

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA
CURSO DE ESPECIALIZAÇÃO EM TECNOLOGIA JAVA**

GILSO MANFREDI

APLICATIVO ANDROID PARA CONTROLE DE CALORIAS INGERIDAS

MONOGRAFIA DE ESPECIALIZAÇÃO

**PATO BRANCO
2015**

GILSO MANFREDI

APLICATIVO ANDROID PARA CONTROLE DE CALORIAS INGERIDAS

Monografia apresentada na disciplina de Metodologia da Pesquisa, do Curso de Especialização em Tecnologia Java, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco, como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. Robson Cris Brito

PATO BRANCO
2015

APLICATIVO ANDROID PARA CONTROLE DE CALORIAS INGERIDAS

Por

Gilso Manfredi

Esta monografia foi apresentada às 16h30 do dia 27 de outubro de 2015 como requisito parcial para a obtenção do título de ESPECIALISTA, no III Curso de Especialização em Tecnologia Java, da Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. O acadêmico foi arguido pela Banca Examinadora composto pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

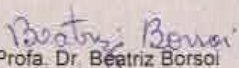

Prof. Msc. Robinson Cris Brito
Orientador

UTFPR – Câmpus Pato Branco


Prof. Msc. Rubia Eliza de Oliveira Schutz Ascari


Banca

UTFPR – Câmpus Pato Branco


Profa. Dr. Beatriz Borsol

Banca

UTFPR – Câmpus Pato Branco


Prof. Msc. Robinson Cris Brito
Coordenador do curso de Especialização

UTFPR – Câmpus Pato Branco

“Não vivemos pela quantidade imediata dos acertos, mas pela capacidade de entendermos aonde estamos errando, transformando as falhas em ajustes consistentes, feitos no tempo adequado para a manutenção das direções, métodos e pessoas.”

Sérgio Dal Sasso

RESUMO

MANFREDI, Gilso Aplicativo Android para controle de calorias ingeridas. 2015. 56 f. Monografia (Trabalho de especialização) – Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2015.

Este trabalho apresenta a modelagem e o desenvolvimento de duas aplicações criadas para auxiliar as pessoas no controle de calorias consumidas, sendo um Webservice REST e um aplicativo *mobile* Android. O objetivo desses sistemas é proporcionar a seus usuários um controle das calorias consumidas no dia-a-dia, por meio de lançamentos diários sobre os alimentos consumidos a cada refeição, apresentando informações como a quantidade de calorias consumidas por dia e por refeição e também a quantidade de calorias que o mesmo poderá ingerir diariamente com base na sua Taxa Metabólica Basal (TMB). O usuário também poderá visualizar o seu Índice de Massa Corporal (IMC). As funcionalidades implementadas nesses sistemas foram elaboradas em conversa com uma nutricionista que apontou as funcionalidades que seriam necessárias para um bom controle de calorias. A partir disso foram elaborados os requisitos que o sistema deve apresentar para contemplar as necessidades de controle das calorias. Foram implementados, então, os módulos de cadastro de usuários do sistema, categorias de alimentos e alimentos, além do módulo de lançamento das calorias consumidas por alimento a cada refeição do dia, e a visualização destes dados lançados em forma de listagens e gráficos.

Palavras-chave: Controle de Calorias. Índice de Massa Corporal. Taxa Metabólica Basal. Android. Computação em nuvem.

ABSTRACT

MANFREDI, Gilso. Calories Control Using an Android Application. 2015. 56 f. Monografia (Trabalho de especialização) – Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Câmpus Pato Branco. Pato Branco, 2015.

This work presents the modeling and the development of two applications created to help people in control of calories consumed, a REST Webservice and an Android mobile application. The purpose of these systems is provide its users a control of the calories consumed in the day-to-day, through journal entries about the foods consumed at each meal, with information such as the amount of calories consumed per day per meal and also the amount of calories that it may ingest daily based on your Basal Metabolic Rate (BMR). The user can also preview your Body Mass Index (BMI). The features implemented in these systems have been prepared in conversation with a nutritionist who pointed out the features that would be necessary for good control of calories. From this have been prepared the requirements that the system must present to contemplate the calories control needs. Were implemented, then the module registration of system users, registration of categories of foods and food, and the launch module of calories consumed for food every meal of the day, and visualization of data launched in the form of lists and charts.

Keywords: Calories control. Body Mass Index. Basal Metabolic Rate. Android. Cloud computing.

LISTA DE FIGURAS

FIGURA 1 - DIAGRAMA DE CASO DE USO.....	33
FIGURA 2 - DIAGRAMA DE CLASSE.....	34
FIGURA 3 - DIAGRAMA DE ENTIDADE E RELACIONAMENTO	35
FIGURA 4 – TELA DE <i>LOGIN</i>	36
FIGURA 5 – TELA DE INFORMAÇÕES CADASTRAIS	37
FIGURA 6 – TELA DE DADOS DO USUÁRIO.....	38
FIGURA 7 – GERENCIADOR DE CALORIAS	39
FIGURA 8 – TELA DE SELEÇÃO DE CATEGORIAS DE PRODUTOS	40
FIGURA 9 – TELA DE SELEÇÃO DE PRODUTOS.....	40
FIGURA 10 – TELA DE LANÇAMENTOS.....	41
FIGURA 11 – TELA DE CADASTRO DE NOVAS CATEGORIAS	42
FIGURA 12 – TELA DE CADASTRO DE NOVOS PRODUTOS	42
FIGURA 13 – LISTAGEM DE CALORIAS CONSUMIDAS POR REFEIÇÃO.....	43
FIGURA 14 – GRÁFICO DE CALORIAS CONSUMIDAS DIARIAMENTE POR REFEIÇÃO.....	44
FIGURA 15 – GRÁFICO DE CALORIAS CONSUMIDAS X NECESSÁRIAS POR REFEIÇÃO.....	44

LISTA DE QUADROS

QUADRO 1 - FÓRMULAS PARA CÁLCULO DA TAXA METABÓLICA BASAL.....	17
QUADRO 2 – FERRAMENTAS E TECNOLOGIAS UTILIZADAS	22
QUADRO 3 – REQUISITOS FUNCIONAIS.....	31
QUADRO 4 – REQUISITOS NÃO FUNCIONAIS	31
QUADRO 5 – CASOS DE USO.....	32

LISTA DE TABELAS

TABELA 1 – NÍVEIS DE ATIVIDADE FÍSICA	17
TABELA 2 – GRAU DE GORDURA SOBRE IMC PARA ADULTOS.....	18
TABELA 3 – GRAU DE GORDURA SOBRE IMC PARA IDOSOS	18
TABELA 4 – GRAU DE GORDURA SOBRE IMC PARA ADOLESCENTES DO SEXO MASCULINO	19
TABELA 5 – GRAU DE GORDURA SOBRE IMC PARA ADOLESCENTES DO SEXO FEMININO	20

LISTAGENS DE CÓDIGOS

LISTAGEM 1 – CLASSE GENÉRICA NO WEBSERVICE DE EXPOSIÇÃO DE SERVIÇOS REST	45
LISTAGEM 2 – CLASSE GENÉRICA NO ANDROID PARA CONSUMO DE SERVIÇOS REST	46
LISTAGEM 3 – CLASSE NO WEBSERVICE QUE EXPÕE O SERVIÇO REST	47
LISTAGEM 4 – CLASSE NO ANDROID QUE CONSOME O SERVIÇO REST	47
LISTAGEM 5 – CLASSE QUE REPRESENTA <i>THREAD</i> ASSÍNCRONA PARA CONSUMO DE SERVIÇOS REST.....	48
LISTAGEM 6 – TRATAMENTO DE EXCEÇÕES COM EXCEÇÕES GENÉRICAS	49
LISTAGEM 7 – CÁLCULO DA TAXA METABÓLICA BASAL	50
LISTAGEM 8 – CÁLCULO DO ÍNDICE DE MASSA CORPORAL.....	51

LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
BI	<i>Business Intelligence</i>
DAO	<i>Data Access Object</i>
DCU	Diagrama de Casos de Uso
DER	Diagrama de Entidades e Relacionamentos
IDE	<i>Integrated Development Environment</i>
IMC	Índice de Massa Corporal
JDK	<i>Java Development Kit</i>
POM	<i>Project Object Model</i>
REST	<i>Representational State Transfer</i>
SDK	<i>Software Development Kit</i>
SGBD	Sistema Gerenciador de Banco de Dados
SQL	<i>Structured Query Language</i>
UML	<i>Unified Modeling Language</i>
TMB	Taxa Metabólica Basal
TI	Tecnologia da Informação
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1 INTRODUÇÃO	12
1.1 CONSIDERAÇÕES INICIAIS	12
1.2 OBJETIVOS	13
1.2.1 Objetivo Geral	13
1.2.2 Objetivos Específicos	13
1.3 JUSTIFICATIVA	14
1.4 ESTRUTURA DO TRABALHO.....	14
2 REFERENCIAL TEÓRICO.....	16
2.1 Calorias	16
2.2 Taxa Metabólica Basal	16
2.3 Índice de Massa Corporal.....	17
2.4 Computação em nuvem	21
3 MATERIAIS E MÉTODO.....	22
3.1 MATERIAIS	22
3.1.1 Astah Community	23
3.1.2 DB Visual Architect Community Edition.....	23
3.1.3 Java.....	23
3.1.4 Android.....	24
3.1.5 Eclipse Luna.....	24
3.1.6 Android Studio	25
3.1.7 Apache Maven	25
3.1.8 Gradle.....	25
3.1.9 PostgreSQL.....	26
3.1.10 SQLite	26
3.1.11 Hibernate.....	26
3.1.12 Wildfly.....	27
3.2 MÉTODO.....	27
3.2.1 Análise.....	27
3.2.2 Projeto	27
3.2.2 Implementação.....	28
4 RESULTADOS.....	29
4.1 ESCOPO DO SISTEMA	29
4.2 MODELAGEM DO SISTEMA	30
4.2.1 Requisitos.....	30
4.2.2 Casos de Uso.....	31
4.2.3 Diagrama de Classes	34
4.2.4 Modelagem de Dados	35
4.3 APRESENTAÇÃO DO SISTEMA.....	36
4.4 IMPLEMENTAÇÃO DO SISTEMA	45
5 CONCLUSÃO	52
REFERÊNCIAS.....	53

1 INTRODUÇÃO

Este capítulo apresenta as considerações iniciais do trabalho, os seus objetivos e a justificativa. O capítulo é finalizado com a apresentação dos capítulos subsequentes.

1.1 CONSIDERAÇÕES INICIAIS

Controlar a quantidade de calorias consumidas diariamente é trabalhoso e custoso. Isso porque é comum as pessoas anotarem a quantidade de calorias consumidas em papéis ou em planilhas eletrônicas, que podem ser facilmente perdidas. Muitas vezes há o esquecimento de realizar essas anotações, fazendo com que seus dados fiquem incompletos, ou ainda, nem sempre que se ingere calorias, a pessoa tem acesso ao registro (está longe de um computador, por exemplo). Há, ainda, as pessoas que tem interesse em controlar as calorias consumidas, porém não conhecem a quantidade de calorias de cada alimento.

Para facilitar o controle de calorias consumidas seria útil que as pessoas pudessem anotar os dados das calorias consumidas em um local seguro, diretamente em seus aparelhos celulares, por exemplo, enquanto ainda consomem o alimento. Assim evitariam o esquecimento e manteriam um controle atualizado das calorias consumidas, além de manterem os dados para acessos posteriores.

Para que o controle de calorias possa ser efetivo, é necessário que a pessoa tenha acesso a um aplicativo *mobile* que mantenha suas informações para acesso pela Internet. E apresente informações relevantes, como quantidade de calorias necessárias para consumo diário, a quantidade já consumida e o Índice de Massa Corporal (IMC). Desta maneira o desenvolvimento de um sistema *mobile* é uma alternativa para o controle das calorias ingeridas.

Com esse sistema instalado em *smartphone* ou *tablet*, as pessoas poderão registrar várias informações referentes ao seu consumo de calorias, como, o alimento consumido, a quantidade de porções consumidas do mesmo e calcular a quantidade de calorias, por refeição e por dia, além de ter esses dados armazenados para consulta a qualquer momento.

Além disso, o usuário poderá consultar a sua Taxa Metabólica Basal (TMB), ou seja, a quantidade de calorias que poderá consumir durante o dia e o seu Índice de Massa Corporal (IMC), que indica se o peso da pessoa está adequado em relação a sua altura. Para facilitar a inserção dos dados, a interface do sistema será simples e de uso fácil.

