



**UNIVERSIDADE COMUNITÁRIA DA REGIÃO DE CHAPECÓ
ÁREA DE CIÊNCIAS EXATAS E AMBIENTAIS
CURSO DE SISTEMAS DE INFORMAÇÃO
(BACHARELADO)**

**PROTÓTIPO DE UM APLICATIVO MOBILE PARA AUXÍLIO DE
CONTROLE NUTRICIONAL DE UMA DIETA PESSOAL**

MARCEL BRAGHINI

CHAPECÓ, JUNHO DE 2016

**UNIVERSIDADE COMUNITÁRIA DA REGIÃO DE CHAPECÓ
ÁREA DE CIÊNCIAS EXATAS E AMBIENTAIS
CURSO DE SISTEMAS DE INFORMAÇÃO
(BACHARELADO)**

**PROTÓTIPO DE UM APLICATIVO MOBILE PARA AUXÍLIO DE
CONTROLE NUTRICIONAL DE UMA DIETA PESSOAL**

**Relatório do Trabalho de Conclusão de
Curso submetido à Universidade
Comunitária da Região de Chapecó
para obtenção do título de bacharelado
no curso de Sistemas de Informação.**

MARCEL BRAGHINI

Orientador(a): Prof. Marcos Moretto, Esp.

CHAPECÓ, JUNHO DE 2016

**PROTÓTIPO DE UM APLICATIVO MOBILE PARA AUXÍLIO DE
CONTROLE NUTRICIONAL DE UMA DIETA PESSOAL**

MARCEL BRAGHINI

**ESTE RELATÓRIO, DO TRABALHO DE CONCLUSÃO DE CURSO, FOI
JULGADO ADEQUADO PARA OBTENÇÃO DO TÍTULO DE:**

BACHAREL EM SISTEMAS DE INFORMAÇÃO

Prof. Marcos Moretto, Esp.
Orientador

BANCA EXAMINADORA:

Prof. Fabio Dallazen, Esp.
Unochapecó

Prof. José Carlos Toniazzo, Esp.
Unochapecó

Prof. Sandro Silva de Oliveira, MSc.
Supervisor de TCC

Prof. Viviane Duarte Bonfim, MSc.
Coordenador de Curso

Chapecó, Julho de 2016

LISTA DE ILUSTRAÇÕES

FIGURA 1 -	Sistema Digestório	7
FIGURA 2 -	Estado Nutricional Ideal.....	9
FIGURA 3 -	Processo e Modelo de Cuidado Nutricional.....	11
FIGURA 4 -	Diário Alimentar	12
FIGURA 5 -	IMC	13
FIGURA 6 -	Versão de Plataforma	16
FIGURA 7 -	Ciclo de Vida de uma Activity.....	18
FIGURA 8 -	AVD	22
FIGURA 9 -	JSON	24
FIGURA 10 -	<i>Activity Splash</i>	34
FIGURA 11 -	<i>Activity Login</i>	35
FIGURA 12 -	<i>Fragment IMC</i>	38
FIGURA 13 -	<i>Navigation Drawer</i>	39
FIGURA 14 -	<i>Fragment início</i>	40
FIGURA 15 -	<i>Fragment nutricionista</i>	41
FIGURA 16 -	<i>Fragment dieta</i>	41
FIGURA 17 -	<i>Fragment dicas</i>	42
TABELA 1 -	Classificação Básica do IMC	14
TABELA 2 -	Requisitos Funcionais	28
TABELA 3 -	Requisitos Não Funcionais.....	29
QUADRO 1 -	Fórmula para o Cálculo do IMC	13
QUADRO 2 -	Diagrama Entidade Relacionamento.....	30
QUADRO 3 -	Diagrama de Fluxo de Dados	31
QUADRO 4 -	Caso de Uso.....	32
QUADRO 5 -	Código da Classe <i>JSONParser</i>	36
QUADRO 6 -	Código contido no servidor php	36
QUADRO 7 -	Código do Botão Calcular IMC	37
QUADRO 8 -	Código da Seleção <i>Navigation Drawer</i>	39
QUADRO 9 -	<i>Wireframe</i> Cadastro Cliente	43
QUADRO 10 -	<i>Wireframe</i> Cadastro Dieta	43

LISTA DE SIGLAS

API	- <i>Application Programming Interface</i>
AVD	- <i>Android Virtual Device</i>
BD	- Banco de Dados
DER	- Diagrama Entidade Relacionamento
HTTP	- <i>HyperText Transfer Protocol</i>
IDE	- <i>Integrated Development Environment</i>
IMC	- Índice de Massa Corporal
JDK	- <i>Java Development Kit</i>
JRE	- <i>Java Runtime Enviroment</i>
JSON	- <i>JavaScript Object Notation</i>
PCN	- Processo de Cuidado Nutricional
REST	- <i>Representational Static Transfer</i>
RPC	- Chamada de Procedimento Remoto
SDK	- <i>Software Development Kit</i>
SGBD	- Sistema Gerenciador de Banco de Dados
SGI	- Sistema Gastro Intestinal
SO	- Sistema Operacional
SOAP	- <i>Simple Object Acess Protocol</i>
SQL	- <i>Structured Query Language</i>
UML	- Linguagem Unificada de Modelagem
URI	- Recurso Identificado por Identificador Recurso Universal
URL	- <i>Uniform Resource Locator</i>
WWW	- <i>World Web Wide</i>

XML - *Extensible Markup Language*

RESUMO

Sabe-se que nos últimos anos houve uma significativa evolução tecnológica principalmente nos que diz respeito aos dispositivos móveis. Um exemplo dessa evolução são os smartphones, que evoluíram bastante nos últimos anos, tornando o ser humano dependente a eles devido à sua versatilidade e multifunções. Nesse contexto pode ser favorável o desenvolvimento de aplicações para esses dispositivos móveis visto que há um grande crescimento do mercado nessa área. Também é notável a busca por uma melhor qualidade de vida quando se fala de hábitos alimentares saudáveis, todavia atingir seus objetivos pode ser difícil para algumas pessoas sem o suporte de um nutricionista. Sendo assim este estudo pretende criar um protótipo de aplicativo para a plataforma Android para uso mútuo do nutricionista e do paciente, no qual o profissional poderá alimentar um banco de dados com informações acerca de uma dieta personalizada além de dicas sobre um hábito alimentar saudável a fim de auxiliar o paciente a atingir seus próprios objetivos.

Palavras-chave: Nutrição, Tecnologias Móveis, Android e Programação.

ABSTRACT

It's known that in the last years there was a significant technologic evolution, mainly in mobile devices. An examples of that evolution are the smartphones, which have evolved considerably in recent years, making human being subordinated to them due to their versatility and multifunctions. In this scenario, developing applications for those mobile devices can be favorable as the Market grows in this area. Also it is remarkable the searching for a better life quality when we talk about healthy feeding behavior, however achive their goals may be hard for some people without a nutritionist help. Therefore this study intend to create a Android prototype app for nutritionist and patient mutual use, where the professional may feed it's database with informations about a customized diet and tips for a healthier feeding behavior in order to help the patient to achieve his/her own goals.

Keywords: Nutrition, Mobile Technologies, Android and Programming.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	IV
LISTA DE SIGLAS	V
RESUMO.....	VII
ABSTRACT	VIII
1 CONTEXTUALIZAÇÃO	1
1.1 Delimitação do Problema	2
1.2 Justificativa.....	3
1.3 Objetivos.....	3
1.3.1 <i>Objetivo geral</i>	3
1.3.2 <i>Objetivos específicos</i>	4
1.4 Procedimentos Metodológicos	4
1.5 Estrutura do Trabalho	4
2 NUTRIÇÃO HUMANA	6
2.1 A Alimentação e Nutrição	6
2.2 O Sistema Digestório.....	6
2.3 A Necessidade Energética	8
2.4 O Papel do Nutricionista	10
2.5 O Índice IMC.....	12
2.6 Considerações do Capítulo	14
3 DESENVOLVIMENTO EM ANDROID	15
3.1 O Sistema Operacional Android.....	15
3.2 SDK	16
3.3 Android Studio	17
3.4 Elementos do Desenvolvimento	17
3.4.1 <i>Activity</i>	17
3.4.2 <i>Fragment</i>	19
3.4.3 <i>Intent</i>	19
3.4.4 <i>Service</i>	20
3.4.5 <i>Broadcast receivers</i>	20
3.4.6 <i>Content provider</i>	20
3.4.7 <i>AndroidManifest.xml</i>	21
3.4.8 <i>Classe R</i>	21
3.5 AVD.....	21

3.6 Mysql e SQLite	22
3.7 Servidor Web	23
3.7.1 REST	23
3.7.2 JSON.....	24
3.8 Back-end	25
3.9 API.....	25
3.10 Considerações do Capítulo	25
4 DESENVOLVIMENTO DO PROTÓTIPO.....	27
4.1 Requisitos	27
4.1.1 Requisitos Funcionais	27
4.1.2 Requisitos Não Funcionais.....	28
4.2 Modelagem	29
4.2.1 Diagrama DER.....	29
4.2.2 Diagrama de Fluxo de Dados	30
4.2.3 Caso de Uso.....	31
4.3 Desenvolvimento do Protótipo	32
4.3.1 Criação do AVD	33
4.3.2 Tela de Splash.....	33
4.3.3 Tela de Login.....	34
4.3.4 Acesso ao Web Service	35
4.3.5 Fragment IMC	37
4.3.6 Navigation Drawer e Fragments.....	38
4.3.7 Telas de Inicio e Nutricionista.....	40
4.3.8 As Telas Dieta e Dicas.....	41
4.3.9 Wireframe de telas Web.....	42
4.4 Considerações do Capítulo	44
5 CONCLUSÕES.....	45
5.1 Considerações Finais	45
5.2 Trabalhos Futuros	46
REFERÊNCIAS BIBLIOGRÁFICAS	47

1 CONTEXTUALIZAÇÃO

Nesta seção é apresentada uma abordagem geral da sobre o tema bem como delimitação do problema, justificativa, objetivos gerais e específicos, procedimentos metodológicos e a estrutura do trabalho.

Há algum tempo já fala em alimentação saudável, em dieta, mas algumas pessoas encontram dificuldades para acompanhar uma dieta nutricional, ou em alguns casos se alimentam de maneira incorreta. Então há uma necessidade de controlar sua alimentação, com o objetivo de melhorar a qualidade de vida. Muitas pessoas não possuem tempo para planejar e pensar suas refeições, muitos iniciam um acompanhamento médico a fim de controlar sua dieta, mas nem todos conseguem organizar-se de maneira correta, sendo assim o que fazer diante desta situação?

Discutir sobre saúde é algo importante e que pode levar as pessoas a pensar em força ou energia, por exemplo, porém pode significar mais que isso, a saúde pode ser o reflexo de vários fatores tais como uma alimentação de qualidade ou exercício físico. Em alguns casos apenas melhorar a alimentação pode não melhorar a saúde, mas um conjunto de outros fatores como o exercício físico é possível obter-se melhores resultados.

É importante que nosso corpo absorva os nutrientes necessários para que possa manter a integridade estrutural e funcional, caso contrário nosso corpo pode enfraquecer. Deve-se saber que problemas de saúde, em muitos casos, podem ser oriundos da ausência ou baixa disposição de nutrientes necessários para o organismo, influenciando diretamente nas funções do corpo. Grande parte destes nutrientes pode ser absorvida pelo organismo através da alimentação, então se percebe que é possível regular ou tratar certas doenças por meio de uma dieta nutricional específica visando o abastecimento dos nutrientes que estão em falta.

