar da complexidade do sistema, para oito respondentes a aplicação mostrou-se de fácil entendimento e um informa que a usabilidade do aplicativo é parcialmente atendida, mostrando que a aplicação poderia ser disponibilizada para um público maior.

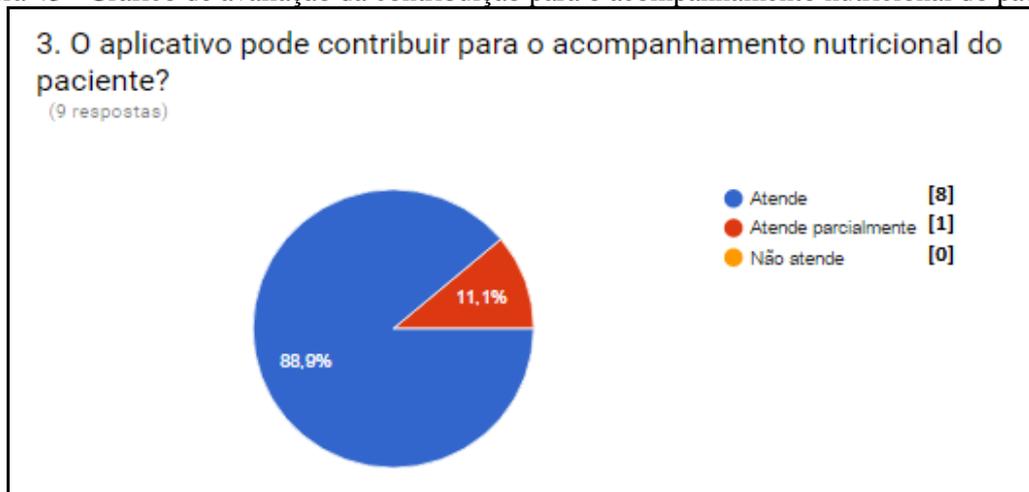
Figura 42 - Gráfico de avaliação da usabilidade do sistema



Fonte: elaborado pelo autor.

A figura 43 apresenta que dos nove questionados, apenas um respondeu que o aplicativo atenderia parcialmente como contribuição a um acompanhamento nutricional do paciente.

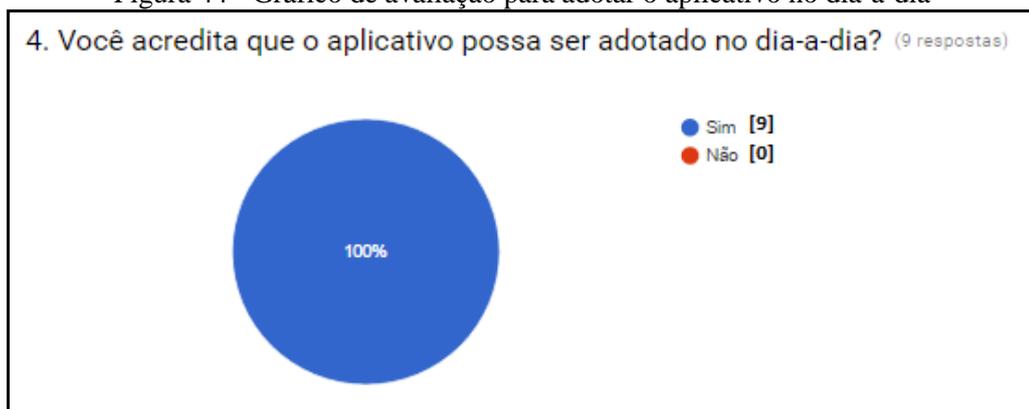
Figura 43 - Gráfico de avaliação da contribuição para o acompanhamento nutricional do paciente



Fonte: elaborado pelo autor.

A figura 44 apresenta que todos os respondentes afirmam que acreditam que o aplicativo possa ser adotado no dia-a-dia.

Figura 44 - Gráfico de avaliação para adotar o aplicativo no dia-a-dia



Fonte: elaborado pelo autor.

Conforme a figura 45, a última pergunta que foi realizada como um opcional do questionário referente as sugestões e críticas foram respondidas por cinco pessoas. Elas apresentam três sugestões relacionadas as refeições executadas pelo paciente, uma sugestão para a tela de cadastro de anamnese e duas sugestões para os gráficos apresentados.

Figura 45 - Questionário opcional de sugestão ou crítica ao aplicativo

5. (Opcional) Você teria alguma sugestão ou crítica ao aplicativo? (5 respostas)

Como sugestões utilizando o perfil paciente seria a inclusão dos alimentos que não estão na dieta ao realizar o registro do acompanhamento, sendo possível assim verificar o que o paciente ingeriu além do que estava em seu plano alimentar.

Opção de escolher o alimento equivalente ao sugerido.

Seria interessante poder anotar as medidas do paciente

Gráfico de pizza ficaria mais fácil de entender

Evolução através de gráficos de dados antropométricos.
Adequação do consumo alimentar a partir de grupos alimentares.

Fonte: elaborado pelo autor.

Quanto aos trabalhos correlatos, o estudo de Cruz (2008) apenas fez o levantamento de informações. Apesar de não ter sido desenvolvido, tiveram além dos diferenciais, como a possibilidade de funcionar em computadores convencionais por ser um *applet* Java. Alguns comparativos a este trabalho podem ser realizados como, por exemplo, a criação de módulos separados para o nutricionista e o paciente, e a possibilidade de utilizar os smartphones para realização e gerenciamento do atendimento nutricional.