Uma aplicação *mobile* é uma aplicação que possa ser utilizada a partir de aparelhos móveis como *smartphones* e *tablets*, que executem em um sistema operacional. Uma forma de implementar um aplicativo *mobile* é por meio da plataforma de programação Android. O Android é um sistema operacional que está presente em uma grande quantidade de aparelhos móveis. Segundo TECHCRUNCH (2015) já existem 1.4 bilhões de aparelhos Android ativos no mundo. Esse sistema operacional possui uma linguagem de programação específica para o desenvolvimento de aplicativos *mobile* que possam ser instalados nele.

Como forma de facilitar o controle de calorias ingeridas por uma pessoa utilizando a linguagem e programação Android, este trabalho apresenta o desenvolvimento de um aplicativo *mobile* que se utiliza desta tecnologia e se destina a pessoas que queiram controlar seu consumo de calorias diárias.

1.2 OBJETIVOS

A seguir são apresentados os objetivos da realização deste trabalho.

1.2.1 Objetivo Geral

Implementar um sistema Android e um Webservice Java para controle de calorias consumidas diariamente, bem como apresentar informações ao usuário com sua Taxa Metabólica Basal e seu Índice de Massa Corporal.

1.2.2 Objetivos Específicos

- Facilitar o controle e armazenamento de calorias consumidas pelas pessoas utilizando uma aplicação Android e conceitos de computação em nuvem.
- Prover informações úteis à saúde do usuário como a quantidade de calorias diárias a serem consumidas pela Taxa Metabólica Basal e o nível de gordura corporal por meio do Índice de Massa Corporal.
- Apresentar em forma de gráfico informações referentes aos lançamentos de calorias consumidas.
- Prover uma interface simplificada, visando facilitar o uso do aplicativo.

1.3 JUSTIFICATIVA

As pessoas que costumam controlar as calorias que consomem durante o dia necessitam de uma ferramenta que possibilite de maneira simples, rápida e fácil realizar os lançamentos das calorias consumidas. E, ainda que essa informação seja armazenada e facilmente recuperada para consultas.

Com o desenvolvimento desse aplicativo as pessoas interessadas terão vários benefícios, como o controle das calorias que consomem durante o dia, a apresentação da quantidade de calorias que necessitam consumir durante o dia, a informação sobre o seu IMC, além da possibilidade de visualização das informações lançadas em formas de listas e gráficos.

A opção de utilizar do sistema Android para implementação do sistema decorre de o mesmo ser uma linguagem de programação estável e amplamente utilizada, e por ser um dos sistemas operacionais móveis mais utilizados na atualidade.

A utilização da linguagem Java para desenvolvimento de um Webservice para a comunicação das informações do usuário com a nuvem, se dá pelo fato de também ser uma das linguagens mais utilizadas e pela gama de recursos disponíveis que facilitam o desenvolvimento deste tipo de aplicação.

O armazenamento dos dados em nuvem se faz necessário pela segurança e confiabilidade nos dados, assim se diminui o risco de perda de dados como poderia ocorrer se os mesmos estivessem diretamente no aparelho.

1.4 ESTRUTURA DO TRABALHO

Este texto está organizado em capítulos, dos quais este é o primeiro e apresenta a necessidade, a ideia e o contexto do sistema, incluindo os objetivos e a justificativa.

O Capítulo 2 contém o referencial teórico que fundamenta a proposta conceitual do sistema desenvolvido, está centrado em documentar o consumo de calorias diárias, o conceito e aplicação das fórmulas de TMB e IMC, conceito de computação em nuvem e no desenvolvimento de aplicações Android.

No Capítulo 3 estão os materiais e o método utilizados no desenvolvimento deste trabalho. Os materiais se referem às tecnologias e ferramentas utilizadas e o método contém as principais atividades realizadas para o desenvolvimento deste trabalho.

No Capítulo 4 se encontram os resultados do desenvolvimento da aplicação, com exemplos dos documentos de modelagem e de implementação das aplicações. A modelagem é apresentada por documentos de análise do projeto. Na implementação são apresentados as principais interfaces e os principais códigos das aplicações propostas.

No Capítulo 5 está a conclusão com as considerações finais.

2 REFERENCIAL TEÓRICO

O referencial teórico apresenta a fundamentação conceitual do trabalho, nele serão apresentados tópicos dos principais assuntos tratados neste documento, como as calorias, a Taxa Metabólica Basal, o Índice de Massa Corporal e computação em nuvem.

2.1 Calorias

A vida do ser humano depende de uma fonte de energia chamada caloria, as calorias estão contidas nos alimentos consumidos. Quando são ingeridas pelo organismo, elas são metabolizadas, gerando energia para a sobrevivência dos seres humanos.

Todos os alimentos possuem calorias, cada um com uma quantidade distinta. Os alimentos gordurosos (por exemplo, carnes gordas e laticínios) contêm mais calorias. Os carboidratos possuem as calorias mais fáceis de serem absorvidas e metabolizadas, sendo boas fontes de energia (MEDICALNEWTODAY, 2015).

O número de calorias que o corpo consome a cada dia varia de pessoa para pessoa. Em algumas tabelas de informações nutricionais dos alimentos baseiam-se em uma dieta de duas mil calorias, um valor médio do que as pessoas comem em um dia. Mas o corpo de uma pessoa pode precisar de mais ou menos calorias que esse valor. A altura, o peso, o sexo, a idade e quantidade de atividade física praticada diariamente afetam a quantidade de calorias que devem ser ingeridas diariamente (HSW, 2015).

O mínimo de calorias que uma pessoa deve consumir todos os dias é baseado na taxa metabólica basal. Comer menos que essa taxa significa que seu corpo não terá energia suficiente para manter suas funções gerais (EHOW, 2015).

2.2 Taxa Metabólica Basal

A Taxa de Metabolismo Basal é a quantidade mínima de energia (calorias) necessária para manter as funções vitais do organismo em repouso (MCARDLE; KATCH; KATCH, 1992).

Para ter conhecimento da necessidade calórica diária de uma pessoa foi desenvolvida uma fórmula com base na equação de Harris Benedict (CDOF, 2009), que calcula a taxa metabólica basal. Esta é calculada com base no peso, idade, altura e sexo. O cálculo realizado por esta equação pode ser observado no Quadro 1, onde P é o peso em quilogramas, A é a

altura em centímetros e I é a idade em anos.

Homens	$66,47 + (13,75 * P) + (5,00 * A) - (6,76 * I)$
Mulheres	$655,1 + (9,56 * P) + (1,85 * A) - (4,68 * I)$

Quadro 1 - Fórmulas para cálculo da Taxa Metabólica Basal

Fonte: CDOF (2009)

Para chegar ao consumo de gasto calórico total, deve-se somar a taxa metabólica basal com a taxa ocupacional, representada pelo nível de atividade física diária do indivíduo, que corresponde a um percentual sobre a Taxa Metabólica Basal, que representará a quantidade de calorias necessárias para manter as funções vitais do organismo com a prática dessas atividades. O cálculo realizado pode ser observado com a fórmula (TMB * Nível de Atividade Física), onde deve-se multiplicar pelo fator correspondente a taxa metabólica basal

Os níveis de atividade física podem ser visualizados na Tabela 1.

Tabela 1 – Níveis de Atividade Física

Situação	Nível de exercício	Fator de Multiplicação
Sedentário	Pouco ou nenhum exercício físico	1,2
Pouco ativo	De 1 a 3 dias por semana	1,375
Moderadamente ativo	De 3 a 5 dias por semana	1,65
Muito ativo	De 6 a 7 dias por semana	1,725
Extremamente Ativo	Diariamente uma ou mais vezes por dia	1,9

Fonte: CDOF (2009)

2.3 Índice de Massa Corporal

O Índice de Massa Corporal é um dos métodos utilizados internacionalmente para avaliar o estado nutricional de um indivíduo. Trata-se de uma fórmula matemática que usa duas informações, o peso (em quilos) e a altura (em metros) da pessoa. O cálculo é simples: basta dividir o peso pela altura ao quadrado ($IMC = \text{peso} / \text{altura}^2$). O resultado obtido serve de indicador sobre como está a manutenção do peso de cada pessoa (NESTLE, 2015a).

O IMC reflete a massa corporal como um todo, podendo ter seu valor alto em casos de acúmulo excessivo de gordura, mas também em casos de massa muscular e óssea elevada. Sendo assim, sempre se recomendam outros parâmetros antropométricos para confirmação de diagnóstico nutricional (HOSPITAL ALBERT EINSTEIN, 2015).

Os valores de IMC são independentes da idade e as mesmas para ambos os sexos. No entanto, IMC pode não coincidir com o mesmo grau de gordura em diferentes faixas etárias (WHO, 2006).

Embora o valor do IMC seja calculado da mesma forma para crianças, adolescentes, adultos e idosos, os critérios utilizados para interpretar o significado do valor do IMC para crianças e adolescentes são diferentes dos utilizados para adultos, e também diferentes dos utilizados em idosos. Para as crianças e adolescentes, os valores de IMC por idade e sexo-específicos são usados por duas razões, a alteração da quantidade de gordura corporal com a idade e a quantidade de gordura corporal que se difere entre meninos e meninas (PEDIATRIAONLINE, 2013).

A classificação diferenciada de IMC para crianças e adolescentes é utilizada na faixa etária de 5 a 18 anos, após os 18 anos é utilizada a classificação de adultos (WHO, 2006). Para idosos não existem diferenças por sexo, abrangendo uma classificação também diferenciada para faixa etária acima de 65 anos (HOSPITAL ALBERT EINSTEIN, 2015).

A seguir serão apresentadas as tabelas para resultado do IMC, estas estão separadas por faixas etárias conforme citado anteriormente, como adulto, idoso, criança/adolescente este último separado pelo sexo. As Tabelas 2, 3, 4 e 5 possuem as informações sobre as medições e a classificação do peso da pessoa com base no valor gerado pelo cálculo do IMC.

Tabela 2 – Grau de Gordura sobre IMC para Adultos

Abaixo de 18,5	Abaixo do peso
Entre 18,5 e 24,9	Peso adequado à altura
Entre 25 e 29,9	Sobrepeso
Acima de 30	Obesidade

Fonte: NESTLE (2015a).

Tabela 3 – Grau de Gordura sobre IMC para Idosos

Abaixo de 22	Abaixo do peso
Entre 22 e 27	Peso adequado
Acima de 27	Obesidade

Fonte: NESTLE (2015b, p.3).

