

UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUCAS JOSÉ ZANATTA PINTO

**PROTÓTIPO PARA ANDROID COM FOCO NO APRENDIZADO
DE NUTRIÇÃO PARA MUSCULAÇÃO E EDUCAÇÃO ALIMENTAR**

UNESP – Bauru
Março de 2015

LUCAS JOSÉ ZANATTA PINTO

**PROTÓTIPO PARA ANDROID COM FOCO NO APRENDIZADO
DE NUTRIÇÃO PARA MUSCULAÇÃO E EDUCAÇÃO ALIMENTAR**

Orientador: Prof. Dr. João Pedro Albino

Monografia apresentada junto à disciplina
Projeto e Implementação de Sistemas II, do curso
de Bacharelado em Ciência da Computação,
Faculdade de Ciências, Unesp, campus de Bauru,
como parte do Trabalho de Conclusão de Curso.

**UNESP – Bauru
Março de 2015**

Lucas José Zanatta Pinto

**PROTÓTIPO PARA ANDROID COM FOCO NO APRENDIZADO DE NUTRIÇÃO
PARA MUSCULAÇÃO E EDUCAÇÃO ALIMENTAR**

BANCA EXAMINADORA

Orientador Prof. Dr. João Pedro Albino.
Instituição: Universidade Estadual Paulista.

Assinatura: _____

Nome: Prof. Dr. Eduardo Martins Morgado.
Instituição: Universidade Estadual Paulista.

Assinatura: _____

Nome: Prof. Dr. Wilson Massashiro Yonezawa.
Instituição: Universidade Estadual Paulista.

Assinatura: _____

Universidade Estadual Paulista
18 de março de 2015

Dedico este trabalho especialmente à minha família que sempre torceu pelo meu sucesso e, mesmo nos momentos mais difíceis, nunca deixou de acreditar na minha capacidade.

Agradecimentos

Primeiramente agradeço à Ele por sempre guiar meu caminho, me proteger dos males, me abençoar e me dar a sabedoria para encarar os problemas. À meus pais por terem me dado a educação, motivação e apoio necessários. À meu irmão pelo conhecimento e personalidade transmitidos durante todo esse tempo. À meus professores e amigos que, juntos, contribuíram para minha formação acadêmica e pessoal. Agradeço também a UNESP Bauru por ter me dado a oportunidade de estudar numa universidade renomada e de conhecer pessoas brilhantes ao longo da jornada.

Enfim, agradeço a todos que direta ou indiretamente me ajudaram a concluir essa importante etapa da vida.

"You can't be paralyzed by fear of failure or you will never push yourself. You keep pushing because you believe in yourself and in your vision and you know that it is the right thing to do, and success will come. So don't be afraid to fail."

Arnold Schwarzenegger

RESUMO

A obesidade é um problema grave que a saúde pública enfrenta atualmente no Brasil e em outros países do mundo. Segundo pesquisa realizada pelo Ministério da Saúde, 48,5% da população brasileira estava acima do peso em 2011. Qual a razão para esse número tão alarmante? Refletindo sobre a evolução humana, há 50 mil anos, o homem tinha muitas dificuldades para encontrar comida e, portanto, estocava no seu organismo o máximo de calorias possível a fim de usá-las em períodos de escassez. Porém, hoje o cenário é diferente: com o desenvolvimento e a industrialização, as pessoas conseguem conservar alimentos por semanas. A propaganda condicionada à falta de conhecimento incitam a população a ingerir alimentos altamente calóricos por preços bastante acessíveis. Além disso, boa parte das pessoas é sedentária, o que acaba contribuindo para o quadro de sobrepeso que a sociedade se encontra. Levando em consideração tais questões, o presente trabalho teve como objetivo a construção de um protótipo para ambientes do sistema operacional móvel *Android* que busca apresentar os principais conceitos de nutrição ao usuário e o auxilia na criação de dietas para musculação, sejam elas voltadas para emagrecimento ou ganho de massa muscular. Foi feita uma série de pesquisas sobre o funcionamento da plataforma *Android* e os principais conceitos de nutrição com o intuito de aplicá-los no *software*. Por fim, os objetivos foram alcançados e o aplicativo foi construído com êxito.

Palavras-chave: Obesidade. Sedentarismo. *Android*. Nutrição.

ABSTRACT

At the present time, the obesity is a serious matter that the public health care system faces in Brazil and other countries of the world. According to a research made by the Ministério da Saúde, 48,5% of brazilian population was overweight in 2011. What is the reason for such and alarming number? Reflecting on the human evolution, 50 thousand years ago, men had many difficulties to find food and, therefore, stocked maximum of calories possible in their organism so that they could use it in periods of shortage. although, today's scenario is different: with the development and the industrialization, people are able to conserve food for weeks. Advertisement added to the lack of knowledge encourage the population to eat high-calorie foods for reasonable prices. Beyond that, a great portion of people is sedentary, which contributes to the overweight state that society is currently on. Taking those questions in consideration, the present paper had as the objective the construction of a prototype for android mobile operational system environment that aims to present the main concepts of nutrition to the user and assists him/her in the creation of diets for weight lifting, they being for weight loss or lean body mass gain. A series of researchs about the operation of the Android platform were done aswell as the main concepts of nutrition with the purpose of applying them in the software. Ultimately, the objectives were reached and the application was build succesfully.

Palavras-chave: Obesity. Sedentary Lifestyle. Android. Nutrition.

Lista de Figuras

Figura 1: Aminoácidos Essenciais e Não Essenciais.....	18
Figura 2: Consumo Total de Água Diária Recomendada.....	21
Figura 3: Fatia de Mercado Mundial de <i>Smartphones</i>	23
Figura 4: Requisição HTTP.....	25
Figura 5: Arquitetura <i>Android</i>	28
Figura 6: Layout para Uma Aplicação <i>Android</i>	32
Figura 7: Ciclo de Vida de Uma <i>Activity</i>	33
Figura 8: Exemplo de Utilização dos <i>Fragments</i>	34
Figura 9: Funcionamento do <i>Navigation Drawer</i>	35
Figura 10: Função do PHP em páginas dinâmicas.....	37
Figura 11: Método <i>downloadUrlGET</i>	39
Figura 12: Verificação de Conectividade com a Internet.....	40
Figura 13: <i>HomeFragment</i>	41
Figura 14: Tela de Cadastro de Dietas.....	42
Figura 15: Gráfico de Evolução.....	43
Figura 16: Notificação.....	44
Figura 17: <i>WaterConsumptionFragment</i>	45

Lista de Quadros

Quadro 1: Equações Utilizadas para Mulheres.....	15
Quadro 2: Equações Utilizadas para Homens	16
Quadro 3: Principais Plataformas dos Dispositivos Móveis.....	22

Sumário

1.	Introdução	12
1.1.	Problema	13
1.2.	Justificativa	13
1.3.	Objetivo Geral	14
1.4.	Objetivos Específicos	14
1.5.	Método de Pesquisa	14
2.	Fundamentação Teórica	15
2.1.	Taxa Metabólica Basal	15
2.2.	Macro nutrientes e sua Distribuição na Dieta	16
2.2.1.	Carboidratos	16
2.2.2.	Proteínas	17
2.2.3.	Gorduras	18
2.3.	Quantidades	19
2.4.	Consumo de Água e sua Importância para o Organismo	20
2.5.	Dispositivos Móveis	21
2.5.1.	Plataformas para Desenvolvimento	22
2.6.	Banco de Dados	23
2.7.	HTTP	24
2.7.1.	Verbos	25
3.	Materiais e Métodos	27
3.1.	Materiais	27
3.1.1.	<i>Android</i>	27
3.1.1.1.	Arquitetura	27
3.1.1.1.1.	Aplicações	28
3.1.1.1.2.	<i>Framework</i> de Aplicação	28
3.1.1.1.3.	Bibliotecas	29

3.1.1.1.4.	<i>Android Runtime</i>	29
3.1.1.1.5.	<i>Linux Kernel</i>	29
3.1.1.2.	Requerimentos para Desenvolvimento.....	30
3.1.1.3.	Configurando um Aparelho para Desenvolver.....	30
3.1.1.4.	SDK.....	31
3.1.1.5.	<i>User Interface - Layout</i>	31
3.1.1.6.	<i>Activity</i>	32
3.1.1.7.	<i>Fragment</i>	33
3.1.1.8.	<i>Navigation Drawer</i>	35
3.1.2.	MySQL	35
3.1.3.	PHP	36
3.2.	Implementação.....	38
3.2.1.	Comunicação com o Banco de Dados	38
3.2.2.	<i>Log In / Sign In</i>	40
3.2.3.	Home.....	40
3.2.4.	Módulos	42
3.2.4.1.	Dieta	42
3.2.4.2.	Evolução.....	43
3.2.4.3.	Consumo de Água	44
4.	Considerações Finais	46
	REFERÊNCIAS	48
	APÊNDICES	50
	APÊNDICE A - Modelo Entidade Relacionamento.....	50
	APÊNDICE B - Diagrama de Caso de Uso.....	51

1. Introdução

Sabe-se que hoje em dia o *smartphone* está em ascensão no mundo da tecnologia. Devido a sua alta diversidade, possui preços variados, o tornando um objeto bastante acessível e de extrema importância para o cotidiano das pessoas. Sua importância se dá pela enorme quantidade de utilitários disponíveis. Tais utilitários, também chamados de aplicativos ou *apps*, trazem facilidade, diversão e muitas vezes comodidade para o usuário. Para se ter uma noção em números, Fabrício Tarouco (2013) apresenta dados importantes.

Atualmente acredita-se que mantendo este ritmo contínuo de crescimento no número de downloads, se atingirá no ano de 2016 aproximadamente 44 bilhões de aplicativos baixados e será neste ano que as mensagens através de *apps* irão ultrapassar as mensagens de texto.

Uma pesquisa realizada em 2012 pela Ericsson previu que o número de *smartphones* no mundo deverá triplicar até o ano de 2018, chegando a 3,3 bilhões de aparelhos móveis em uso. (TAROUCO, 2013, p. 3 e 4).

Atualmente, os aplicativos se encontram nas mais variadas formas e abordam os mais diversos assuntos. Um tema bastante interessante que merece uma maior relevância por parte dos desenvolvedores é a saúde.

Com a popularização e acessibilidade do *fast food*, a educação alimentar das pessoas está cada vez pior. A obesidade é uma doença que vem ganhando bastante atenção. Em 2002, por exemplo, ela foi considerada uma epidemia pelos especialistas. Nos EUA, a vida sedentária e alimentação desregrada contribuíam com 300.000 mortes naquele ano. (ADES; KERBAUY, 2002).

Para buscar solucionar estes problemas causados pela obesidade, existem algumas soluções tais como reeducação alimentar (dieta) e atividade física. Entretanto, em alguns casos mais graves, as mudanças alimentares e a prática de atividades físicas são impossíveis de serem implementadas. Nestas situações, apenas uma intervenção médica mais efetiva, como a cirurgia bariátrica (cirurgia para redução do tamanho do estômago), deve resolver o problema. São necessários, posteriormente, uma reeducação alimentar e o ingresso a uma academia.

A combinação musculação/dieta balanceada só traz benefícios.

A prática regular de atividade física tem sido apontada como um dos mais importantes fatores de prevenção, reversão e controle de

diversas enfermidades, especialmente as doenças crônicas não transmissíveis como enfermidades cardiovasculares, *diabetes mellitus*, câncer de cólon e de mama, entre outras.