Em relação a aplicação de Zimmermann (2011), também houveram comparações e diferenças mensuráveis. Apesar do seu propósito ser semelhante, o trabalho de Zimmermann (2011) teve como foco uma aplicação para computadores convencionais, diferente deste trabalho que propôs mobilidade no compartilhamento do atendimento e acompanhamento nutricional. Porém, como semelhança, o trabalho de Zimmermann (2011) também dispôs da possibilidade de criar diferentes dietas e cardápios elaborados em macronutrientes e micronutrientes por paciente.

Por fim, o protótipo de Kleinschmidt (2013) permitiu com que o conjunto de trabalhos correlatos aqui apresentados, influenciassem na criação uma nova aplicação voltada para o mercado atual de smartphones. Em relação ao seu trabalho, a aplicação desenvolvida permitiu também o acompanhamento de um profissional e um paciente durante o PAN (Processo de Assistência Nutricional) através de um *smartphone* com o compartilhamento de informações, e também a notificação das refeições a serem executadas pelo paciente. O diferencial, foi mencionado como extensão do trabalho de Kleinschmidt (2013) e foi alcançado. Eles permitiram que sejam adicionados micronutrientes (vitaminas e nutrientes) e unidades de medidas (porções) aos alimentos informados nas refeições. O maior controle das notificações executadas pelos pacientes, que podem agora alterar o horário que desejam ser notificados de suas refeições. Por fim, o acompanhamento gráfico baseado na análise das práticas alimentares dos pacientes, que permitem saber a evolução do peso do paciente por data de

anamneses realizadas, e também os gráficos das quilocalorias e consumos das refeições executadas pelo paciente.

Concluindo esta seção de discussões, o quadro 20 apresenta um comparativo com os trabalhos correlatos.

Quadro 20 - Comparativo entre o sistema desenvolvido e os trabalhos correlatos

Funcionalidades e características	Cruz (2008)	Zimmermann (2011)	Kleinschmidt (2013)	Sistema desenvolvido
Linguagem do sistema	JAVA	PHP	JAVA	JAVA
Módulo para <i>desktop</i>	Possui	Possui	Não possui	Não possui
Módulo para <i>smartphone</i>	Possui	Não possui	Possui	Possui
Cadastro de medidas do paciente	Não informado	Possui	Não possui	Não possui
Cardápios baseados em macronutrientes.	Não informado	Possui	Possui	Possui
Cardápios baseados em micronutrientes.	Não informado	Possui	Não possui	Possui
Compartilhamento de informações com o paciente.	Possui	Não possui	Possui	Possui
Notificação de refeições.	Não possui	Não possui	Possui	Possui
Personalizar horário de notificações	Não possui	Não possui	Não possui	Possui
Gerar acompanhamento gráfico do atendimento.	Não possui	Não possui	Não possui	Possui

Fonte: elaborado pelo autor.

4 CONCLUSÕES

Com o avanço da tecnologia e o crescimento do mercado de smartphones segundo Amador (2015), cada vez mais a tecnologia fará parte do dia-a-dia de diversos ramos profissionais. Conforme Kleinschmidt (2013, p. 57), as pessoas estão cada vez mais preocupadas com a qualidade de vida, aumentando assim estimativa de vida do ser humano.

A reeducação alimentar é um fator muito importante na qualidade de vida e para ser alcançada precisa ser acompanhada de uma forma que evite o paciente cometa deslizos no seu acompanhamento, por mais difícil que seja devido as pressões sociais ou familiares, conforme ressalta Medeiros (2010, p. 32).

Levando estes fatores em consideração, o sistema desenvolvido com o levantamento de informações sobre o PAN (Processo de Assistência Nutricional) conforme mencionado por Bon et al (2013, p. 85) e também com o estudo dos trabalhos correlatos de Cruz (2008), Zimmermann (2011) e Kleinschmidt (2013), permitiu o compartilhamento de informações sobre a saúde alimentar entre paciente e nutricionista, cada um tendo a sua visão do tratamento e possibilitando o cadastro e consultas sobre atendimentos baseados em macronutrientes e micronutrientes. Também se apresentou a personalização do horário das notificações do paciente e o acompanhamento gráfico com a análise das anamneses e refeições consumidas fazendo assim com que os objetivos deste trabalho fossem alcançados.

Como maiores dificuldades, destacam-se o tempo dedicado ao estudo e conclusão do *webservice*, e alguns problemas com o desenvolvimento para a plataforma Android. Como problemas com a plataforma Android temos a limitação do campo de visão e as funcionalidades de lista personalizada com os campos para informar quantidade de cada item da lista conforme apresentado nas telas de cadastro de alimento, cadastro de refeição e cadastro de consumo, que possuem certas particularidades em sua sintaxe para funcionarem através das chamadas *views*. E por fim, a criação da lógica para desenvolver os gráficos relacionados à anamnese e as refeições junto a biblioteca *GraphView*.

Este trabalho propôs além das extensões de Kleinschmidt (2013) um novo aplicativo para a plataforma Android utilizando a ferramenta Android Studio, que através de diversos recursos em conjunto com *webservice*, oferece o compartilhamento de informações entre pacientes e nutricionistas, além de notificações com horários personalizáveis e acompanhamentos gráficos.