Tabela 4 – Grau de Gordura sobre IMC para Adolescentes do sexo Masculino

Sexo	Masculino			
Idade	Abaixo do Peso	Normal	Sobrepeso	Obesidade
5 Anos	Abaixo de 14.1	Entre 14.1 e 16.6	Entre 16.6 e 18.3	Acima de 18.3
5,5 Anos	Abaixo de 14.1	Entre 14.1 e 16.7	Entre 16.7 e 18.4	Acima de 18.4
6 Anos	Abaixo de 14.1	Entre 14.1 e 16.8	Entre 16.8 e 18.5	Acima de 18.5
6,5 Anos	Abaixo de 14.1	Entre 14.1 e 16.9	Entre 16.9 e 18.7	Acima de 18.7
7 Anos	Abaixo de 14.2	Entre 14.2 e 17.0	Entre 17.0 e 19.0	Acima de 19.0
7,5 Anos	Abaixo de 14.3	Entre 14.3 e 17.2	Entre 17.2 e 19.3	Acima de 19.3
8 Anos	Abaixo de 14.4	Entre 14.4 e 17.4	Entre 17.4 e 19.7	Acima de 19.7
8,5 Anos	Abaixo de 14.5	Entre 14.5 e 17.7	Entre 17.7 e 20.1	Acima de 20.1
9 Anos	Abaixo de 14.6	Entre 14.6 e 17.9	Entre 17.9 e 20.5	Acima de 20.5
9,5 Anos	Abaixo de 14.8	Entre 14.8 e 18.2	Entre 18.2 e 20.9	Acima de 20.9
10 Anos	Abaixo de 14.9	Entre 14.9 e 18.5	Entre 18.5 e 21.4	Acima de 21.4
10,5 Anos	Abaixo de 15.1	Entre 15.1 e 18.8	Entre 18.8 e 21.9	Acima de 21.9
11 Anos	Abaixo de 15.3	Entre 15.3 e 19.2	Entre 19.2 e 22.5	Acima de 22.5
11,5 Anos	Abaixo de 15.5	Entre 15.5 e 19.5	Entre 19.5 e 23.0	Acima de 23.0
12 Anos	Abaixo de 15.8	Entre 15.8 e 19.9	Entre 19.9 e 23.6	Acima de 23.6
12,5 Anos	Abaixo de 16.1	Entre 16.1 e 20.4	Entre 20.4 e 24.2	Acima de 24.2
13 Anos	Abaixo de 16.4	Entre 16.4 e 20.8	Entre 20.8 e 24.8	Acima de 24.8
13,5 Anos	Abaixo de 16.7	Entre 16.7 e 21.3	Entre 21.3 e 25.3	Acima de 25.3
14 Anos	Abaixo de 17.0	Entre 17.0 e 21.8	Entre 21.8 e 25.9	Acima de 25.9
14,5 Anos	Abaixo de 17.3	Entre 17.3 e 22.2	Entre 22.2 e 26.5	Acima de 26.5
15 Anos	Abaixo de 17.6	Entre 17.6 e 22.7	Entre 22.7 e 27.0	Acima de 27.0
15,5 Anos	Abaixo de 18.0	Entre 18.0 e 23.1	Entre 23.1 e 27.4	Acima de 27.4
16 Anos	Abaixo de 18.2	Entre 18.2 e 23.5	Entre 23.5 e 27.9	Acima de 27.9
16,5 Anos	Abaixo de 18.5	Entre 18.5 e 23.9	Entre 23.9 e 28.3	Acima de 28.3
17 Anos	Abaixo de 18.8	Entre 18.8 e 24.3	Entre 24.3 e 28.6	Acima de 28.6
17,5 Anos	Abaixo de 19.0	Entre 19.0 e 24.6	Entre 24.6 e 29.0	Acima de 29.0
18 Anos	Abaixo de 19.2	Entre 19.2 e 24.9	Entre 24.9 e 29.2	Acima de 29.2
18,5 Anos	Abaixo de 19.4	Entre 19.4 e 25.2	Entre 25.2 e 29.5	Acima de 29.5

Fonte: WHO Multicentre Growth Reference Study Group (2006).

Tabela 5 – Grau de Gordura sobre IMC para Adolescentes do sexo Feminino

Sexo	Feminino			
Idade	Abaixo do Peso	Normal	Sobrepeso	Obesidade
5 Anos	Abaixo de 13.9	Entre 13.9 e 16.9	Entre 16.9 e 18.9	Acima de 18.9
5,5 Anos	Abaixo de 13.9	Entre 13.9 e 16.9	Entre 16.9 e 19.0	Acima de 19.0
6 Anos	Abaixo de 13.9	Entre 13.9 e 17.0	Entre 17.0 e 19.2	Acima de 19.2
6,5 Anos	Abaixo de 13.9	Entre 13.9 e 17.1	Entre 17.1 e 19.5	Acima de 19.5
7 Anos	Abaixo de 13.9	Entre 13.9 e 17.3	Entre 17.3 e 19.8	Acima de 19.8
7,5 Anos	Abaixo de 14.0	Entre 14.0 e 17.5	Entre 17.5 e 20.1	Acima de 20.1
8 Anos	Abaixo de 14.1	Entre 14.1 e 17.7	Entre 17.7 e 20.6	Acima de 20.6
8,5 Anos	Abaixo de 14.3	Entre 14.3 e 18.0	Entre 18.0 e 21.0	Acima de 21.0
9 Anos	Abaixo de 14.4	Entre 14.4 e 18.3	Entre 18.3 e 21.5	Acima de 21.5
9,5 Anos	Abaixo de 14.6	Entre 14.6 e 18.7	Entre 18.7 e 22.0	Acima de 22.0
10 Anos	Abaixo de 14.8	Entre 14.8 e 19.0	Entre 19.0 e 22.6	Acima de 22.6
10,5 Anos	Abaixo de 15.1	Entre 15.1 e 19.4	Entre 19.4 e 23.1	Acima de 23.1
11 Anos	Abaixo de 15.3	Entre 15.3 e 19.9	Entre 19.9 e 23.7	Acima de 23.7
11,5 Anos	Abaixo de 15.6	Entre 15.6 e 20.3	Entre 20.3 e 24.3	Acima de 24.3
12 Anos	Abaixo de 16.0	Entre 16.0 e 20.8	Entre 20.8 e 25.0	Acima de 25.0
12,5 Anos	Abaixo de 16.3	Entre 16.3 e 21.3	Entre 21.3 e 25.6	Acima de 25.6
13 Anos	Abaixo de 16.6	Entre 16.6 e 21.8	Entre 21.8 e 26.2	Acima de 26.2
13,5 Anos	Abaixo de 16.9	Entre 16.9 e 22.3	Entre 22.3 e 26.8	Acima de 26.8
14 Anos	Abaixo de 17.2	Entre 17.2 e 22.7	Entre 22.7 e 27.3	Acima de 27.3
14,5 Anos	Abaixo de 17.5	Entre 17.5 e 23.1	Entre 23.1 e 27.8	Acima de 27.8
15 Anos	Abaixo de 17.8	Entre 17.8 e 23.5	Entre 23.5 e 28.2	Acima de 28.2
15,5 Anos	Abaixo de 18.0	Entre 18.0 e 23.8	Entre 23.8 e 28.6	Acima de 28.6
16 Anos	Abaixo de 18.2	Entre 18.2 e 24.1	Entre 24.1 e 28.9	Acima de 28.9
16,5 Anos	Abaixo de 18.3	Entre 18.3 e 24.3	Entre 24.3 e 29.1	Acima de 29.1
17 Anos	Abaixo de 18.4	Entre 18.4 e 24.5	Entre 24.5 e 29.3	Acima de 29.3
17,5 Anos	Abaixo de 18.5	Entre 18.5 e 24.6	Entre 24.6 e 29.4	Acima de 29.4
18 Anos	Abaixo de 18.6	Entre 18.6 e 24.8	Entre 24.8 e 29.5	Acima de 29.5
18,5 Anos	Abaixo de 18.6	Entre 18.6 e 24.9	Entre 24.9 e 29.6	Acima de 29.6

Fonte: WHO Multicentre Growth Reference Study Group (2006).

2.4 Computação em nuvem

Segundo Taurion (2009), o termo computação em nuvem surgiu em 2006 em uma palestra de Eric Schmidt, da Google, sobre como sua empresa gerenciava seus data centers.

Computação em nuvem é uma das palavras chaves da indústria de Tecnologia da Informação (TI). A nuvem é uma representação para a Internet, baseada em uma abstração que oculta a complexidade de infraestrutura. Cada parte desta é provida por meio de serviços, e estes são normalmente alocados em data-centers, utilizando hardware compartilhado para computação e armazenamento (SOUSA, MOUREIRA e MACHADO, 2010).

Após estudo foram verificadas várias definições distintas para o conceito de computação na nuvem. Pode-se considerar que nuvens são grandes repositórios de recursos virtualizados, tais como hardware, plataformas de desenvolvimento e software, que são facilmente acessíveis. Estes recursos podem ser configurados dinamicamente de modo a ajustar-se a diferentes cargas de trabalho com a intenção de otimizar sua utilização (VAQUERO et al., 2008).

Para BUYYA, YEO e VNUGOPAL (2008), uma nuvem é um tipo de sistema paralelo e distribuído que consiste de uma coleção de computadores virtualizados e interconectados que são provisionados de forma dinâmica e apresentados como um ou mais recursos computacionais unificados. Estes recursos são disponibilizados e controlados por meio de acordos relacionados aos serviços que são estabelecidos entre um prestador e um consumidor sendo definidos a partir de negociações entre as partes.

3 MATERIAIS E MÉTODO

Este capítulo apresenta os materiais e os métodos utilizados para a realização deste trabalho. Os materiais se referem às tecnologias como linguagens e ferramentas para a modelagem e a implementação do sistema. O método contém as etapas com os principais procedimentos utilizados para o desenvolvimento do sistema, abrangendo do levantamento dos requisitos à implementação das aplicações.

3.1 MATERIAIS

O Quadro 2 apresenta as ferramentas e as tecnologias utilizadas para a modelagem e a implementação do sistema.

Ferramenta / Tecnologia	Versão	Referência	Finalidade	Utilização
Astah Community	7.0	http://astah.net/editions/community	Modelagem UML	WebService
DB Visual Architect Community Edition	1.0.7	http://www.sqlpower.ca	Modelagem do banco de dados	WebService
Java Platform (JDK)	8u45	http://www.oracle.com/	Linguagem de Programação	WebService
Android (SDK)	API 22	https://developer.android.com	Linguagem de Programação <i>Mobile</i>	Aplicativo Mobile
Eclipse Luna	4.4.2	https://www.eclipse.org/	Ambiente de desenvolvimento Webservice	WebService
Android Studio	1.2.2	https://developer.android.com	Ambiente de desenvolvimento <i>Mobile</i>	Aplicativo Mobile
Apache Maven	3.3.3	https://maven.apache.org/	Gerenciamento de dependências	WebService
Gradle	2.7	http://gradle.org/	Gerenciamento de dependências	Aplicativo Mobile
PostgreSQL	9.4	http://www.postgresql.org/	Banco de dados	WebService
SQLite	3.8.6	http://www.sqlite.org	Banco de dados	Aplicativo Mobile
Hibernate	4.3.5	http://hibernate.org/	Mapeamento Objeto Relacional	WebService
Wildfly	8.2	http://wildfly.org/	Servidor de aplicação	WebService

Quadro 2 – Ferramentas e tecnologias utilizadas

3.1.1 Astah Community

Para a diagramação UML (*Unified Modeling Language*) dos aplicativos foi utilizada a ferramenta Astah Community. Esta ferramenta possibilita a elaboração de diversos diagramas como os utilizados no sistema.

Astah Community é uma ferramenta para modelagem de sistemas com suporte para a *Unified Modeling Language 2* (UML) (ASTAH, 2015a). Além da modelagem de diagramas básicos como diagrama de classes, caso de uso, sequência, atividade, comunicação entre outros, a ferramenta permite a impressão e a exportação das imagens dos diagramas.

O Astah Community é uma ferramenta gratuita para o usuário final, possuindo uma versão paga com mais alguns recursos, como a criação de diagramas a partir de códigos das linguagens Java, C e C++, e também a exportação de código a partir dos diagramas para esta linguagem (ASTAH, 2015b).

3.1.2 DB Visual Architect Community Edition

Para a modelagem de banco de dados foi utilizada a ferramenta *Power Architect*. De acordo com a *SQL Power Software* (2015a), esta ferramenta conecta-se a banco de dados de múltiplas fontes ao mesmo tempo, além disso, lembra a origem de cada coluna e também permite realizar engenharia reversa e consultas SQL (*Structured Query Language*).

Fundada em 1988 o grupo SQL Power é uma grande organização especializada em aplicações de BI (*Business Intelligence*), mineração de dados e modelagem de bancos de dados (SQL POWER SOFTWARE, 2015b).

3.1.3 Java

Para o desenvolvimento do Webservice será utilizada a linguagem de programação Java. Java é uma linguagem de programação e plataforma computacional lançada pela Sun Microsystems em 1995. O Java é rápido, seguro e confiável. De laptops a datacenters, consoles de games a supercomputadores científicos, telefones celulares à Internet, o Java está em todos os lugares (JAVA, 2015a).

Inicialmente chamada de Oak (do inglês árvore de carvalho), em 1995 foi rebatizada como Java e nesse ano lançada pela Sun Microsystems. É uma tecnologia que possibilita a criação de aplicativos de alta qualidade (JAVA, 2015b). Em 2009 a Oracle comprou a Sun Microsystems e hoje possui os direitos sobre o Java, bem como realiza a manutenção e implementação de novos recursos.

3.1.4 Android

Para o desenvolvimento do aplicativo *mobile* será utilizada a plataforma Android. Android é um conjunto de softwares para dispositivos móveis que inclui um sistema operacional, aplicações e *middleware* (sistemas de software que comunicam as aplicações com o sistema operacional). O Android foi adquirido pela Google em julho de 2005, e é controlado por ela até os dias atuais (SUPERINTERESANTE, 2015).

Para o desenvolvimento é utilizado o Android SDK que é o responsável por fornecer ferramentas e APIs (*Application Programming Interface*) necessárias para desenvolvimento na plataforma Android. Através da sua utilização é possível a criação de aplicativos inovadores e ricos em design e componentes (ANDROID, 2015).

