

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Projeto e Análise de Algoritmos de Sugestão de Alimentos em Dispositivos Móveis: Uma abordagem Orientada a Critérios de Qualidade

Autor: Victor Cotrim de Lima
Orientador: Prof. Dr Mauricio Serrano

Brasília, DF
2016



Victor Cotrim de Lima

Projeto e Análise de Algoritmos de Sugestão de Alimentos em Dispositivos Móveis: Uma abordagem Orientada a Critérios de Qualidade

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr Mauricio Serrano

Coorientador: Profa. Dra Milene Serrano

Brasília, DF

2016

Victor Cotrim de Lima

Projeto e Análise de Algoritmos de Sugestão de Alimentos em Dispositivos Móveis: Uma abordagem Orientada a Critérios de Qualidade/ Victor Cotrim de Lima. – Brasília, DF, 2016-

118 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr Mauricio Serrano

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2016.

1. Algoritmo. 2. Sugestão de Alimentos. I. Prof. Dr Mauricio Serrano. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Projeto e Análise de Algoritmos de Sugestão de Alimentos em Dispositivos Móveis: Uma abordagem Orientada a Critérios de Qualidade

CDU 02:141:005.6

Victor Cotrim de Lima

Projeto e Análise de Algoritmos de Sugestão de Alimentos em Dispositivos Móveis: Uma abordagem Orientada a Critérios de Qualidade

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Prof. Dr Mauricio Serrano
Orientador

Profa. Dra Milene Serrano
Coorientador

Prof. Dr Edson Alves da Costa Junior
Convidado 1

Profa. Dra Carla Silva Rocha Aguiar
Convidado 2

Brasília, DF
2016

*Este trabalho é dedicado a minha companheira que,
sempre me apoiou quando mais precisei.
Aos meus amigos que me apoiaram a continuar sem hesitar.
E ao OnFit, que me deu a oportunidade de crescer.*

Agradecimentos

Agradeço a minha família que sempre esteve ao meu lado. Aos meus amigos, em especial o meu iOS Tutor Pedro pela ajuda com o material de ótima qualidade fornecido para estudo. Aos meus companheiros do OnFit que sempre me deixavam mais animados a cada encontro. E em especial aos meus orientadores de TCC e professores da FGA que me ajudaram sempre que uma dificuldade aparecia. Sou grato a todos.

*“Sempre comece algo importante com música,
nada disso seria tão importante se o som não estivesse ali.
Inclusive, não ouviria qualquer um.*

Resumo

Considerando a questão da reeducação alimentar os interessados sobre o assunto procuram por ajuda profissional de como adequar sua dieta alimentar rotineira para emagrecimento, manutenção de uma vida mais saudável e até mesmo ganho de massa, levando em conta os aspectos de perfil destes interessados: sexo, IMC(Índice de Massa Corpóreo), quantidade de dias no qual realiza uma atividade física durante a semana. Diante do exposto e dado que hoje temos a presença em massa de dispositivos móveis no cotidiano das pessoas, estudos sobre o uso desses dispositivos como meios de prover informações quanto à reeducação alimentar tornam-se relevantes, ampliando a difusão do conhecimento sobre boas práticas alimentares. Este cenário cabe alguns cuidados como: as informações providas pelo aplicativo em dispositivo móvel devem ter sido validadas com profissionais da área médica, no caso nutricionistas; as limitações de capacidade de memória e processamento, intrínsecas nos dispositivos móveis; e estudos quanto ao desempenho, à qualidade das recomendações, à usabilidade e demais critérios de qualidade associados ao contexto em questão. Portanto, este trabalho propõem o desenvolvimento de um algoritmo capaz de auxiliar na recomendação de alimentos, levando em conta: o perfil do interessado, a qualidade da dieta alimentar e questões de desempenho – dado que o ambiente utilizado será o de dispositivos móveis.

Palavras-chaves: *Knapsack*. Otimização. Desempenho em tempo e espaço. Engenharia de Software. Nutrição. Desenvolvimento em dispositivo móvel. Plataforma iOS.

Abstract

Considering the issue of re-education food concerned about it looking for professional help to adjust your routine diet for weight loss, maintaining a healthier life and even gain weight, taking into account the profile aspects of these stakeholders: sex, BMI (body mass index), number of days in which performs a physical activity during the week. Given the above and given that today we have the massive presence of mobile devices in daily life, studies on the use of these devices as a means of providing information about the nutritional education become relevant, expanding the dissemination of knowledge about good dietary practice. This scenario fits some care as the information provided by the mobile device application must have been validated with medical professionals, nutritionists in the case; the memory capacity constraints and processing inherent in mobile devices; and studies of the performance, the quality of recommendations, usability and other quality criteria linked to the context in question. Therefore, this paper proposes the development of an algorithm to assist in food recommendation, taking into account: the person's profile, the quality of diet and performance issues - as the environment used will be the mobile device.

Key-words: *Knapsack*. Optimization. Time and Space Performances. Software Engineering. Nutrition. Mobile development. iOS Platform.