Este projeto tem como objetivo a criação de um protótipo mobile, que pode ser utilizado através de um smartphone. O protótipo deve auxiliar na parte de controle da dieta, utilizando dados inseridos por um profissional responsável.

1.1 Delimitação do Problema

Levando em consideração a saúde, diversas pessoas buscam uma qualidade de vida ideal, para enfrentar com energia o seu dia-a-dia, vive-se em uma época muito cômoda, facilitada pelo avanço da tecnologia onde se exercita pouco e alimenta-se descontroladamente.

Em muitos casos a alimentação pode sair do controle de forma que o indivíduo não volta ao seu estado nutricional ideal sem o auxílio de um profissional, neste caso não existe um balanço nutricional entre os nutrientes gastos em seu dia entre os nutrientes absorvidos no mesmo período podendo ocasionar o descontrole do metabolismo que pode ser refletido na falta de disposição ou por diversas doenças.

Diante de casos como estes alguns indivíduos precisam consultar um nutricionista, que é um profissional capacitado de conhecimento que organiza a nutrição conforme a necessidade específica de cada paciente através de exames e questionários específicos.

O que pode acontecer é que algumas pessoas que passam pelo acompanhamento de um nutricionista não tem um cuidado especial em relação às informações a serem seguidas principalmente no que diz respeito aos cuidados com a nova dieta que pode ser repesada pelo profissional.

Em função disso, muitas pessoas não chegam ao seu objetivo quando, por exemplo, espera-se uma condição física saudável, tratamento ou prevenção de alguma doença, podendo causar sua insatisfação.

Espera-se resolver a dificuldade de disciplinar o paciente de maneira que ele cumpra as metas e os objetivos traçados previamente na consulta, utilizando o protótipo que será acessado pelo seu smartphone com a intenção de facilitar o acesso a sua dieta.

1.2 Justificativa

Diante da busca inerente de qualidade de vida, as pessoas se voltam para a nutrição alimentar, buscando uma alimentação mais adequada e saudável, portanto, observa-se um aumento na busca por uma alimentação saudável, deste modo à procura por orientação profissional, a fim de, promover o bem estar pessoal. O profissional de nutrição tem possibilidade de não somente indicar alimentos com vários nutrientes, mas sim alimentos com os nutrientes que cada organismo necessita para melhorar sua condição.

O fato é que muitas vezes a dieta nutricional indicada é seguida de maneira incorreta, por vários motivos como: falta de conhecimento específico sobre alguma alimentação, esquecimento, preguiça, dentre outros.

As pessoas têm o conhecimento que é preciso alimentar-se de maneira correta, isso é um ponto positivo, porém quando elas precisam empenhar-se para realizar suas atividades, muitas vezes caem no comodismo e simplesmente deixam de lado o empenho necessário para alcançar seus objetivos. O que mais tarde pode ser refletido em uma insatisfação por não cumprimento da dieta.

Diante disso o protótipo pode auxiliar no controle nutricional, onde é resgatado o comprometimento com a dieta que é previamente inserida pelo profissional de nutrição, de maneira que facilite o acesso do usuário às informações fornecidas e que o ajude a lembrar de sua alimentação diária, por meio de uma aplicação que é executada em um smartphone com Android.

1.3 Objetivos

1.3.1 *Objetivo geral*

Desenvolver um protótipo mobile para auxílio de controle nutricional de uma dieta pessoal.

1.3.2 Objetivos específicos

Tem-se como objetivos específicos:

- a) Auxiliar no controle de uma dieta nutricional;
- b) Permitir o acompanhamento de dietas com nível nutricional adequado conforme um perfil previamente cadastrado por um nutricionista;
- c) Consumir dados de um servidor por *Json* e inseri-los no protótipo.

1.4 Procedimentos Metodológicos

Para o desenvolvimento desta monografia foi utilizada a pesquisa com objetivos exploratórios de maneira qualitativa, com a intenção de compreender a maneira que a nutrição é apresentada e a relação para com o individuo bem como algumas características.

Em continuidade a pesquisa bibliográfica foi intensivamente sobre a plataforma Android, mais precisamente sobre sua JDK (*Java Development Kit*), suas AVDs (*Android Virtual Devices*) que seria o Emulador, a IDE (*Integrated Development Environment*) o próprio Android Studio e alguns elementos presentes na aplicação, gerenciador de Banco de Dados SQLite Studio juntamente com Mysql, Apache como servidor e *Json* para a troca de informações.

Após esta etapa de pesquisa foi iniciado o levantamento de requisitos funcionais, não funcionais, casos de uso e por fim feito o desenvolvimento do protótipo.

1.5 Estrutura do Trabalho

Este trabalho é dividido entre cinco capítulos que são: Contextualização, Nutrição Humana, Desenvolvimento em Android, Desenvolvimento do Protótipo e Conclusões.

O primeiro capítulo que é abordado nesta monografia é a contextualização dos assuntos levantados neste trabalho, bem como organização e objetivos esperados.

Já no segundo capítulo que se chama Nutrição Humana é onde se aborda o assunto desta monografia ao que se refere em nutrição pessoal comentando o que é e como funciona.

O capítulo adiante tem como objetivos levantar as informações que são necessárias para o desenvolvimento de um protótipo em Android que será usado no próximo capítulo.

O quarto capítulo trata de colocar em prática as informações levantadas no capítulo anterior, ou seja, onde é realmente desenvolvido o protótipo. Mostra detalhadamente os processos de criação do que foi desenvolvido.

Por fim, as conclusões finais do autor, bem como trabalhos futuros e a referência bibliográfica utilizada na monografia.

2 NUTRIÇÃO HUMANA

Neste capítulo é abordado o tema da Nutrição Humana contendo também assuntos pertinentes à alimentação, relacionando o sistema digestório, a necessidade energética, o papel do nutricionista e o Índice de Massa Corporal (IMC).

2.1 A Alimentação e Nutrição

“A nutrição balanceada ideal é um determinante importante da saúde. Pode ser utilizada para promover a saúde e o bem-estar, para evitar doenças e promover seu tratamento.” (GIBNEY, HESTER e KOK, 2005, p. 1).

Sob a condição de a nutrição ser importante para manter nossa saúde em boas condições, ela também é muito importante para manter o organismo com capacidade de defesa suficiente para tentar evitar algumas doenças, “influencia a função do corpo, protege contra doenças, restaura a saúde e determina a resposta do indivíduo a mudanças no meio ambiente.” (GIBNEY, HESTER e KOK, 2005, p. 1).

Quando se fala sobre nutrição não se pode deixar de mencionar que ela é dependente da alimentação, isto é, a alimentação é o alvo da nutrição, "organelas celulares, células, tecidos, órgãos, sistemas e o corpo como um todo obtêm e utilizam as substâncias necessárias obtidas dos alimentos (nutrientes) para manter a integridade estrutural e funcional." (GIBNEY, HESTER e KOK, 2005, p. 2).

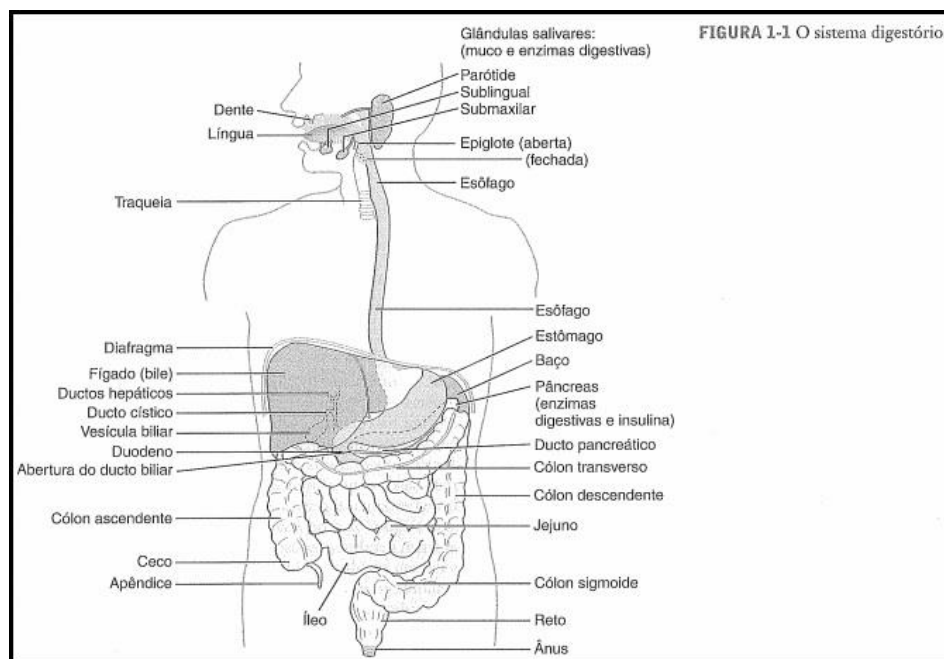
2.2 O Sistema Digestório

Para um melhor entendimento sobre o processo envolvido em nossa alimentação os autores Mahan, Escott-Stump e Raymond (2012) afirmam que o

processo inicia-se na boca onde o alimento é reduzido a partículas misturadas com a saliva, o esôfago transporta o alimento para o estômago, lá é misturado com o líquido ácido para a digestão, logo em seguida repassa ao intestino delgado onde a absorção de nutrientes é maior até chegar ao cólon. Sendo que a maioria dos nutrientes absorvidos chega ao fígado pela circulação do sangue, onde o restante é descartado por defecação pelo reto e o ânus, conforme pode ser visto na Figura 1.

Pode-se ver na Figura 1 como é o sistema digestório.

FIGURA 1 - Sistema Digestório



Fonte: Gibney, Hester, Kok (2005).

Percebe-se que o caminho percorrido pela alimentação é longo e que cada órgão tem uma função essencial para ocorrer à digestão, o corpo absorve os nutrientes de cada alimento a partir do estômago, este conjunto pode ser denominado como sistema gastrointestinal.

"O sistema gastrointestinal (SGI) serve para (1) digerir as proteínas, carboidratos e lipídios dos alimentos e bebidas ingeridos; (2) absorver fluidos, micronutrientes e oligoelementos; e (3) fornecer uma barreira física e imunológica para micro-organismos, corpos estranhos e possíveis antígenos consumidos com o alimento ou formados durante a passagem do alimento pelo SGI." (MAHAN, ESCOTT-STUMP, RAYMOND, 2012, p. 1).

As células do SGI (Sistema Gastro Intestinal) são encarregadas de

absorverem os nutrientes contidos na alimentação, "As células captam nutrientes (por meios de mecanismos complexos através das membranas celulares) do meio ambiente imediato, também conhecido como ambiente interno corporal." (GIBNEY, HESTER e KOK, 2005, p. 2).

Tão logo ocorra a absorção dos nutrientes pelos organismos celulares os mesmos são lançados a corrente sanguínea para então distribuir para todos os sentidos do corpo, "Os nutrientes e o oxigênio são fornecidos para o ambiente interno através do sangue circulante, que também remove resíduos metabólicos e substâncias nocivas deste ambiente para serem excretados através da pele, dos rins e do intestino grosso." (GIBNEY, HESTER e KOK, 2005, p. 2).