O Android SDK oferece aos desenvolvedores de software móvel o necessário para a construção de aplicações, bem como um dispositivo totalmente funcional de emulador para testá-los quando o aplicativo estiver terminado. Android SDK fornece um ambiente de desenvolvimento rápido e confiável (TECHTUDO 2015).

3.1.5 Eclipse Luna

Para o desenvolvimento do Webservice será utilizada a ferramenta Eclipse Luna. Eclipse Luna é uma ferramenta *open source*, onde os projetos estão focados na construção de uma plataforma de desenvolvimento aberta composta por extensíveis frameworks, ferramentas e tempos de execução para a construção, implantação e gerenciamento do software (ECLIPSE, 2015).

Criado originalmente pela IBM em novembro de 2001 o projeto Eclipse foi apoiado por alguns fornecedores de software. Os principais membros que formaram a organização Eclipse foram: Rational Software, IBM, Borland, Red Hat, SuSE entre outros. A Eclipse Foundation teve origem em janeiro de 2004, sendo uma corporação sem fins lucrativos para administrar a comunidade Eclipse (ECLIPSE, 2015).

3.1.6 Android Studio

Para o desenvolvimento da aplicação *mobile* foi utilizada a ferramenta Android Studio. Android Studio é a IDE (*Integrated Development Environment*) oficial para desenvolvimento de aplicativos Android, ela é baseada na IDE IntelliJ IDEA (ANDROID STUDIO, 2015).

Segundo INTELLIJ IDEA (2015) esta IDE fornece soluções rápidas e práticas para o desenvolvimento de aplicações, facilitando a vida do usuário com sugestões de melhorias de código e sugestões de plugins de desenvolvimento que auxiliarão no desenvolvimento de suas aplicações.

A IDE IntelliJ IDEA foi criada em 2001 e mantida até hoje pela JetBrains, esta fornece aos desenvolvedores ferramentas inteligentes profissionais para ajudá-los a escrever código de qualidade, limpo e oferecer uma solução rápida (JETBRAINS, 2015).

O desenvolvimento de software profissional tornou-se uma experiência mais agradável e produtiva (JETBRAINS, 2015)

3.1.7 Apache Maven

O Apache Maven é uma ferramenta de gerenciamento de projetos de software (MAVEN, 2015). O Maven faz gerencia de dependências e gerencia o ciclo de vida do projeto e as etapas para construção do mesmo. O Maven utiliza um arquivo *Extensible Markup Language* (XML) no formato projeto modelo de objeto (*Project Object Model* (POM)) para descrever o projeto de software, suas dependências, seus módulos e componentes externos, a ordem de compilação, diretórios e plug-ins necessários.

3.1.8 Gradle

O Gradle torna possível o impossível, de maneira fácil e elegante (GRADLE, 2015). O Gradle é um sistema avançado de automatização de builds que une o melhor da flexibilidade do Ant com o gerenciamento de dependências e as convenções do Maven.

Os arquivos de build do Gradle são scripts escritos na linguagem Groovy, ao contrário dos formatos de construção do Ant e Maven que usam arquivos XML para a configuração. Por serem baseados em scripts, os arquivos do Gradle permitem que sejam realizadas tarefas de programação em seu arquivo de configuração. O Gradle ainda conta com um sistema de *plugins* que adicionam funcionalidades extras ao seu *core*.

3.1.9 PostgreSQL

Para o armazenamento dos dados será utilizada a ferramenta de banco de dados PostgreSQL. PostgreSQL é um sistema de banco de dados *open source* objeto-relacional, é um Sistema Gerenciador de Banco de Dados (SGBD) multiplataforma, altamente confiável e estável, suporta um grande número de dados, é altamente escalável tanto na quantidade de dados gerenciados como no número de usuários simultâneos que podem ser alocados (POSTGRESQL, 2015).

3.1.10 SQLite

Para o armazenamento dos dados do sistema *mobile* será utilizada a ferramenta de banco de dados SQLite. SQLite é uma biblioteca em linguagem C que implementa um banco de dados transacional SQL embutido, de código aberto, livre para uso para qualquer finalidade, comercial ou privada (SQLITE, 2015).

Diferentemente da maioria dos outros bancos de dados, SQLite não tem um servidor separado. SQLite lê e escreve diretamente em arquivos no disco. Um completo banco de dados SQL com múltiplas tabelas, índices, triggers e views, está contido em um arquivo único no disco (SQLITE, 2015).

3.1.11 Hibernate

O Hibernate é um *framework* para realizar o mapeamento objeto-relacional que facilita o mapeamento dos atributos entre um banco de dados relacional e o modelo de objetos das classes de uma aplicação (HIBERNATE, 2015).

O objetivo do Hibernate é diminuir a complexidade das aplicações orientadas a objeto, com banco de dados relacional, transformando os dados do banco de dados em objetos e os objetos em registros no modelo objeto-relacional.

Sua principal característica são as operações de recuperação e persistência dos dados são realizadas por métodos do Hibernate, sendo assim, o programador não precisa se preocupar com instruções SQL como selects, join, insert, delete e etc., sendo o framework capaz até de resolver as peculiaridades que cada SGDB impõe.

3.1.12 Wildfly

O WildFly (WILDFLY, 2015), anteriormente conhecido como JBoss Application Server, ou simplesmente JBoss, é um servidor de aplicativos da JBoss, agora desenvolvido pela Red Hat.

O WildFly é escrito em Java, e implementa a especificação Java Enterprise Edition (Java EE). Por ser escrito em Java possui compatibilidade com múltiplas plataformas.

3.2 MÉTODO

No método utilizado para o desenvolvimento do sistema são consideradas as três fases que são análise, projeto e implementação, propostas por (RUMBAUGH, *et al.*, 1997).

3.2.1 Análise

A análise do sistema foi elaborada juntamente com a primeira etapa do trabalho, onde ocorreu o levantamento bibliográfico, incluindo os conceitos descritos no Capítulo 2 e as tecnologias descritas na Seção 3.1 do Capítulo 3.

3.2.2 Projeto

Nesta etapa foram definidas todas as tarefas a serem realizadas antes da implementação do sistema. Sendo:

- Definir requisitos funcionais e não funcionais;
- Criar diagrama de casos de uso;
- Criar diagrama de classes com seus atributos e métodos;
- Criar diagrama de entidade e relacionamento (DER) com as definições de tabelas, atributos, *constraints* e restrições;

- Gerar scripts para a criação do banco de dados;
- Preparar ambiente para desenvolvimento, como banco de dados e as IDEs utilizadas.

3.2.2 Implementação

A implementação do sistema Webservice foi realizada utilizando a IDE Eclipse Luna, com a linguagem Java, banco de dados PostgreSQL e gerenciamento de dependências com o apache Maven.

A implementação do sistema mobile foi realizada utilizando a IDE Android Studio, com linguagem Android, Banco de dados SQLite.

4 RESULTADOS

Este capítulo apresenta os resultados da implementação do sistema de gerenciamento de calorias utilizando um Webservice e aplicativo Android. Nele são apresentados o escopo, a modelagem, a apresentação e por fim a implementação do sistema.

4.1 ESCOPO DO SISTEMA

O sistema de controle de calorias tem como objetivo controlar as calorias consumidas por seus usuários, bem como apresentar informações como o IMC, a Taxa Metabólica Basal e informações sobre o seu consumo de calorias a cada refeição.

O usuário poderá lançar a cada refeição uma lista de alimentos consumidos e a quantidade de cada um deles. Cada alimento cadastrado possui uma quantidade de calorias, com base no cálculo da quantidade de calorias multiplicada pela quantidade consumida, o sistema irá apresentar ao usuário a quantidade de calorias que o mesmo consumiu.

Ainda é possível cadastrar novos alimentos caso não encontre um alimento específico que esteja procurando. Para o cadastro basta informar a descrição, a quantidade em gramas ou em unidades do alimento para cada porção e o valor calórico.

O cadastro de alimentos ficará armazenado na nuvem, assim quando um usuário efetuar o cadastro todos os demais poderão utilizá-lo.

O sistema apresentará também informações relevantes ao usuário, como o Índice de Massa Corporal e a Taxa Metabólica Basal. No que diz respeito ao IMC, será apresentado o valor e como o mesmo é classificado (ex. abaixo do peso, peso ideal, sobrepeso ou obeso).

A TMB representa a quantidade de calorias necessárias para a sobrevivência de uma pessoa, além dela será apresentada a quantidade de calorias diárias que o usuário deve consumir, usando a TMB e o nível de atividade física do usuário.

Por fim o sistema ainda apresentará em forma de relatórios e gráficos a quantidade de calorias consumidas pelo usuário por refeição e por dia, assim o mesmo poderá ter um controle das calorias consumidas.

Para que tudo isso seja possível foram implementados dois sistemas, um Webservice e uma aplicação Android. O sistema Android fará toda a interação com o usuário e apresentará as informações, já o webservice será o responsável pelo controle das informações, como o tratamento dos dados, a gravação e pesquisa e os cálculos citados de TMB e IMC.

4.2 MODELAGEM DO SISTEMA

Para a modelagem do sistema foram levantados os requisitos funcionais e não funcionais, diagrama de caso de uso, diagrama de classe e diagrama de entidade e relacionamento, apresentados na sequência.

4.2.1 Requisitos

Segundo Bezerra (2007, p.26), “Nesta atividade, o foco de interesse é tentar construir uma estratégia de solução sem se preocupar com a maneira como essa estratégia será realizada”.

Requisitos são as características que um sistema a ser desenvolvido deve apresentar, são propriedades ou comportamentos desejáveis de um aplicativo. Os requisitos são divididos em funcionais e não funcionais.

Os requisitos funcionais são definidos pelas funções ou serviços que um software deve possuir. Ex.: cadastros, geração de relatórios e controles. Os requisitos não funcionais não se referem a funcionalidades dos sistemas, mas estão relacionados a essas funcionalidades do software. Ex.: confiabilidade, rapidez, segurança.

O levantamento de requisitos foi feito a partir de uma conversa com uma nutricionista, onde a mesma descrevia as funcionalidades que ela e seus clientes buscam em um aplicativo de gerenciamento de calorias.

No Quadro 3 são descritos os requisitos funcionais.

Código	Nome	Descrição
RF-01	Cadastrar usuário	Permite cadastro dos usuários do sistema.
RF-02	Cadastrar produtos	Permite cadastro dos produtos a serem utilizados nos lançamentos.
RF-03	Cadastrar categorias de produtos	Permite cadastro das categorias de produtos para melhor organização na pesquisa de produtos.
RF-04	Lançar consumo de calorias nas refeições	Permite lançar os produtos consumidos em cada refeição.

RF-05	Visualizar os dados lançados	Permite visualizar a quantidade de calorias consumidas em cada refeição.
RF-06	Obter informações em forma de gráficos	Permite visualizar em forma de gráficos a quantidade de calorias consumidas em cada refeição.
RF-07	Calcular o IMC do usuário	Possibilita a visualização da informação de IMC do usuário.
RF-08	Calcular a Taxa Metabólica Basal do usuário	Possibilita a visualização da quantidade de calorias necessárias diárias para o usuário.

Quadro 3 – Requisitos funcionais

No Quadro 4 são descritos os requisitos não funcionais.

Código	Nome	Descrição
RNF-01	Acesso ao sistema	Permite acesso ao sistema após informar <i>login</i> e senha válidos.
RNF-02	Agilidade	O sistema deverá apresentar bom desempenho.
RNF-03	Usabilidade	O sistema deverá ser de fácil entendimento para o usuário.

Quadro 4 – Requisitos não funcionais

4.2.2 Casos de Uso

“Um caso de uso é a especificação de uma sequência completa de interações entre um sistema e um ou mais agentes externos a esse sistema” (BEZERRA 2007, p. 54).

A partir de um caso de uso é possível verificar quais são os usuários que irão interagir com o sistema e quais as funcionalidades e resultados fornecidos pelo mesmo.

Diagrama de caso de uso é uma visão externa do sistema, é uma representação dos atores, casos de uso e relacionamento entre os atores e casos de uso.