Além disso, a atividade física pode trazer benefícios ao estado nutricional dos indivíduos, prevenindo e diminuindo a ocorrência da obesidade e suas comorbidades, uma vez que um estilo de vida sedentário limita a quantidade de calorias a serem ingeridas diariamente e pode ocasionar ganho de peso indesejado. Mura, Silva (2011, p. 237).

Além dos benefícios apresentados a cima, melhora o bem estar, postura e auto estima do praticante. A musculação pode ser efetuada por pessoas de praticamente todas as idades. Utilizando os idosos como exemplo, segundo Pereira, Souza, Lisbôa (2007) “Atualmente, a questão da participação de idosos em programas de exercícios regulares parece ser uma forma efetiva de reduzir ou prevenir o declínio funcional associado ao envelhecimento. [...], a atividade física é um fator determinante no sucesso do processo de envelhecimento saudável e com qualidade de vida.”.

A fim de prestar um serviço a uma geração dependente da tecnologia e ao mesmo tempo sedentária e poluída por propagandas de *fast food*, surgiu a ideia da construção de um aplicativo que estimula o usuário a melhorar sua educação alimentar auxiliando na criação de dietas voltadas para musculação.

1.1. Problema

Levando as questões de obesidade e suas morbidades na sociedade atual, o problema de pesquisa consiste no desenvolvimento de um aplicativo que ensine e auxilie um usuário em sua dieta, e também, o ajude a obter os conhecimentos necessários em nutrição para que ele o utilize com a garantia de que as informações apresentadas no dispositivo possuem embasamento teórico.

1.2. Justificativa

Como citado na introdução, é de conhecimento geral que a sociedade de hoje possui sérios problemas com relação à saúde, muitos desses originados pelo sedentarismo e alimentação de má qualidade. A partir dessa motivação, foi proposta uma aplicação *Android* que, mesmo de forma indireta, ajuda a amenizar a crise do sobrepeso que o mundo

(principalmente países desenvolvidos) vêm passando. O software busca atingir não só o perfil de pessoas com problemas, mas sim todos: homens, mulheres, recém ingressantes na academia, veteranos e até idosos.

1.3. Objetivo Geral

Criar um protótipo para plataforma *Android* que auxilie um usuário em sua educação alimentar e nas criações de dietas voltadas para musculação.

1.4. Objetivos Específicos

- Compreender os conceitos de nutrição necessários para o desenvolvimento do aplicativo.
- Aprender as linguagens de programação utilizadas no desenvolvimento de aplicativos para a plataforma *Android*.
- Implementar um módulo que auxilie/armazene uma dieta adequada para o usuário.
- Criar um gráfico a fim de mostrar a evolução do usuário.
- Armazenar as informações da aplicação em um banco de dados remoto utilizando o MySQL.
- Criar um mecanismo de notificações que auxilie o usuário em seu consumo diário de água.

1.5. Método de Pesquisa

Foram realizadas pesquisas para se ter um fundamento teórico dos conteúdos que envolvem nutrição a fim de aplicá-los no desenvolvimento do aplicativo.

O procedimento da coleta de dados se deu por pesquisas realizadas na biblioteca central do Campus de Bauru, da UNESP e através da internet. Posteriormente, já na fase de desenvolvimento, foram realizadas pesquisas experimentais, os chamados testes.

A equipe para este projeto consistiu de um orientador e um programador.

Os materiais necessários foram: computadores, uma IDE (Ambiente Integrado para Desenvolvimento) compatível para desenvolver o aplicativo e um *smartphone* para testes.

2. Fundamentação Teórica

Este capítulo apresenta uma sucinta descrição dos conceitos e teorias pesquisados que foram necessários para o entendimento e desenvolvimento do projeto.

2.1. Taxa Metabólica Basal

De acordo com os estudos de Wahrlich e Anjos, a Taxa Metabólica Basal (TMB) consiste na quantidade de energia (caloria) necessária para manter as funções vitais do organismo. Ela é medida, idealmente, em jejum, repouso físico e mental, em ambiente tranquilo com controle de temperatura, iluminação e sem ruído. A medição da TMB é feita através da quantidade de calor produzida pelo organismo ou pelo consumo de oxigênio/excreção de gás carbônico. Sabendo que os dados de tais medições são extremamente difíceis de se obter por pessoas ordinárias (não especialistas e sem ferramentas), Harris & Benedict começaram um estudo em 1919 a fim de desenvolver equações de predição da TMB a partir de medidas antropométricas, ou seja, dimensões e proporções do corpo (WAHRLICH; DOS ANJOS, 2001). É importante notar que as equações geram uma aproximação e não a quantidade de calorias exata. Nos quadros 1 e 2 é possível observar que já existem várias para esse fim. (MC = massa corporal, EST = estatura e ID = idade).

Quadro 1: Equações Utilizadas para Mulheres

Equações utilizadas para mulheres	
Harris & Benedict (1919)	$655 + (9,6 \times MC) + (1,85 \times EST) - (4,676 \times ID)$
Schofield (1985)	18-30 anos $[(0,062 \times MC) + 2,036]$ 30-60 anos $[(0,034 \times MC) + 3,538]$
FAO/WHO/UNU (1995)	18-30 anos $(14,7 \times MC) + 496$ 30 - 60 anos $(8,7 \times MC) + 829$
Henry & Rees (1991)	18-30 anos $[(0,048 \times MC) + 2,562]$ 30-60 anos $[(0,048 \times MC) + 2,448]$

(fonte: Adaptado de WAHRLICH; DOS ANJOS, 2001)

Quadro 2: Equações Utilizadas para Homens

Equações utilizadas para homens	
Harris & Benedict (1919)	$TMB = 66,47 + (13,75 \times MC) + (5,00 \times EST) - (6,76 \times ID)$
Schofield (1985)	18-30 anos $(0,063 \times MC + 2,896)$ 30-60 anos $(0,048 \times MC + 3,653)$
FAO/WHO/UNU (1995)	18-30 anos $(15,3 \times MC + 679)$ 30-60 anos $(11,6 \times MC + 879)$
Henry & Rees (1991)	18-30 anos $(0,056 \times MC + 2,800)$ 30-60 anos $(0,046 \times MC + 3,160)$

(fonte: Adaptado de WAHRLICH; DOS ANJOS, 2001)

2.2. Macro nutrientes e sua Distribuição na Dieta

A hipertrofia muscular é o processo através do qual se dá o aumento da massa muscular. Este aumento é, essencialmente, um aumento no diâmetro ou no comprimento das fibras musculares. Segundo Moreira (2010), “O exercício de resistência é considerado um estímulo eficaz para a ocorrência da hipertrofia. No entanto, tal não é suficiente para a promoção da deposição proteica no músculo, sendo essencial a interação do exercício com a adequada ingestão alimentar”. Surge, então, a necessidade de distribuir os macro nutrientes na dieta; nutrientes esses que se dividem em, basicamente: proteína, carboidrato e gordura. Sabe-se que simplesmente quantificá-los adequadamente não é suficiente. O anabolismo (construção) muscular depende de vários outros fatores como: propriedades digestivas das proteínas, *timing* da ingestão de nutrientes em relação ao período de exercício, co-ingestão de outros nutrientes e ingestão energética total (MOREIRA, 2010).

2.2.1. Carboidratos

Os carboidratos são a principal fonte de energia para os seres humanos, podendo ser encontrados em frutas, vegetais, leguminosas, cereais, leite e mel. Fornecem a maior parte da energia necessária para a pessoa se movimentar, executar trabalhos e viver, contribuindo com um percentual de 50 a 80% das calorias totais da alimentação. São digeridos e absorvidos no intestino delgado e fermentados por bactérias no intestino grosso (COSTA, et al, 2008). Além do simples fornecimento de energia, agem como economizadores de proteínas e gorduras. O combustível usado na contração muscular são os carboidratos. Como as reservas de

glicogênio no músculo são pequenas, se um indivíduo, ao praticar uma atividade física prolongada, não fizer uma refeição rica em glicídios, o corpo passará a converter proteínas e gorduras em glicose, ou seja, estes macro nutrientes deixarão de exercer suas funções específicas no corpo para servirem como combustível (COSTA, et al, 2008).

Os glicídios são classificados em 3 grandes classes:

- Monossacarídeos: Constituem os chamados açúcares simples. Os principais são a glicose e frutose. A glicose é o carboidrato mais importante para o corpo humano, pois além de ser absorvida diretamente, algumas células específicas, como as hemácias, não conseguem usar outro tipo de combustível. É a melhor forma de açúcar quando o corpo necessita de um suprimento imediato de energia (COSTA, et al, 2008).
- Dissacarídeos: São representados pela sacarose (açúcar da cana), maltose (açúcar do malte) e lactose (açúcar do leite) e compostos por dois monossacarídeos. Como o corpo só consegue absorver monossacarídeos, estes carboidratos são hidrolisados (reação química que quebra os dissacarídeos com o auxílio de água) pelas enzimas digestivas (COSTA, et al, 2008).
 - Sacarose = glicose + frutose
 - Maltose = glicose + glicose
 - Lactose = glicose + galactose
- Polissacarídeos: Com o objetivo de maximizar a energia potencial dos carboidratos, as plantas e animais os armazenam em formas complexas, ou seja, em moléculas compostas por várias glicoses. A exemplo de polissacarídeos tem-se o glicogênio (armazenamento dos animais), amido, celulose (encontrada nas plantas, não digerível pelos homens) etc (COSTA, et al, 2008).

2.2.2. Proteínas

Segundo Costa et al (2008), a palavra vem do grego *protos* (primeiro). Foi o primeiro macro nutriente considerado essencial para o organismo e é o principal e mais abundante componente estrutural e funcional das células do corpo humano.

As proteínas são moléculas grandes formadas por aminoácidos. Uma proteína difere de outra pela quantidade e sequência de aminoácidos em sua composição (estima-se que

exista cerca de 200 aminoácidos na natureza). Proteínas diferentes possuem funções diferentes.

Os aminoácidos podem ser classificados em 2 grupos:

- **Dispensáveis ou não essenciais:** São aqueles que o organismo consegue produzir por conta própria. Vale ressaltar que aqueles que são usados mais rapidamente do que produzidos passam a ser considerados essenciais.
- **Indispensáveis ou essenciais:** São aqueles que o organismo não consegue produzir por conta própria e, portanto, é necessária a ingestão de certos alimentos que supram sua deficiência; alimentos esses que provem exclusivamente de origem animal, uma vez que os de origem vegetal são deficientes em um ou mais aminoácidos.

Ainda existem os condicionalmente indispensáveis que são aqueles que são essenciais em alguma parte do desenvolvimento e crescimento ou em situações específicas, como uma doença. A figura 1 traz os nomes dos aminoácidos essenciais e não essenciais.

Figura 1: Aminoácidos Essenciais e Não Essenciais.