2.3 A Necessidade Energética

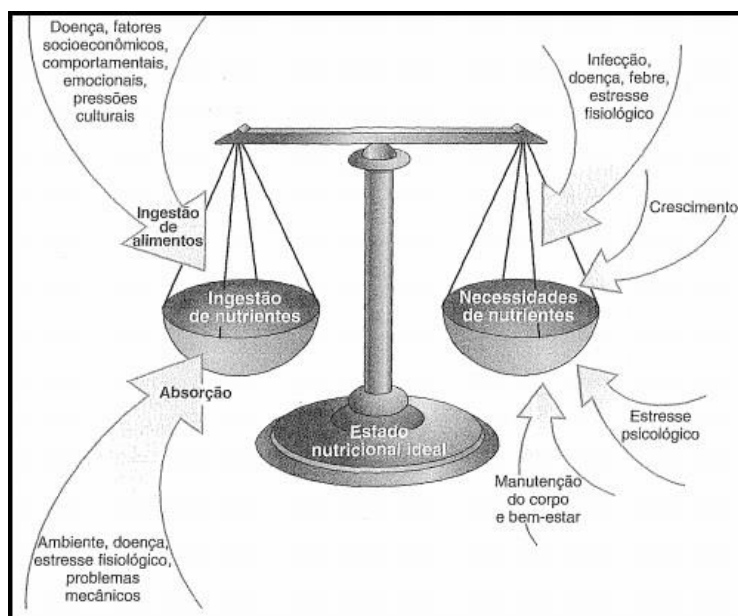
Para que o organismo seja capaz de viver ele precisa de nutrientes que servem de combustível para o corpo, estes nutrientes são absorvidos pelos organismos celulares e distribuídos na corrente sanguínea onde a mesma trata de distribuir para as demais partes do corpo. Para Gibney, Hester e Kok (2005, p. 2) "A saúde ou a doença dos diferentes órgãos e sistemas determinará a necessidade de nutrientes do corpo como um todo".

Diante disso, nota-se que cada organismo pode ter uma necessidade nutricional, dependendo de vários fatores característicos, por exemplo, o indivíduo que tenha câncer, para Mahan, Escott-Stump e Raymond (2012) que citam "A nutrição pode modificar o processo carcinogênico, defesa das células e do hospedeiro, diferenciação celular e crescimento do tumor.". Da mesma maneira uma paciente gestante tem características nutricionais específicas para tal situação, ou uma pessoa com doenças cardiovasculares.

De maneira objetiva os autores Gibney, Hester e Kok (2005) dizem que deve haver um equilíbrio entre a necessidade energética e os nutrientes absorvidos pelo corpo, e que o gasto energético total diário deve ser igual ou maior ao ganho nutricional naquele dia, este é o processo ideal para ocorrer um balanço energético.

A ingestão adequada de nutrientes promove crescimento e desenvolvimento, mantendo a saúde em bom estado, protegendo o corpo contra doenças e enfermidades a Figura 2 define o estado nutricional ideal (MAHAN; ESCOTT-STUMP; RAYMOND, 2012).

FIGURA 2 - Estado Nutricional Ideal



Fonte: Mahan, Escott-Stump, Raymond, (2012).

Para Gibney, Hester e Kok (2005, p. 26), "a maioria dos indivíduos saudáveis conseguem manter um equilíbrio notável entre a energia consumida e a gasta, resultando, portanto, em um estado de balanço energético no corpo."

Já Mahan, Escott-Stump e Raymond (2012, p. 19) dizem que "As necessidades energéticas são definidas como a ingestão de energia dietética necessária para o crescimento ou a manutenção em pessoas de idade, sexo, peso e estatura definidos, e o grau de atividade física desempenhada por elas."

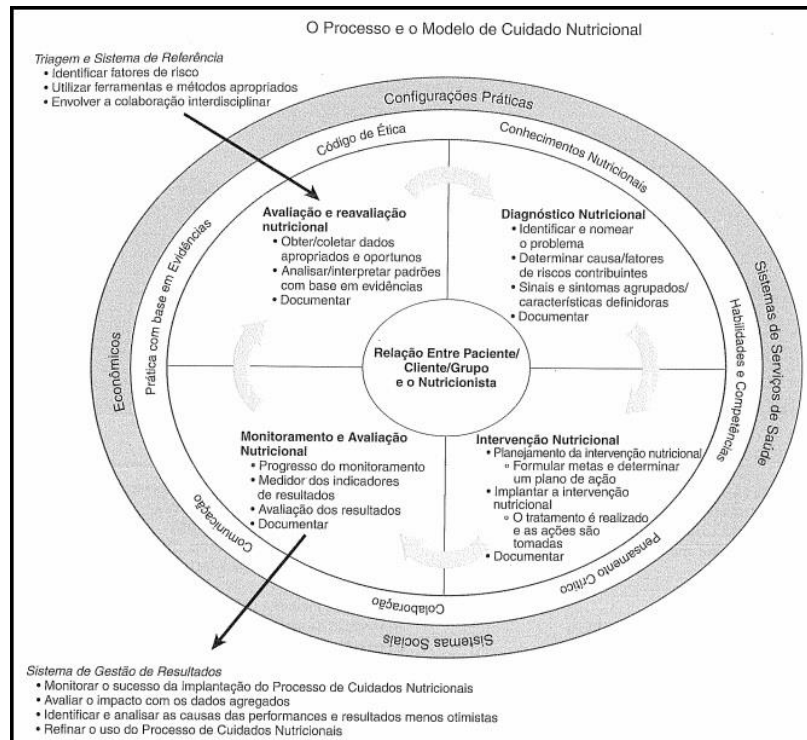
Quando as reservas nutricionais estão baixas não satisfazendo a necessidade do corpo desenvolve-se um estado de desnutrição e quando o consumo alimentar é feito em excesso também pode ocasionar problemas, como a obesidade (MAHAN; ESCOTT-STUMP; RAYMOND, 2012).

2.4 O Papel do Nutricionista

O nutricionista pode ter um papel fundamental no estado nutricional de seu paciente, pois é ele quem irá fazer um estudo sobre a necessidade nutricional de um indivíduo através de exames e questionários. Depois de obter os dados necessários os mesmos são analisados e organizados de maneira que permita ao profissional organizar uma dieta nutricional. "O principal objetivo dos nutricionistas é aplicar os princípios da nutrição para promover saúde e o bem-estar, evitar doença e/ ou restaurar a saúde." (GIBNEY, HESTER e KOK, 2005, p. 6).

Porém não existe um meio ideal para medir o estado nutricional de um indivíduo, o que se faz é analisar alguns fatores característicos, desta maneira o que pode ser feito é uma avaliação nutricional oriunda de históricos médicos e exames de laboratório (GIBNEY; HESTER; KOK, 2005).

Qualquer pessoa pode procurar auxílio nutricional, seja para tratar alguma doença, entrar em boa condição física, controle de peso, gravidez, envelhecimento, fase de crescimento, transtornos alimentares ou simplesmente para manter a saúde. De acordo com a fase da vida existe uma nutrição ideal a ser seguida, por exemplo, durante a gravidez, durante a infância, na adolescência e idade adulta. É comum também uma nutrição específica para saúde e ginástica, transtornos alimentares, desempenho esportivo, saúde óssea e dental, conforme é visto na Figura 3 que demonstra as etapas de um processo de estudo nutricional que é dividido em quatro etapas (MAHAN; ESCOTT-STUMP; RAYMOND, 2012).

FIGURA 3 - Processo e Modelo de Cuidado Nutricional

Fonte: Mahan, Escott-Stump, Raymond, (2012).

O PCN (Processo do Cuidado Nutricional) estabelecido pela *American Dietetic Association* padronizou os cuidados nutricionais. Dividido basicamente em quatro etapas: avaliação nutricional, diagnóstico nutricional, intervenção nutricional e monitoramento e avaliação. (MAHAN; ESCOTT-STUMP; RAYMOND, 2012).

Logo após estas etapas o nutricionista pode passar ao seu paciente o cronograma ou uma dieta organizacional de como deve ser sua alimentação, que geralmente possui dias, horários e alimentos a serem ingeridos, observa-se na Figura 4 um exemplo de diário alimentar para utilização do mesmo.

FIGURA 4 - Diário Alimentar

Diário Alimentar: DIA _____				
REFEIÇÃO	Alimentos (lista)	QUANTIDADE INGERIDA	MODO DE PREPARO	LOCAL (Casa, trabalho etc.)
Café da manhã: (desjejum)				
Lanche:				
Almoço:				
Jantar:				
Ceia:				

Suplementos Alimentares Latas/dia: _____ Nome: _____
 Suplemento de Vitaminas/Minerais: _____

Fonte: Mahan, Escott-Stump, Raymond, (2012).

De acordo com Mahan, Escott-Stump e Raymond (2012, p. 131) "Um registro alimentar diário, ou diário alimentar, envolve documentar a ingestão dietética conforme ela é realizada e é frequentemente utilizado no atendimento ambulatorial."

2.5 O Índice IMC

O IMC é utilizado para determinar se o peso de uma pessoa é compatível com sua altura, podendo indicar uma subnutrição, $IMC = \text{Peso (kg)} \div \text{Altura}^2 \text{ (m)}$. (MAHAN; ESCOTT-STUMP; RAYMOND, 2012). Conforme é observado no Quadro 1.

QUADRO 1 - Fórmula para o Cálculo do IMC

$$\text{IMC} = \frac{\text{Peso}}{\text{Altura}^2}$$

Fonte: Elaboração do autor, (2016).

Pode-se dizer que o índice de massa corporal pode determinar se o peso de uma pessoa é compatível com sua altura, ou seja, se há uma desnutrição ou subnutrição. O IMC não mede a gordura corporal diretamente, mas se correlaciona com as medidas de gordura corporal tais como passagem debaixo da água e absorciometria de energia dupla de raios-X, o cálculo para se chegar ao IMC pode ser visto na Figura 5 (KEYS et al., 1972; MEI et al., 2002 apud MAHAN; ESCOTT-STUMP; RAYMOND, 2012, p. 166).

FIGURA 5 - IMC

VISÃO CLÍNICA

Calculando o IMC e Determinando o Peso Corporal Apropriado

Exemplo: Mulher que tenha 1,72 m (68 pol) de altura e 84 kg (185 libras) de peso

Passo 1: Calcular o IMC atual:

Fórmula: $\text{Peso (kg)} \div \text{Altura (m}^2\text{)}$
 (Métrica) $(1,72 \text{ m}) \times (1,72 \text{ m}) = 2,96 \text{ m}^2$
 $\text{IMC} = 84 \div 2,96 = 28,4 = \text{sobrepeso}$

Passo 2: Variação de peso apropriada para ter um IMC que fique entre 18,5 e 24,9

18,5 $(18,5) \times (2,96) = 54,8 \text{ kg} = 121 \text{ libras}$
 24,9 $(24,9) \times (2,96) = 73,8 \text{ kg} = 162 \text{ libras}$
 Variação de peso apropriada = 121 – 162 libras ou 54,8 – 73,8 kg

Fórmula (Inglesa) $\text{Peso (libras)} \div (\text{Altura [pol]} \times \text{Altura [pol]}) \times 703 = \text{IMC}$
 IMC, Índice de massa corporal.

Fonte: Mahan, Escott-Stump, Raymond, (2012).

Para calcular o índice é necessário fazer o cálculo demonstrado na FIGURA

5, ao término deste cálculo deve-se observar o seu resultado para posteriormente se analisar a Tabela 1.