Lista de ilustrações

| | |
|--|----|
| Figura 1 – Ligação peptídica | 34 |
| Figura 2 – Função dos Minerais | 35 |
| Figura 3 – Pirâmide alimentar brasileira com porções alocadas pelo TecnoNutri . . | 49 |
| Figura 4 – Distribuição das porções em relação as refeições | 50 |
| Figura 5 – Fonte: (CHACON; STRAUB, 2016, p. 95) Figura 3-17 | 52 |
| Figura 6 – Trello usado pelos time de desenvolvimento OnFit | 53 |
| Figura 7 – Visão geral da metodologia | 56 |
| Figura 8 – Metodologia Scrum segundo (SCHWABWER; SUTHERLAND, 2013) . | 59 |
| Figura 9 – Trello sendo utilizado como Kanbam | 61 |
| Figura 10 – Utilização de <i>Card</i> como História de Usuário | 62 |
| Figura 11 – Estratégia para análise dos dados | 63 |
| Figura 12 – Trello sendo utilizado para controle de <i>Sprint</i> | 81 |
| Figura 13 – Esquemático de módulos e como se relacionam | 85 |
| Figura 14 – Cobertura de código provida pelo XCode e XCTest <i>framework</i> | 85 |
| Figura 15 – Testes de desempenho para o knapsack realizado ao final do ciclo 2 . . | 86 |
| Figura 16 – Gráfico tempo de execução X entradas do algoritmo, com e sem otimização | 87 |

Lista de tabelas

| | |
|---|-----|
| Tabela 1 – Valor de Atividade física em relação ao nível de atividade na semana . | 39 |
| Tabela 2 – Cálculo de ajuste empirico mediante objetivos dos pacientes | 39 |
| Tabela 3 – Faixa de valor aceitável diário de macronutrientes | 39 |
| Tabela 4 – Ingestão Adequada diária de fibras | 39 |
| Tabela 5 – Valores em porções | 40 |
| Tabela 6 – Tabela de itens | 43 |
| Tabela 7 – Exemplo de perfil de paciente para geração da lista de alimentos sugeridos | 62 |
| Tabela 8 – Exemplo de lista sugerida de alimentos conforme pefil de paciente . . . | 63 |
| Tabela 9 – Total de macro nutrientes a serem consumidos em relação ao EER . . . | 67 |
| Tabela 10 – Fracionamento das gramas em cada refeições | 67 |
| Tabela 11 – Exemplo da formação dos itens para o <i>knapsack</i> conforme alimentos . . | 68 |
| Tabela 12 – Exemplo de valoração dos itens | 70 |
| Tabela 13 – Campos e tamanhos da estrutura para cada item | 70 |
| Tabela 14 – Máscara de bit possíveis | 75 |
| Tabela 15 – Histórias de Usuário levantadas | 79 |
| Tabela 16 – Completude das Histórias de Usuário. Tarefas feitas por tarefas levantadas. | 81 |
| Tabela 17 – Configurações iPhone 5C | 82 |
| Tabela 18 – Requisitos não funcionais: Título e Descrição | 82 |
| Tabela 19 – Requisitos não funcionais: Interdependências, Conflitos e Impactos . . | 83 |
| Tabela 20 – Cenários utilizados para balanceamento de valores | 88 |
| Tabela 21 – Vantagens e Desvantagens entre métodos retrospectivos | 104 |
| Tabela 22 – Classificação Estado nutricional de acordo com IMC | 106 |
| Tabela 23 – Valor de Atividade física em relação ao nível de atividade na semana . | 110 |
| Tabela 24 – Cálculo de ajuste empirico mediante objetivos dos pacientes | 111 |
| Tabela 25 – Faixa de valor aceitável diário de macronutrientes | 111 |
| Tabela 26 – Ingestão Adequada diária de fibras | 111 |
| Tabela 27 – Valores em porções | 112 |
| Tabela 28 – Histórias de Usuário e suas respectivas tarefas | 115 |

Código

| | | |
|-----|--|----|
| 5.1 | Configuração inicial tabela <i>knapsack</i> | 72 |
| 5.2 | Recuperação de itens do <i>knapsack</i> comumente usada | 73 |
| 5.3 | Método de recuperação de itens | 74 |
| 5.4 | Recuperação de itens a partir dos bytes da última célula | 76 |

Lista de abreviaturas e siglas

| | |
|--------------|--|
| <i>BPMN</i> | Business Process Model and Notation |
| <i>EER</i> | Estimativa de Energia Recomendada |
| <i>IMC</i> | Índice de Massa Corporeo |
| <i>GEB</i> | Gasto Energético Basal |
| <i>FAF</i> | Fator Atividade Física |
| <i>BEPiD</i> | Brazilian Education Programm for iOS Development |
| <i>DRI</i> | <i>Dietary Reference Intake</i> |
| <i>IA</i> | Ingestão Adequada |
| <i>IM</i> | Ingestão Máxima Tolerável |
| <i>QFCA</i> | Questionário de Frequência de Consumo Alimentar |
| <i>TCC</i> | Trabalho de Conclusão de Curso |
| <i>CSV</i> | Coma Separated Value |

Lista de símbolos

| | |
|-----|-----------------------|
| P | Peso em Kilogramas |
| A | Altura em Metros |
| h | Altura em centímetros |
| t | Idade em anos |