Conclui-se com a realização deste trabalho o aumento dos conhecimentos sobre o tema nutrição, o desenvolvimento e funcionamento do *webservice* em JAVA com a biblioteca JAX-

RS e o desenvolvimento para plataforma Android na ferramenta Android Studio em conjunto a biblioteca GraphView. Também proporcionou o aumento do conhecimento na área de programação, visto que as tecnologias utilizadas não fazem parte do dia-a-dia do autor.

4.1 EXTENSÕES

Como sugestões para trabalhos futuros pode-se elencar as seguintes:

- a) desenvolver uma funcionalidade para recuperação de senha do aplicativo;
- b) desenvolver uma funcionalidade para enviar cardápios por e-mail;
- c) desenvolver uma funcionalidade para adicionar alimentos fora da refeição cadastrada pelo nutricionista;
- d) desenvolver uma funcionalidade para permitir listas alternativas de alimentos por refeição;
- e) desenvolver uma funcionalidade para o funcionamento *off-line* do aplicativo;
- f) desenvolver o aplicativo em outra plataforma, como iOS, Windows Phone, ou através da ferramenta PhoneGap, unindo HTML5, CSS e Javascript.

REFERÊNCIAS

- ABLESON, W. Frank et al. **Android em ação**. 3ª Edição. Rio de Janeiro: Elsevier Brasil, 2012.
- AMADOR, João Gabriel. **Estudo mostra crescimento no uso de dispositivos móveis e domínio Android**: Pesquisa realizada com mais de 40 mil internautas em 32 países, incluindo Brasil, revela aumento de uso da internet em smartphones e tablets. Brasília, [2015]. Disponível em: <http://www.correiobraziliense.com.br/app/noticia/tecnologia/2015/01/17/interna_tecnologia,466691/estudo-mostra-crescimento-no-uso-de-dispositivos-moveis-e-dominio-andr.shtml>. Acesso em: 14 jan. 2016.
- BITTENCOURT, Patrícia Afonso de et al. Estratégias de atuação do nutricionista em consultoria alimentar e nutricional da família. **Revista de nutrição**. Campinas, v. 22, n. 6, p. 919-927, dez. 2009. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-52732009000600013&lng=en&nrm=iso>. Acesso em: 14 mar. 2016.
- BON, Avany Maria Xavier et al. **Atendimento nutricional**: uma visão prática adultos e idosos. São Paulo: M. Books do Brasil Editora Ltda, 2013.
- BRUNET, Daniel. **Procura por nutricionista aumentou 32,5% ano passado**. Rio de Janeiro, [2014]. Disponível em: <<http://blogs.oglobo.globo.com/blog-emergencia/post/procura-por-nutricionista-aumentou-32-5-ano-passado-535395.html>>. Acesso em: 14 mar. 2016.
- CALDEIRA, Vânia. **Alimentação e obesidade**. Torres Vedras, [2007]. Disponível em: <http://coursejournal_medicina.blogs.sapo.pt/19005.html>. Acesso em: 14 maio 2016.
- CARBONIERI, Fernando. **O futuro da medicina está em seu smartphone**. Londrina, [2015]. Disponível em: <<https://academiamedica.com.br/o-futuro-da-medicina-esta-em-seu-smartphone/>>. Acesso em: 14 maio 2016.
- CONSELHO FEDERAL DE NUTRICIONISTAS, **Resolução CFN Nº 419/2008 Capítulo I da responsabilidade técnica ART. 1º**: A Responsabilidade Técnica exercida pelo Nutricionista é o compromisso profissional e legal na execução de suas atividades, compatível com a formação e os princípios éticos da profissão, visando a qualidade dos serviços prestados a sociedade. Brasília, 2008a. 110 p. Disponível em: <<http://www.cfn.org.br/novosite/pdf/res/2008/res419.pdf>>. Acesso em: 14 mar. 2016
- CONDE, Jéssica. **Anamnese alimentar**. Araçatuba, [2012]. Disponível em: <<http://jeehconde.blogspot.com.br/2012/04/anamnese-alimentar.html>>. Acesso em: 14 mar. 2016.
- CRUZ, Eduardo Flávio. **Proposta de modelagem para um sistema de acompanhamento nutricional através do telefone celular**. 2008. 13 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro de Ciências Exatas e Naturais, Universidade Presidente Antônio Carlos, Barbacena.
- DURÃO, Catarina. **Anamnese**: Uma oportunidade para aumentar a aderência à intervenção nutricional. [2003]. Disponível em: <<http://www.apn.org.pt/documentos/revistas/Doc3.pdf>>. Acesso em: 14 mar. 2016.
- FISBERG, Regina Mara et al. **Inquéritos alimentares**: Métodos e bases científicas. Barueri, SP: Editora Manole Ltda, 2005.