Os padrões classificam um IMC para um adulto com menos de 18,5 como abaixo do peso, um IMC entre 18,5 e 24,9 como sobrepeso e um IMC maior de 25 como obesos. Um IMC saudável para adultos é considerado entre 18,5 e 24,9 (CDC, 2009 apud MAHAN; ESCOTT-STUMP; RAYMOND, 2012, p. 166).

Demonstra-se na Tabela 1 como é a classificação básica do IMC.

TABELA 1 - Classificação Básica do IMC

Abaixo de peso	< 18,5
Saudável	= > 18,5 e < 25
Sobrepeso	= > 25 e < 30
Obeso	> 30

Fonte: Adaptado de Mahan; Escott-Stump; Raymond, (2012). Elaboração do autor, (2016).

2.6 Considerações do Capítulo

É muito importante que a nutrição se mantenha balanceada para que desta maneira ela promova a saúde e o bem estar o qual é necessário para ter uma qualidade de vida ideal, podendo deixar o organismo resistente a doenças ou restaurando a saúde por exemplo. Os nutrientes que o organismo precisa encontram-se nos alimentos, quando estes são ingeridos passam pelo sistema gastrointestinal onde ao final do processo os nutrientes são absorvidos e o restante dos alimentos descartados.

O corpo humano consome energia ao longo do dia, afinal a energia é o combustível do organismo, por isso é importante manter o equilíbrio entre os nutrientes ingeridos e a energia gasta. Pois quando existe uma sobra de nutrientes ou até mesmo a falta por consequência é possível que cause algum distúrbio.

Cabe ao nutricionista então identificar tais necessidades que são específicas de cada organismo, e deixa-las em equilíbrio de maneira que a quantidade de energia gasta seja equivalente à quantidade de energia adquirida.

3 DESENVOLVIMENTO EM ANDROID

O mercado de dispositivos móveis cresce diariamente e aliado a isso os Sistemas Operacionais (SO) também estão conquistando seu espaço no mercado, há uma grande demanda pela facilidade em que estes dispositivos trazem quando se precisa tirar uma foto, conectar a internet seja a trabalho ou diversão por exemplo.

3.1 O Sistema Operacional Android

“O sistema operacional Android foi desenvolvido pela Android, Inc., qual foi adquirida pela Google em 2005. Em 2007, foi formada a *Open Handset Alliance™*” (DEITEL, 2015, p. 7).

Formado por um grupo grandes empresas de tecnologia, criado com a intenção de criar um ambiente único e comum com a ideia de aperfeiçoamento do SO. “Entre alguns integrantes do grupo estão nomes consagrados como HTC, LG, Motorola, Samsung, Sony Ericsson, Toshiba, HTC, Huawei, Sprint Nextel, China Mobile, T-Mobile, ASUS, Intel, Acer, Dell, Garmin” (LECHETA, 2013, p. 23), liderados pela Google como comentado anteriormente.

O SO Android é de código fonte aberto, permitindo seus usuários a visualizarem seu código e como ele é implementado. (DEITEL, 2015). Essa plataforma de desenvolvimento para dispositivos móveis é baseada em LINUX, com uma interface visual rica e um ambiente de criação poderoso em que é possível que seja usada a linguagem de desenvolvimento Java (LECHETA, 2013).

Conforme ainda Lecheta (2013) cada versão do Android é chamada de plataforma identificada por um código chamado de API (*Application Programming Interface*), muito importante na hora do desenvolvimento e identificação de versões

nos dispositivos, as quais podem ser vistas na Figura 6, bem como sua API e sua porcentagem de distribuição. Nota-se que no momento do desenvolvimento desta monografia API atual de desenvolvimento em Android é a 23.

FIGURA 6 - Versão de Plataforma

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	2.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.2%
4.1.x	Jelly Bean	16	7.8%
4.2.x		17	10.5%
4.3		18	3.0%
4.4	KitKat	19	33.4%
5.0	Lollipop	21	16.4%
5.1		22	19.4%
6.0	Marshmallow	23	4.6%

Fonte: Android, (2016a).

3.2 SDK

Para que o SDK (*Software Development Kit*) seja utilizado no desenvolvimento é necessário que se tenha também o JDK (*Java Development Kit*) e o IDE (*Integrated Development Environment*) sendo que o JDK oferece ferramentas como o compilador e JRE (*Java Runtime Environment*) que permite a execução de programas em Java e a IDE onde será utilizado para a criação e programação (MEDNIEKS et al, 2012).

“O SDK (*Software Development Kit*) do Android fornece as ferramentas necessárias para construir aplicativos Android. Ele está disponível gratuitamente no

site Android Developers.” (DEITEL, 2015, p. 13).

“O SDK do Android é um conjunto de arquivos: bibliotecas, executáveis, scripts, documentação e ferramentas.” (MEDNIEKS et al, 2012, p. 25).

O SDK é utilizado para desenvolver aplicações em Android, que contém um emulador que imita um smartphone, ferramentas e uma API completa em Java, também é possível plugar o celular e configurá-lo para executar a aplicação diretamente no celular (LECHETA, 2013).

3.3 Android Studio

Conforme Android Developer (2016d) o Android Studio é a IDE oficial para o desenvolvimento em Android, fornecendo ferramentas rápidas para a criação de aplicativos em diferentes dispositivos, editor de código, depuração, ferramentas de desempenho, sistema de construção flexível e um sistema de compilação no instante da codificação.

Os códigos inseridos nos arquivos XML são instantaneamente executados em uma espécie de emulador para validação e exibição, possui o recurso IntelliJ que mostra opções sobre o que for inserido facilitando a programação. Também é possível que seja desenvolvido aplicações para telefones, tablets, relógios, automóveis e TVs.

3.4 Elementos do Desenvolvimento

3.4.1 Activity

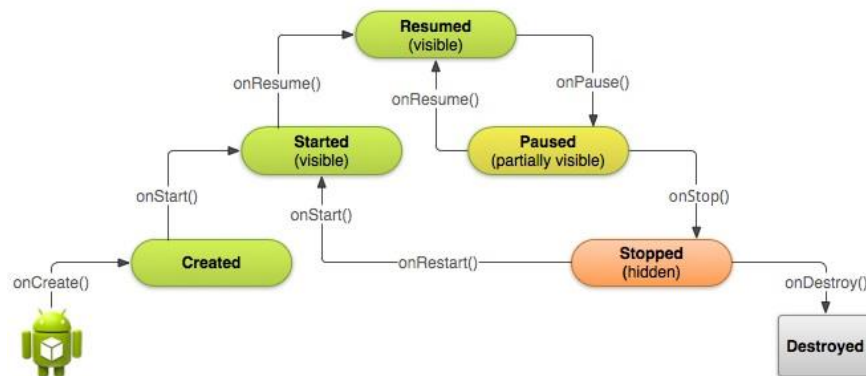
“Geralmente representa uma tela de aplicação e é responsável por tratar os eventos gerados nessa tela, como por exemplo, quando o usuário pressiona um botão ou quando um item de menu é escolhido.” (LECHETA, 2013, p. 113).

Utilizada por uma tela de aplicação, um dos componentes mais utilizados que é composta por *Views*.

Uma *Activity* possui um ciclo de vida específico que pode ser visto na Figura 7. (Pereira, Silva, 2012) indica os métodos:

- - *onCreate()*: método chamado quando a atividade é inicialmente criada.
- - *onStart()*: chamado quando a atividade torna-se visível para o usuário.
- - *onResume()*: é o topo da pilha de atividade, chamado quando vai iniciar uma interação com o usuário.
- - *onPause()*: chamado quando o sistema esta perto de começar uma próxima atividade. Geralmente é utilizado para gravar as informações que ainda não foram salvas.
- - *onStop()*: é chamado quando a atividade não estiver mais sendo utilizada pelo usuário e perdeu o foco para outra atividade.
- - *onDestroy()*: pode ser chamado quando a atividade terminou ou quando o sistema precisa finalizar atividades para liberação de recursos.
- - *onRestart()*: chamado quando sua atividade estiver interrompida e prestes a ser acionada pelo usuário novamente.

FIGURA 7 - Ciclo de Vida de uma Activity



Fonte: Android, (2016b).

Para Pereira e Silva (2012, p. 16) “São quatro os estados em que uma atividade pode se encontrar”:

- Executando: a atividade que estiver ativa no visor do dispositivo.
- Parada: uma atividade que perdeu o foco para outra atividade, mantém todas as informações de estado, porém não está interagindo com o usuário. Pode ser finalizada em situações de baixo nível de memória disponível.
- Interrompida: caso uma atividade não esteja sendo utilizada pelo usuário,

ela mantém suas informações de estado, porém são muitas vezes finalizadas quando uma recuperação de memória seja necessária, perdendo as informações.

- Finalizada: o sistema pode remover uma atividade da memória, caso esta esteja interrompida temporariamente ou estiver parada. O estado anterior só pode ser restaurado se os métodos tiverem sido implementados pelo desenvolvedor.

3.4.2 *Fragment*

Os *Fragments* podem ser chamados também de fragmento de código, chamados e controlados por uma *Activity*. Introduzidos para atender os tablets que utilizam uma tela maior, neste sentido ele aproveita melhor o espaço na tela (DEV MEDIA, 2016b).

Conforme Devmedia (2016c) o *Fragment* é mais do que apenas um pedaço de tela, ele é um componente visual que permite modular e reutilizar partes do aplicativo.

3.4.3 *Intent*

Para Lecheta (2013, p. 152), “A tradução de *Intent* para português é “intenção”, de forma que se pode dizer que uma *Intent* representa a intenção da aplicação de realizar determinada tarefa”.

Afirma ainda que de maneira prática ela é enviada ao SO como mensagem que pode ser chamada como broadcast, também pode ser utilizado para: abrir uma nova janela de aplicação, solicitar ao SO que faça a ligação para determinado numero, abrir um navegador, exibir alguma rota no Google *Maps*, solicitar algum processo em outra aplicação ou abrir o Google Play para baixar uma aplicação (LECHETA, 2013).

3.4.4 Service

“A classe *android.app.Service* é utilizada no Android para executar um processamento em segundo plano por tempo indeterminado, chamado popularmente de serviço. Um serviço geralmente faz um alto consumo de recursos.” (LECHETA, 2013, p. 348).

Afirma, ainda, que rodam em *background* sem um ciclo de vida, somente encerram quando existir uma solicitação e quando conectado se comunica por meio de uma interface apresentada ao usuário (PEREIRA, SILVA, 2012).

Como exemplo é possível citar um reprodutor de música em um celular, você pode deixar uma música rodando enquanto navega na internet ou executa alguma outra tarefa.

3.4.5 Broadcast receivers

Comenta que esta classe é executada em segundo plano sem utilizar interface gráfica, fazendo suas ações sem que o usuário perceba, isso pode ser importante para integrar algumas aplicações na troca de informações, (LECHETA, 2013).

Já Pereira e Silva (2013) relatam que quando a aplicação recebe um evento externo de uma *Intent*, um *Broadcast Receiver* pode ser acionado para fazer alguma ação, ele não possui interface sendo que pode ser acessado mesmo se o aplicativo estiver inativo.

3.4.6 Content provider

O provedor de conteúdo conforme Lecheta (2013) permite que um pacote seja compartilhado em diferentes aplicações, a classe *android.content.ContentProvider* deixa determinar quais informações podem ser públicas, sendo possível que seja consultado, inserido, alterado ou excluído algum dado.