Aminoácidos	
Essenciais	Não-Essenciais
Fenilalanina	Ácido Aspártico
Histidina	Ácido Glutâmico
Isoleucina	Alanina
Leucina	Arginina
Lisina	Asparagina
Metionina	Cisteína
Treonina	Glicina
Triptofano	Glutamina
Valina	Prolina
	Serina
	Tirosina

(fonte: http://diogobiologo.blogspot.com.br/2011_12_01_archive.html)

2.2.3. Gorduras

Para Costa et al (2008), os lipídios, diferente dos carboidratos e proteínas, não são polímeros, ou seja, suas moléculas não são compostas por repetições de uma unidade básica (aminoácidos no caso da proteína, monossacarídeos no caso do carboidrato). Possuem importantes funções biológicas como:

- Reserva de energia: Apesar do carboidrato ser a principal fonte de energia do organismo, a gordura é o macro nutriente que possui mais energia em sua composição, cada 1g possui 9 kcal.
- Componente estrutural das membranas.
- Isolamento térmico, elétrico e mecânico para proteção de células e órgãos.
- Participam da formação de alguns hormônios sexuais.

2.3. Quantidades

Para Mura e Silva (2011, p. 246): “As necessidades nutricionais devem ser minuciosamente calculadas com o auxílio de protocolos apropriados à idade, sexo, características hereditárias, peso e composição corporal, condicionamento físico, tipo e fase de treinamentos de atletas e praticantes de atividade física”. Sabendo disso, é importante destacar que os dados apresentados neste capítulo bem como em todos deste trabalho relacionados à nutrição são aproximações, e que os estudos relacionados a esse assunto ainda não chegaram num consenso, uma vez que fontes diferentes não apontam para quantidades exatamente iguais. Portanto, o acompanhamento e consulta de profissionais continuam sendo insubstituíveis.

As recomendações para pessoas ativas na musculação apontam para 1.4-2.1 g/kg de proteínas diárias. Mais importante do que a quantidade é o tipo e quando tais nutrientes serão ingeridos. Vale ressaltar que elevar as quantidades apresentadas para 2,5-3.0 g/kg, por exemplo, será um completo desperdício, uma vez que a síntese protéica é limitada (MOREIRA, 2010). Já Mura e Silva mencionam que a recomendação é de 1,4-1,8 g/kg e que, apesar do que muitos atletas e treinadores sugerem, não há evidências científicas de que dietas com quantidades muito altas de proteína resultarão num ganho de massa adicional ou promoverão aumento do desempenho. (MURA; SILVA, 2011).

Quanto às gorduras, pouco se sabe sobre quantidades, porém elas estão associadas aos níveis de testosterona do corpo, um dos principais hormônios relacionados ao anabolismo, o que torna a gordura um macro nutriente essencial na dieta. (MOREIRA, 2010). Mura e Silva indicam que uma dieta voltada para praticantes de atividade física não deve ultrapassar a razão de 1g para cada kg de massa corporal. Dietas restritas em lipídios podem ocasionar em hipovitaminoses (falta de vitaminas no corpo). Por outro lado, seu consumo elevado leva a perda de desempenho. (MURA; SILVA, 2011).

Além da ingestão protéica, a ingestão energética é um fator determinante para obter resultados. Quando o objetivo é hipertrofia muscular, a recomendação é seguir uma dieta hiperenergética. Uma dieta se diz hiperenergética quando possui uma quantidade de calorias superior a apresentada pelo Gasto Energético Total (GET). Este cálculo consiste na soma da TMB com o gasto das atividades físicas feitas durante o dia e o efeito térmico dos alimentos. Foram criados multiplicadores (gradientes) da TMB a fim de facilitar seu cálculo. Os multiplicadores variam de 1.1 (pouco ativo) a 2.0 (muito ativo). (MACHADO). A quantidade de energia excedente varia de acordo com o sexo do praticante, porém, uma alimentação com 300-1000 kcal a mais do que o que é fornecido pelo GET é considerada hiperenergética.

2.4. Consumo de Água e sua Importância para o Organismo

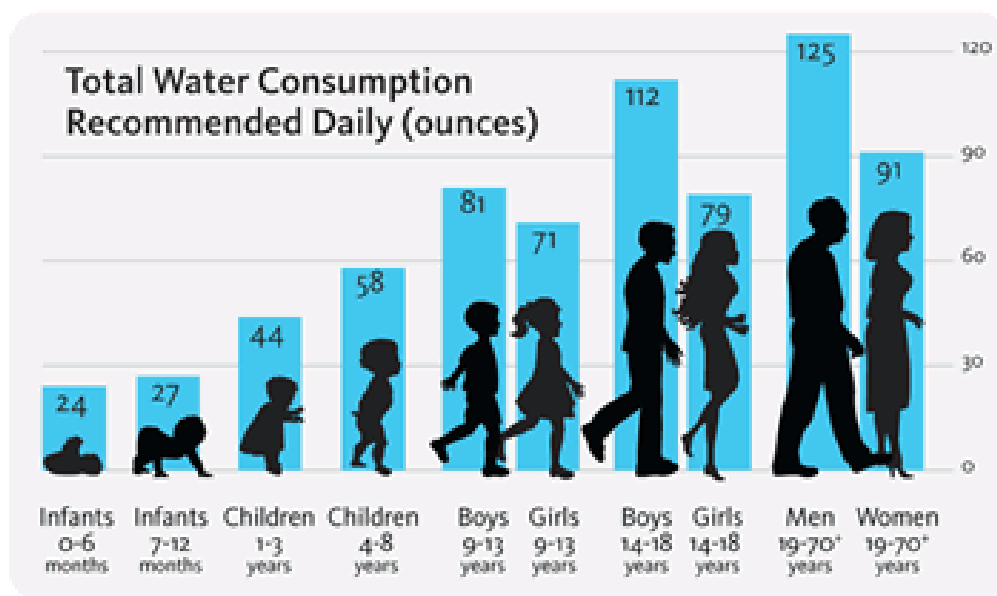
Sabe-se que a água é o constituinte mais abundante de todos os tecidos. No organismo humano, ela representa, em geral, cerca de 60% do peso corporal de um adulto. Essa porcentagem varia de acordo com idade, gênero, composição corporal etc. A gordura do corpo possui de 10 a 40% de água enquanto a massa magra possui de 70 a 75%, ou seja, indivíduos com uma maior quantidade de massa magra necessitam de mais água. (MURA; SILVA, 2011). É uma substância essencial para todos os tecidos do organismo. Possui inúmeras funções, dentre elas, destacam-se:

- Participante de todas as reações químicas do corpo.
- Transportador de todas as substâncias (nutrientes, toxinas e resíduos metabólicos).
- Essencial para a digestão, uma vez que absorve e excreta os resíduos metabólicos e substâncias nocivas em excesso.
- Termorregulador.

De 20 a 25% da água total ingerida vem através do consumo de alimentos e 75 a 80% vem de água pura e bebidas. (MURA; SILVA, 2011).

A molécula da água possui uma instabilidade muito grande e o corpo humano não possui um compartimento capaz de armazená-la. Sendo assim, a reposição hídrica se faz necessária todos os dias. A recomendação do quanto consumir é bastante difícil de ser estabelecida, pois depende de vários fatores como: gênero, idade, composição da dieta etc. Apesar disso, foram formuladas recomendações pelo *Institute of Medicine* que são ilustradas na Figura 2 (observação: 1 *ounce* é aproximadamente 0,03 litros) (MURA; SILVA, 2011).

Figura 2: Consumo Total de Água Diária Recomendada



(fonte: <http://livehealthy.aquasana.com/2012/03/how-much-water-should-i-drink/>)

Além dessa referência, foi estabelecida uma razão de acordo com o consumo calórico. Para adultos, por exemplo, a recomendação é 1 *mL* para cada *kcal*.

2.5. Dispositivos Móveis

Atualmente, a constante necessidade das pessoas contarem com o acesso imediato às informações pessoais e à internet fez com que a indústria da computação trabalhasse em cima de equipamentos que suprissem a necessidade do consumidor. Tais equipamentos chamados de dispositivos de computação móvel sofreram, ao longo dos anos, uma grande transformação.

Segundo os estudos de Gonçalves (2012), os dispositivos móveis se dividem em, basicamente, 4 grupos:

- *Notebooks* e *Netbooks*: são dispositivos portáteis com a capacidade de processamento equivalente a um computador pessoal;
- PDAs (*Personal Digital Assistants*): capazes de rodar aplicativos desenvolvidos em linguagem de alto nível, possuem recursos de multimídia e são capazes de fazer conexões com redes sem fio (*wi-fi*), porém com processamento menor que *notebooks*;

- Celulares: têm acesso à rede *bluetooth*, possuem tamanho reduzido e recursos de multimídia, porém com uma capacidade de processamento reduzida. Atualmente estão praticamente obsoletos;
- *Smartphones e Tablets*: são celulares com recursos de PDAs. O *tablet* difere do *smartphone* na capacidade de processamento e tamanho. Apesar de antes ser uma tecnologia exclusiva do *tablet*, hoje ambos possuem tela sensível ao toque.

2.5.1. Plataformas para Desenvolvimento

Devido ao crescimento da necessidade de se ter um dispositivo móvel nos dias de hoje (especialmente *smartphone* e/ou *tablet*), diferentes empresas de tecnologia passaram a investir no setor a fim de conquistar sua parcela no mercado. O Quadro 3 apresenta as principais plataformas da história desses dispositivos e suas principais características:

Quadro 3: Principais Plataformas dos Dispositivos Móveis

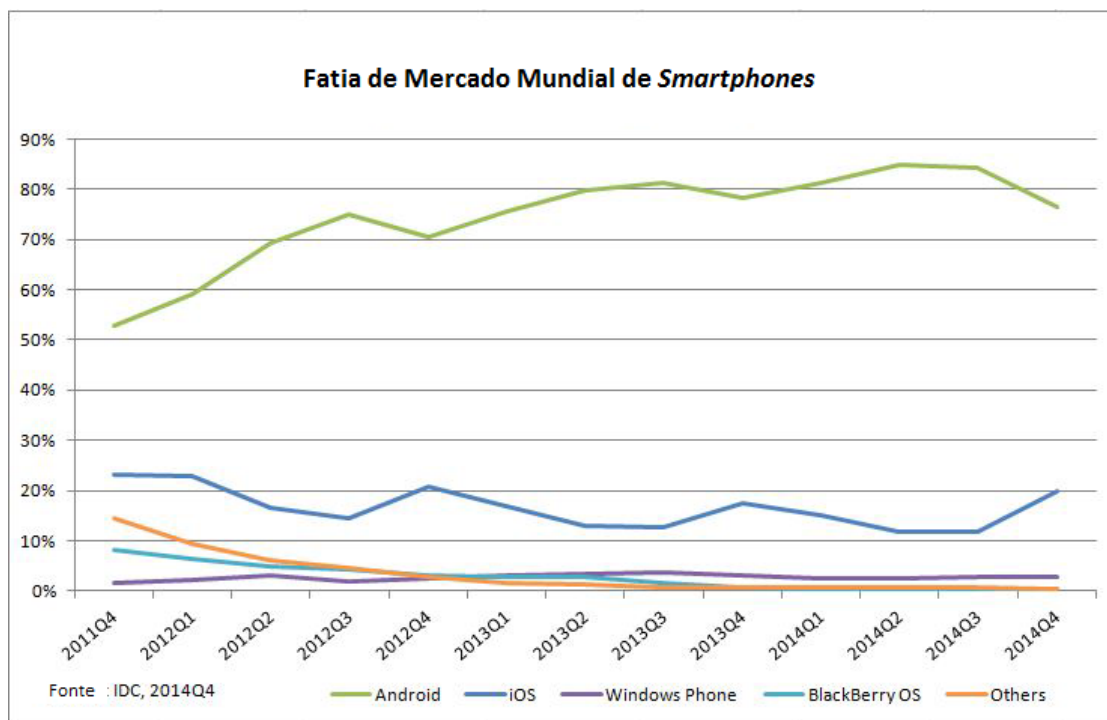
	Android	iOS	Windows Phone	Blackberry	Symbian (obsoleto)
Mantenedora	Google	Apple	Microsoft	RIM	Nokia
Desenvolvimento	Java, C, Javascript	Objective C	C#	JavaME, BB API	C++, Ruby, .NET
Código Aberto	Sim	Não	Não	Não	Sim
Kernel	Linux	Mac OSX	Windows	BlackBerry OS	EPOC
Loja de aplicativos	Play Store	App Store	Windows Store	App World	OVI

(fonte: Elaborado pelo autor)

De acordo com pesquisas feitas pelo IDC (*International Data Corporation*), o mercado de *smartphones* cresceu 27.2% no segundo trimestre de 2014 se comparado ao mesmo período do ano anterior. Os produtos comercializados são, em sua grande maioria,

vinculados ao *Android*, cerca de 85%, seguido pelo *iOS* com aproximadamente 10% (fonte: <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>). A Figura 3 ilustra a fatia no mercado de cada plataforma (de 2011 a 2014).