- KLEINSCHMIDT, Ana Claudia. **Protótipo de um aplicativo Android para compartilhamento de informações nutricionais entre profissional – paciente**. 2013. 77 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- LECHETA, Ricardo. **Android essencial**. São Paulo: Novatec Editora, 2016.
- MEDEIROS, Fernando. **Um mar de possibilidades: A medicina no passado, presente e futuro**. São Paulo: Editora Biblioteca 24 Horas, 2010.
- MICROSYSTEMS, Sun. **RESTful web services developer’s guide**. Santa Clara, [2009]. Disponível em: <<https://docs.oracle.com/cd/E19776-01/820-4867/820-4867.pdf>>. Acesso em: 5 nov. 2016
- PAPINI, Hevoise Fátima. **O que é reeducação alimentar**. São Paulo, [2007]. Disponível em: <<http://hevoise.blogspot.com.br/2007/12/o-que-reeducacao-alimentar-reeducacao.html>>. Acesso em: 14 mar. 2016.
- POLÔNIO, Maria Lúcia Teixeira; PERES, Frederico. Consumo de aditivos alimentares e efeitos à saúde: desafios para a saúde pública brasileira. **Cad. Saúde Pública**, Rio de Janeiro, v. 25, n. 8, p. 1653-1666, Ago. 2009. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-311X2009000800002&lng=en&nrm=iso>. Acesso em: 14 mar. 2016.
- SIZER, Frances Sienkiewicz; WHITNEY, Eleanor. **Nutrição: oitava edição conceitos e controvérsias**. Barueri, SP: Editora Manole Ltda, 2003.
- SMITH, Ben. **JSON Básico: Conheça o formato de dados preferido da web**. São Paulo: Novatec Editora, 2015.
- VARELLA, Dr. Drauzio. **Obesidade e nutrição**. São Paulo, SP: Gold Editora Ltda, 2009.
- VILLELA, Flávia. **Número de brasileiros com celular cresce 107%, diz IBGE**. Rio de Janeiro, [2013]. Disponível em: <<http://exame.abril.com.br/tecnologia/noticias/numero-de-brasileiros-com-celular-cresce-107-diz-ibge>>. Acesso em: 14 maio. 2016.
- WHO, World Health Organization. **Nutrition**. [2016]. Disponível em: <<http://www.who.int/topics/nutrition/en/>>. Acesso em 14 mar. 2016.
- ZIMMERMANN, Natasha. **Sistema de Informação para elaboração de cardápios nutricionais**. 2011. 56 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

APÊNDICE B – Descrição dos principais casos de uso

Esta seção da proposta apresenta a descrição dos principais casos de uso apresentados na seção de especificação deste trabalho.

No quadro 21 é apresentado a descrição do caso de uso “manter o paciente”.

Quadro 21 – UC01 - Manter o paciente

UC01 – Manter o paciente

Permite ao usuário nutricionista manter tabela de paciente. Serão cadastrados os seguintes dados: nome, idade, data de nascimento, CPF, cidade, endereço, bairro, número, complemento, CEP, e-mail, telefone, celular, sexo, usuário e senha.

Ator: Usuário Nutricionista

Pré-condição: O usuário deve possuir autorização para uso do aplicativo.

Cenário Principal

1. O usuário nutricionista seleciona a opção paciente no menu do nutricionista.
2. O sistema apresenta o menu de opções do paciente: cadastrar e consultar.
3. O usuário nutricionista seleciona a opção cadastrar.
4. O sistema apresenta a tela de cadastro de paciente.
5. O usuário nutricionista preenche as informações relacionadas ao paciente.
6. O sistema faz a verificação dos campos obrigatórios (nome, idade, data de nascimento, CPF, CEP, telefone, sexo, usuário, senha e confirmação de senha) através da persistência da entidade preenchida pelo usuário nutricionista.
7. O sistema grava a informação verificada.

Fluxo Alternativo 1

No passo 2 do cenário principal, o usuário nutricionista seleciona a opção consultar do menu de opções do paciente.

- 2.1. O sistema apresenta a tela de consulta, exibindo uma lista com os pacientes cadastrados.

2.2. O usuário nutricionista seleciona um registro da tela de consulta.

2.3. O sistema deve apresentar a tela menu do paciente, exibindo ao nutricionista as opções: anamnese, dieta, cardápio, refeição, acompanhar e alterar cadastro.

2.4. O usuário nutricionista seleciona a opção alterar cadastro.

2.5. Retorna ao passo 4 do cenário principal.

Fluxo Alternativo 2

No passo 6 do cenário principal, o sistema verifica na persistência que existem informações obrigatórias não preenchidas.

6.1. O sistema informa qual campo deve ser preenchido.

6.2. O usuário nutricionista preenche o campo informado.

6.3. Retorna ao passo 7 do cenário principal.

Fonte: elaborado pelo autor.

No quadro 22 é apresentado a descrição do caso de uso “manter os alimentos”.

Quadro 22 - UC02 - Manter os alimentos

UC02 – Manter os alimentos

Permite ao usuário nutricionista manter tabela de alimentos. Serão cadastrados os seguintes dados: nome, descrição, grupo de alimento, porção do alimento, quantidade de proteínas, quantidade de carboidratos, quantidade de lipídios, quantidade de quilocalorias e vitaminas/nutrientes.

Ator: Usuário Nutricionista

Pré-condição: O usuário deve possuir autorização para uso do aplicativo.

Cenário Principal

1. O usuário nutricionista seleciona a opção alimento no menu do nutricionista.

2. O sistema apresenta o menu de opções do alimento: cadastrar e consultar.

3. O usuário nutricionista seleciona a opção cadastrar.

4. O sistema apresenta a tela de cadastro de alimento.

5. O usuário nutricionista preenche as informações relacionadas ao alimento.

6. O sistema faz a verificação dos campos obrigatórios (nome, grupo de alimento, porção do alimento) através da persistência da entidade preenchida pelo usuário nutricionista.

7. O sistema grava a informação verificada.

Fluxo Alternativo 1

No passo 2 do cenário principal, o usuário nutricionista seleciona a opção consultar do menu de opções do alimento.

2.1. O sistema apresenta a tela de consulta, exibindo uma lista com os alimentos cadastrados.

2.2. O usuário nutricionista seleciona um registro da tela de consulta.

2.3. Retorna ao passo 4 do cenário principal.

Fluxo Alternativo 2

No passo 6 do cenário principal, o sistema verifica na persistência que existem informações obrigatórias não preenchidas.