Sumário

| | | |
|----------|---------------------------------------|-----------|
| | Código | 19 |
| 1 | INTRODUÇÃO | 29 |
| 1.1 | Contextualização | 29 |
| 1.2 | Questão de Desenvolvimento | 30 |
| 1.3 | Justificativa | 31 |
| 1.4 | Objetivos | 31 |
| 1.4.1 | Geral | 31 |
| 1.4.2 | Específicos | 31 |
| 1.5 | Considerações Finais | 32 |
| 2 | REFERENCIAL TEÓRICO | 33 |
| 2.1 | Nutrição | 33 |
| 2.1.1 | Macronutrientes | 33 |
| 2.1.2 | Micronutrientes | 35 |
| 2.1.3 | Estudo Nutricional | 36 |
| 2.1.4 | Avaliação Nutricional | 37 |
| 2.2 | Engenharia de Software | 40 |
| 2.2.1 | Requisitos de Software | 40 |
| 2.2.2 | Métricas | 41 |
| 2.2.3 | Algoritmos Gulosos | 42 |
| 2.2.4 | Programação Dinâmica | 42 |
| 2.2.5 | Complexidade | 43 |
| 2.2.6 | Scrum | 45 |
| 2.3 | Considerações Finais | 45 |
| 3 | SUPORTE TECNOLÓGICO | 47 |
| 3.1 | Aplicativo de Nutrição | 47 |
| 3.1.1 | Dieta e Saúde | 48 |
| 3.1.2 | TecnoNutri | 48 |
| 3.2 | Ferramentas de Engenharia de software | 49 |
| 3.2.1 | Xcode | 49 |
| 3.2.2 | Git | 51 |
| 3.2.3 | Bonita | 52 |
| 3.2.4 | Trello | 52 |
| 3.2.5 | PunchTime | 52 |

| | | |
|------------|---|-----------|
| 3.3 | Considerações Finais | 53 |
| 4 | METODOLOGIA | 55 |
| 4.1 | Metodologia para Desenvolvimento TCC | 55 |
| 4.1.1 | Revisão Bibliográfica | 55 |
| 4.1.2 | Prova de Conceito | 56 |
| 4.1.3 | Escrita | 57 |
| 4.2 | Metodologia para Desenvolvimento do Software | 57 |
| 4.2.1 | Adaptação Adotada | 59 |
| 4.2.2 | Suporte Ferramental de Metodologia | 60 |
| 4.3 | Metodologia de Análise dos Resultados | 61 |
| 4.3.1 | Obtenção e Tabulação dos Dados | 62 |
| 4.3.2 | Estratégia da Análise de Dados | 63 |
| 4.3.3 | Resultados da Análise de Dados | 63 |
| 4.3.4 | Conclusão da Análise de Dados | 64 |
| 4.4 | Considerações Finais | 64 |
| 5 | ALGORITMO PARA SUGESTÃO DE ALIMENTOS | 65 |
| 5.1 | Problema | 65 |
| 5.2 | Solução | 65 |
| 5.2.1 | Preparação dos Dados | 66 |
| 5.2.2 | Formulação dos Itens Conforme Alimentos Escolhidos | 68 |
| 5.2.3 | Cálculo do Valor dos Itens | 68 |
| 5.2.4 | Uso e Análise do <i>knapsack</i> | 70 |
| 5.3 | Considerações Finais | 77 |
| 6 | RESULTADOS OBTIDOS | 79 |
| 6.1 | Análise dos Requisitos | 79 |
| 6.1.1 | Requisitos Funcionais | 79 |
| 6.1.2 | Requisitos Não-Funcionais | 82 |
| 6.2 | Módulos do Código | 84 |
| 6.3 | Testes Unitários | 84 |
| 6.4 | Testes de Desempenho | 86 |
| 6.5 | Análise de Complexidade | 86 |
| 6.6 | Ciclos de Ajuste do Algoritmo | 88 |
| 6.6.1 | Ciclo 1 | 89 |
| 6.6.2 | Ciclo 2 | 89 |
| 6.6.3 | Ciclo 3 | 91 |
| 6.6.4 | Ciclo 4 | 92 |
| 6.6.5 | Conclusão dos Ciclos | 92 |

| | | |
|-------|--|------------|
| 6.7 | Considerações Finais | 92 |
| 7 | CONCLUSÃO | 95 |
| 7.1 | Sugestão de Trabalhos Futuros | 96 |
| | REFERÊNCIAS | 97 |
| | APÊNDICES | 101 |
| | APÊNDICE A – REFERENCIAL TEÓRICO | 103 |
| A.1 | Nutrição | 103 |
| A.1.1 | Métodos Retrospectivos | 103 |
| A.1.2 | Avaliação Antropométrica | 105 |
| A.1.3 | Avaliação Bioquímica | 108 |
| A.1.4 | Avaliação Nutricional | 108 |
| | APÊNDICE B – HISTÓRIAS DE USUÁRIO | 115 |

1 Introdução

Este capítulo tem como intuito introduzir o leitor deste Trabalho de Conclusão de Curso, em uma explicação do que há de ser este trabalho.

Na Seção 1.1 é feita a contextualização deste trabalho, no âmbito da área de nutrição no Brasil e da área de aplicativos mobile. A Seção 1.2 apresenta a questão de pesquisa, que evidencia o problema referenciado na contextualização. A Seção 1.3 mostra a justificativa deste trabalho, na qual a relevância e importância da proposta é feita em conformidade com a contextualização. E na última Seção, 1.4, são apresentados os objetivos geral e específicos deste trabalho, alinhados ao problema e a proposta deste trabalho.

1.1 Contextualização

A má educação alimentar é uma realidade no contexto brasileiro. Esta situação é refletida nas estatísticas feitas pela VIGITEL (Vigilância de Fatores de Risco e Proteção para Doenças Crônicas por Inquérito Telefônico) em 2013, que segundo a pesquisa, 50,8% da população brasileira está acima do peso ideal. Adicionalmente, nesse grupo, tem-se ainda que 17,5% são obesos.