Da mesma maneira Pereira e Silva (2012) quando é necessário o compartilhamento de algum tipo de dado entre algum aplicativo pode ser usado o *Content Provider* permitindo armazenar e recuperar dados no mesmo repositório.

3.4.7 *AndroidManifest.xml*

Para Pereira e Silva (2012) o *AndroidManifest.xml* é um arquivo principal que descreve a aplicação, ou seja, cada componente definindo os dados de cada elemento e inclusive as especificações de permissões.

Já Lecheta (2013) define que este arquivo é a base para uma aplicação em Android, deve conter as configurações necessárias para a aplicação ser executada, como nome do pacote e classes de cada *Activity*.

Conforme Mednieks (2012) um aplicativo deve descrever o ambiente em que o espera, seu nome, intentos, permissões e outras informações importantes da aplicação, estes dados são armazenados no arquivo chamado *AndroidManifest.xml*.

3.4.8 *Classe R*

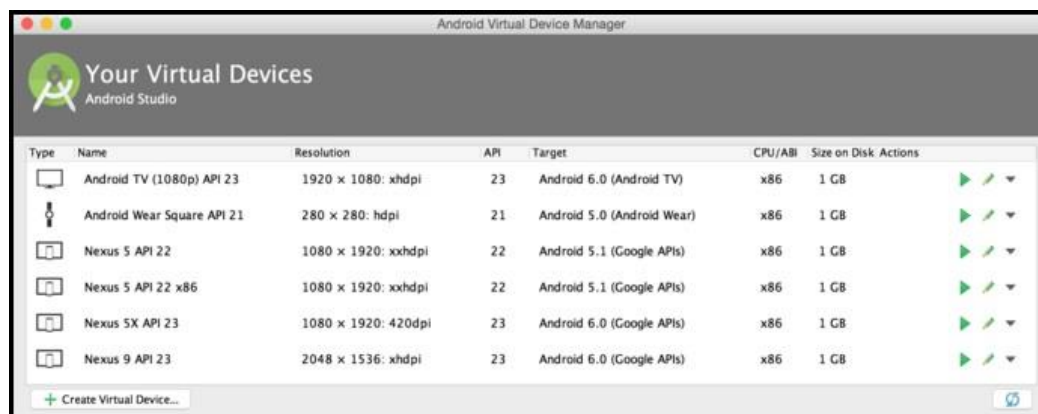
Esta classe é gerada automaticamente e possui constantes que acessam diversos recursos do projeto podendo ser uma imagem ou um arquivo XML (*extensible markup language*) de definição. Sempre que algum arquivo for criado a subpastas como: *drawable*, *layout*, *menu* e *values* uma nova constante será criada na Classe R automaticamente (LECHETA, 2013).

3.5 AVD

O conceito de AVD foi criado a partir do SDK 1.5 para a simulação de um celular em funcionamento idêntico ao real, que é importante para a configuração de um ambiente de desenvolvimento já que se deve analisar a API e se ela é compatível com a versão emulada (LECHETA, 2013).

Criado antes da execução ele é conhecido como emulador do Android e permite o teste de aplicativos em um computador ao invés do celular real, muito útil caso não tenha um aparelho para testar a aplicação, a Figura 8 representa a tela onde é possível incluir uma nova AVD (DEITEL, 2015).

FIGURA 8 - AVD



Fonte: Android, (2016c).

3.6 Mysql e SQLite

Afirma, ainda, que o BD (Banco de Dados) é uma boa solução para armazenamento de dados que possam ter vários registros, sendo que um BD relacional se mostra eficiente, estruturado e genérico para a administração de dados (MEDNIEKS et al, 2012).

Segundo Mednieks (2012) pode ser necessário dependendo de tipo de aplicação um Banco de Dados para armazenamento de registros, permitindo que estes dados possam ser inseridos, modificados ou excluídos conforme a necessidade.

O Mysql é um banco de dados de código aberto, possibilita uma simples entrega de dados confiável, com alto desempenho, redimensionável, com base na web, é de fácil uso e tem a confiança da Oracle, com recursos avançados, ferramentas de gerenciamento e suporte para desenvolver, programar e gerenciar BD (ORACLE, 2016).

Conforme Devmedia (2016d) o Mysql é um Sistema de Gerenciador de Banco de Dados (SGDB) que utiliza a linguagem SQL (*Strutured Query Language*) como interface, onde pode ser possível ter: portabilidade, compatibilidade, excelente desempenho, pouco exigente quanto ao *hardware* de fácil uso, software livre, suporta controle transacional, *triggers*, *cursors*, suporta *stored procedures* e *funcions* e interfaces gráficas.

“O Android oferece o mecanismo de Banco de Dados relacional SQLite, que

é leve e poderoso, para a persistência de dados.” (MEDNIEKS et al, 2012, p. 295).

O SQLite é uma ferramenta simples para armazenamento de dados em tabelas para manipulação por meio de comandos SQL, funcionando como um SGBD, criando um arquivo em disco e servindo como um servidor (PEREIRA, SILVA, 2012).

3.7 Servidor Web

Observa-se que é possível utilizar o Banco de Dados no Android, mas muitas vezes estes dados já podem estar armazenados em outro BD por algum motivo, sendo assim se faz necessário que o dispositivo se comunique com o banco, neste caso entra em ação o *Servlet* que fica no servidor web e aguarda uma requisição (LECHETA, 2013).

O conceito fundamental de serviços web é que eles permitem compor RPC (Chamada de Procedimento Remoto) de um objeto para a internet ou alguma rede, podem usar padrões como HTTP (*Hypertext Transfer Protocol*) e XML (HENDRICKS et al, 2002).

Os dispositivos devem fazer requisições HTTP para o Web Server que consulta ou envia informações que podem retornar em arquivos XML ou JSON (*JavaScript Object Notation*) (LECHETA, 2013).

Quando a comunicação for feita pela internet é recomendado utilizar o protocolo HTTP, que faz a troca de mensagem em XML através do protocolo SOAP (*Simple Object Access Protocol*), ou seja, o Web Service que se comunica pelo protocolo HTTP recebe as solicitações do SOAP (HENDRICKS et al, 2002).

3.7.1 REST

De acordo com NetBeans (2016) o REST (*Representational State Transfer*) pode ser definido como um estilo de arquitetura distribuído parecido com o WWW (*World Wide Web*), onde se pode manipular recursos sobre sua interface. É uma alternativa para uso do SOAP na implementação de serviços por ser de leve transmissão e em HTTP.

Segundo Caelum (2016) o REST permite o endereçamento de uma forma padronizada podem-se citar algumas vantagens: o URI (recurso identificado por identificador de recurso universal) são *bookmarkable* e a conexão entre os sistemas pode ter uma melhor visão da transmissão.

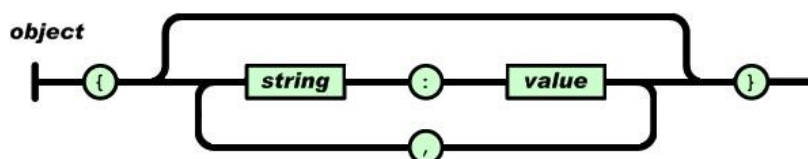
Se baseia em três conceitos, o Recurso que são os nomes dos recursos no sistema que devem ser substantivos, Operações são definidas utilizando os métodos *GET, POST, PUT, DELETE, HEAD, OPTIONS e TRACE* e a representação pode ser explicado como quando uma aplicação é feita ela não trafega pela rede apenas uma apresentação. (CAELUM, 2016).

É importante comentar que no desenvolvimento deste protótipo foi utilizado como padrão o estilo de arquitetura REST.

3.7.2 JSON

Pode-se dizer que JSON é uma formação simples de troca de dados, produzida em formato texto que pode ser independente de qualquer linguagem, basicamente constituído em duas estruturas: uma coleção de pares com nome/valor e uma lista ordenada de valores. Em JSON os dados são apresentados conforme se pode ver na Figura 9.

FIGURA 9 - JSON



Fonte: Json, (2016).

Conforme o artigo de Corrêa (2016) JSON é um modelo de armazenamento e transmissão de dados em formato de texto, simples e que pode ser utilizado em aplicações web. Sua representação é que para cada valor apresentado é necessário definir um nome, representação derivada do Javascript. Pode-se dizer que é uma espécie de corrente do XML na área de troca de informações.

3.8 Back-end

De acordo com Devmedia (2016f) *back-end* descreve um tipo de ação a ser realizada, pode ser descrito também como a parte da aplicação que contém a lógica do negócio e acesso as informações. Onde os dados definitivamente são processados depois de recebidos de um formulário por exemplo.

3.9 API

A API ou Interface de Programação de Aplicativos é um conjunto de padrões de programação que permite o desenvolvimento de aplicativos, grande parte das vezes ela se comunica com outros códigos criando diversas funções em um aplicativo. Com a utilização de uma API é possível criar sistemas melhores e minimizar o entendimento deles, podemos citar como exemplo a API dos SO que possuem métodos que se comunicam com diversos processos do Sistema Operacional (DEV MEDIA, 2016e).

Conforme Tecmundo (2016) a API é uma interface que roda por trás de tudo, conforme você utiliza um aplicativo, por exemplo, ele pode comunicar-se com diversos outros sistemas ou aplicativos sem que você perceba.

3.10 Considerações do Capítulo

Unido ao grande desenvolvimento dos dispositivos móveis, os sistemas operacionais também estão em crescimento, principalmente o Android que foi desenvolvido por grandes empresas de tecnologias e que é de código fonte aberto.

Para ser possível o desenvolvimento em Android são necessárias algumas ferramentas como o SDK que é um kit de ferramentas indispensáveis para a programação, o AVD que é um emulador que simula um dispositivo para serem feitos os testes e o Android Studio que é a ferramenta onde a programação é feita, o banco de dados SQLite e também Mysql, o servidor web que gera os dados em *Json*, dados estes que são consumidos pelo protótipo.

São importantes também alguns elementos usados na programação: *Activitys*, *Fragments*, *Intents*, *Services*, *Broadcast Receivers*, *Content Provider*, *Android Manifest* e *Classe R*.

4 DESENVOLVIMENTO DO PROTÓTIPO

Depois da finalização da etapa de revisão da bibliografia é possível integrar as informações e analisar os pontos importantes pesquisados para coloca-los em prática no desenvolvimento. Este se dá por meio da criação de um protótipo nutricional onde o profissional irá relaciona-se com seu cliente.

A parte desenvolvida neste protótipo é a aplicação para o dispositivo móvel que é executado no Android, apresentando os principais requisitos funcionais, não funcionais, diagrama DER (Diagrama Entidade Relacionamento), diagrama de fluxo e caso de uso.

Depois foi iniciada a programação do protótipo em Android, criação do BD, do servidor e a comunicação entre eles, porém não será feito o desenvolvimento das telas web que seriam para o cadastro das informações pelo Nutricionista.

4.1 Requisitos

4.1.1 Requisitos Funcionais

Conforme Sommerville (2007) os requisitos funcionais podem ser representados pelo que o sistema irá fornecer, o que é feito conforme as entradas no sistema e a maneira como ele deve se comportar, ou seja, os recursos que o sistema deve poder apresentar.

De acordo com o objetivo proposto nesta monografia pode-se selecionar alguns dos principais requisitos funcionais que se pode visualizar na Tabela 2.