6.1. O sistema informa qual campo deve ser preenchido.

6.2. O usuário nutricionista preenche o campo informado.

6.3. Retorna ao passo 7 do cenário principal.

Fonte: elaborado pelo autor.

No quadro 23 é apresentado a descrição do caso de uso “manter as anamneses”.

Quadro 23 - UC06 - Manter as anamneses

UC06 – Manter as anamneses

Permite ao usuário nutricionista manter tabela de anamnese. Serão cadastrados os seguintes dados: descrição, motivo, objetivo, data inicial, data final, peso, peso ideal, altura e doenças.

Ator: Usuário Nutricionista

Pré-condição: O usuário deve possuir autorização para uso do aplicativo.

Pré-condição: Deve estar cadastrado no sistema pelo menos um paciente.

Cenário Principal

1. O usuário nutricionista seleciona a opção anamnese no menu do paciente.
2. O sistema apresenta o menu de opções da anamnese: cadastrar e consultar.
3. O usuário nutricionista seleciona a opção cadastrar.
4. O sistema apresenta a tela de cadastro de anamnese.
5. O usuário nutricionista preenche as informações relacionadas a anamnese.
6. O sistema faz a verificação dos campos obrigatórios (descrição, motivo, objetivo, data inicial, data final, peso, peso ideal e altura) através da persistência da entidade preenchida pelo usuário nutricionista.
7. O sistema grava a informação verificada.

Fluxo Alternativo 1

No passo 2 do cenário principal, o usuário nutricionista seleciona a opção consultar

do menu de opções da anamnese.

- 2.1. O sistema apresenta a tela de consulta, exibindo uma lista com as anamneses cadastradas.
- 2.2. O usuário nutricionista seleciona um registro da tela de consulta.
- 2.3. Retorna ao passo 4 do cenário principal.

Fluxo Alternativo 2

No passo 6 do cenário principal, o sistema verifica na persistência que existem informações obrigatórias não preenchidas.

- 6.1. O sistema informa qual campo deve ser preenchido.
- 6.2. O usuário nutricionista preenche o campo informado.
- 6.3. Retorna ao passo 7 do cenário principal.

Fonte: elaborado pelo autor.

No quadro 24 é apresentado a descrição do caso de uso “manter as dietas”.

Quadro 24 - UC010 - Manter as dietas

UC010 – Manter as dietas

Permite ao usuário nutricionista manter tabela de dieta. Serão cadastrados os seguintes dados: anamnese, data de referência, observação, quantidade a consumir por semana.

Ator: Usuário Nutricionista

Pré-condição: O usuário deve possuir autorização para uso do aplicativo.

Pré-condição: Deve estar cadastrado no sistema pelo menos um paciente.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma anamnese.

Cenário Principal

1. O usuário nutricionista seleciona a opção dieta no menu do paciente.
2. O sistema apresenta o menu de opções da dieta: cadastrar e consultar.
3. O usuário nutricionista seleciona a opção cadastrar.
4. O sistema apresenta a tela de cadastro de dieta.
5. O usuário nutricionista preenche as informações relacionadas a dieta.
6. O sistema faz a verificação dos campos obrigatórios (anamnese, data de referência e observação) através da persistência da entidade preenchida pelo usuário nutricionista.
7. O sistema grava a informação verificada.

Fluxo Alternativo 1

No passo 2 do cenário principal, o usuário nutricionista seleciona a opção consultar

do menu de opções da dieta.

2.1. O sistema apresenta a tela de consulta, exibindo uma lista com as dietas cadastradas.

2.2. O usuário nutricionista seleciona um registro da tela de consulta.

2.3. Retorna ao passo 4 do cenário principal.

Fluxo Alternativo 2

No passo 6 do cenário principal, o sistema verifica na persistência que existem informações obrigatórias não preenchidas.

6.1. O sistema informa qual campo deve ser preenchido.

6.2. O usuário nutricionista preenche o campo informado.

6.3. Retorna ao passo 7 do cenário principal.

Fonte: elaborado pelo autor.

No quadro 25 é apresentada a descrição do caso de uso “manter os cardápios”.

Quadro 25 - UC011 - Manter os cardápios

UC011 – Manter os cardápios

Permite ao usuário nutricionista manter tabela de cardápio. Serão cadastrados os seguintes dados: dieta, quilocalorias e observação.

Ator: Usuário Nutricionista

Pré-condição: O usuário deve possuir autorização para uso do aplicativo.

Pré-condição: Deve estar cadastrado no sistema pelo menos um paciente.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma anamnese.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma dieta.

Cenário Principal

1. O usuário nutricionista seleciona a opção cardápio no menu do paciente.

2. O sistema apresenta o menu de opções do cardápio: cadastrar e consultar.

3. O usuário nutricionista seleciona a opção cadastrar.

4. O sistema apresenta a tela de cadastro de cardápio.

5. O usuário nutricionista preenche as informações relacionadas ao cardápio.

6. O sistema faz a verificação dos campos obrigatórios (todos) através da persistência da entidade preenchida pelo usuário nutricionista.

7. O sistema grava a informação verificada.

Fluxo Alternativo 1

No passo 2 do cenário principal, o usuário nutricionista seleciona a opção consultar do menu de opções do cardápio.