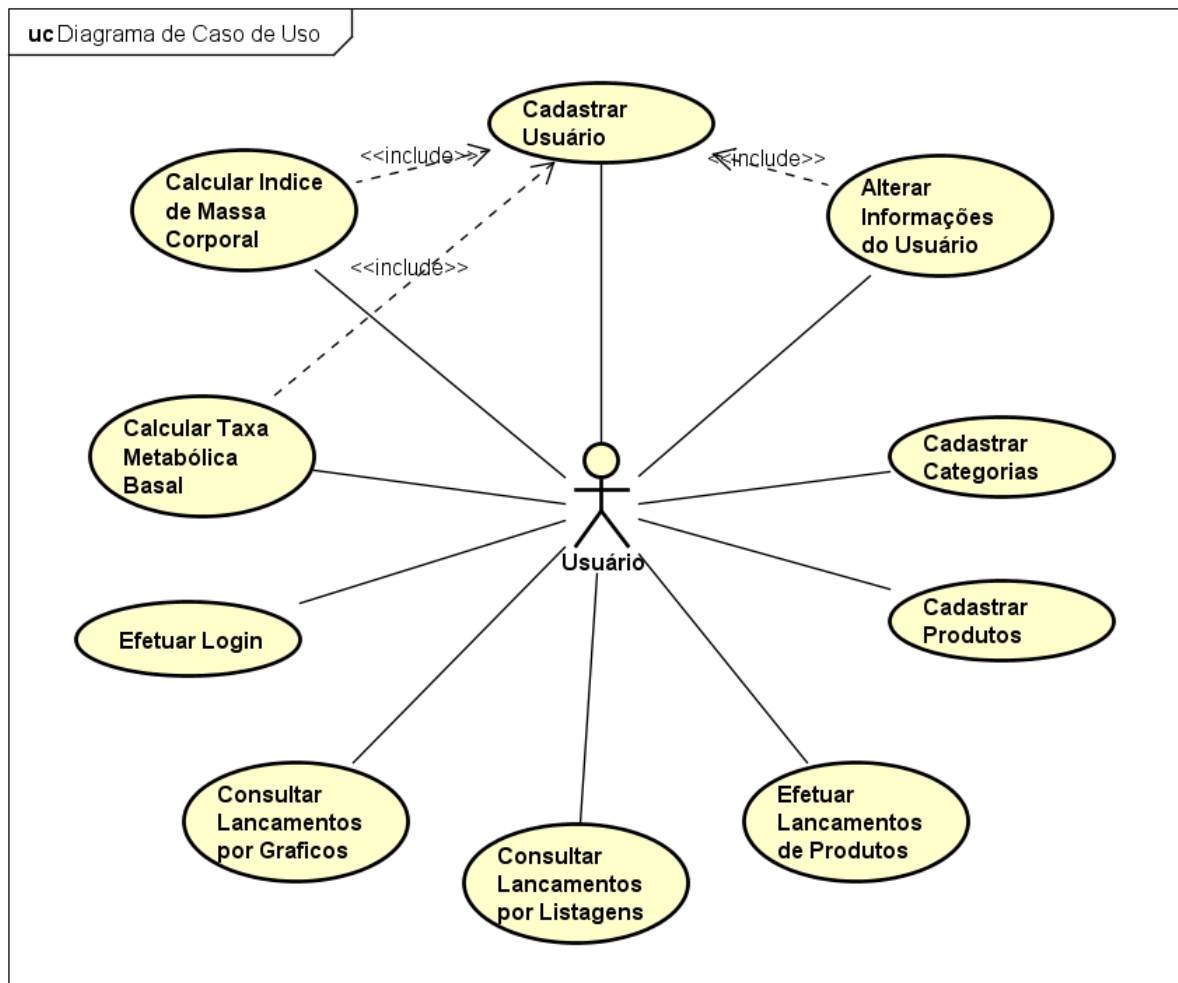
“O DCU (Diagrama de Casos de Uso) tem o objetivo de ilustrar em um nível alto de abstração quais elementos externos interagem com que funcionalidades do sistema” (BEZERRA 2007, p.70).

No Quadro 5 são apresentados os casos de uso.

Nome	Ator	Descrição
Cadastrar Usuário	Usuário	Processo de cadastro de usuários.
Efetuar <i>Login</i>	Usuário	Processo de acesso ao sistema.
Alterar Informações do Usuário	Usuário	Processo de alteração dos dados dos usuários.
Calcular Índice de Massa Corporal	Usuário	Processo de cálculo do valor de IMC do usuário.
Calcular Taxa Metabólica Basal	Usuário	Processo de cálculo do valor de TMB do usuário.
Cadastrar Categorias	Usuário	Processo de cadastro de categorias de produtos.
Cadastrar Produtos	Usuário	Processo de cadastro de produtos.
Efetuar Lançamentos de Produtos	Usuário	Processo de lançamento de produtos consumidos.
Consultar Lançamentos de Produtos por Listagens	Usuário	Processo de consulta de lançamentos por meio de listagens.
Consultar lançamentos de produtos por gráficos	Usuário	Processo de consulta de lançamentos por meio de gráficos.

Quadro 5 – Casos de Uso

A Figura 1 ilustra o diagrama de caso de uso.



powered by Astah

Figura 1 - Diagrama de Caso de Uso

Esse diagrama representa as funcionalidades necessárias para o bom funcionamento dos sistemas, nele estão listados todos os recursos disponíveis ao usuário, desde o seu cadastro, cadastro de produtos e lançamentos até as consultas disponíveis.

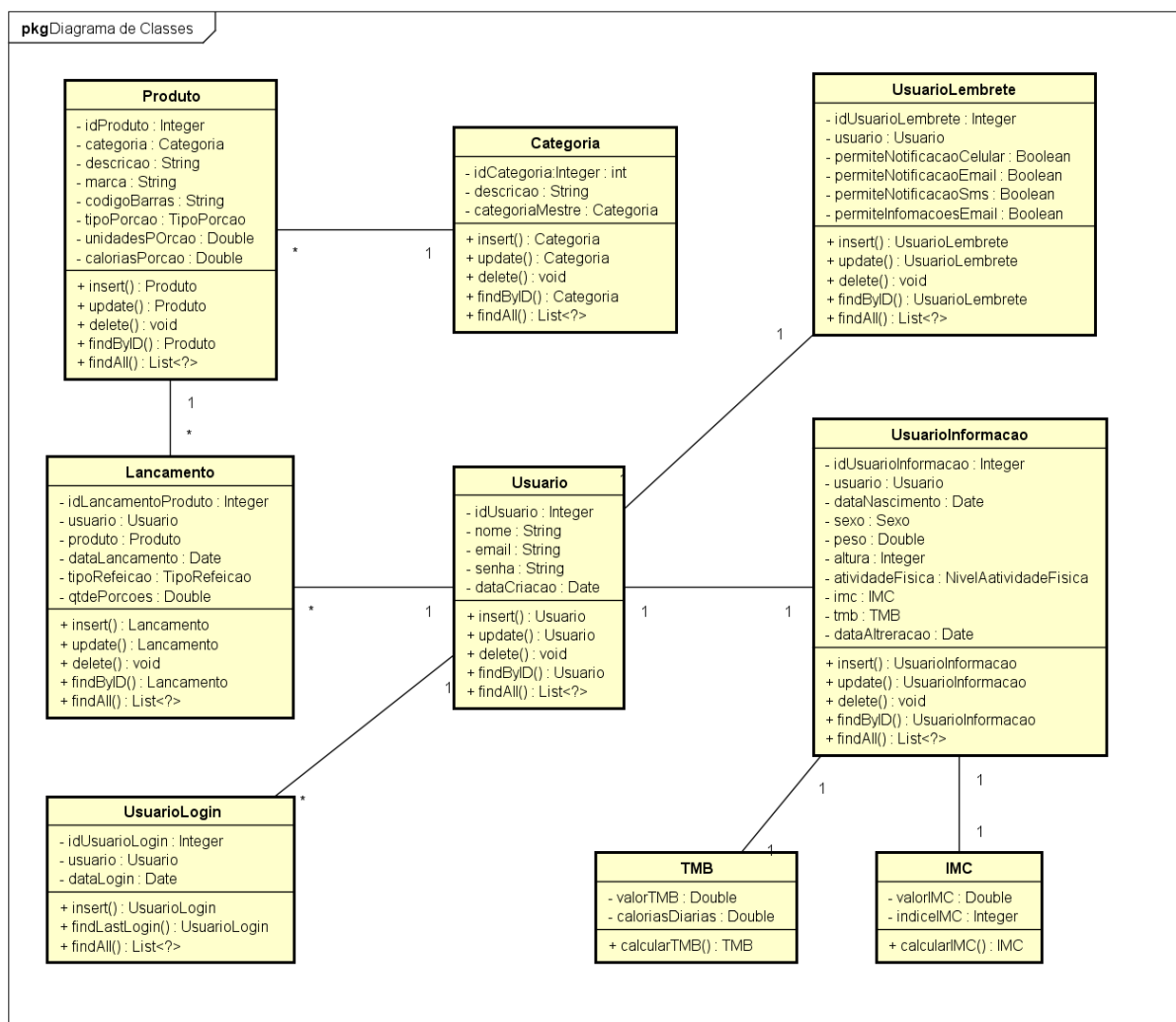
Cada caso de uso representa uma funcionalidade do sistema como pode-se ver no caso de uso Cadastrar Usuário, nele o próprio usuário pode efetuar o seu cadastro informando os dados necessários. O caso de uso de Efetuar *Login* representa a funcionalidade de acesso ao sistema. Outro exemplo é o caso de uso Consultar Lançamentos por Gráficos onde o usuário poderá visualizar em forma de gráficos os dados de seus lançamentos no sistema.

4.2.3 Diagrama de Classes

De acordo com Fowler (*apud* QUADROS 2002, p.180) “Os diagramas de classe descrevem os tipos de objetos do sistema e os vários tipos de relacionamento estáticos que existem entre eles”.

Este é um dos diagramas mais utilizados no desenvolvimento de um sistema, por meio dele podem ser encontradas diversas informações, como: classes, atributos e métodos das classes, além da integração entre as mesmas.

A Figura 2 ilustra o diagrama de classe.



powered by Astah

Figura 2 - Diagrama de Classe

Este diagrama de classes representa as classes de entidade do sistema, são elas as que dão suporte às principais classes de Controllers e de DAOs do sistema, também os objetos gerados por elas serão os dados persistidos no banco de dados.

Nele também são representados os relacionamentos entre classes que reflete o relacionamento de entidades do banco de dados.

4.2.4 Modelagem de Dados

Modelagem de dados corresponde a criação de um modelo com o objetivo de explicar características sobre o comportamento e funcionamento de um sistema a ser criado, facilitando assim sua compreensão e desenvolvimento.

O diagrama de entidade e relacionamento é de extrema importância quando se quer conhecer a estrutura do banco de dados por um diagrama. Com ele é possível visualizar as tabelas e seus atributos, e a relação entre cada uma no banco de dados.

“Um Diagrama de Entidades e Relacionamentos (DER), nada mais faz que exibir as tabelas do banco de dados e seus inter-relacionamentos (chaves-estrangeiras).” (QUADROS 2002, p.193).

A Figura 3 ilustra o diagrama de entidade e relacionamento.