Ainda considerando a questão da educação alimentar, indicam-se consultas a profissionais, normalmente nutricionistas, visando obter recomendações quanto à adequada dieta alimentar, dado o perfil do interessado e as condições para que este se mantenha saudável. Apesar dessa consulta ao profissional ser adequada, outras formas de acesso à sugestão alimentar poderiam permitir aos interessados adquirir conhecimentos sobre a importância de uma dieta alimentar balanceada para manter uma vida saudável.

Diante do exposto e dado que hoje temos a presença em massa de dispositivos móveis no cotidiano das pessoas, estudos sobre o uso desses dispositivos como meios de prover informações quanto à reeducação alimentar tornam-se relevantes, ampliando a difusão do conhecimento sobre boas práticas alimentares.

Entretanto, cabem alguns cuidados nessa abordagem de facilidade de acesso, dentre eles: (i) as informações providas pelo aplicativo no dispositivo móvel devem ter sido validadas por profissionais devidamente capacitados da área médica, no caso nutricionistas, psicólogos e outros; (ii) as limitações de capacidade de memória e processamento, intrínsecas nos dispositivos móveis, devem ser contornadas com base no projeto e na análise de algoritmos específicos para o fim proposto, e (iii) os estudos quanto ao desempenho, à qualidade das sugestões, à usabilidade e demais critérios de qualidade associados

ao contexto em questão demandam esforços por parte dos engenheiros de software.

O aplicativo OnFit (ANDRADE; BERNARDES; COTRIM, 2015) é um exemplo de iniciativa nesse contexto. Esse aplicativo foi proposto pelos desenvolvedores Lucas Andrade, Thiago Bernardes e Victor Cotrim (autor desse TCC), todos associados ao projeto BEPiD¹ (Universidade Católica de Brasília, 2015), ao longo do período compreendido entre 30 de Março de 2014 e 27 de Março de 2015. Como foi dito anteriormente, alguns cuidados devem ser tomados quanto a facilidade de acesso, sendo um destes a equipe médica de suporte especializado. No caso do aplicativo, essa equipe foi formada por: uma nutricionista, uma endocrinologista e um médico orto-molecular.

Atualmente, o aplicativo OnFit encontra-se em sua primeira versão, disponível na App Store². Apesar do aplicativo ter sido disponibilizado para uso, o mesmo ainda demanda refinamentos significativos quanto às questões de: (i) limitações de capacidade de memória e processamento, e (ii) qualidade no que tange o desempenho, a adequação das sugestões alimentares, a usabilidade e outros critérios de qualidade.

1.2 Questão de Desenvolvimento

A análise dos alimentos a serem consumidos, quando feita por um nutricionista, sempre leva em conta os seus macro e micro nutrientes (ONFIT, 2014). Essa análise nem sempre é facilmente obtida, uma vez que o nutricionista deverá considerar, dentre outros aspectos do perfil do paciente: sexo, IMC(Índice de Massa Corpóreo), quantidade de dias no qual realiza as atividades físicas durante a semana, e o objetivo do paciente – emagrecimento, ganho de massa ou manutenção de peso.

Portanto, o trabalho proposto procurará responder a seguinte questão de desenvolvimento: Como desenvolver um algoritmo capaz de auxiliar na sugestão de alimentos, levando em conta o perfil do interessado, a qualidade da sugestão fornecida, questões de desempenho e demais critérios de qualidade relevantes para acesso em dispositivos móveis? E se possível, em tempo hábil, como adaptar o algoritmo de auxílio na sugestão de alimentos sob o ponto de vista dos nutricionistas, auxiliando-os na análise dos alimentos recomendados?

Ainda é pretendido, se possível em tempo hábil, adaptar a solução proposta sob o ponto de vista dos nutricionistas, permitindo aos mesmos a análise da sugestão de alimentos considerando especificamente cada interessado. Nesse cenário, seria possível o nutricionista analisar e alterar os alimentos em quantidade, e substituir esses últimos por outros alimentos com propriedades alimentares semelhantes, caso haja necessidade.

¹ Brazilian Education Programm for iOS Development

² Link para download: <https://itunes.apple.com/us/app/onfit/id964375064?l=pt&ls=1&mt=8>

Ainda é pertinente ser dito que a produção de um algoritmo de sugestão de alimentos não substitui a consulta com um profissional da nutrição.

1.3 Justificativa

Considerando estudos quanto à dieta alimentar dos Brasileiros, tem-se que as recomendações devem se basear em alimentos mais do que em nutrientes (SICHIERI et al., 2000) . O Ministério da Saúde, em 2014, disponibilizou um guia alimentar (SAÚDE; BÁSICA, 2014) no intuito de orientar a população Brasileira sobre alimentação saudável e balanceada.

Esse guia informa sobre a necessidade de uma reeducação alimentar no âmbito nacional, salientando a importância de pesquisas nessa área e procurando ampliar as vias de sugestão de alimentos. Outro fator que corrobora com essa necessidade é a questão da obesidade sendo vista como uma epidemia, não apenas no Brasil, também no mundo (PEREIRA, 2007).

Nesse cenário, pretende-se, com a proposta aqui apresentada, refinar o aplicativo OnFit, visando adequadas sugestões de alimentos aos interessados, via dispositivos móveis, e alinhadas aos objetivos específicos destes interessados - emagrecimento, ganho de massa ou manutenção de peso.