2.1. O sistema apresenta a tela de consulta, exibindo uma lista com os cardápios cadastrados.

2.2. O usuário nutricionista seleciona um registro da tela de consulta.

2.3. Retorna ao passo 4 do cenário principal.

Fluxo Alternativo 2

No passo 6 do cenário principal, o sistema verifica na persistência que existem informações obrigatórias não preenchidas.

6.1. O sistema informa qual campo deve ser preenchido.

6.2. O usuário nutricionista preenche o campo informado.

6.3. Retorna ao passo 7 do cenário principal.

Fonte: elaborado pelo autor.

No quadro 26 é apresentado a descrição do caso de uso “manter as refeições”.

Quadro 26 - UC012 - Manter as refeições

UC012 – Manter as refeições

Permite ao usuário nutricionista manter tabela de refeição. Serão cadastrados os seguintes dados: cardápio, nome, horário, observação e alimentos.

Ator: Usuário Nutricionista

Pré-condição: O usuário deve possuir autorização para uso do aplicativo.

Pré-condição: Deve estar cadastrado no sistema pelo menos um paciente.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma anamnese.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma dieta.

Pré-condição: Deve estar cadastrado no sistema pelo menos um cardápio.

Cenário Principal

1. O usuário nutricionista seleciona a opção refeição no menu do paciente.
2. O sistema apresenta o menu de opções da refeição: cadastrar e consultar.
3. O usuário nutricionista seleciona a opção cadastrar.
4. O sistema apresenta a tela de cadastro de refeição.
5. O usuário nutricionista preenche as informações relacionadas a refeição.
6. O sistema faz a verificação dos campos obrigatórios (cardápio, nome e horário) através da persistência da entidade preenchida pelo usuário nutricionista.
7. O sistema grava a informação verificada.

Fluxo Alternativo 1

No passo 2 do cenário principal, o usuário nutricionista seleciona a opção consultar do menu de opções do cardápio.

- 2.1. O sistema apresenta a tela de consulta, exibindo uma lista com os cardápios cadastrados.
- 2.2. O usuário nutricionista seleciona um registro da tela de consulta.
- 2.3. Retorna ao passo 4 do cenário principal.

Fluxo Alternativo 2

No passo 6 do cenário principal, o sistema verifica na persistência que existem informações obrigatórias não preenchidas.

- 6.1. O sistema informa qual campo deve ser preenchido.
- 6.2. O usuário nutricionista preenche o campo informado.
- 6.3. Retorna ao passo 7 do cenário principal.

Fonte: elaborado pelo autor.

No quadro 27 é apresentado a descrição do caso de uso “analisar a quantidade de alimentos consumidos informada pelo paciente”.

Quadro 27 - UC014 - Analisar a quantidade de alimentos consumidos informada pelo paciente

UC014 – Analisar a quantidade de alimentos consumidos informada pelo paciente

Permite ao usuário nutricionista analisar a quantidade de alimentos consumidos informada pelo paciente.

Ator: Usuário Nutricionista

Ator: Webservice

Pré-condição: O usuário deve possuir autorização para uso do aplicativo.

Pré-condição: Deve estar cadastrado no sistema pelo menos um paciente.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma anamnese.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma dieta.

Pré-condição: Deve estar cadastrado no sistema pelo menos um cardápio.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma refeição com alimentos.

Pré-condição: Deve estar cadastrado no sistema pelo menos um registro de consumo do paciente.

Cenário Principal

1. O usuário nutricionista seleciona a opção refeição no menu do paciente.
2. O sistema apresenta o menu de opções da refeição: cadastrar e consultar.
3. O usuário nutricionista seleciona a opção consultar.
4. O sistema apresenta a tela de consulta da refeição com a lista de refeições cadastradas do paciente.
5. O usuário nutricionista seleciona uma refeição da lista.
6. O sistema apresenta a tela de cadastro de refeição.
7. O usuário nutricionista seleciona a opção consultar consumo.
8. O sistema apresenta a tela de consulta de refeições consumidas com a lista de

datas de consumo.

9. O usuário nutricionista seleciona um item da lista de refeições consumidas.

10. O sistema apresenta a tela com as informações de consumo da data selecionada.

Fonte: elaborado pelo autor.

No quadro 28 é apresentado a descrição do caso de uso “gerar gráficos com informações das anamneses do paciente”.

Quadro 28 - UC015 - Gerar gráficos com informações das anamneses do paciente

UC015 – Gerar gráficos com informações das anamneses do paciente

Permite ao usuário nutricionista gerar gráficos com informações das anamneses do paciente.

Ator: Usuário Nutricionista

Pré-condição: O usuário deve possuir autorização para uso do aplicativo.

Pré-condição: Deve estar cadastrado no sistema pelo menos um paciente.

Pré-condição: Deve estar cadastrado no sistema pelo menos duas anamneses.

Cenário Principal

1. O usuário nutricionista seleciona a opção acompanhar no menu do paciente.
2. O sistema apresenta o menu de acompanhamento com as opções: anamnese e refeição.
3. O usuário nutricionista seleciona a opção anamnese.
4. O sistema apresenta a tela de acompanhamento de anamnese.
5. O usuário nutricionista preenche a data inicial e a data final.
6. O usuário nutricionista seleciona a opção filtrar.
7. O sistema apresenta a quantidade de anamneses encontrada no período e o gráfico em linhas informando o peso do paciente em cada data de anamnese.

Fluxo Alternativo 1

No passo 5 do cenário principal, o usuário nutricionista preenche a data inicial, a data final e seleciona a opção exibir primeira e última anamnese somente.