Figura 3: Fatia de Mercado Mundial de *Smartphones*



(fonte: Adaptado de <http://www.idc.com/prodserv/smartphone-os-market-share.jsp>)

2.6. Banco de Dados

Todo aplicativo/programa que exige armazenamento de dados precisa de uma estrutura de armazenamento. Estruturas essas que se dividem em, basicamente: arquivos e banco de dados. É de conhecimento geral que para aplicações que movimentam muitas informações, banco de dados é a melhor pedida, por ser uma estrutura mais robusta, oferecer mais segurança e ser mais íntegro.

Segundo Date (2004), “Um banco de dados é uma coleção de dados persistentes, usadas pelos sistemas de aplicação de uma determinada entidade”. Por persistentes, ele quer dizer que assim que os dados entram no banco, eles só saem por meio de uma requisição explícita do SGBD¹. Um banco de dados consiste de tabelas e tabelas consistem de atributos.

¹ É um conjunto de softwares que fica entre o banco de dados físico e os usuários do sistema. São eles que tratam todas as requisições de acesso ao banco de dados, garantem sua segurança etc.

Para pesquisas e manipulação é utilizada uma linguagem universal, a chamada SQL (Linguagem de Consulta Estruturada).

O *Android* fornece uma ferramenta simples e eficiente capaz de criar e manipular um banco de dados integrado com a aplicação: o denominado *SQLite*. Esta ferramenta possui a classe *SQLiteDatabase* que por sua vez contém métodos para interagir com o suposto objeto. (PEREIRA; SILVA, 2009).

2.7. HTTP

HTTP significa *Hyper Text Transfer Protocol* (em português, Protocolo de Transferência de Hipertexto). Trata-se de um protocolo de rede responsável pela transferência de dados e pela comunicação entre cliente e servidor no ambiente da Internet, o qual permite a transferência de dados em hipermídia (texto, imagem e som). Segundo Mitchell (2013), o HTTP é composto de várias partes: o endereço de rede onde se encontra o recurso requisitado (URL), o método (ou verbo) usado, alguns cabeçalhos (*headers*), códigos de status (*status code*) e o corpo das respostas (o que normalmente é mostrado na tela). Um usuário comum navega pela web clicando em *links* e abrindo imagens sem o conhecimento do que realmente acontece. Basicamente, clicar em um *link* implica em uma requisição/resposta. O cliente (navegador) faz uma requisição para o servidor e este manda uma resposta.

Quando um usuário faz uma requisição (<http://google.com>, por exemplo), o que ele vê é somente o corpo da resposta do HTTP. Porém, como citado anteriormente, o protocolo é composto, também, por *headers*, *status code* e verbo. A Figura 4 ilustra as outras partes de uma suposta requisição HTTP.

Figura 4: Requisição HTTP

Request header:

```
GET / HTTP/1.1
User-Agent: Mozilla/5.0 (X11; Linux i686) AppleWebKit/537.8 (KHTML, like Gecko)
Chrome/23.0.1246.0 Safari/537.8
Host: oreilly.com
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-GB,en-US;q=0.8,en;q=0.6
```

Response header:

```
HTTP/1.1 200 OK
Date: Thu, 15 Nov 2012 09:36:05 GMT
Server: Apache
Last-Modified: Thu, 15 Nov 2012 08:35:04 GMT
Accept-Ranges: bytes
Content-Length: 80554
Content-Type: text/html; charset=utf-8
Cache-Control: max-age=14400
Expires: Thu, 15 Nov 2012 13:36:05 GMT
Vary: Accept-Encoding
```

(fonte: Mitchell, 2013, p.2)

Através da Figura 4, é possível observar que o método utilizado foi o GET e o *status code* da resposta foi 200, além de muitas outras informações úteis como: *host*, servidor, tipo de conteúdo etc.

2.7.1. Verbos

O verbo (ou método) é uma parte muito importante do protocolo, pois é ele quem consegue transmitir as intenções da requisição. Eles se resumem em, basicamente, dois: GET e POST (apesar de existirem outros tipos, este trabalho foca apenas nos citados).

De acordo com Mitchell (2013), o método GET possui umas particularidades como:

- Suas requisições podem ser cacheadas no navegador;
- Permanecem no histórico do navegador;
- Podem ser incluídas na lista de favoritos, pois as requisições feitas por esse método são enviadas via URL;
- Possuem restrições quanto ao tamanho da requisição;
- Nunca devem ser usadas para fornecer conteúdo.

Enquanto o método POST:

- Suas requisições nunca são cacheadas;
- Não ficam no histórico do navegador;
- Não podem ser "favoritadas", pois as requisições feitas por esse método são enviadas no corpo da requisição;
- Não possuem restrições de tamanho.

A partir das características apresentadas, é possível definir quando e onde é adequado usar o método POST, assim como pro método GET. Por exemplo, requisições de busca devem ser feitas via GET, pois o usuário tem a possibilidade de incluí-las na lista de favoritos. Já o método POST pode ser usado para cadastro, uma vez que é considerado um método mais seguro.

3. Materiais e Métodos

3.1. Materiais

Este sub capítulo consiste de uma breve introdução dos materiais e tecnologias utilizados para o desenvolvimento bem como as justificativas das escolhas.

3.1.1. *Android*

Segundo Pereira, Silva (2009) “O *Android* é uma plataforma para tecnologia móvel completa, envolvendo um pacote com programas para celulares, já com um sistema operacional, *middleware*², aplicativos e interface do usuário”. Ele é o primeiro projeto de uma plataforma *open source* para dispositivos móveis em conjunto com a *Open Handset Alliance* (OHA)³. Foi criado com a finalidade de fornecer aos desenvolvedores as ferramentas necessárias para explorar todas as funções do celular. Por exemplo, um aplicativo pode fazer chamadas, utilizar a câmera, mandar mensagens, utilizar o GPS etc. Por ser *open source*, pode ser sempre adaptado e melhorado de acordo com a ascensão de novas tecnologias, ou seja, sempre estará em evolução. Foi desenvolvido com base no sistema operacional Linux.

Apesar de *open source*, o Android tem uma capacidade de segurança muito forte. Como é executado num núcleo Linux, toda vez que um aplicativo é instalado, cria-se um novo usuário Linux especialmente para aquele programa, com diretórios que serão usados pelo aplicativo, mas somente para aquele usuário Linux. Como os aplicativos ficam isolados uns aos outros, qualquer tentativa de acesso a informações de outro aplicativo é barrada na negação (ou não) do usuário. Caso uma aplicação faça uso do GPS, por exemplo, é preciso estar explícito para o usuário tal condição.

3.1.1.1. Arquitetura

O *Android* possui uma arquitetura dividida em 5 camadas: Aplicações, *Framework* de Aplicação, Bibliotecas, *Android Runtime* e Linux *Kernel*, conforme ilustrado na Figura 5.

² É um mediador que permite a comunicação entre diferentes sistemas ou programas independente de sistema operacional ou protocolos de comunicação.

³ Consiste de todas as empresas envolvidas no processo de telefonia móvel: operadoras de celular, fabricantes de aparelhos, empresas de software, empresas de comercialização etc.

Figura 5: Arquitetura *Android*



(fonte: PEREIRA; SILVA, 2009, p.5)

A seguir é explicado brevemente o papel de cada camada.

3.1.1.1.1. Aplicações

O *Android* inclui em seu sistema operacional alguns aplicativos que interagem com o celular e dão a funcionalidade básica do sistema. Entre eles estão: navegadores, programa de SMS, gerenciador de contatos, agenda, calendários etc. (PEREIRA; SILVA, 2009).

3.1.1.1.2. Framework de Aplicação

Framework é um conjunto de classes que incorporam um *design* abstrato a fim de prover uma solução para uma família de problemas semelhantes.

Sendo assim, o *Framework* de Aplicação do *Android* contém todas as APIs (Interface de Programação de Aplicativos) e recursos utilizados pelos aplicativos, as chamadas classes visuais (listas, grades, caixas de texto). Possui também o provedor de conteúdo (*Content Provider*), que faz com que uma aplicação possa acessar ou até mesmo compartilhar

informações de outra. Gerenciador de localização e notificação etc. (PEREIRA; SILVA, 2009).

3.1.1.1.3. Bibliotecas

A camada de bibliotecas possui um conjunto de bibliotecas C/C++ que acabam por poupar trabalho do desenvolvedor. Algumas rotinas fundamentais como: abertura de arquivo, mostrar mensagens, cálculo aritmético já estão implementadas. Possui também bibliotecas das áreas de multimídia, funções para gráficos, funções para navegadores web e muitos outros utilitários (PEREIRA; SILVA, 2009).

3.1.1.1.4. *Android Runtime*

Assim como o Java, o *Android* possui uma máquina virtual. Esta chamada de *Dalvik Virtual Machine* (DVM) é responsável por rodar todos os aplicativos no celular. Por se tratar de dispositivos móveis cujo desempenho é extremamente limitado se comparado a computadores pessoais, ela precisou se adaptar a tais limitações, o que acabou tornando-a muito eficiente, pois consegue trabalhar em sistemas com baixa frequência de UCP (processador) e pouca memória RAM disponível. Além disso, a Dalvik é otimizada para consumir pouca bateria.

O código do aplicativo é compilado em um *.class*, depois transformado em arquivo binário (*.dex*) que é interpretado pela Dalvik. Quem faz essa transformação é uma ferramenta chamada DX que é disponibilizada no SDK do *Android* (Kit de Desenvolvimento de Software). É importante apontar que é exatamente por isso que os aplicativos podem ser rodados em qualquer dispositivo *Android*, independentemente da arquitetura do seu processador. (PEREIRA; SILVA, 2009).

3.1.1.1.5. *Linux Kernel*

Utiliza a versão 2.6 do *kernel* (núcleo) do Linux. O núcleo de um SO (sistema operacional) atua como uma camada de abstração entre o hardware e as camadas seguintes da arquitetura. É ele que trata dos serviços fundamentais do sistema. Serviços como: segurança, gestão de memória, gestão de processos, modelo de *drivers* e pilha de protocolos de rede. Um

recurso interessante encontrado nesse *kernel* é o *Binder*. Este consiste num mecanismo de comunicação de processos. (PEREIRA; SILVA, 2009).