1.4 Objetivos

Seguem os objetivos, geral e específicos, da proposta.

1.4.1 Geral

Projetar e analisar algoritmos específicos para sugestão de alimentos, usando uma abordagem orientada a critérios de qualidade, na qual destacam-se estudos quanto ao desempenho em dispositivos móveis, qualidade da sugestão recomendada e usabilidade do aplicativo.

1.4.2 Específicos

1. Investigar possíveis soluções candidatas para viabilizar a sugestão de alimentos, levando em conta o perfil e os objetivos desejados pelo interessado bem como considerando questões de desempenho e usabilidade em dispositivos móveis;
2. Realizar experimentos para seleção de uma solução adequada à questão de desenvolvimento, considerando as soluções candidatas identificadas no objetivo anterior;

3. Documentar essa solução, gerando artefatos orientados às boas práticas da Engenharia de Software em diferentes níveis de abstração: requisitos, projeto, codificação e teste.

1.5 Considerações Finais

Foi mostrada neste capítulo como os problemas são alinhados a questão de pesquisa e desenvolvidos de forma coerente aos objetivos geral e específicos deste Trabalho de Conclusão de Curso, na qual uma justificativa é feita para reforçar a necessidade e relevância do tema proposto.

Os capítulos seguintes contemplam as explicações referentes a contextualização para este TCC.

O Capítulo 2 contempla às explicações acerca do referencial teórico tanto para a área de Nutrição quanto para a área de Engenharia de Software.

O Capítulo 3 contempla às ferramentas utilizadas como suporte tecnológico para desenvolvimento deste TCC.

O Capítulo 4 contempla a metodologia utilizada para desenvolvimento deste TCC, desenvolvimento do Software e análise dos resultados.

O Capítulo 5 contempla a solução feita para demonstrar como os objetivos específicos são abordados e completos. Neste capítulo é utilizado todo o contexto abordado nos capítulos 2, 3 e 4.

O Capítulo 6 contempla os resultados obtidos para este TCC em relação ao algoritmo desenvolvido e a evolução do mesmo.

O Capítulo 7 contempla a conclusão deste TCC e mostra quais são os próximos passos a serem seguidos.

2 Referencial Teórico

Este capítulo tem como intuito mostrar os fundamentos teóricos em relação a área de Nutrição e da área de Engenharia de Software que são utilizados no decorrer da leitura deste Trabalho de Conclusão de Curso.

Na Seção 2.1 é abordado sobre a fundamentação teórica acerca da área de Nutrição. Nesta seção são definidos: os conceitos dos Macronutrientes e Micronutrientes, o estudo nutricional relacionado aos pacientes - com o intuito de investigar e obter as informações dos mesmos -, os métodos para obtenção dos dados antropométricos relacionados aos pacientes e como ocorre o processo para a geração de dietas.

Na Seção 2.2 é abordado acerca das fundamentações teóricas referentes a área de Engenharia de Software. Nesta seção são definidos: os grupos de requisitos de software a serem utilizados, as métricas e seus tipos de escalas, os tipos de estratégia de programação utilizadas e a complexidade existente em termos de algoritmo.

2.1 Nutrição

Esta seção trata sobre como a área da Nutrição aborda sobre as definições mais básicas, desde dos Macronutrientes e Micronutrientes aos passos de como uma dieta pode ser gerada.

2.1.1 Macronutrientes

Os macronutrientes são macromoléculas existentes em "[...] estruturas vegetal e animal que podem ser digeridas, absorvidas e utilizadas por um outro organismo [...] como substratos para a síntese de carboidratos, gorduras e proteínas necessárias para manter a integridade da célula do sistema", segundo (MAHAN; ESCOTT-STUMP, 2010, p. 41). E ainda, os macronutrientes podem ser divididos em:

- Carboidratos;
- Proteínas;
- Lipídeos;

Segundo (MAHAN; ESCOTT-STUMP, 2010) os Carboidratos são produzidos pelos vegetais e são uma importante fonte de energia na dieta, compreendendo cerca de metade do total de calorias. (PHILIPPI, 2008) diz que a classificação dos Carboidratos

por muito tempo foram divididas em simples e complexos, mediante a sua cadeia carbônica, entretanto a classificação atual indica que a mesma deva ser feita através do seu grau de polimerização¹ dividindo-os em três principais grupos: açúcares, oligossacarídeos e polissacarídeos.

As Proteínas são compostas por ligações peptídicas² entre os aminoácidos formadores dela, vide Figura (1). As funções desempenhadas, segundo (PHILIPPI, 2008, p. 109) vão desde proteínas estruturais a formação de hormônios, proteínas do sistema imune, enzimas e nucleoproteínas. Elas podem ser classificadas em até 4 níveis estruturais: Estrutura primária, Estrutura secundária, Estrutura terciária e Estrutura quaternária.