5.1. O usuário nutricionista seleciona a opção filtrar.

5.2. O sistema apresenta a quantidade de anamneses encontrada no período e o gráfico em linhas informando o peso do paciente em cada data de anamnese, mas só exibe no eixo X a data da primeira e a última anamnese.

Fluxo Alternativo 2

No passo 5 do cenário principal, o usuário nutricionista preenche a data inicial, a data final e seleciona a opção exibir o gráfico em barras.

5.3. O usuário nutricionista seleciona a opção filtrar.

5.4. O sistema apresenta a quantidade de anamneses encontrada no período e o gráfico em barras informando o peso do paciente em cada data de anamnese.

Fluxo Alternativo 3

No passo 6 do cenário principal, o usuário nutricionista informa um período que não possui registros ou possui somente uma anamnese.

6.1. O sistema apresenta a mensagem informando se não foram encontrados registros ou se somente uma anamnese foi encontrada.

6.2. Retorna ao passo 5 do cenário principal.

Fonte: elaborado pelo autor.

No quadro 29 é apresentado a descrição do caso de uso “gerar gráficos com informações das refeições consumidas pelo paciente”.

Quadro 29 - UC016 - Gerar gráficos com informações das refeições consumidas pelo paciente

UC016 – Gerar gráficos com informações das refeições consumidas pelo paciente.

Permite ao usuário nutricionista gerar gráficos com informações das anamneses do paciente.

Ator: Usuário Nutricionista

Pré-condição: O usuário deve possuir autorização para uso do aplicativo.

Pré-condição: Deve estar cadastrado no sistema pelo menos um paciente.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma anamnese.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma dieta.

Pré-condição: Deve estar cadastrado no sistema pelo menos um cardápio.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma refeição com alimentos.

Pré-condição: Deve estar cadastrado no sistema pelo menos um registro de consumo do paciente.

Cenário Principal

1. O usuário nutricionista seleciona a opção acompanhar no menu do paciente.
2. O sistema apresenta o menu de acompanhamento com as opções: anamnese e refeição.
3. O usuário nutricionista seleciona a opção refeição.
4. O sistema apresenta a tela de consulta de refeição do paciente com a lista de refeições cadastradas pelo nutricionista.
5. O usuário nutricionista seleciona um item da lista de refeições do paciente.
6. O sistema apresenta a tela com o menu de acompanhamento da refeição selecionada, este contendo a opção acompanhamento geral e abaixo uma lista contendo as datas de consumo da refeição.
7. O usuário nutricionista seleciona a opção acompanhamento geral.
8. O sistema apresenta a tela de acompanhamento geral de refeições.
9. O usuário nutricionista preenche a data inicial e a data final.
10. O usuário nutricionista seleciona a opção filtrar.
11. O sistema apresenta a quantidade de quilocalorias a consumir dos alimentos e a quantidade de quilocalorias consumidas dos alimentos no período através de um gráfico com duas linhas em cada data da refeição consumida.

Fluxo Alternativo 1

No passo 6 do cenário principal, o usuário seleciona um item da lista de datas de consumo da refeição selecionada.

6.1. O sistema apresenta a tela de consumo por data individual.

6.2. O usuário nutricionista preenche a data inicial e a data final.

6.3. O usuário nutricionista seleciona a opção filtrar.

6.4. O sistema apresenta a quantidade de quilocalorias a consumir e a quantidade de quilocalorias consumidas da data de consumo específica exibindo um gráfico de duas linhas em cada alimento da refeição consumida.

Fluxo Alternativo 2

No passo 6.2 do cenário alternativo 1, o usuário nutricionista preenche a data inicial, a data final e seleciona a opção exibir o gráfico em barras.

6.2.1. O usuário nutricionista seleciona a opção filtrar.

6.2.2. O sistema apresenta a quantidade de quilocalorias a consumir e a quantidade de quilocalorias consumidas da data de consumo específica exibindo um gráfico com duas barras em cada alimento da refeição consumida.

Fluxo Alternativo 3

No passo 6.2 do cenário alternativo 1, o usuário nutricionista preenche a data inicial, a data final e seleciona a opção exibir o gráfico em consumos.

6.2.3. O usuário nutricionista seleciona a opção filtrar.

6.2.4 O sistema apresenta a quantidade a consumir e a quantidade consumida da data de consumo específica exibindo um gráfico com duas linhas em cada alimento da refeição consumida.

Fluxo Alternativo 4

No passo 9 do cenário principal, o usuário nutricionista preenche a data inicial, a data final e seleciona a opção exibir primeira e última refeição consumida somente.

9.1. O usuário nutricionista seleciona a opção filtrar.

9.2. O sistema apresenta a quantidade de refeições encontradas no período, apresenta a quantidade total de quilocalorias dos alimentos a consumir e a quantidade

total de quilocalorias dos alimentos consumidos no período através de um gráfico com duas linhas em cada data da refeição consumida, mas só exibe no eixo X a data da primeira e a última refeição.

Fluxo Alternativo 5

No passo 9 do cenário principal, o usuário nutricionista preenche a data inicial, a data final e seleciona a opção exibir o gráfico em barras.

9.3. O usuário nutricionista seleciona a opção filtrar.

9.4. O sistema apresenta a quantidade de refeições encontradas no período e o gráfico em duas barras, informando a quantidade de quilocalorias totais dos alimentos a consumir e a quantidade de quilocalorias totais dos alimentos consumidos em cada data da refeição consumida.