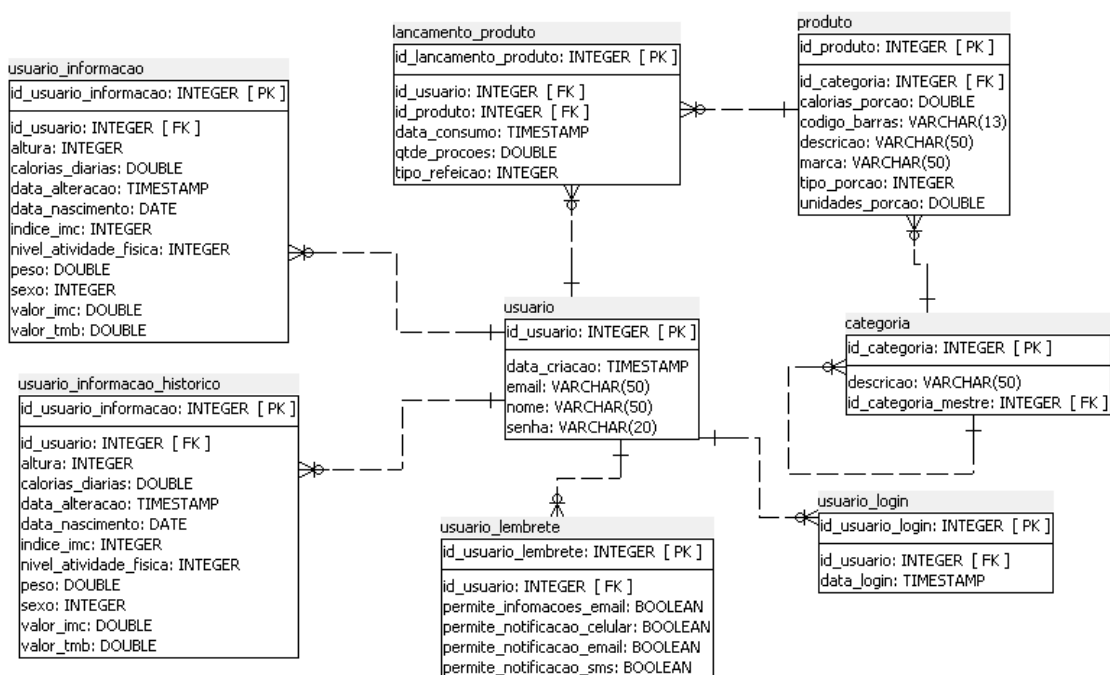


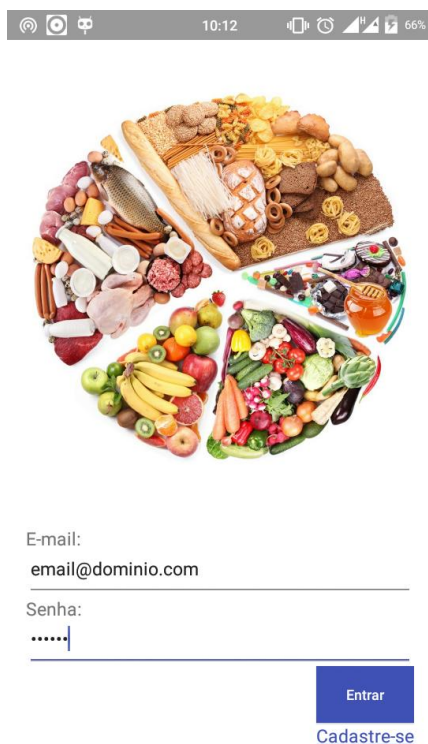
Figura 3 - Diagrama de Entidade e Relacionamento

Este diagrama representa o modelo do bando de dados, os relacionamentos entre as tabelas e a forma que os dados serão gravados, nele estão representadas todas as tabelas e seus respectivos nomes e atributos, e informações como chaves primárias e estrangeiras.

4.3 APRESENTAÇÃO DO SISTEMA

A primeira tela do sistema é o *login*, a partir desta tela o usuário poderá acessar a aplicação, informando seu e-mail e senha, e caso ainda não possua um cadastro no sistema será possível acessá-lo por meio desta tela.

A Figura 4 apresenta a tela de *login*.



E-mail:
email@dominio.com

Senha:
.....

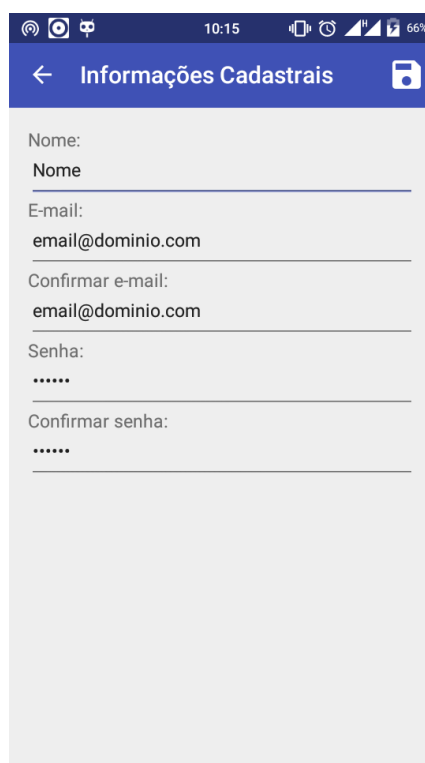
Entrar

[Cadastre-se](#)

Figura 4 – Tela de Login

Ao abrir a tela de cadastro o usuário encontrará apenas alguns campos que são nome, e-mail, confirmação de e-mail, senha e confirmação de senha. Estes dados são apenas os dados essenciais para acesso ao sistema, após a conclusão deste pré-cadastro o usuário será direcionado a outra tela onde deverá informar os demais dados do seu cadastro.

A Figura 5 apresenta a tela de informações cadastrais do usuário.



A imagem mostra uma captura de tela de um aplicativo móvel. No topo, há uma barra de status com ícones de notificação, bateria e sinal, e o horário 10:15. Abaixo, uma barra azul contém um ícone de seta para trás, o texto "Informações Cadastrais" e um ícone de salvar. O formulário principal, em um fundo cinza claro, contém os seguintes campos:

- Nome: Nome
- E-mail: email@dominio.com
- Confirmar e-mail: email@dominio.com
- Senha:
- Confirmar senha:

Figura 5 – Tela de Informações Cadastrais

Após se cadastrar o usuário visualizará uma tela onde deverá informar dados como data de nascimento, sexo, altura, peso e nível de atividade física. Estas são as informações necessárias para o cálculo da Taxa Metabólica Basal, Calorias necessárias e Índice de Massa Corporal.

A Figura 6 apresenta a tela de dados do usuário.

Sexo:
 Feminino Masculino

Data de nascimento:
30/09/1990

Altura em cm:
185

Peso em kg:
85.00

Nível de atividade física:
Sedentário

Índice de Massa Corporal (IMC)	
Resultado	Situação
24.84	Peso Ideal

Taxa Metabólica Basal (TMB)	
Taxa Metabólica	Calorias Diárias
1985.50	2382.60

Figura 6 – Tela de Dados do Usuário

Após o cadastro realizado o usuário poderá dar início a utilização do sistema. O mesmo visualizará a tela principal do gerenciador de calorias onde serão apresentadas as informações das calorias consumidas a cada refeição.

A Figura 7 apresenta a tela do gerenciador de calorias, esta é a principal tela do sistema.

Calorias Diárias		
2382.60		
Consumidas	A Consumir	
0.00	2382.60	

Café da Manhã		
Necessarias	Consumidas	A Consumir
476.52	0.00	476.52

Lanche da Manhã		
Necessarias	Consumidas	A Consumir
238.26	0.00	238.26

Almoço		
Necessarias	Consumidas	A Consumir
833.91	0.00	833.91

Figura 7 – Gerenciador de Calorias

Para efetuar o lançamento do consumo de um alimento o usuário deve selecionar a categoria a qual o alimento pertence, nesta tela são apresentadas todas as categorias disponíveis para seleção, com a possibilidade de filtrar as categorias para um resultado mais rápido.

Após selecionar uma categoria de alimento será exibida a tela de seleção de alimentos, onde o usuário deverá selecionar o alimento consumido. Nesta tela serão apresentados todos os produtos da categoria selecionada, para facilitar a seleção o usuário também pode filtrar os produtos por descrição.

Após a seleção do alimento o usuário será direcionado a tela de lançamento de calorias, onde o usuário deverá informar a quantidade de porções consumidas do alimento selecionado. O cálculo das calorias será feito com base nas informações do cadastro do alimento, onde o mesmo já possui a quantidade de calorias para cada porção, assim basta multiplicar a quantidade de porções consumidas pela quantidade de calorias de cada porção.

As Figuras 8, 9 e 10 apresentam as telas de seleção de categoria de alimentos, seleção de alimentos e lançamento das calorias consumidas.

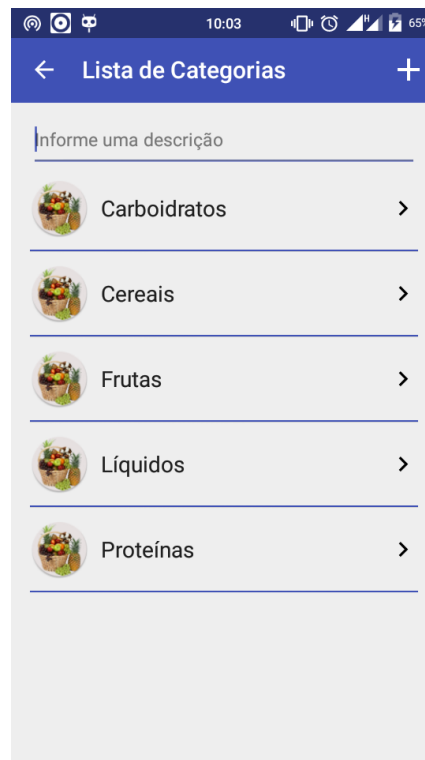


Figura 8 – Tela de Seleção de Categorias de Produtos

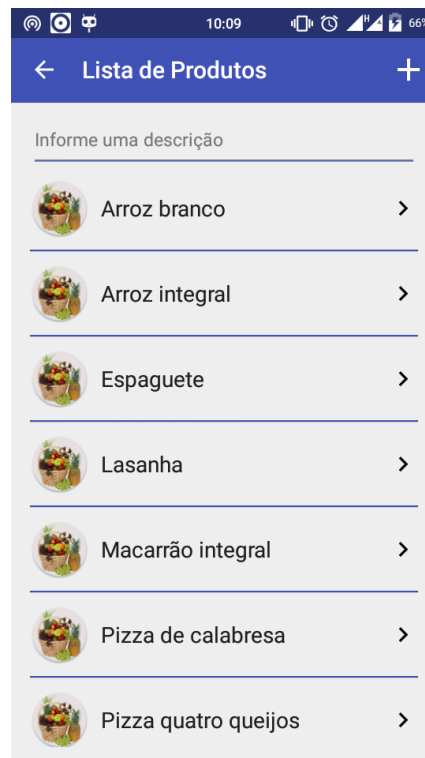


Figura 9 – Tela de Seleção de Produtos

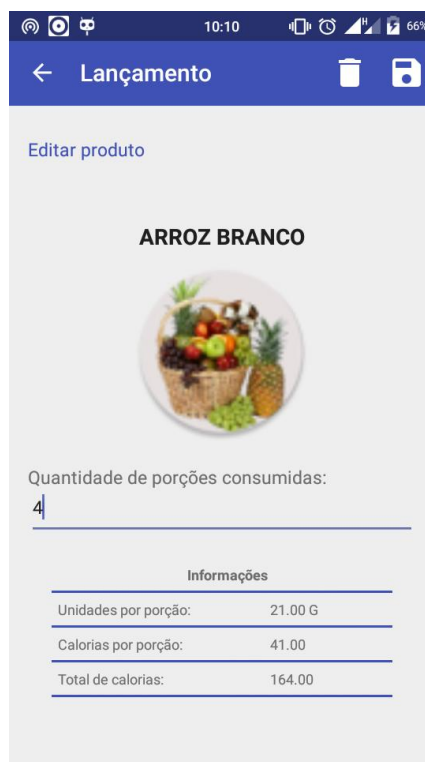
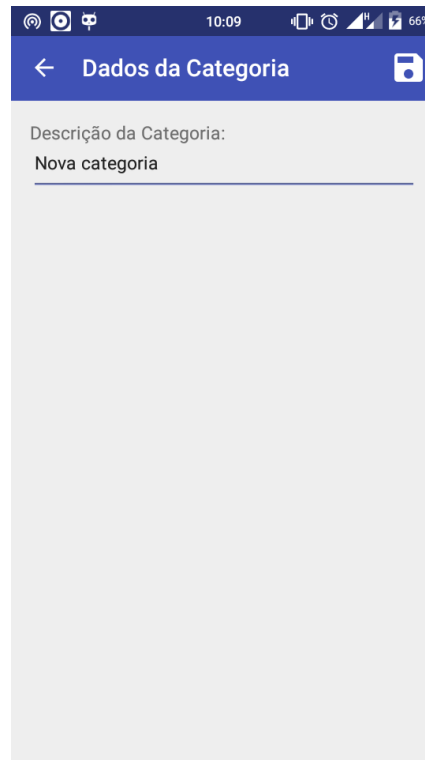


Figura 10 – Tela de Lançamentos

Caso o usuário não encontre uma categoria ou alimento específico ele poderá efetuar o cadastro de uma nova categoria ou produto.

As Figuras 11 e 12 apresentam as telas de cadastro de categoria de alimentos, e cadastro de alimentos.