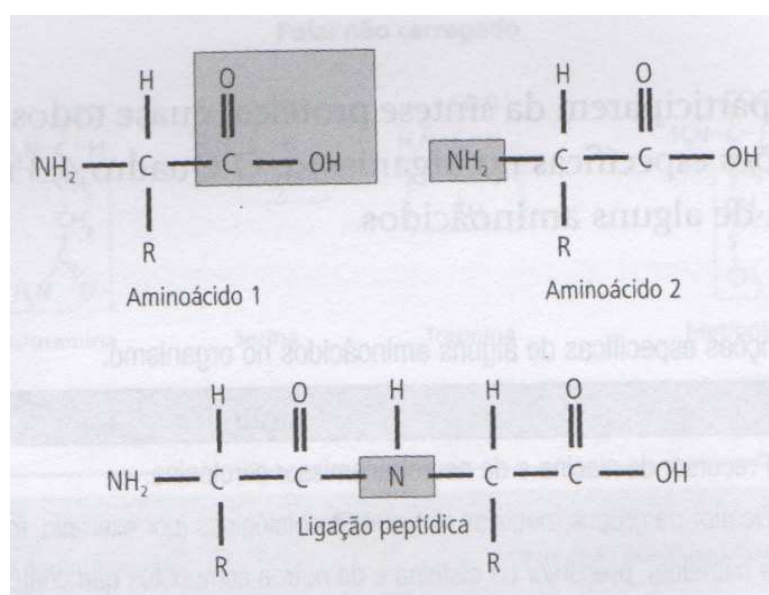


Figura 1: Ligação peptídica

Fonte: (PHILIPPI, 2008, p. 108)

Os Lipídios são constituídos por um conjunto de substâncias orgânicas existentes em óleos e gorduras, sendo insolúvel em água e solúvel em solventes orgânicos tais como éter e benzeno. Os óleos são provenientes de alimentos de origem vegetal, sendo extraídos de sementes ou frutas de várias plantas, em seguida são refinados e usados para a obtenção de margarinas e gordura vegetal hidrogenada. As gorduras são de origem animal, na qual o consumo da mesma pode acontecer integrado ao alimento. Tanto óleos quanto gorduras são importantes na alimentação, pois são essenciais para o fornecimento de energia e transporte de vitaminas lipossolúveis (PHILIPPI, 2008)

¹ Quantidade de ligações glicosídicas presente na molécula

² "Ligação formada entre o grupo carboxila de um aminoácido com o grupo amina do próximo aminoácido, com perda de uma molécula de água, obtendo assim os resíduos dos aminoácidos ligados" (PHILIPPI, 2008)

2.1.2 Micronutrientes

Diferentemente dos macronutrientes, os micronutrientes são compostos por vitaminas e minerais. Tanto as vitaminas quanto os minerais "são essenciais à manutenção do processo metabólico fazendo parte de algumas enzimas de hormônios que regulam a atividade fisiológica" (SILVA; MURA, 2007).

As vitaminas não são utilizadas como fonte de energia, porém tem um papel importante na regulação das reações de produção de energia.

De acordo com (MAHAN; ESCOTT-STUMP, 2010, p. 68):

O termo **vitaminas** descreve um grupo de micronutrientes que geralmente preenchem os seguintes critérios: (1) compostos orgânicos distintos (ou classe de compostos) das gorduras, carboidratos ou proteínas, (2) componentes naturais dos alimentos; normalmente presente em quantidades mínimas, (3) não são sintetizadas pelo corpo em necessidades adequadas para suprir as necessidades fisiológicas normais.

Os minerais são os responsáveis tanto no processo de produção de energia quanto na modulação do catabolismo de Carboidratos, Lipídeos e Proteínas, vide Figura 2. Assim como as vitaminas, os minerais são essenciais a manutenção dos processos metabólicos. (SILVA; MURA, 2007, P.226) ainda diz, "Além disso, os minerais atuam na manutenção das estruturas celulares e teciduais e estão envolvidas na contração muscular e na resposta nervosa"

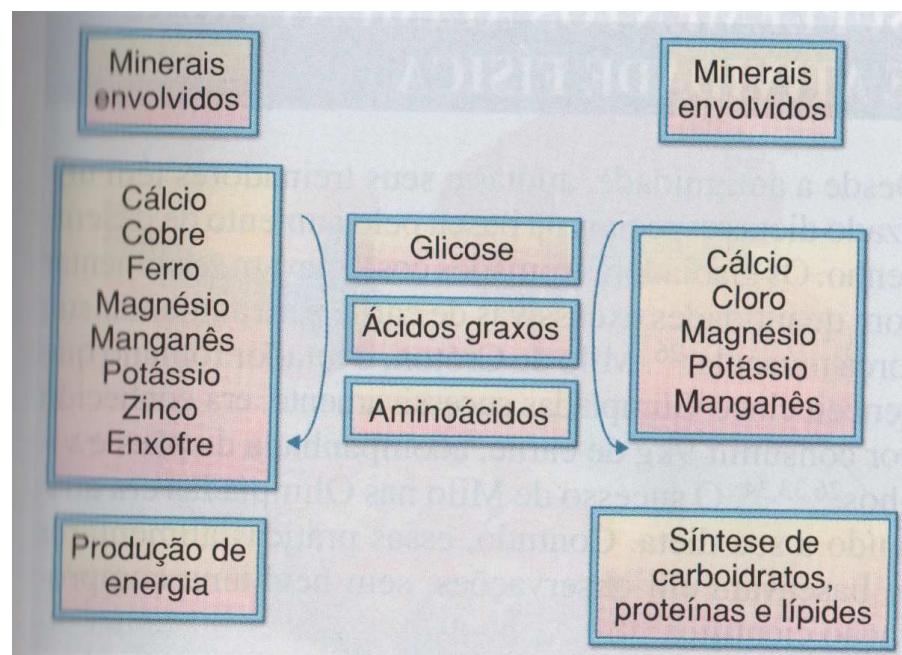


Figura 2: Função dos Minerais

Fonte: (SILVA; MURA, 2007, p. 227)

2.1.3 Estudo Nutricional

Este trabalho tem como intuito a produção de um algoritmo que faça a recomendação alimentar. Para esta tarefa é necessário o conhecimento do processo seguindo por um profissional da área de nutrição, desde o conhecimento da atual dieta seguida por seus pacientes até a entrega da dieta, explicando o porque das mudanças de alimentos, funcionalidades e a sua devida importância(educação nutricional).