TABELA 2 - Requisitos Funcionais

RF001	É necessário que a aplicação mostre a dieta a ser seguida pelo usuário, conforme cadastrado pelo profissional de nutrição.	Essencial
RF002	É essencial que se cadastre uma dieta, ou seja, uma relação de alimentos que devem estar cadastrados no sistema como: descrição, hora da alimentação, data da alimentação, tipo de refeição, alimento, categoria, tipo de dieta.	Essencial
RF003	Deve ser possível o cadastramento de alguns dados importantes de um usuário como: nome, peso, altura, usuário e senha.	Essencial
RF004	Devem ser cadastradas algumas dicas de saúde para o usuário.	Desejável
RF005	A aplicação pode calcular com base em dados previamente inseridos o IMC do usuário.	Importante
RF006	Visualização de dados do nutricionista como contato.	Essencial
RF007	Todos os dados devem ficar registrados no Banco de Dados.	Essencial
RF008	O cadastro da dieta deve ser feito pelo profissional de nutrição.	Essencial
RF009	O usuário do protótipo deve apenas visualizar dados.	Essencial
RF010	A única tela onde o usuário pode alterar algum dado é para cálculo do IMC.	Essencial

Fonte: Elaborado pelo autor, (2016).

Fica evidente a importância de se levantar todos os requisitos antes do início do desenvolvimento para que sejam levantadas todas as possibilidades ou ao menos as principais funcionalidades que a aplicação pode ter como recurso. Muitas vezes menos no meio do processo de desenvolvimento ainda é necessário incluir novos requisitos ou alterar algum já existente.

4.1.2 Requisitos Não Funcionais

Da mesma forma que com os requisitos funcionais é apresentado uma tabela com os requisitos não funcionais presentes na monografia, pode-se observar a Tabela 3 que mostra a relação destes requisitos.

TABELA 3 - Requisitos Não Funcionais

RNF001	O protótipo deve rodar em plataforma Android a partir da versão Android 6.0 Marshmallow API 23.0.3	Essencial
RNF002	As informações são resgatadas de um Web Service, onde serão feitos os cadastros, quando o usuário fazer login no protótipo seus dados serão atualizados na aplicação, isso deve ser feito via internet.	Desejável
RNF003	Deve ser possível que os dados sejam acessados mesmo sem rede, já que podem ser baixados na aplicação.	Desejável
RNF004	Caso o nutricionista faça alguma alteração no perfil de algum cliente, é necessário que o mesmo faça login usando a rede para que o protótipo resgate as informações do Web Service.	Desejável

Fonte: Elaborado pelo autor, (2016).

Os requisitos não funcionais são restrições ou funções oferecidas pelo sistema podendo restringir alguns processos que não estão diretamente ligados as principais funções do sistema. Mais relacionados com confiabilidade, tempo de resposta ou espaço de armazenamento (SOMMERVILLE, 2007).

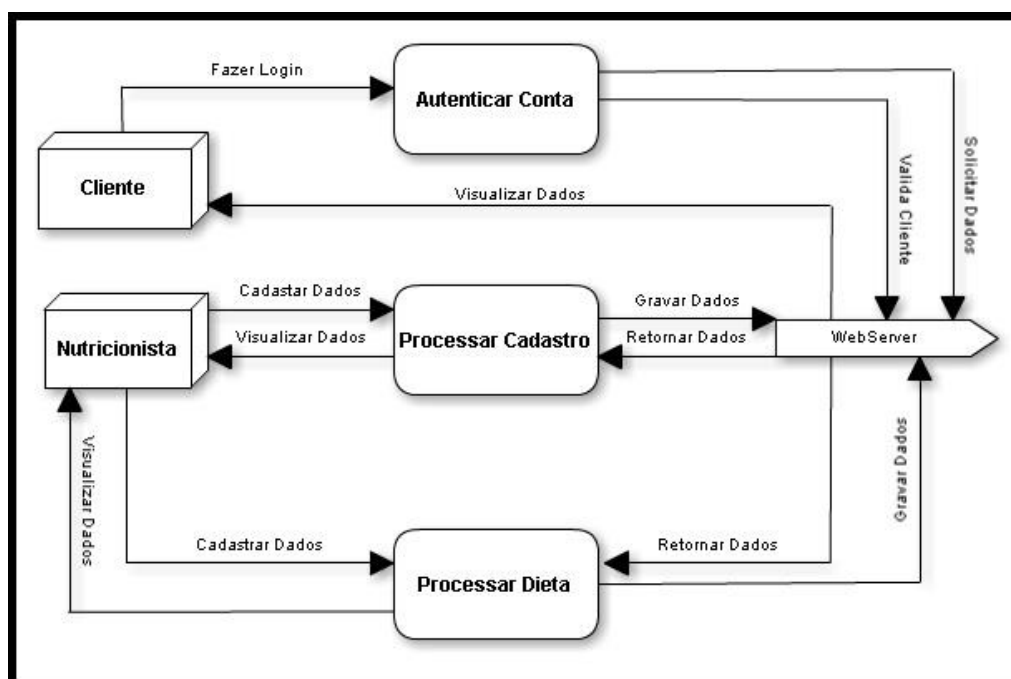
4.2 Modelagem

4.2.1 Diagrama DER

O DER (Diagrama Entidade Relacionamento) representa as estruturas de dados que podem ser abstraídos de um levantamento de requisitos. É aplicado na fase do modelo lógico representado também por um mapeamento de dados ou processos que serão utilizados (TONSIG, Sérgio Luiz, 2013).

Então o DER pode ser criado com base nos principais requisitos necessários para funcionamento do aplicativo, também é possível que nesta etapa apareça problemas estruturais ou até mesmo necessidades ainda não percebidas no levantamento de requisitos, pode-se ver no Quadro 2 o diagrama DER.

QUADRO 3 - Diagrama de Fluxo de Dados



Fonte: Elaborado pelo autor, (2016).

De acordo com Devmedia (2016a) este diagrama é capaz de demonstrar a lógica de um determinado processamento, por sequencia e símbolos que facilita o entendimento de um processamento. Busca a padronização e representação dos métodos e facilitando a leitura e o entendimento.

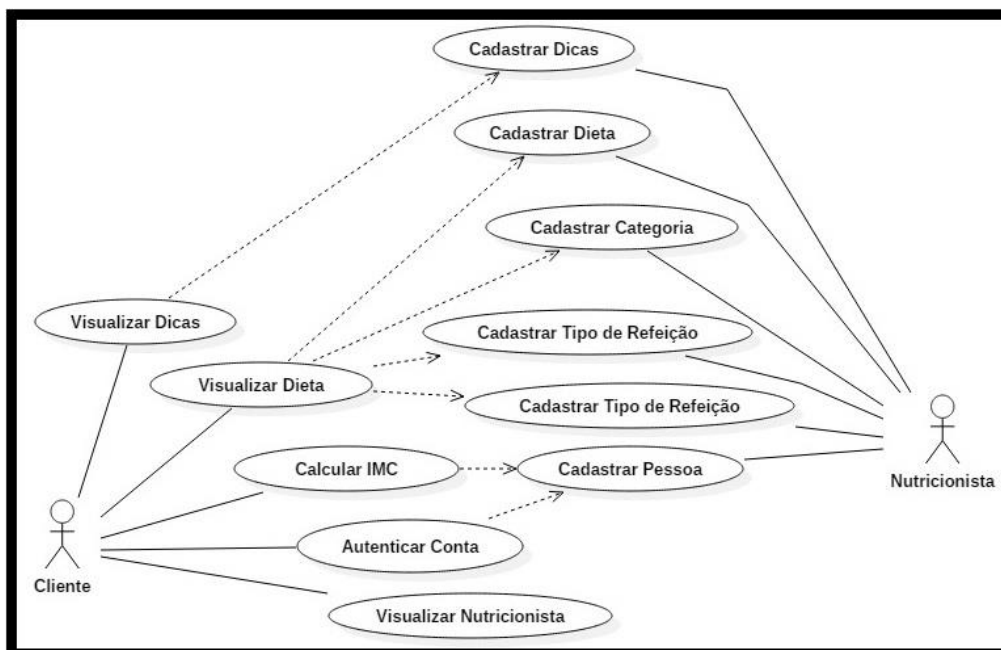
4.2.3 Caso de Uso

De acordo com Jacobsen, et al. apud Sommerville (2007) o Caso de Uso é uma técnica baseada em cenários para a representação de requisitos. Pode ser definido como uma característica básica do UML (*Unified Modeling Language*) descrevendo modelos orientados a objeto.

O Caso de Uso documenta as atividades sobre enfoque do usuário, a ligação de iterações com processos e outros usuários do sistema. Neste Caso de Uso pode aparecer um cenário ou use case, seus atores e comunicação (DEV MEDIA, 2016).

De acordo com os requisitos levantados tendo em vista a necessidade do protótipo é possível definir no caso de uso a seguinte representação conforme o Quadro 4.

QUADRO 4 - Caso de Uso



Fonte: Elaborado pelo autor, (2016).

Os elementos essenciais presentes no caso de uso podem ser os atores, representados por bonecos e interações que são representadas por elipses identificando cada interação (SOMMERVILLE, 2007).

4.3 Desenvolvimento do Protótipo

Foi definido antes de iniciar a prototipação, que devido a grande pesquisa que fora feito para obter o conhecimento que vem ser aplicado nesta monografia, é interessante dividir o desenvolvimento em algumas etapas que podemos ver a seguir.

Para a criação do protótipo é usado o Android Studio uma das principais ferramentas para desenvolvimento em Android, com SQLite que é o Banco de Dados, o Apache com PHP e Mysql para a parte do Web Server.

O processo de criação iniciou com a criação de uma AVD, em seguida o desenvolvimento de uma tela de *Splash*, logo depois uma tela de *Login*, em seguida os *Fragments* que são acessados pelo menu que por sua vez encontra-se em *Activitys* e logo depois a parte do banco de dados.

4.3.1 Criação do AVD

A AVD tem um papel importante no desenvolvimento do protótipo por que através dele é possível testar ou executar a aplicação, para podermos descobrir se há erros ou se realmente a execução é perfeita. Neste protótipo utilizaremos a AVD Nexus 4 API 23 com Android 6.0.

Ele demora até carregar por completo pois simula um smartphone real, com todos as funções que um celular por exemplo, é importante que logo ele seja iniciado para que durante a programação já esteja carregado para a execução do código.

4.3.2 Tela de Splash

Pode-se chamar a tela *Splash Screen* de tela de abertura, já que esta pode ser utilizada nos aplicativos e geralmente traz alguma informação relevante ao mesmo. Seja algo relacionado ao objetivo do aplicativo ou simplesmente uma tela que é gerada para ocupar o tempo enquanto o software é carregado em segundo plano.

No caso deste protótipo a tela de *Splash* é usada para carregar a logomarca da aplicação juntamente como o nome do aplicativo, ou seja, fazer uma breve apresentação. Podendo despertar o interesse de um usuário utilizar ou não o protótipo conforme pode ser visto na Figura 10.

FIGURA 10 - Activity Splash

Fonte: Elaborado pelo autor, (2016).

Na execução do protótipo pode-se ver uma animação na logo da Figura 11, ela gira 360° para a direita implementada pela função *AnimationListener* em java e no XML da *Activity* utiliza-se o `<rotate>`.