3.1.1.2. Requerimentos para Desenvolvimento

Como é de conhecimento geral, as aplicações *Android* são desenvolvidas em cima da plataforma Java, conseqüentemente, se faz necessária a instalação dela, do JDK (*Java Development Kit*), que contém o compilador, *debugger* e outras ferramentas para o desenvolvimento do software, bem como a JRE (*Java Runtime Environment*), responsável por executar tais ferramentas.

Atualmente, a fim de poupar o tempo do programador com instalações e reduzir a chance de erros, a Google disponibiliza o *Android Development Tools – Bundle* (ADT). Como o próprio nome diz, ele é um pacote que contém os componentes essenciais do SDK (*Software Development Kit* - será explicado mais detalhadamente em capítulos seguintes); a IDE Eclipse; a última plataforma *Android* etc. Nota-se que antes do ADT *Bundle*, todos os componentes eram instalados manualmente.

3.1.1.3. Configurando um Aparelho para Desenvolver

Quase todos os aparelhos *Android* podem ser usados para desenvolvimento. Para habilitá-los, é necessário acessar a área desenvolvedor/programador (o local específico onde tal área se encontra depende da versão do SO, porém ela normalmente se encontra dentro de Configurações) e habilitar a opção Depuração USB (USB Debugging). Em seguida é necessária a instalação do *driver* do ADB (*Android Debug Bridge*). Esta consiste em uma ferramenta que permite a comunicação com um emulador ou aparelho *Android*. Trata-se de um programa que inclui 3 componentes: cliente, que roda na máquina de desenvolvimento; servidor, que é executado como um processo em segundo plano na máquina de desenvolvimento; *daemon*, que roda como um processo em segundo plano no aparelho ou emulador.

3.1.1.4. SDK

Como definido anteriormente, o SDK (*Software Development Kit*) é essencial para o desenvolvimento. Ele consiste de um monte de componentes de software agregados. Normalmente inclui várias bibliotecas, documentação da API, códigos de amostra, IDE etc.

O *Android* SDK tem uma particularidade: possui uma estrutura modular, ou seja, é composto de pacotes. Isso implica em benefícios, uma vez que o programador pode instalar apenas os componentes necessários para sua aplicação, desperdiçando menos espaço. É dividido em:

- *SDK Tools*: contém ferramentas para *debug* e testes mais outros utilitários requeridos na construção de um aplicativo.
- *Platform Tools*: são ferramentas adicionais que são atualizadas conforme a versão do *Android*.
- *SDK Platform*: há uma *SDK Platform* pra cada versão do SO. Ela inclui um *android.jar* que contém uma biblioteca *Android*.
- *Google APIs*: bibliotecas necessárias para utilizar os recursos exclusivos do *Google*, i.e, *Google Cloud Messaging* ou *Maps*.
- *Samples and Documentation*: contém amostras de projetos com suas respectivas documentações para cada versão da plataforma (PEREIRA; SILVA, 2009).

3.1.1.5. User Interface - Layout

Assim como em páginas web, a interface com o usuário em um aplicativo é determinante para sucesso ou fracasso do mesmo. O layout de uma aplicação *Android* é criado a partir de um mecanismo simples e intuitivo: arquivos XML (*Extensible Markup Language*). Por serem separados do código da aplicação, o desenvolvedor pode mudar o *layout* sem ter que recompilar ou editar a lógica do programa. Ele é feito através da hierarquia entre dois objetos:

- *View*: normalmente são os *widgets* (caixa de texto, botões).
- *ViewGroups*: objetos invisíveis (*containers*, *forms*) que definem como os objetos filhos (*Views*) serão exibidos na tela (PEREIRA; SILVA, 2009).

A Figura 6 mostra um arquivo XML básico para uma aplicação *Android*. É possível notar a hierarquia entre os objetos e a facilidade de entendimento.