Nesta seção é feita a discussão sobre o paciente, conhecendo suas manias, cotidiano, horários, hábitos alimentares, queixas e objetivos. Sendo este processo análogo ao processo de licitação de requisitos para um projeto de software.

Os dados providos da Ingestão Alimentar podem ser de caráter qualitativos e/ou quantitativos, nos quais são obtidos utilizando métodos prospectivos ou retrospectivos³, segundo (SILVA; MURA, 2007).

Métodos Prospectivos

Existe um pequeno *gap*⁴ para conseguir com certa exatidão quanto a quantidade de comida realmente ingerida. Um método prospectivo não requer grande esforço por parte do entrevistado quanto a utilização de memória, isto é uma vantagem, uma vez que este método procura ser realizado no cotidiano do entrevistado. Um método muito utilizado é o Registro Alimentar.

(SILVA; MURA, 2007) diz que o Registro Alimentar é de utilização no momento de consumo, pois o entrevistado irá registrar todos os alimentos e bebidas ingeridas, bom como as suas medidas caseiras. A quantidade de dias no qual este método será utilizado somente depende do objetivo do investigador, mas atualmente o mais comum é o uso do registro de três dias: 1 dia do fim de semana e 2 dias da semana alternados. Este método requer do entrevistado um treinamento para anotação correta das medidas caseiras e da forma de preparo dos alimentos.

A grande vantagem, como já foi citado, é a não utilização da memória do entrevistado. O trecho abaixo mostra a desvantagem na visão do autor:

Como desvantagem o método pode interferir no padrão alimentar usual, levando principalmente à subestimativa do consumo. [...]. Alguns indivíduos podem fazer um registro tendencioso nos dias de coleta dos dados. Os índices mais altos de sub-relato foram identificados entre indivíduos com Índice de Massa Corpórea mais elevados, principalmente mulheres. Esse efeito pode ser um reflexo do maior hábito de seguimento de dietas entre indivíduos mais pesados.

³ Vide seção A.1.1 para maiores esclarecimentos

⁴ Vide nota de rodapé anterior

2.1.4 Avaliação Nutricional

Fazer uma análise do paciente nos quesitos de: "o que ele come?" e "qual o estado que o seu corpo se encontra?", pode ser feito através de um estudo nutricional. Este estudo engloba avaliações antropométrica⁵ e bioquímica⁶. Além das avaliações já ditas, ainda são considerados a análise do histórico clínico, social, bioquímicos e a interação entre drogas e nutrientes (SILVA; MURA, 2007).

Após o estudo nutricional como um todo é necessário fazer a orientação e planejamento dietético do paciente, ou seja, a avaliação nutricional.

Alguns passos são considerados ao se fazer o estudo desta avaliação nutricional. Em alto nível pode ser citado os seguintes passos:

1. Estudo das patologias do paciente e seu estado nutricional;
2. Análise dos resultados provenientes da avaliação dietética;
3. Cálculo das necessidades calóricas, de macros e micros nutrientes;
4. Montagem da dieta de acordo com normas de saúde, necessidades e preferências do paciente;
5. Montagem da lista de substituição de acordo com a dieta gerada;
6. Conversão das medidas usadas para cálculos das porções de cada alimento para medidas caseiras;
7. Entrega da dieta com explicações acerca das mudanças de alimentos, suas funcionalidades e importância;

Para este TCC, serão tomados os passos 3 e 4 como essenciais⁷. Pois estes são os tópicos necessários para o norteamento dos objetivos e fundamentação da proposta.

Passo 3: Cálculo das Necessidades

Neste passo são feitos os cálculos, baseados nas medidas antropométricas, das necessidades calóricas e de macro e micro nutrientes. O objetivo deste passo é tentar manter aquilo que o paciente ingere habitualmente com a nova dieta que será montada, assim esta adaptação de dieta será melhor aceita.

O nutricionista pode manter algumas alimentos que o paciente habitualmente ingere, mas se o mesmo faz mal não é indicado fazer a retirada completa do alimento e sim fazer a diminuição do mesmo para não haver um impacto tão grande em relação a dieta.

⁵ Vide seção A.1.2 para mais informações

⁶ Vide seção A.1.3 para mais informações

⁷ Para mais informações sobre todos os passos vide seção A.1.4

Os nutricionistas usam as chamadas DRIs⁸ como referência de valores. A concepção de DRI é "um termo global concebido para englobar os quatro tipos específicos de nutrientes (ingestão adequada [IA], necessidade média estimada [EAR], cota dietética recomendada [EAR] e grau de ingestão máxima tolerável[IM])" (MAHAN; ESCOTT-STUMP, 2010).

Para (SILVA; MURA, 2007) ainda existe mais uma referência de valor chamada de estimativa das necessidades de energia (EER). Ainda é dito que destas 5 medidas, 4 são utilizadas para prescrição das dietas : EER, RDA, EAR e AI. A medida de EAR somente deve ser usada se conhecido o desvio-padrão do grupo (SILVA; MURA, 2007, p. 385).