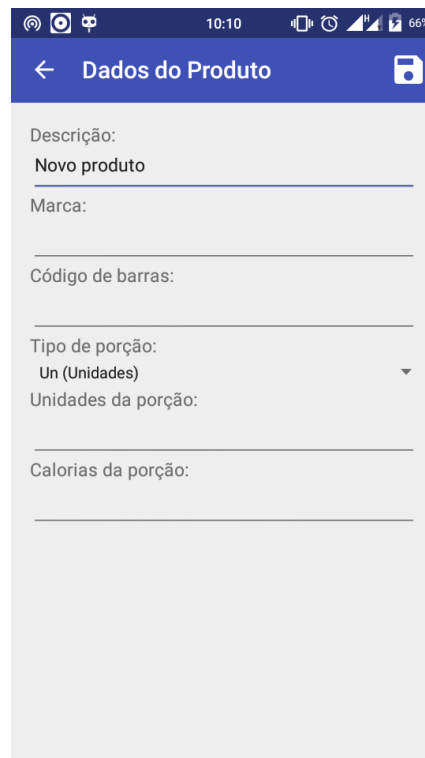


10:09 66%

← Dados da Categoria

Descrição da Categoria:
Nova categoria

Figura 11 – Tela de Cadastro de Novas Categorias



10:10 66%

← Dados do Produto

Descrição:
Novo produto

Marca:

Código de barras:

Tipo de porção:
Un (Unidades)

Unidades da porção:

Calorias da porção:

Figura 12 – Tela de Cadastro de Novos Produtos

O usuário ainda poderá conferir as calorias consumidas diariamente por meio de listagens e gráficos, estas informações serão apresentadas separadas por refeição.

A Figura 13 apresenta a tela de listagem das calorias consumidas.

Almoço		
Necessárias	Consumidas	A Consumir
833.91	735.00	98.91

Lancamentos		
Produto:		
Qtde Porções	Cal. Porção	Cal. Total
Arroz branco		
4.00	41.00	164.00
Lasanha		
1.00	139.00	139.00
Pizza quatro queijos		
1.00	432.00	432.00

Figura 13 – Listagem de Calorias Consumidas por Refeição

As Figuras 14 e 15 apresentam as telas onde são apresentadas as informações das calorias consumidas por refeição em forma de gráficos.

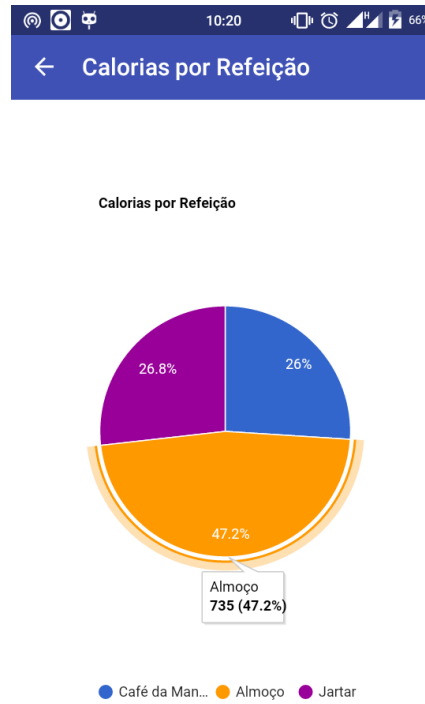


Figura 14 – Gráfico de Calorias Consumidas Diariamente por Refeição

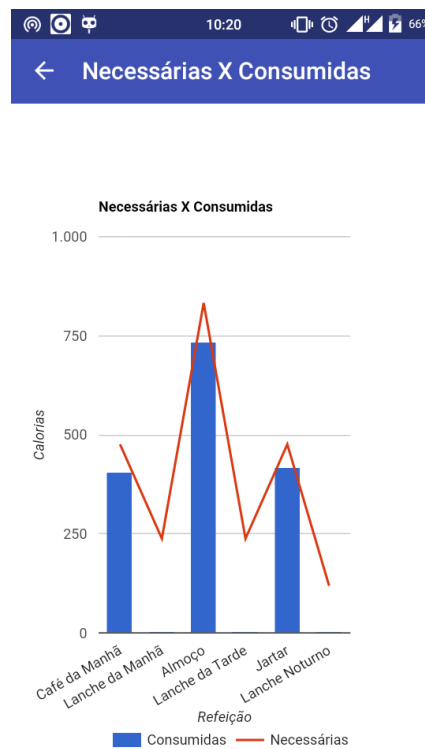


Figura 15 – Gráfico de Calorias Consumidas X Necessárias por Refeição

4.4 IMPLEMENTAÇÃO DO SISTEMA

Conforme apresentado na seção 4.1 foram implementados dois sistemas, um Webservice utilizando conceitos de *cloud computing* e uma aplicação *mobile* utilizando a plataforma Android. A comunicação entre os dois sistemas é feita por meio de disponibilização e consumo de serviços REST, para isso foram implementadas classes genéricas de disponibilização e consumo destes serviços. A seguir são apresentadas listagens de código com exemplos das classes genéricas e classes que estendem das genéricas.

A Listagem 1 apresenta um trecho de código da classe genérica de exposição de serviços REST no Webservice.

```
public class GenericRestImpl<T> implements GenericRest<T> {

    public static final String APPLICATION_JSON_MEDIA_TYPE = "application/json;
charset=UTF-8";

    @POST
    @Consumes( GenericRestImpl.APPLICATION_JSON_MEDIA_TYPE )
    @Produces( GenericRestImpl.APPLICATION_JSON_MEDIA_TYPE )
    @Override
    public Response insert( T obj ) {
        return Response.status( Status.FORBIDDEN ).build();
    }

    /**
     * Código abaixo omitido
     */
}
```

Listagem 1 – Classe genérica no Webservice de exposição de serviços REST

A Listagem 2 apresenta o trecho de código da classe genérica para consumo de serviços REST no aplicativo Android.

```
public class RestClient {

    public final RestResponse post(RestRequest request) {

        RestResponse response = new RestResponse();
        try {
            HttpPost httpPost = new HttpPost(new URI(request.getUrl()));
            httpPost.setHeader("Accept", "application/json");
            httpPost.setHeader("Content-type", "application/json");
            StringEntity sEntity = new
StringEntity(request.getJsonRequest(), "UTF-8");
            httpPost.setEntity(sEntity);

            HttpResponse httpResponse =
HttpClientSingleton.getInstance().execute(httpPost);
            HttpEntity entity = httpResponse.getEntity();

            if (entity != null) {
```

```

        response.setStatus( httpResponse.getStatusLine()
        .getStatusCode() );
        InputStream instream = entity.getContent();
        response.setJsonResponse( getJson( instream ) );
        instream.close();
        Log.d("post", "Result : " + response.getStatus() + " : " +
response.getJsonResponse() );
    } else {
        throw new Exception("Retorno nulo");
    }
} catch (Exception e) {
    Log.e("post", "Falha ao acessar serviço rest", e);
    response.setStatus(500);
    response.setJsonResponse( e.getMessage() );
} finally {
    return response;
}
}

/**
 * Código abaixo omitido
 */
}

```

Listagem 2 – Classe genérica no Android para consumo de serviços REST

A Listagem 3 apresenta a implementação do serviço REST no Webservice, este é estendido do código genérico citado na Listagem 1.

```

@Path( UsuarioRestImpl.ENDPOINT )
public class UsuarioRestImpl extends GenericRestImpl<UsuarioDTO> implements
UsuarioRest {

    /**
     * Descrição do Endpoint.
     */
    public static final String ENDPOINT = "usuario";

    /**
     * Injeção do Controlador.
     */
    @EJB
    private UsuarioController usuarioController;

    @POST
    @Override
    public Response insert( UsuarioDTO usuario ) {

        try {
            usuario = usuarioController.insert( usuario );
            return Response.status( Status.OK ).entity( usuario ).build();
        } catch ( DatabaseException | ControllerException e ) {
            return Response.status( Status.INTERNAL_SERVER_ERROR ).entity(
e.getMessage() ).build();
        }

    }

    /**
     * Código abaixo omitido
     */
}

```

```

    */
}

```

Listagem 3 – Classe no Webservice que expõe o serviço REST

A Listagem 4 apresenta o código para implementação do consumo de serviço REST no Android. Esse trecho complementa código genérico para consumo de serviços citado na Listagem 2.

```

public class UsuarioREST {

    private static final String URL_ENDIPOINT =
RestClient.URL_CONTEXTO.concat("usuario/");

    public UsuarioDTO save(UsuarioDTO usuario) throws Exception {

        String usuarioJSON =
RestClient.getInstance().objectToJson(usuario);

        RestRequest request = new RestRequest();
        request.setUrl(URL_ENDIPOINT);
        request.setJsonRequest(usuarioJSON);

        RestResponse response = RestClient.getInstance().post(request);

        if (response.getStatus().equals(200)) {

            return transformaEmObjeto(response.getJsonResponse());
        } else {

            throw new Exception(response.getJsonResponse());
        }
    }

    /**
     * Código abaixo omitido
     */
}

```

Listagem 4 – Classe no Android que consome o serviço REST

No consumo de serviços REST em aplicações Android, para que o aplicativo não fique travado até o retorno das informações é necessário utilizar *threads*. Estas farão um processamento paralelo liberando o aplicativo para uso. Uma boa prática é utilizar *threads* assíncronas para um melhor resultado, na Listagem 5 é apresentado um exemplo de utilização de *threads* assíncronas para consumo dos serviços REST por meio dos códigos listados nas listagens anteriores.

```

public class UsuarioSave extends AsyncTask<UsuarioDTO, Void, Object[]> {

    private Activity activity;

    public UsuarioSave(Activity activity) {
        this.activity = activity;
    }
}

```



```

@Override
protected Object[] doInBackground(UsuarioDTO... params) {

    Object[] retorno = new Object[2];
    try {
        UsuarioDTO usuario = params[0];
        usuario = new UsuarioREST().save(usuario);
        retorno[0] = usuario;

    } catch (Exception e) {
        retorno[1] = e.getMessage();
    } finally {
        return retorno;
    }
}

@Override
protected void onPostExecute(Object[] retorno) {

    UsuarioDTO usuario = (UsuarioDTO) retorno[0];
    String erro = String.valueOf(retorno[1]);
    if (usuario != null) {
        ToastUtil.makeToast(activity, R.string.usuario_cadastrado_com_
sucesso);

        Intent intent = new Intent(activity, LoginActivity.class);
        activity.startActivity(intent);
        activity.finish();
    } else {
        ToastUtil.makeToast(activity, erro);
    }
}
}

```

Listagem 5 – Classe que representa *thread* assíncrona para consumo de serviços REST

Como pode-se perceber nas listagens anteriores a utilização de classes genéricas ajuda muito na redução e evita redundância de código, além de deixá-lo mais limpo e fácil de visualizar.

Outro tratamento genérico implementado são as *exceptions*, como pode-se ver na listagem seguinte são utilizadas apenas duas classes para lançamento de exceções (que são `DatabaseException` e `ControllerException`), fazendo com que as exceções possam ser melhor controladas, e em caso de manutenção as alterações ficam mais simplificadas.