Figura 6: Layout para Uma Aplicação Android

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">
    <EditText android:id="@+id/edit_message"
        android:layout_weight="1"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:hint="@string/edit_message" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button_send" />
</LinearLayout>
```

(fonte: Documentação oficial do *Android*. Disponível em <http://developer.android.com/guide/topics/ui/declaring-layout.html>)

3.1.1.6. Activity

Segundo o Android Developers (2015), *Activity* é o componente principal de uma aplicação. Ela é composta de uma tela onde é desenhada uma interface. O usuário pode interagir com essa interface através de ações como, por exemplo, enviar email, tirar foto, visualizar mapa etc. O fluxo de uma aplicação *Android* começa por uma *activity*.

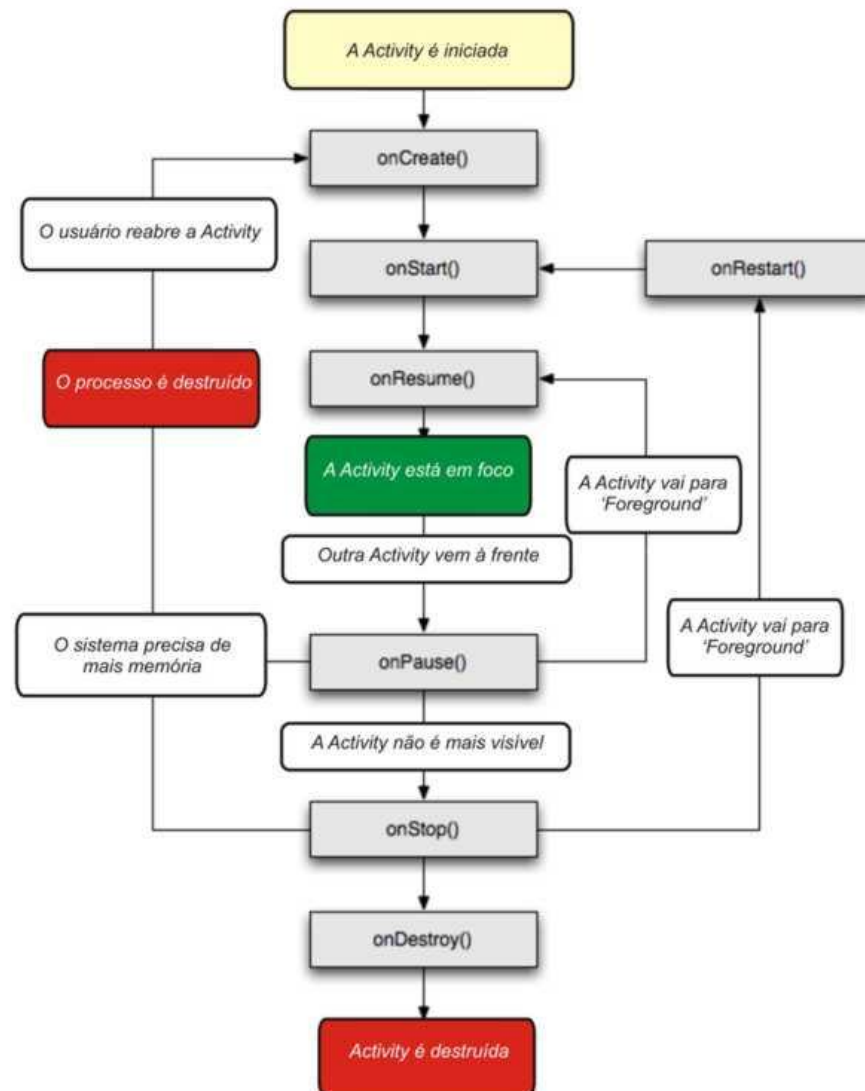
A fim de criar uma *activity*, é necessário criar uma subclasse da classe *Activity* e implementar alguns métodos. Tais métodos compõem o ciclo de vida da *activity*. Sabendo disso, alguns deles se tornam obrigatórios dependendo do comportamento da *activity* no sistema. São eles:

- *onCreate*: esse método é o único que deve ser implementado em todas as ocasiões, pois é chamado quando a *activity* está sendo criada. Para se ter uma idéia de sua importância, é nele que é definido qual *layout* será utilizado para a *activity* em questão;
- *onRestart*: chamado quando a *activity* foi pausada e está prestes a ser chamada novamente. Esse método é sempre precedido pelo *onStart*;
- *onStart*: chamado quando a *activity* se torna visível para o usuário;
- *onResume*: chamado quando a *activity* está prestes a se interagir com o usuário;
- *onPause*: chamado quando o sistema está prestes a retornar a *activity* anterior;

- *onStop*: chamado quando a *activity* não está mais visível para o usuário;
- *onDestroy*: chamado antes da *activity* ser destruída. A razão da chamada pode ser ou liberação de espaço ou término da mesma.

A Figura 7 ilustra o ciclo de vida de uma *activity* e a interação entre os métodos.

Figura 7: Ciclo de Vida de Uma Activity



(fonte: <http://www.felipesilveira.com.br/2010/05/activity-o-que-e-isso/>)

3.1.1.7. *Fragment*

De acordo com o Android Developers (2015), os *Fragments* representam um comportamento ou um pedaço de interface de uma *activity*. É possível combinar vários *fragments* dentro de uma *activity*. O intuito em usá-los no sistema se deve ao fato de serem

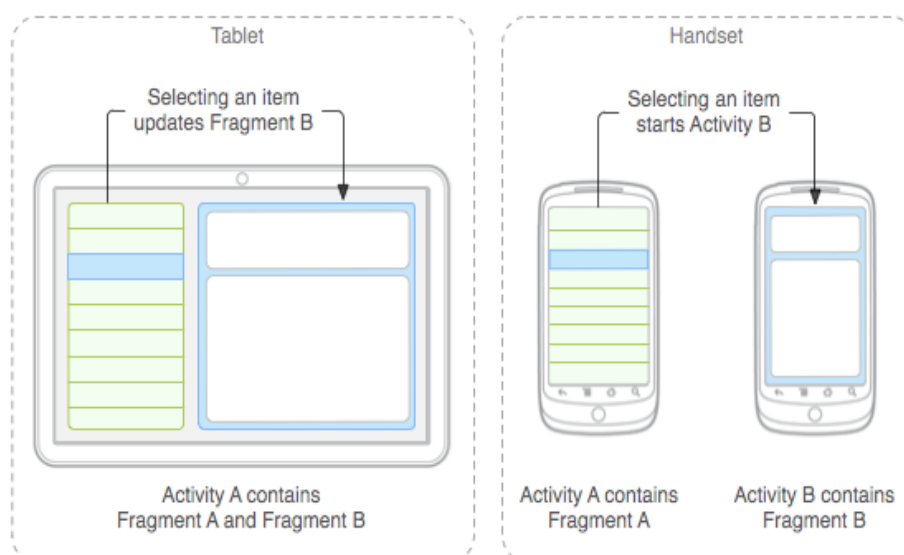
reutilizáveis, ou seja, um mesmo *fragment* pode ser incorporado em várias *activities*. É importante notar que, apesar de ter um ciclo de vida próprio (muito similar ao de uma *activity*, com métodos de chamada do tipo: *onCreate*, *onPause*, *onStart* etc), um *fragment* depende totalmente do ciclo de vida de sua respectiva *activity*. Por exemplo, se uma *activity* é pausada, todos os *fragments* relacionados a ela serão pausados também.

Esse componente foi introduzido na versão 3.0 para suportar interfaces mais flexíveis e dinâmicas em dispositivos com telas grandes, como os *tablets*. Os *fragments* permitem fazer o *design* de telas sem a necessidade de mudanças complexas no código. Ao dividir o *layout* de uma *activity* em vários desses componentes, fica muito mais fácil e dinâmico modificar sua aparência.

Em suma, os *fragments* devem ser projetados de forma modular e reutilizável, pois isso possibilita o desenvolvimento de aplicativos com suporte para vários *layouts* diferentes, ou seja, o posicionamento desses elementos são diferentes de acordo com o tamanho da tela do dispositivo.

A imagem da Figura 8 representa um exemplo de uso dos *fragments*. O *tablet*, por possuir uma tela maior, comporta os dois *fragments* numa única *activity*. Já o *handset*, por possuir uma tela com tamanho reduzido, necessita de duas *activities* para mostrar os dois *fragments* em questão.

Figura 8: Exemplo de Utilização dos *Fragments*

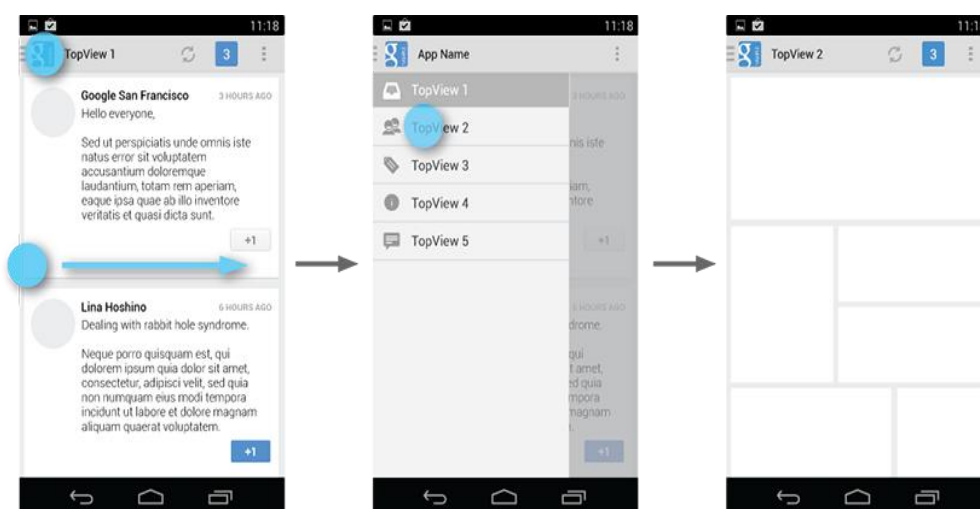


(fonte: Documentação oficial do *Android*. Disponível em <http://developer.android.com/guide/components/fragments.html>)

3.1.1.8. *Navigation Drawer*

É um tipo de menu muito utilizado nos dias de hoje. Ele é representado por 3 traços horizontais. Acaba por poupar muito espaço da tela, uma vez que inicialmente fica escondido e só é acionado quando o usuário toca no ícone ou o arrasta da lateral da tela. De acordo com o Android Developers (2015), é recomendado quando o menu contém mais de 3 itens. A Figura 9 demonstra um exemplo de *navigation drawer*.

Figura 9: Funcionamento do *Navigation Drawer*



(fonte: Documentação oficial do Android. Disponível em <https://developer.android.com/design/patterns/navigation-drawer.html>)

É importante notar que esse componente faz uso dos *fragments* em sua construção. Basicamente, a tela em si (primeira imagem da figura 9) é uma *activity* que contém um layout, o menu (inicialmente escondido) é um *fragment* dessa *activity* e, ao entrar em um dos itens do menu (segunda imagem), a aplicação procura pelo *fragment* correspondente (terceira imagem). Ou seja, trata-se de uma única *activity* que varia de *fragment* de acordo com a escolha do usuário.

3.1.2. MySQL

Com mais de 10 milhões de instalações registradas (incluindo alguns líderes da indústria como Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia e Booking.com – fonte tirada do site mysql.com), o MySQL é provavelmente o SGBD mais popular para

servidores web nos dias de hoje (NIXON, 2014). Hoje é desenvolvido e distribuído pela Oracle. Seu sucesso se deve a algumas características. Segundo (SUEHRING, 2002):

- Roda em várias plataformas;
- Possui um baixo TCO (Custo total de posse);
- Documentação intuitiva;
- Consegue rodar no mais básico dos hardwares;
- A licença do *software* é gratuita.

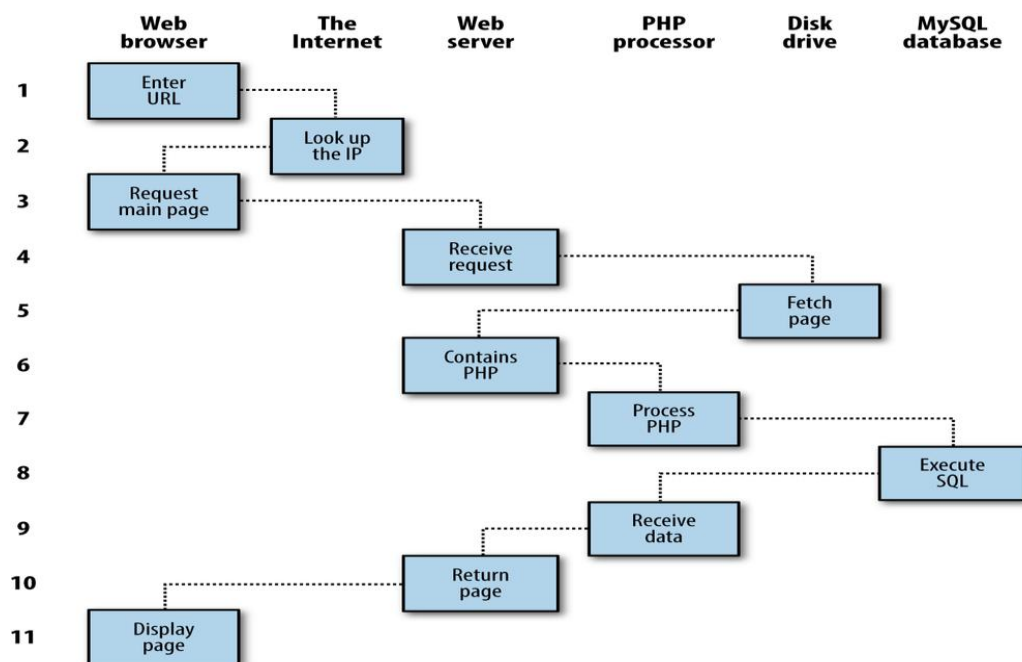
3.1.3. PHP

Segundo o PHP.NET, PHP é uma linguagem de programação de ampla utilização, interpretada, (ou seja, o código não é compilado, o interpretador o executa diretamente) especialmente interessante para desenvolvimento para a *web*. Seu objetivo principal é gerar páginas dinamicamente.

O capítulo anterior tratou de conceitos do protocolo HTTP, porém não foi explicitado como fazer e tratar uma requisição. A linguagem PHP é uma das formas (talvez a mais famosa hoje em dia). Quando fala-se de uma requisição entre cliente e servidor, o PHP pode estar ao lado do cliente e/ou do servidor. Para melhor compreensão, um exemplo do PHP sendo o cliente seria a sua utilização na busca pelos *tweets* mais recentes. Pegando o mesmo exemplo, o PHP seria a parte do servidor se ele fosse o responsável por mandar os *tweets* mais recentes para um suposto cliente.

A Figura 10 ilustra o papel do PHP no processo de retornar uma página dinâmica ao usuário.

Figura 10: Função do PHP em páginas dinâmicas



(fonte: NIXON, 2014, p.4).

1. Usuário entra com `http://server.com` no browser.
2. *Browser* procura o endereço IP relacionado a url.
3. *Browser* faz uma requisição para esse endereço.
4. Servidor recebe a requisição.
5. É procurada a página relacionada no disco do servidor.
6. O servidor percebe que existe código PHP no arquivo e passa o mesmo para o interpretador.
7. O interpretador executa o código.
8. O código do arquivo contém declarações MySQL. Sendo assim o banco de dados entra em ação.
9. MySQL retorna os resultados de banco ao interpretador PHP.
10. O interpretador retorna os resultados do código executado juntamente com os resultados do banco pro servidor.
11. Servidor retorna a página ao usuário.

Assim como o MySQL, existem várias características que justificam o uso massivo do PHP hoje em dia:

- Roda em várias plataformas;

- Compatível com quase todos os servidores usados hoje em dia;
- Gratuito;
- Fácil de aprender.

3.2. Implementação

O diagrama de caso de uso (apêndice B) e o MER (apêndice A) conseguem transmitir uma idéia básica da aplicação. Este capítulo apresenta uma abordagem mais detalhada e técnica de seu funcionamento.

3.2.1. Comunicação com o Banco de Dados

Como todos os dados da aplicação são armazenados remotamente num banco de dados MySQL, optou-se por utilizar a linguagem PHP e o protocolo HTTP como meio de transferência e comunicação com o BD.

Basicamente, o *Android* manda uma requisição HTTP para uma URL, o arquivo PHP referente àquele endereço trata os dados e faz a comunicação com o banco. O banco retorna para o arquivo PHP, que por sua vez manda a resposta para a aplicação (vide capítulo 3.1.3, com a ressalva de que, no trabalho em questão, o cliente é o aplicativo e não o navegador).

O *Android* possui uma classe chamada *HttpURLConnection* que executa requisições para uma dada URL. Nela é possível estabelecer qual método (verbo) utilizar. Este trabalho buscou obedecer as boas práticas apresentadas no capítulo 2.7.1.

A documentação oficial do *Android* sugere que todo processo que envolve a conexão com o banco de dados remoto seja feito de forma paralela, ou seja, em forma de *thread*. Isso pode ser realizado, no caso dessa aplicação, através da classe *PageDownloader* que herda de *AsyncTask*. A Figura 11 mostra a implementação de um dos métodos da classe em questão, o *downloadUrlGET*. Ele faz a requisição com o verbo GET e processa o conteúdo da página php referente a dada URL (o método *downloadUrlPost* também foi implementado. Este é responsável por fazer requisições com o verbo POST).

Figura 11: Método *downloadUrlGET*

```

private String downloadUrlGET(String myurl) throws IOException {
    InputStream is = null;
    int len;

    try {
        URL url = new URL(myurl);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setReadTimeout(10000 /* milliseconds */);
        conn.setConnectTimeout(15000 /* milliseconds */);
        conn.setRequestMethod("GET");
        conn.setDoInput(true);
        // Starts the query
        conn.connect();

        is = conn.getInputStream();
        len = is.available();

        // Convert the InputStream into a string
        String contentAsString = readIt(is, len);
        return contentAsString;

        // Makes sure that the InputStream is closed after the app is
        // finished using it.
    } finally {
        if (is != null) {
            is.close();
        }
    }
}

```

(fonte: Elaborada pelo autor)

O método *onPostExecute* é chamado quando toda a computação é feita. Ele possui um parâmetro *string* que contém o resultado da requisição. No caso dessa aplicação, os resultados consistem de mensagens do tipo "tmb cadastrada, login sucesso, usuário ou senha incorretos" etc. bem como os dados requeridos. *onPostExecute* é composto de várias condições e, dependendo da mensagem de resposta, toma um rumo determinado na aplicação.

A parte do PHP é formada por um arquivo que recebe todas as requisições HTTP do sistema. A comunicação com o banco de dados varia de acordo com os parâmetros passados na requisição em questão. Por exemplo, se o PHP enxerga um parâmetro "*login = 1*", é feita uma consulta na tabela *Users* onde o nome e senha são iguais aos passados pela parte do *Android*.

3.2.2. Log In / Sign In

A primeira tela da aplicação é a de cadastro do usuário. Como o *software* faz uso de um banco de dados MySQL remoto, a conexão com a internet se faz necessária todo tempo, portanto, primeiramente é feita essa verificação. A Figura 12 ilustra o código fonte responsável pela checagem.

Figura 12: Verificação de Conectividade com a Internet