4.3.3 Tela de Login

A tela onde a autenticação é feita, ou tela de *Login*, possui extrema importância em relação a este aplicativo já que o cliente ou o usuário irá baixar a aplicação em seu Google Play. Porém vai precisar de um usuário previamente cadastrado pelo nutricionista para o acesso a aplicação, ou seja, é necessário que ele esteja previamente cadastrado. Pode-se ver na Figura 11 o exemplo.

FIGURA 11 - Activity Login

Fonte: Elaborado pelo autor, (2016).

De acordo com os objetivos propostos nesta monografia a aplicação não vai definir e melhor dieta para seu usuário, mas sim, o nutricionista é ele quem avalia o estado de seu paciente podendo este, fazer a escolha que melhor se adaptar a necessidade de cada indivíduo. O objetivo desta ferramenta é servir de suporte para que haja um elo de comunicação de tarefas entre as duas fontes, cliente e nutricionista.

4.3.4 Acesso ao Web Service

Foi desenvolvido uma classe chamada de *JSONParser* onde recebe como atributo a URL (*Uniform Resource Locator*) que é o caminho onde o Web Service pode ser acessado, no caso desse aplicativo é *localhost* pois esta sendo utilizado o *apache* como servidor como pode-se ver no Quadro 5.

Também é possível observar no Quadro 6 o código contido no servidor em PHP.

QUADRO 5 - Código da Classe *JSONParser*

```

public JSONObject getJSONFromUrl(String url) {
    StrictMode.ThreadPolicy policy = new StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);

    // Making HTTP request
    try {
        // defaultHttpClient
        DefaultHttpClient httpClient = new DefaultHttpClient();
        HttpPost httpPost = new HttpPost(url);

        HttpResponse httpResponse = httpClient.execute(httpPost);
        HttpEntity httpEntity = httpResponse.getEntity();
        is = httpEntity.getContent();

    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

    try {
        BufferedReader reader = new BufferedReader(new InputStreamReader(
            is, "iso-8859-1"), 8);
        StringBuilder sb = new StringBuilder();
        String line = null;
        while ((line = reader.readLine()) != null) {
            sb.append(line + "\n");
        }
        is.close();
        json = sb.toString();
    } catch (Exception e) {
        Log.e("Buffer Error", "Error converting result " + e.toString());
    }

    // try parse the string to a JSON object
    try {
        jsonObj = new JSONObject(json);
    } catch (JSONException e) {
        Log.e("JSON Parser", "Error parsing data " + e.toString());
    }

    // return JSON String
    return jsonObj;
}

```

Fonte: Elaborado pelo autor, (2016).

QUADRO 6 - Código contido no servidor php

```

<?php
$host="localhost";
$username="root";
$password="";
$db_name="nutrilife";

$con=mysql_connect("$host", "$username", "$password")or die("cannot connect");
mysql_select_db("$db_name")or die("cannot select DB");
$sql = "select * from dieta";

$result = mysql_query($sql);
$json = array();

if(mysql_num_rows($result)){
    while($row=mysql_fetch_assoc($result)){
        $json['dieta'][]=$row;
    }
}
mysql_close($con);
echo json_encode($json);
?>

```

Fonte: Elaborado pelo autor, (2016).

A classe do Quadro 6 retorna um objeto que carrega os dados consumidos no servidor, o código que se encontra no servidor vai até o banco de dados Mysql e traz as informações gravadas até o servidor, que mais tarde são solicitadas pelo protótipo.

4.3.5 Fragment IMC

Nesta janela é possível calcular o IMC do usuário, através de dados que foram preenchidos por seu nutricionista, porém o campo de peso e altura esta liberado para troca de dados que pode ser feita pelo usuário, porém estes não ficam salvos no dispositivo.

Conforme os dados de altura e peso o protótipo é capaz de calcular o IMC por uma simples função já apresentada nesta monografia, é utilizado alguns *ifs* e *else ifs* para a escolha do resultado ideal como também já apresentado nesta monografia, como é possível observar no Quadro 7.

QUADRO 7 - Código do Botão Calcular IMC

```
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.btnCalcular:
            //Criando variáveis para cálculo
            float res, pes, alt;
            //Passando de STRING -> FLOAT
            pes = Float.parseFloat(Peso.getText().toString());
            alt = Float.parseFloat(Altura.getText().toString());
            //Cálculo do IMC
            res = (pes / (alt * alt));
            //Testando no que se enquadra o resultado do peso:
            if (res < 18.5) {
                //Abaixo do Peso
                Situacao.setText("Você está abaixo do Peso!");
            } else if (res < 25 && res >= 18.5) {
                //Saudável
                Situacao.setText("Sua situação é Saudável!");
            } else if (res >= 25 && res <= 30) {
                //Sobrepeso
                Situacao.setText("Você está com Sobrepeso!");
            } else if (res > 30) {
                //Obeso
                Situacao.setText("Você pode estar Obeso!");
            }

            //Código que deixa aparecer somente duas casas decimais após a virgula
            float f = res;
            NumberFormat formatter = NumberFormat.getNumberInstance();
            formatter.setMinimumFractionDigits(2);
            formatter.setMaximumFractionDigits(2);
            String output = formatter.format(f);

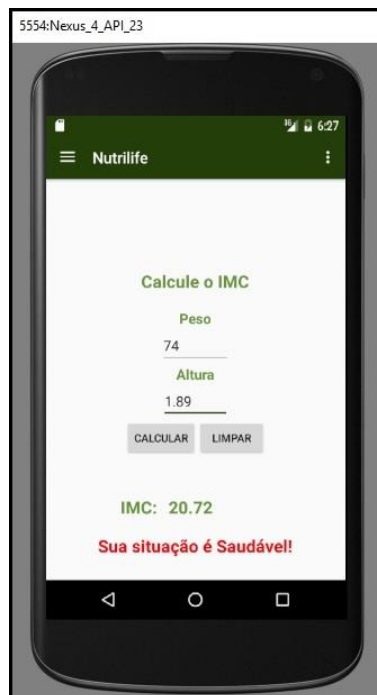
            //Passando de FLOAT -> STRING
            Resultado.setText(output);

            break;
        case R.id.btnLimpar:
            //Limpendo os dados da tela
            Altura.setText("");
            Peso.setText("");
            Resultado.setText("");
            Situacao.setText("");
            break;
    }
}
```

Fonte: Elaborado pelo autor, (2016).

Os campos de resultado apresentados nesta *Fragment* podem ser definidos como IMC e a situação que o individuo encontra-se conforme seus dados cadastrados, conforme é observado ver na Figura 12.

FIGURA 12 - *Fragment* IMC

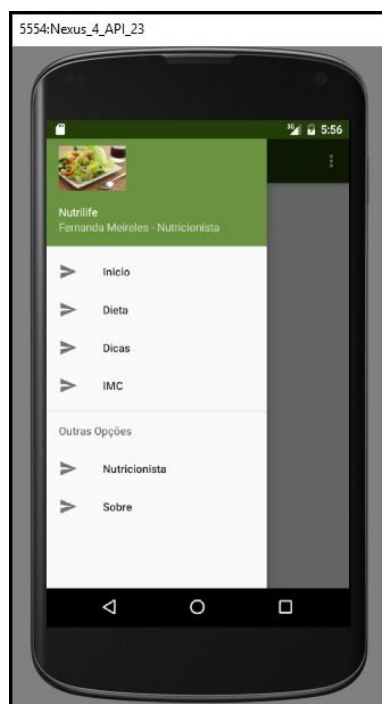


Fonte: Elaborado pelo autor, (2016).

4.3.6 *Navigation Drawer e Fragments*

O *Navigation Drawer* pode ser definido como uma *Activity* do Android utilizada para a criação de um menu deslizante lateral, ele já vem com o Android Studio, é leve e de fácil implementação, cria um menu que pode ser acessado por algum outro recurso.

Neste protótipo foram utilizados em conjunto com ele os *Fragments* que são módulos ou pequenas partes de tela, que são utilizados em casos como deste protótipo. Cada uma das divisões do menu acessa um *Fragment* que é acessado na visualização da tela principal. Dessa forma a cada seleção no menu é acessando ele ao invés de uma nova *Activity*. Pode-se ver na Figura 13 como fica o menu feito com o *Navigation Drawer*.

FIGURA 13 - *Navigation Drawer*

Fonte: Elaborado pelo autor, (2016).

Observa-se o Quadro 8 onde consta a seleção da *Activity*.

QUADRO 8 - Código da Seleção *Navigation Drawer*

```
@SuppressWarnings("StatementWithEmptyBody")
@Override
public boolean onNavigationItemSelected(MenuItem item) {

    android.app.FragmentManager fm = getFragmentManager();
    int id = item.getItemId();

    if (id == R.id.nav_início) {
        //Fazendo a ligação entre o fragmento e a opção selecionada
        fm.beginTransaction().replace(R.id.content_frame, new MainFragment()).commit();
    } else if (id == R.id.nav_dieta) {
        fm.beginTransaction().replace(R.id.content_frame, new DietaFragment()).commit();
    } else if (id == R.id.nav_dicas) {
        fm.beginTransaction().replace(R.id.content_frame, new DicasFragment()).commit();
    } else if (id == R.id.nav_imc) {
        fm.beginTransaction().replace(R.id.content_frame, new ImcFragment()).commit();
    } else if (id == R.id.nav_nutricionista) {
        fm.beginTransaction().replace(R.id.content_frame, new NutricionistaFragment()).commit();
    } else if (id == R.id.nav_sobre) {
        fm.beginTransaction().replace(R.id.content_frame, new SobreFragment()).commit();
    }

    DrawerLayout drawer = (DrawerLayout) findViewById(R.id.drawer_layout);
    drawer.closeDrawer(GravityCompat.START);
    return true;
}
```

Fonte: Elaborado pelo autor, (2016).

Estes *Fragments* do Quadro 8 são acessados por uma função chamada *onNavigationSelected* onde são criados *ifs* e *else ifs* que fazem a escolha do menu conforme o item selecionado.

4.3.7 Telas de Início e Nutricionista

A primeira *Fragment* que aparece do protótipo é a tela de início onde é apresentado boas vindas e um breve texto sobre o aplicativo conforme pode ser visto na Figura 14.

FIGURA 14 - *Fragment* início



Fonte: Elaborado pelo autor, (2016).

O protótipo apresenta um *Fragment* chamado de nutricionista onde se pode ver na Figura 15, onde constam informações sobre o profissional que o cliente possui vínculo. Podem ser encontrados nesta janela alguns dados importantes para o relacionamento entre paciente e nutricionista.

FIGURA 15 - *Fragment nutricionista*

Fonte: Elaborado pelo autor, (2016).

4.3.8 As Telas Dieta e Dicas

Pode ser na Figura 16 e Figura 17 as *Fragments* dieta e dicas, estas são as telas que mostram o conteúdo cadastrado como dica e como dieta.

FIGURA 16 - *Fragment dieta*

 A screenshot of a mobile application interface showing a table of diet records. The app is titled 'Nutrilife'. The table has four columns: 'Data', 'Refeição', 'Hora', and 'Dieta'. The data rows are as follows:

Data	Refeição	Hora	Dieta
		Quant./Porção	Alimento
2016-06-15	Café	06:00:00 3	Dieta Rica Torrada
2016-06-15	Almoco	12:00:00 1	Dieta Rica Arroz
2016-06-16	Lanche	09:00:00 1	Dieta B Iogurte

 The status bar at the top shows the time as 1:13.

Fonte: Elaborado pelo autor, (2016).