Fluxo Alternativo 6

No passo 9 do cenário principal, o usuário nutricionista preenche a data inicial, a data final e seleciona a opção exibir o gráfico em consumos.

9.5. O usuário nutricionista seleciona a opção filtrar.

9.6. O sistema apresenta a quantidade de refeições encontradas no período e a quantidade total a consumir e a quantidade total consumida dos alimentos, exibindo um gráfico de duas linhas em cada data da refeição consumida.

Fluxo Alternativo 7

No passo 10 do cenário principal, o usuário nutricionista informa um período que não possui registros ou possui somente uma refeição consumida.

10.1. O sistema apresenta a mensagem informando se não foram encontrados registros ou se somente uma refeição consumida foi encontrada.

10.2. Retorna ao passo 9 do cenário principal.

Fonte: elaborado pelo autor.

No quadro 30 é apresentado a descrição do caso de uso “notificar o paciente com as refeições cadastradas nos respectivos horários”.

Quadro 30 - UC022 - Notificar o paciente com as refeições cadastradas nos respectivos horários

UC022 – Notificar o paciente com as refeições cadastradas nos respectivos horários

Permite ao paciente ser notificado das refeições cadastradas pelo nutricionista em seus respectivos horários.

Ator: Usuário Paciente

Pré-condição: O usuário paciente deve estar cadastrado no sistema por um nutricionista.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma anamnese.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma dieta.

Pré-condição: Deve estar cadastrado no sistema pelo menos um cardápio.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma refeição.

Cenário Principal

1. O usuário paciente informa seu usuário e senha cadastrados pelo nutricionista na tela de acesso do sistema

2. O sistema faz a validação do cadastro do paciente.

3. O sistema busca as refeições do paciente cadastradas pelo nutricionista.

4. O sistema verifica se as refeições ainda não foram consumidas.

5. O sistema cria as notificações para serem exibidas no respectivo horário cadastrado.

6. O sistema notifica o paciente com o áudio padrão configurado no *smartphone*, com vibração e apresenta a notificação visual no topo da tela do dispositivo.

Fluxo Alternativo 1

No passo 2 do cenário principal, o sistema não encontra o cadastro do paciente

2.1. O sistema apresenta uma mensagem informando que o usuário não está cadastrado.

2.2. Retorna ao passo 1 do cenário principal.

No quadro 31 é apresentado a descrição do caso de uso “informar a quantidade de alimentos consumidos”.

Quadro 31 - UC023 - Informar a quantidade de alimentos consumidos

UC023 – Informar a quantidade de alimentos consumidos

Permite ao paciente informar a quantidade de alimentos consumidos das refeições cadastradas pelo nutricionista

Ator: Usuário Paciente

Ator: Webservice

Ator: Usuário Nutricionista

Pré-condição: O usuário paciente deve estar cadastrado no sistema por um nutricionista.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma anamnese.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma dieta.

Pré-condição: Deve estar cadastrado no sistema pelo menos um cardápio.

Pré-condição: Deve estar cadastrado no sistema pelo menos uma refeição.

Cenário Principal

1. O usuário paciente informa seu usuário e senha cadastrados pelo nutricionista na tela de acesso do sistema
2. O sistema faz a validação do cadastro do paciente.
3. O sistema busca as refeições do paciente cadastradas pelo nutricionista.
4. O sistema verifica se as refeições ainda não foram consumidas.
5. O sistema cria as notificações para serem exibidas no respectivo horário cadastrado.
6. O sistema notifica o paciente com o áudio padrão configurado no *smartphone*, com vibração e apresenta a notificação visual no topo da tela do dispositivo.
7. O paciente seleciona a notificação exibida.

8. O sistema apresenta a tela de cadastro de refeição consumida.

9. O paciente informa a quantidade consumida de cada alimento da lista de alimentos da refeição específica.

10. O sistema envia através do Webservice as informações cadastradas pelo paciente, que são armazenadas no banco de dados.

11. O nutricionista consegue na tela de consulta de consumos da refeição do paciente, conferir as informações referentes ao seu consumo.

Fluxo Alternativo 1

No passo 2 do cenário principal, o sistema não encontra o cadastro do paciente

2.1. O sistema apresenta uma mensagem informando que o usuário não está cadastrado.

2.2. Retorna ao passo 1 do cenário principal.

Fonte: elaborado pelo autor.

APÊNDICE C – Questionário aplicado

A figura 47 apresenta o questionário de avaliação do aplicativo realizado com uma professora do curso de Nutrição da FURB, uma aluna do terceiro semestre do mesmo curso, e outras sete pessoas.

Figura 47 - Questionário para avaliação da usabilidade do aplicativo Nutrisys

Questionário para avaliação da usabilidade do aplicativo Nutrisys

Este questionário serve como avaliação para o Trabalho de Conclusão de Curso: Aplicativo Android para compartilhamento de informações sobre saúde alimentar. Desenvolvido na aplicação Nutrisys.

*Obrigatório

1. Você conhece outro sistema móvel com este propósito? *

Sim

Não

2. Como você classifica a usabilidade deste aplicativo? *

Atende

Atende parcialmente

Não atende

3. O aplicativo pode contribuir para o acompanhamento nutricional do paciente? *

Atende

Atende parcialmente

Não atende

4. Você acredita que o aplicativo possa ser adotado no dia-a-dia? *

Sim

Não

5. (Opcional) Você teria alguma sugestão ou crítica ao aplicativo?

Sua resposta _____

Fonte: elaborado pelo autor.