```
public void myClickHandler(View view) {
    ...
    ConnectivityManager connMgr = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo networkInfo = connMgr.getActiveNetworkInfo();
    if (networkInfo != null && networkInfo.isConnected()) {
        // fetch data
    } else {
        // display error
    }
    ...
}
```

(fonte: Documentação oficial do *Android*. Disponível em <http://developer.android.com/training/basics/network-ops/connecting.html>)

Em caso de sucesso, o usuário é direcionado para a tela de *Login* e entra com um *nickname* e uma senha. Ao tocar no botão "cadastrar", é instanciado um objeto da classe *PageDownloader* com a requisição HTTP. Primeiramente é feita uma busca por cadastros com o *nickname* apresentado. Se o BD já tiver o dado nome, é mostrada uma mensagem "Usuário já registrado". Caso contrário, é feita a inserção.

Em caso de falha, o usuário é direcionado para uma tela onde é mostrada uma mensagem "Sem conexão com a internet" e a aplicação fica ociosa.

3.2.3. Home

Depois de logado, o usuário é direcionado para tela *HomeActivity*. Esta é a tela principal do aplicativo e pode ser dividida em 3 componentes principais:

- *Navigation Drawer*: o menu contém 4 itens: *Home*, *Dieta*, *Evolução* e *Consumo de Água*. De acordo com a explicação dada no capítulo 3.1.1.8, esse tipo de menu procura pelo *fragment* correspondente ao item selecionado. No

caso da aplicação, os *fragments* são: *HomeFragment*, *DietFragment*, *EvolutionFragment*, *WaterConsumptionFragment* e o *fragment* correspondente aos próprios itens, *NavigationDrawerFragment*;

- TMB: como já explicado no capítulo 2.1, a TMB possui várias formulas de predição, a aplicação em questão usou a de Harris & Benedict. Como o seu cálculo é fundamental para o funcionamento do aplicativo, deixá-la na tela principal acaba por facilitar sua atualização. O usuário entra com os dados necessários que são guardados na tabela TMB. Seu cadastro e atualização são feitos de acordo com o que foi apresentado no capítulo 3.2.1;
- Você sabia: a fim de agregar conhecimento ao usuário, o "Você Sabia" é uma caixa de texto localizada na parte superior da tela. Seu conteúdo abrange conceitos importantes abordados no capítulo 2 como: o que é proteína, pra que serve? principais funções do carboidrato, conceito de TMB, principais funções da água no organismo etc. Com o intuito de facilitar sua manutenção e possível expansão, o conjunto de trechos é guardado numa tabela *Nutrition_Concepts*. O usuário recebe um conceito aleatório toda vez que vai pra tela *home* do aplicativo.

A Figura 13 mostra como é a *Home* do aplicativo.

Figura 13: *HomeFragment*

(fonte: Elaborada pelo autor)

Textos muito grandes como o mostrado pela figura 13 possuem um *scroll*.

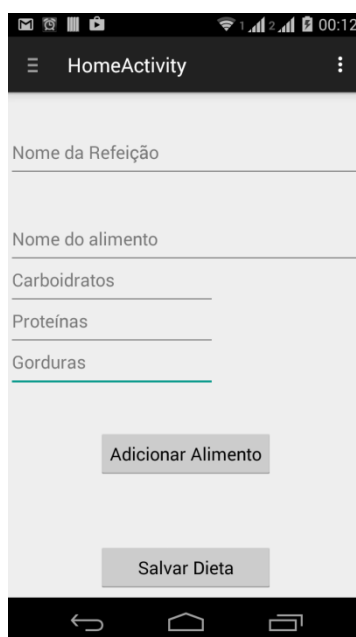
3.2.4. Módulos

Apresentados no capítulo anterior, os módulos da aplicação podem ser divididos em: *Home*, *Dieta*, *Evolução* e *Consumo de Água*.

3.2.4.1. Dieta

No modulo Dieta, o usuário preenche um alimento por vez fornecendo o nome da refeição, nome do alimento, quantidade total de carboidratos, quantidade total de proteínas e quantidade total de gorduras. Ao tocar em "Adicionar Alimento", o *software* guarda as informações em uma variável do tipo *json*⁴ e calcula a quantidade de calorias do mesmo. Isso se repete até o usuário tocar em "Salvar Dieta". Nesse momento, a aplicação retorna quantos gramas de cada macro nutriente a dieta possui, bem como a quantidade de calorias totais. Seguindo as informações apresentadas no capítulo 2.1, o usuário toma conhecimento se as quantidades estão adequadas, além ou abaixo do esperado. A inserção, atualização e exclusão são feitas via HTTP conforme apresentado no capítulo 3.2.1. A Figura 14 apresenta a tela de cadastro de dietas.

Figura 14: Tela de Cadastro de Dietas



(fonte: Elaborada pelo autor)

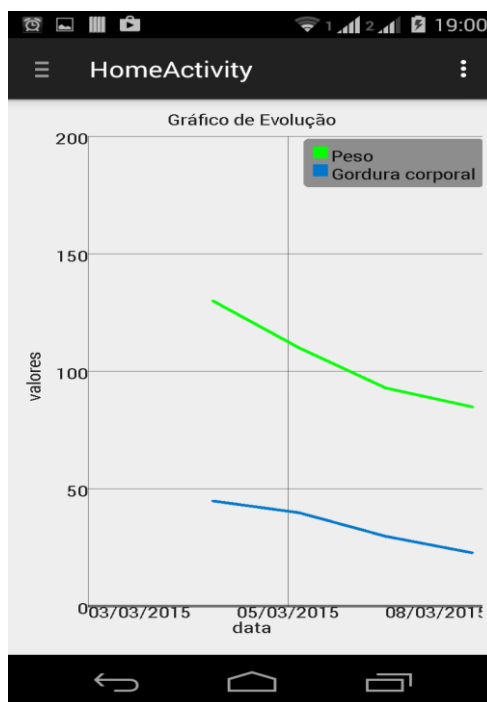
⁴ Trata-se de um tipo de informação de fácil interpretação e escrita. É facilmente analisada por máquinas e as mais diferentes linguagens de programação.

3.2.4.2. Evolução

Informações como "peso" e "porcentagem de gordura" são importantes para se ter uma idéia de progressão. Sabendo disso, o usuário, ao atualizar sua TMB, automaticamente salva esses dados e o aplicativo passa a montar um gráfico baseado nas datas⁵ em que eles foram salvos, ou seja, a coordenada 'x' é preenchida com as datas e a coordenada 'y' com os dados salvos.

O gráfico é gerado com o auxílio da biblioteca *GraphView* (a documentação pode ser encontrada em <http://www.android-graphview.org/>), uma vez que o *Android* não possui uma classe nativa que o faça. A implementação dessa biblioteca é relativamente simples: primeiramente é necessário instanciar um objeto do tipo *GraphView*, que se comporta como a *View* na hierarquia do *layout*. Depois cria-se um (ou mais, dependendo do número de gráficos desejados. No caso da aplicação são 2, um para ilustrar a progressão do peso e outro para gordura corporal) objeto do tipo *LineGraphView* com os pontos do tipo *DataPoint* passados como parâmetro. Como os gráficos são compostos por mais de um ponto, o parâmetro costuma ser um vetor. A Figura 15 ilustra a tela de progressão de um suposto usuário.

Figura 15: Gráfico de Evolução



(fonte: Elaborada pelo autor)

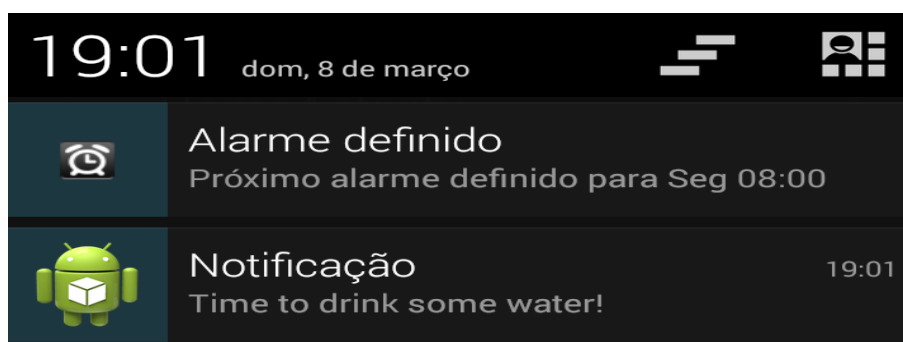
⁵ Basicamente, no momento em que o peso e a porcentagem de gordura são armazenados no banco de dados, é executada a função *NOW()* do MySQL que retorna a data exata da inserção.

3.2.4.3. Consumo de Água

O capítulo 2.4 apresentou as principais funções da água e sua importância para o anabolismo muscular. Também citou que é possível ter uma aproximação do quanto ingerir diariamente sabendo o GET. O módulo Consumo de Água manda notificações ao usuário de acordo com o período escolhido.

Foi utilizada a classe *AlarmManager* na implementação. A fim de notificar o usuário mesmo com o dispositivo em *stand by*, foi escolhido o *AlarmManager* do tipo *ELAPSED_REALTIME_WAKEUP*. Esse tipo de alarme "acorda" o dispositivo no momento da notificação. No caso deste trabalho, o usuário pode configurar a periodicidade das notificações (representadas na figura 16).

Figura 16: Notificação



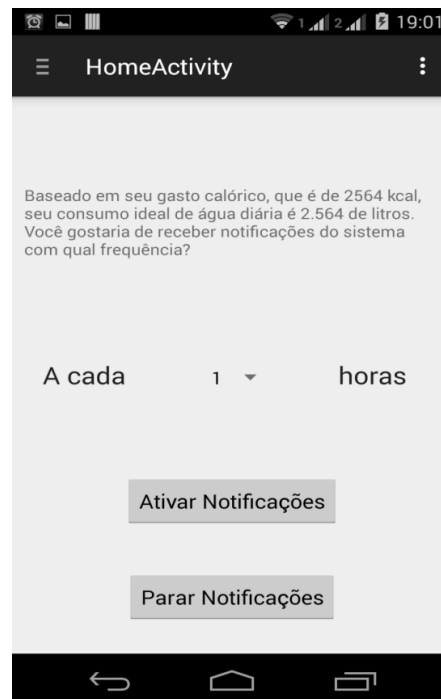
(fonte: Elaborada pelo autor)

Para gerá-las, é necessário implementar uma classe que estenda *BroadcasReceiver*. Um *BroadcastReceiver* é um componente do *Android* que responde para anúncios de *broadcast*⁶ no contexto do dispositivo em geral. Uma aplicação pode iniciar um *broadcast* por meio dessa classe. Por se tratar de uma classe abstrata, é preciso implementar o método *onReceive*.