A Listagem 6 apresenta exemplo de tratamento de exceções com classes genéricas.

```

@Stateless
public class UsuarioControllerImpl extends GenericControllerImpl<UsuarioDTO>
implements UsuarioController {

    /**
     * Código acima omitido
     */

    @Override

```

```

public UsuarioDTO insert( UsuarioDTO usuarioDTO ) throws DatabaseException,
ControllerException {

    try {
        Usuario usuario = usuarioConverter.convertToORM( usuarioDTO );

        validaCampos( usuario );

        validaEmailExistente( usuario );

        calculaIMC( usuario );

        calculaTMB( usuario );

        usuario = usuarioDAO.insert( usuario );

        return usuarioConverter.convertToDTO( usuario );
    } catch ( DatabaseException | ControllerException e ) {
        throw e;
    } catch ( Exception e ) {
        throw new ControllerException( Mensagens.FalhaGravarUsuario, e );
    }
}

/**
 * Código abaixo omitido
 */
}

```

Listagem 6 – Tratamento de exceções com exceções genéricas

Na seção 2 são apresentadas formulas para o cálculo da Taxa Metabólica Basal e Índice de Massa Corporal. As Listagens 7 e 8 apresentam trechos de códigos que representam os respectivos cálculos.

```

private Double getTaxaMetabolica( UsuarioInformacao usuarioInformacao ) {

    TipoSexo tipoSexo = usuarioInformacao.getSexo();

    if ( tipoSexo.equals( TipoSexo.FEMININO ) ) {
        return calculaTaxaMetabolicaFeminino( usuarioInformacao );
    } else {
        return calculaTaxaMetabolicaMasculino( usuarioInformacao );
    }
}

private Double calculaTaxaMetabolicaFeminino( UsuarioInformacao
usuarioInformacao ) {

    // TMB Feminino = 655 + (9,6 * P) + (1,8 * A) - (4,7 * I)

    Double peso = usuarioInformacao.getPeso();
    Integer altura = usuarioInformacao.getAltura();
    Integer idade = getIdade( usuarioInformacao );

    Double tmb = 655 + ( 9.6D * peso ) + ( 1.8D * altura ) - ( 4.7D * idade

```

```

);

    return tmb;

}

private Double calculaTaxaMetabolicaMasculino( UsuarioInformacao
usuarioInformacao ) {

    // TMB Masculino = 66 + (13,7 * P) + (5 * A) - (6,8 * I)

    Double peso = usuarioInformacao.getPeso();
    Integer altura = usuarioInformacao.getAltura();
    Integer idade = getIdade( usuarioInformacao );

    Double tmb = 66 + ( 13.7D * peso ) + ( 5D * altura ) - ( 6.8D * idade );

    return tmb;

}

```

Listagem 7 – Cálculo da Taxa Metabólica Basal

```

@Override
public ImcDTO calculaIMC( UsuarioInformacao usuarioInformacao ) {

    Integer idade = getIdade( usuarioInformacao );

    if ( idade > 60 ) {
        return calculaIMCIidoso( usuarioInformacao );
    } else if ( idade < 18 ) {
        return calculaIMCAdolescente( usuarioInformacao );
    } else {
        return calculaIMCA adulto( usuarioInformacao );
    }

}

private ImcDTO calculaIMCA adulto( UsuarioInformacao usuarioInformacao ) {

    ImcDTO imc = new ImcDTO();

    Double valorIMC = geraIMCGeral( usuarioInformacao );

    imc.setValorIMC( valorIMC );

    if ( valorIMC >= 30 ) {
        imc.setIndiceIMC( IndiceIMC.OBESIDADE.ordinal() );
    } else if ( valorIMC >= 25 ) {
        imc.setIndiceIMC( IndiceIMC.SOBREPESO.ordinal() );
    } else if ( valorIMC >= 18.5 ) {
        imc.setIndiceIMC( IndiceIMC.PESO_ADEQUADO.ordinal() );
    } else {
        imc.setIndiceIMC( IndiceIMC.ABAIXO_DO_PESO.ordinal() );
    }

    return imc;

}

```

```

private ImcDTO calculaIMCIidoso( UsuarioInformacao usuarioInformacao ) {

    ImcDTO imc = new ImcDTO();

    Double valorIMC = geraIMCGeral( usuarioInformacao );

    imc.setValorIMC( valorIMC );

    if ( valorIMC >= 27 ) {
        imc.setIndiceIMC( IndiceIMC.OBESIDADE.ordinal() );
    } else if ( valorIMC >= 22 ) {
        imc.setIndiceIMC( IndiceIMC.PESO_ADEQUADO.ordinal() );
    } else {
        imc.setIndiceIMC( IndiceIMC.ABAIXO_DO_PESO.ordinal() );
    }

    return imc;
}

private ImcDTO calculaIMCAdolescente( UsuarioInformacao usuarioInformacao ) {

    ImcDTO imc = new ImcDTO();

    Double valorIMC = geraIMCGeral( usuarioInformacao );

    ValorIMCAdolescentes imcAdolescente = ValorIMCAdolescentes.getValue(
        getIdade( usuarioInformacao ),
        usuarioInformacao.getSexo() );

    imc.setValorIMC( valorIMC );
    imc.setIndiceIMC( geraIndiceIMCAdolescente( imcAdolescente, valorIMC
    ).ordinal() );

    return imc;
}

private Double geraIMCGeral( UsuarioInformacao info ) {

    Double imc = info.getPeso() / Math.pow( info.getAltura().doubleValue() /
    100, 2 );

    return DoubleUtil.arredonda( imc, 3 );
}

private IndiceIMC geraIndiceIMCAdolescente( ValorIMCAdolescentes
imcAdolescente, Double valorIMC ) {

    if ( valorIMC >= imcAdolescente.getObesidade() ) {
        return IndiceIMC.OBESIDADE;
    } else if ( valorIMC >= imcAdolescente.getSobrepeso() ) {
        return IndiceIMC.SOBREPESO;
    } else if ( valorIMC >= imcAdolescente.getAdequado() ) {
        return IndiceIMC.PESO_ADEQUADO;
    } else {
        return IndiceIMC.ABAIXO_DO_PESO;
    }
}
}

```

Listagem 8 – Cálculo do Índice de Massa Corporal

5 CONCLUSÃO

O principal objetivo deste trabalho foi implementar uma aplicação de controle de calorias consumidas por seus usuários, utilizando conceitos de computação em nuvem e desenvolvimento na plataforma Android.

As tecnologias utilizadas apresentam muitas vantagens ao trabalho, como a linguagem Java que possui uma utilização global e multiplataforma, a disponibilização de serviços por meio de um Webservice REST que pode ser integrado com vários outros sistemas desenvolvidos em outras linguagens. A utilização da plataforma Android trouxe vantagens como a mobilidade.

Durante o desenvolvimento dos sistemas foram encontradas e superadas várias dificuldades, como a integração dos dois sistemas via REST, o desenvolvimento de uma aplicação Android com várias classes assíncronas para que o sistema trabalhe mais fluído enquanto consome os serviços REST.

Entretanto, apesar das dificuldades, o resultado foi satisfatório, os dois sistemas se comunicam fluentemente pelos serviços REST, as classes assíncronas garantem um bom funcionamento da aplicação enquanto são consumidos os serviços.

Os objetivos deste trabalho foram alcançados com o desenvolvimento da aplicação conforme proposto e a aplicação auxiliará as pessoas a controlarem as calorias consumidas durante as refeições de seu dia-a-dia.

REFERÊNCIAS

ANDROID. **A história do Android**. Disponível em <https://www.android.com/intl/pt-BR_br/history/>. Acesso em: 22 ago. 2015.

ANDROID STUDIO. **Android Studio Overview**. Disponível em <<https://developer.android.com/tools/studio/index.html>>. Acesso em: 22 ago. 2015.

ASTAH. **Community**. Disponível em <<http://astah.net/editions/community>>. Acesso em: 22 ago. 2015.a

_____. **Professional**. Disponível em <<http://astah.net/editions/professional>>. Acesso em: 22 ago. 2015b.

BEZERRA, Eduardo. **Princípios de análise e projetos de sistemas com UML**. Rio de Janeiro, 2007.

BUYYA, R.; YEO, C.; VNUGOPAL, S. **Market-oriented cloud computing: vision, hype, and reality for delivering it services as computing utilities**. 2008.

CDOF. **Nutrição**. Disponível em: <<http://www.cdof.com.br/>>. Acessado em 10 nov. 2015.

ECLIPSE. **About the Eclipse Foundation**. Disponível em <<https://eclipse.org/org/>>. Acesso em: 22 ago. 2015.

EHOW. **Qual o mínimo de calorias que devo comer em um dia?**. Disponível em <http://www.ehow.com.br/minimo-calorias-devo-comer-dia-fatos_170063/>. Acesso em: 10 out. 2015.

GRADLE. **#WhyGradle**. Disponível em <<https://gradle.org/whygradle-build-automation/>>. Acesso em: 07 set. 2015.

HIBERNATE. **Hibernate**. Disponível em <<http://hibernate.org/>>. Acesso em: 07 set. 2015.

HOSPITAL ALBERT EINSTEIN. **Saiba o que é IMC – Índice de Massa Corporal**. Disponível em <<http://www.einstein.br/Blog/paginas/post.aspx?post=1260>>. Acesso em: 10 out. 2015.

HSW. **De quantas calorias uma pessoa precisa diariamente?**. Disponível em <<http://saude.hsw.uol.com.br/questao457.htm>>. Acesso em: 10 out. 2015.

INTELLIJ IDEA. **Why IntelliJ IDEA?** Disponível em <<https://www.jetbrains.com/idea/whyidea.html>>. Acesso em: 22 ago. 2015.

JAVA. **O que é a tecnologia Java e porque preciso dela?** Disponível em <http://www.java.com/pt_BR/download/faq/whatis_java.xml>. Acesso em: 07 set. 2015a.

_____. **The history of Java technology.** Disponível em <<http://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html>>. Acesso em: 22 ago. 2015b.

JETBRAINS. **Our mission.** Disponível em <<https://www.jetbrains.com/company/>>. Acesso em: 07 set. 2015.

MAVEN. **Welcome to Apache Maven.** Disponível em <<https://maven.apache.org/>>. Acesso em: 07 set. 2015.

McARDLE, W. D.; KATCH, F. I.; KATCH, V. L. **Fisiologia do exercício; energia, nutrição e desempenho humano.** Rio de Janeiro: Guanabara koogan, 1992.

MEDICALNEWSTODAY. **What are calories? What is a calorie?** Disponível em <<http://www.medicalnewstoday.com/articles/263028.php>>. Acesso em: 10 out. 2015.

NESTLE. **IMC: o que a balança não diz.** Disponível em <https://www.nestle.com.br/site/fazbem/bem-estar/d/imc_o_que_a_balanca_ao_diz.aspx>. Acesso em: 10 out. 2015a.

_____. **Avaliação do estado nutricional de idosos.** Disponível em <<http://www.nestle-nutricaoodomiciliar.com.br/Files/documentos/AVALIACAO%20EST%20NUT.pdf>>. Acesso em: 10 out. 2015b.

PEDIATRAONLINE. **Como calcular o IMC.** Disponível em <http://pediatraonline.com.br/eden_vivianecolla/Artigo/21302>. Acesso em: 10 out. 2015.

POSTGRESQL. **About.** Disponível em <<http://www.postgresql.org/about/>>. Acesso em: 22 ago. 2015.

QUADROS, Moacir. **Gerência de projetos de software: técnicas e ferramentas.** Santa Catarina, 2002.

SOUSA, Flávio R. C.; MOREIRA, Leonardo O.; MACHADO Javam C., **Computação em nuvem: conceitos, tecnologias, aplicações e desafios.** Disponível em <<http://www.ufpi.br/subsiteFiles/ercemapi/arquivos/files/minicurso/mc7.pdf>>. Acesso em: 10 out. 2015.

SQL POWER SOFTWARE. **Data modeling & profiling tool.** Disponível em <<http://www.sqlpower.ca/page/architect>>. Acesso em: 22 ago. 2015a.

_____. **Company overview.** Disponível em <<http://www.sqlpower.ca/page/company>>. Acesso em: 22 ago. 2015b.

SQLITE. **About SQLite.** Disponível em <<https://www.sqlite.org/about.html>>. Acesso em: 22 ago. 2015.

SUPERINTERESANTE. **Conheça a história do Android, o sistema operacional mobile da Google.** Disponível em <<http://super.abril.com.br/conheca-a-historia-do-android-o-sistema-operacional-mobile-da-google>>. Acesso em: 22 ago. 2015.

TAURION, Cezar. **Cloud computing: computação em nuvem: transformando o mundo da tecnologia da informação**. Rio de Janeiro: Brasport, 2009.

TECHCRUNCH. **Android now has 1.4 billion 30-day active users globally**. Disponível em <<http://techcrunch.com/2015/09/29/android-now-has-1-4bn-30-day-active-devices-globally/>>. Acesso em: 30 set. 2015.

TECHTUDO. **Desenvolva aplicações para plataforma móvel Android, do Google**. Disponível em <<http://www.techtudo.com.br/tudo-sobre/android-sdk.html>>. Acesso em: 22 ago. 2015.

VAQUERO, L., RODERO-MERINO, L., CACERES, J., LINDENER, M. **A break in the clouds: towards a cloud definition**. SIGCOMM, 2008.

WHO. **BMI classification**. Disponível em <http://apps.who.int/bmi/index.jsp?introPage=intro_3.html>. Acesso em: 10 out. 2015.

WHO Multicentre Growth Reference Study Group. **WHO Child Growth Standards: Length/height-for-age, weight-for-age, weight-for-length, weight-for-height and body mass index-for-age: methods and development**. Geneva: World Health Organization, 2006.

WILDFLY. **What is WildFly?** Disponível em <<http://wildfly.org/about/>>. Acesso em: 22 ago. 2015.