FIGURA 17 - *Fragment dicas*

Fonte: Elaborado pelo autor, (2016).

As *Fragments* Dieta e Dicas são parecidas, as mesmas retornam dados gravados no *Mysql* que por sua vez é acessado via servidor e gravado em *Json*, o que a aplicação faz é consumir este *Json* e mostrar em uma *Listview* a lista de dados gravados.

4.3.9 Wireframe de telas Web

Foi criado um modelo para a exibição de telas que seriam utilizadas pelo Nutricionista para devidos cadastros, este modelo pode ser chamado também de *Wireframe*, ou seja, um desenho básico.

Conforme já comentado neste trabalho, as telas da parte Web não foram desenvolvidas, porém o que foi feito é o *Wireframe* referente à tela de Cadastro de Pessoa e Cadastro de Dieta conforme visto no Quadro 9 e Quadro 10.

QUADRO 9 - Wireframe Cadastro Cliente

A Web Page
http://www.nutrilife.com.br/cadastrarciente.php

NutriLife | [Home](#) | [Cadastros Gerais](#) | [Cadastrar Dieta](#) | [Cadastar Pessoa](#) | [Editar Nutricionista](#)

Home
Cadastros Gerais
Cadastrar Dieta
Cadastar Cliente
Editar Nutricionista

Cadastro de Cliente

Nome: Endereço:

Peso: Cidade:

Altura: Telefone:

Usuario:

Senha:

Fonte: Elaborado pelo autor, (2016).

QUADRO 10 - Wireframe Cadastro Dieta

A Web Page
http://www.nutrilife.com.br/cadastrardieta.php

NutriLife | [Home](#) | [Cadastros Gerais](#) | [Cadastrar Dieta](#) | [Cadastar Pessoa](#) | [Editar Nutricionista](#)

Home
Cadastros Gerais
Cadastrar Dieta
Cadastar Cliente
Editar Nutricionista

Cadastro de Dieta

Hora: / / Tipo de Refeição:

Data: / / Alimento:

Quantidade: Categoria:

Tipo de Dieta:

Descrição:

Fonte: Elaborado pelo autor, (2016).

Observa-se no Quadro 9 e Quadro 10, as telas com que o Nutricionista deve cadastrar clientes e dietas, porém ainda e se faz necessário que se tenha cadastro de categoria, tipo de refeição, tipo de dieta, alimento, dicas e campo onde pode-se editar dados do Nutricionista, como se mostra no menu superior e menu esquerdo.

4.4 Considerações do Capítulo

Diante do tema proposto nesta monografia se fez necessário uma pesquisa detalhada e elaborada para apresentar um protótipo que pode relacionar o nutricionista a seu paciente.

Foi utilizado um conjunto de *Activitys* e *Fragments* para o desenvolvimento deste protótipo, a *Activity* básica utilizada foi a Navigation Drawer, que cria um menu lateral de navegação para ser utilizado juntamente com as *Fragment*, estas que são acessadas através do menu. Outras *Activitys* também foram usadas para fazer à tela de acesso a aplicação, ou seja, Login, a tela de apresentação da aplicação, chamada de Splash.

Também foi utilizado o servidor em PHP, para acesso aos arquivos cadastrados no Mysql através do *Json*, que é importante para trazer os dados para o protótipo. O apache é o servidor que roda o código em PHP que faz a pesquisa no BD e armazena em arquivo *Json*.

5 CONCLUSÕES

Nesta seção são discutidas em uma abordagem geral todas as informações levantadas durante o processo de desenvolvimento, seja na pesquisa ou na criação do protótipo.

5.1 Considerações Finais

É notável que o ser humano busque uma alimentação saudável desejando uma qualidade de vida ideal, em contrapartida a realidade de muitos casos é alimentar-se mal e não praticar exercícios. Muitas vezes devido a estes exemplos um individuo pode sofrer com distúrbios ou doenças que prejudicam sua saúde.

Então, surge a necessidade de buscar um auxílio profissional para que o ajude nesta situação, eis que entra em ação o Nutricionista, o profissional capacitado que deve analisar a situação de seu paciente e elaborar um plano nutricional específico para determinada pessoa, com base em pesquisas e até mesmo exames.

É importante salientar que este protótipo pretende auxiliar a ligação entre paciente e o profissional, e não, que o protótipo em si, calcule e diga qual o plano nutricional. Pois esta tarefa deve permanecer ao Nutricionista, pois existem diversas variáveis que devem ser analisadas conforme a necessidade.

O protótipo desenvolvido em Android, ferramenta escolhida pela grande quantidade de conteúdo em livros e internet e é claro, pela facilidade de programação. Outro ponto importante é o aumento de sua utilização nos últimos anos, este SO vem ganhando mercado impulsionado pelos aplicativos que são criados e por ser gratuito.

O protótipo é feito com o auxílio de uma *Activity* chamada de *Navigation Drawer* que é nativa do Android Studio, que possibilita a navegação entre menu lateral

esquerdo, onde que por sua vez aciona as chamadas *Fragments* que são partes de tela que podem ser acessadas na mesma visualização, faz comunicação com BD SQLite e Mysql via *Json* e os consome para ser tratado na aplicação.

De acordo com os objetivos propostos no que se diz respeito ao auxílio e acompanhamento buscando alcançar o resultado esperado pelos usuários e pela comunicação com o servidor web percebe-se a viabilidade do protótipo, ou seja, ele poderá ajudar a controlar a dieta, facilitando o relacionamento e repasse de informações entre nutricionista e cliente.

5.2 Trabalhos Futuros

Em continuidade desta monográfica é possível destacar alguns pontos, ou seja, destacar algumas situações que poderiam ser melhoradas ou até mesmo novas ideias:

- Desenvolvimento do sistema web onde o nutricionista iria alimentar os dados de acordo com a necessidade de cada paciente;
- Aprimorar algumas funcionalidades já existentes neste protótipo;
- Tornar este protótipo acessível a outros SO como Windows Phone e IOS;
- Desenvolver um cadastro completo de alimentos com todas as informações nutricionais, cadastro completo de tipos de dieta, tipos de refeição e dicas;
- Criar um acompanhamento de acordo com planejado e cumprido.

REFERÊNCIAS BIBLIOGRÁFICAS

ANDROID. **Activities**. Disponível em: <<http://developer.android.com/intl/pt-br/training/basics/activity-lifecycle/starting.html>>. Acesso em: 25 abr. 2016b.

ANDROID. **Android Studio**. Disponível em: <<https://developer.android.com/studio/index.html>>. Acesso em: 27 mai. 2016d.

ANDROID. **Dashboards**. Disponível em: <<http://developer.android.com/intl/pt-br/about/dashboards/index.html>>. Acesso em: 25 abr. 2016a.

ANDROID. **Managing AVDs with the AVD Manager**. Disponível em: <<http://developer.android.com/intl/pt-br/tools/devices/managing-avds.html>>. Acesso em: 25 abr. 2016c.

CAELUM. **REST**. Disponível em: <<https://www.caelum.com.br/apostila-vraptor-hibernate/rest/#11-2-caracteristicas-e-vantagens>>. Acesso em: 13 mai. 2016.

CORRÊA, Eduardo. **Introdução ao formato JSON**. Disponível em: <<http://www.devmedia.com.br/introducao-ao-formato-json/25275>>. Acesso em: 10 mai. 2016.

DEITEL, Paul; DEITEL, Harvey; DEITEL, Abbey. **Android para Programadores, Uma abordagem baseada em aplicativos**. 2. Ed. Porto Alegre: Bookman, 2015.

DEVMEDIA. **Android Fragments: Criando layouts flexíveis**. Disponível em: <<http://www.devmedia.com.br/android-fragments-criando-layouts-flexiveis/30859>>. Acesso em: 04 junho. 2016c.

DEVMEDIA. **Application Programming Interface: Desenvolvendo APIs de Software**. Disponível em: <<http://www.devmedia.com.br/application-programming-interface-desenvolvendo-apis-de-software/30548>>. Acesso em: 11 julho. 2016e.

DEVMEDIA. **Ciclo de vida do Fragments no Android**. Disponível em: <<http://www.devmedia.com.br/ciclo-de-vida-do-fragments-no-android/33099>>. Acesso em: 04 junho. 2016b.

DEVMEDIA. **Introdução ao MySQL**. Disponível em: <<http://www.devmedia.com.br/introducao-ao-mysql/28438>>. Acesso em: 11 julho. 2016d.

DEVMEDIA. **O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML**. Disponível em: <<http://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>>. Acesso em: 13 mai. 2016a.

DEVMEDIA. **Turbinando aplicações com REST e Backbone.js**. Disponível em: <<http://www.devmedia.com.br/turbinando-aplicacoes-com-rest-e-backbone-js/29692>>. Acesso em: 11 julho. 2016f.

GIARETTA, Gabriel. **Fluxogramas, diagrama de blocos e de Chapin no desenvolvimento de algoritmos**. 2013. Disponível em: <<http://www.devmedia.com.br/fluxogramas-diagrama-de-blocos-e-de-chapin-no-desenvolvimento-de-algoritmos/28550>>. Acesso em: 13 mai. 2016.

GIBNEY, Michael J.; HESTER, Vorster H.; KOK, Frans J.. **Nutrição Humana**. 1. Ed. Rio de Janeiro: Guanabara Koogan, 2005.

HENDRICKS, Mack; GALBRAITH, Ben; IRANI, Romin; MILBERNY, James; MODI, Tarak; TOST, Andre; TOUSSAINT, Alex; BASHA, S. Jeelani; CABLE, Scott. **Profissional Java Web Services**. 1. ed. Rio de Janeiro: Alta Books, 2002.

JSON. **Introdução ao JSON**. Disponível em: <<http://www.json.org/json-pt.html>>. Acesso em: 10 mai. 2016.

LECHETA, Ricardo R. **Google Android: aprenda a criar aplicações para dispositivos móveis com Android SDK**. 3. ed. São Paulo: Novatec, 2013.

MAHAN, L. Kathleen; ESCOTT-STUMP, Sylvia; RAYMOND, Janice L.. **Krause Alimentos, Nutrição e Dietoterapia**. 13. Ed. Rio de Janeiro: Elsevier, 2012.

MEDNIEKS, Zigurd; DORNIN, Laird; MEIKE, G. Blake; NAKAMURA, Masumi. **Programando o Android**. 2. ed. São Paulo: Novatec, 2012.

NETBEANS. **Introdução ao Web Services RESTful**. Disponível em: <https://netbeans.org/kb/docs/websvc/rest_pt_BR.html>. Acesso em: 13 mai. 2016.

ORACLE. **Mysql**. Disponível em: <<http://www.oracle.com/br/products/mysql/overview/index.html>>. Acesso em: 18 jun. 2016.

PEREIRA, Lucio Camilo Oliva; SILVA, Michel Lourenço. **Android para Desenvolvedores**. 2. ed. Rio de Janeiro: Brasport, 2012.

SOMMERVILLE, Ian. **Engenharia de Software**. 8. Ed. São Paulo: Pearson Addison-Wesley, 2007.

TECMUNDO. **O que é API?**. Disponível em: <<http://www.tecmundo.com.br/programacao/1807-o-que-e-api-.htm>>. Acesso em: 11 julho. 2016.

TONSIG, Sergio Luiz. **Engenharia de Software Análise e Projeto de Sistemas. 2.**
Ed. Rio de Janeiro: Editora Ciência Moderna Ltda., 2013.