No caso deste trabalho, *onReceive* faz o celular vibrar e gera uma notificação. A Figura 17 ilustra o *fragment WaterConsumptionFragment*.

⁶ Basicamente é a transmissão de informações em larga escala, ou seja, para vários receptores. No *Android*, os *broadcasts* são feitos no *background* do dispositivo. São comumente utilizados para atualização de banco de dados de uma aplicação.

Figura 17: *WaterConsumptionFragment*



(fonte: Elaborada pelo autor)

4. Considerações Finais

Sabe-se que hoje, mais do que nunca, os dispositivos móveis (especialmente *smartphones* e *tablets*) estão em extrema ascensão no mercado de TI. Sua popularidade é justificável, uma vez que os aplicativos abrangem os mais diferenciados assuntos e acabam por facilitar a vida das pessoas (seja na praticidade, informação ou até mesmo diversão), além de estarem cada vez mais acessíveis financeiramente.

A escolha do tema da aplicação é justificada pela escassez de aplicativos grátis voltados para nutrição no mercado. Foi feita uma pesquisa na *play store* e concluiu-se que, além de escassos, os aplicativos aparentam não fazer uso de muitas das teorias essenciais apresentadas neste trabalho (como o cálculo da TMB, GET etc); além disso, não focam em introduzir tais conceitos aos usuários, ou seja, não têm o intuito de ensiná-los. Apesar básicos e vistos em sua grande maioria no ensino médio, as pessoas não dão a atenção suficiente para o tema ou não são ensinadas de acordo, e como foi apresentado neste trabalho, isso acabou se tornando um problema social nos dias de hoje, vide a obesidade que é considerada uma epidemia.

O projeto consistiu de uma série de pesquisas sobre alguns dos principais conceitos de nutrição aplicados à musculação, bem como uma série de estudos sobre o desenvolvimento na plataforma *Android*. Com a aplicação definida, optou-se por utilizar um banco de dados remoto pensando em futuras modificações.

De acordo com o capítulo 3.2, os objetivos foram alcançados vide capítulo 3.2.4.1 para o módulo Dieta; capítulo 3.2.4.2 para o Gráfico de Evolução; a implementação do Banco de Dados Remoto foi apresentada no capítulo 3.2.1 e o Mecanismo de Notificações no capítulo 3.2.4.3. Sendo assim, o aplicativo comprovou cumprir os objetivos.

Com seu uso, espera-se contribuir com as pessoas que buscam uma vida mais saudável e não tiveram a oportunidade de aprender os assuntos abordados ou não têm condição financeira para contratar um profissional que dê o auxílio necessário.

Espera-se, também, dar continuidade ao trabalho, uma vez que se trata de um protótipo e a aplicação pode agregar muitas outras funcionalidades, como:

- Implementar um módulo Treino, muito parecido com o módulo Dieta;
- Aproveitar o banco de dados remoto e implementar um sistema de *reviews* onde um usuário poderá ver, comentar e avaliar a dieta de outro usuário;
- Aplicar o conceito de *Gamification* (uso de mecanismos de jogos com a finalidade de estimular o usuário);

- Implementar um banco de dados próprio de alimentos;
- Implementar um tipo de integração com redes sociais;
- Melhorar o *design* da aplicação;

Por fim, aspira-se publicar o aplicativo na *Play Store*.

REFERÊNCIAS

ADES, Lia; KERBAUY, Rachel Rodrigues. **Obesidade: Realidades e Indagações**. Psicologia, USP, 2002. Disponível em: <
http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-65642002000100010&lng=pt&nrm=iso#back>

ANDROID DEVELOPERS. *App Components*. 2015. Disponível em: <
<http://developer.android.com/guide/components/index.html> >. Acesso em: 26/02/2015

COSTA, Neuza Maria Brunoro et al. **Nutrição Básica e Metabolismo**. Editora UFV, 2008.

DATE, Christopher J. **Introdução a Sistemas de Banco de Dados**. Elsevier Editora Ltda, 2004. Disponível em: < http://books.google.com.br/books?hl=pt-BR&lr=&id=xBeO9LSIK7UC&oi=fnd&pg=PP23&dq=banco+de+dados+cliente+servidor&ots=xaLxeXDeaG&sig=1qXp_sObMObhagsnW0E5A7IIQtY#v=onepage&q&f=false>.

GONÇALVES, Julio Cesar. **Uso da Plataforma Android em Um Protótipo de Aplicativo Coletor de Consumo de Gás Natural**. UTFPR, 2012

MACHADO, Dalmo Roberto Lopes. **Estimativa do Gasto Energético**. EEFERP, USP. Disponível em: <
<http://sistemas.eeferp.usp.br/myron/arquivos/3396411/316f9dcf5e9599de79d7aa387aa63e6a.pdf>>. Acesso em 16/12/2014

MITCHELL, Lorna Jane. **PHP Web Services**. O'Reilly, 2013.

MOREIRA, Bruna. **Hipertrofia Muscular e Nutrição**. Faculdade de Ciências da Nutrição e Alimentação, Universidade do Porto, 2010. Disponível em: <http://repositorio-aberto.up.pt/bitstream/10216/54607/3/139063_1054TCD54.pdf>.

MURA, Joana D'arc Pereira; SILVA, Sandra Maria Chemin Seabra. **Tratado de Alimentação, Nutrição & Dietoterapia**. Editora Roca, 2011.

NIXON, Robin. **Learning PHP, MySQL, JavaScript, CSS & HTML5**. O'Reilly Media Inc, 2014.

PEREIRA, Isadora Coelho; SOUZA, Ismênia Rosa Dias; LISBÔA, Maria de Fátima. **Perfil Alimentar de Praticantes de Musculação na Maturidade**. Revista Brasileira de Nutrição Esportiva, 2007. Disponível em: <<http://www.rbne.com.br/index.php/rbne/article/download/7/7>>

PEREIRA, Lúcio Camilo Oliva; SILVA, Michel Lourenço. **Android para Desenvolvedores**. Brasport Livros e Multimídia Ltda, 2009. Disponível em: <<http://books.google.com.br/books?hl=pt-BR&lr=&id=8u9wJowXfdUC&oi=fnd&pg=PA1&dq=aplicativo+para+android&ots=LSlp441np2&sig=tWNxfIQ5wdt8w8VbG-5AZGhilhM#v=onepage&q=aplicativo%20para%20android&f=false>>.

PHP.NET, *What is PHP?* Disponível em: <<http://php.net/manual/en/intro-whatis.php>>. Acesso em: 03/08/2014

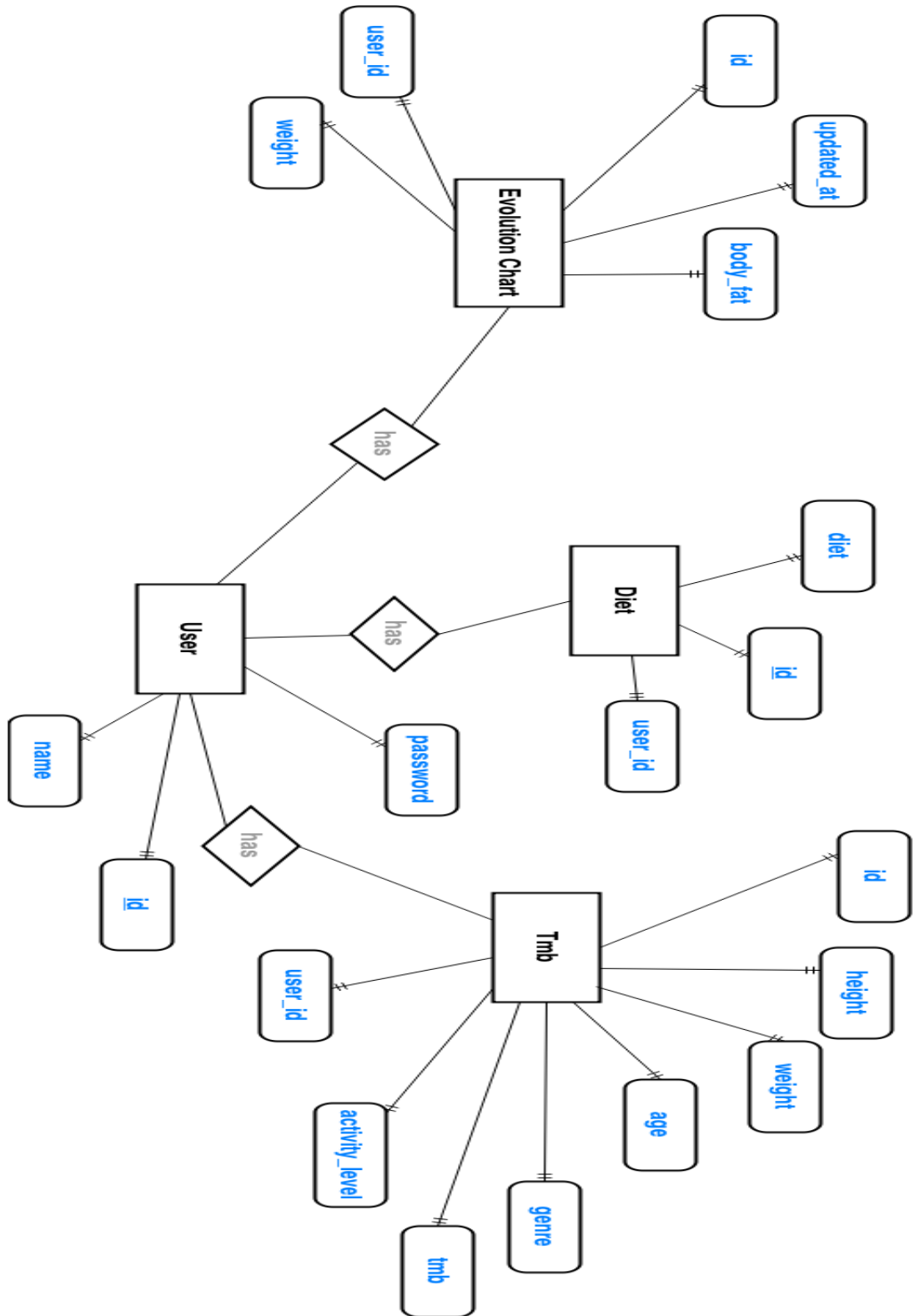
SUEHRING, Steve. **MySQL Bible**. Wiley Publishing Inc, 2002.

TAROUÇO, Fabricio. **A Metrópole Comunicacional e a Popularização dos Apps para Dispositivos Móveis**. Unisinos, 2013. Disponível em: <http://coral.ufsm.br/sipecom/2013/wp-content/uploads/gravity_forms/1-997169d8a192ed05af1de5bcf3ac7daa/2013/09/A-metropole-comunicacional-o-e-a-popularizacao-dos-apps.pdf>

WAHRLICH, Vivian; ANJOS, Luiz Antonio. **Aspectos Históricos e Metodológicos da Medição e Estimativa da Taxa Metabólica Basal: Uma Revisão da Literatura**. Cad. Saúde Pública vol.17 n.4, Rio de Janeiro, Jul./Aug. 2001. Disponível em: <http://www.scielo.org/scielo.php?script=sci_arttext&pid=S0102-311X2001000400015&lng=en&nrm=iso&tlng=pt#tab1>.

APÊNDICES

APÊNDICE A - Modelo Entidade Relacionamento



APÊNDICE B - Diagrama de Caso de Uso