Para (VOLP et al., 2011) o cálculo da EER é baseada no Gasto Energético Basal (GEB). Uma vez feito o cálculo do GEB, o fator de atividade física é aferido na tabela 1 conforme a quantidade de dias na semana especificados pelo paciente e em seguida o valor de EER é ajustado conforme tabela de objetivos (ONFIT, 2015, Entrevista nutricionista Silvia).

Para dar início ao passo seguinte é necessário o cálculo de algumas variáveis como o valor de GEB e EER e as quantidades diárias de carboidratos, proteínas, fibras e lipídeos.

O primeiro cálculo realizado é o de gasto calórico do paciente. O cálculo pode ser feito usando a fórmula 2.1 para homens e a formula 2.2 para mulheres (VOLP et al., 2011, p. 437 Tabela 2).

$$\begin{aligned} P &= \text{Peso}(kg) \\ h &= \text{Altura}(cm) \\ t &= \text{idade}(Anos) \\ GEB &= 66.5 + (13.75 \cdot P) + (5.003 \cdot h) - (6.775 \cdot t) \end{aligned} \tag{2.1}$$

$$GEB = 655.1 + (9.563 \cdot P) + (1.85 \cdot h) - (4.676 \cdot t) \tag{2.2}$$

O nível de atividade física pode ser calculado com base na tabela 1 utilizando o Fator Atividade Física (FAF), sendo os valores adquiridos de forma empírica conforme o nível de atividade física executado durante a semana (ONFIT, 2015, Entrevista nutricionista Silvia).

Utilizando-se das equações 2.1 e 2.2 e tabela 1, a necessidade energética pode ser calculada mediante equação 2.3.

$$EER = GEB \cdot FAF \tag{2.3}$$

⁸ Ingestão dietética de referência, ou no original *Dietary Reference Intake*

Tabela 1: Valor de Atividade física em relação ao nível de atividade na semana

Fonte: (ONFIT, 2015, Entrevista nutricionista Silvia)

| Nível atividade física(dias/semana) | Valor FAF | Classificação Paciente |
|-------------------------------------|-----------|------------------------|
| 0 | 1 | Sedentário |
| 1-2 | 1.2 | Pouco Sedentário |
| 3-4 | 1.375 | Pouco Ativo |
| 4-5 | 1.55 | Moderado |
| 6-7 | 1.725 | Ativo |

Conforme tabela 2, o ajusta da EER é feito em consonância com os objetivo do paciente.

Tabela 2: Cálculo de ajuste empirico mediante objetivos dos pacientes

Fonte: (ONFIT, 2015, Entrevista nutricionista Silvia)

| Objetivos | Fórmula de Ajuste |
|-----------------|-------------------|
| Perder Peso | EER - 500 |
| Ganhar Músculos | EER + 500 |
| Manter Peso | EER |

Ao que diz respeito aos cálculos, (SILVA; MURA, 2007) relacionou o Institute of Medicine quanto as referências relacionadas aos macronutrientes em uma faixa de distribuição aceitável. A tabela 3 relaciona os valores adequados para cada macro, em porcentagem, utilizando como referência o EER.

Tabela 3: Faixa de valor aceitável diário de macronutrientes

| Macronutriente | Faixa de valor aceitável |
|----------------|--------------------------|
| Proteína | 10 a 35% |
| Lípides | 20 a 35% |
| Carboidratos | 45 a 65% |

Para as fibras uma tabela foi concebida, levando em conta estudos epidemiológicos prospectivos (SILVA; MURA, 2007, p. 386). A tabela 4 mostra os valores correspondentes a ingestão diária, dividindo quanto ao gênero e idade.

Sabendo-se as quantidade de cada macro nutriente que é recomendado para o paciente ingerir, agora é possível realizar o próximo passo, a montagem da dieta.

Passo 4: Montagem da Dieta

Para montagem da dieta são utilizados guias alimentares e normas de saúde, levando em conta as preferências, restrições e melhorias da dieta do paciente. (SILVA; MURA, 2007, p. 174) diz,

Tabela 4: Ingestão Adequada diária de fibras

| Categoria | Ingestão Adequada (g/dia) |
|-----------------|---------------------------|
| Homens | |
| 19 - 50 anos | 38 |
| 51 anos ou mais | 30 |
| Mulheres | |
| 19 - 50 anos | 25 |
| 51 anos ou mais | 21 |

No intuito de garantir uma alimentação equilibrada e saudável, a pirâmide foi a representação gráfica escolhida para dispor os alimentos separados em níveis e agrupados por tipos de nutrientes.

A pirâmide alimentar brasileira reflete os grupos alimentares que foram dispostos em ordem de quantidade de porções. A tabela 5 mostra a distribuição da quantidade de porções em relação ao calor calórico cálculo da dieta, o valor do EER.

Tabela 5: Valores em porções

| Grupos | Dieta 1600kcal | Dieta 2200kcal | Dieta 2800kcal |
|--------------------------------------|----------------|----------------|----------------|
| Arroz, massa, pão, batata e mandioca | 5 | 7 | 9 |
| Verduras e legumes | 4 | 4,5 | 5 |
| Frutas | 3 | 4 | 5 |
| Leite, queijo e iogurte | 3 | 3 | 3 |
| Carnes e ovos | 1 | 1,5 | 2 |
| Feijões | 1 | 1 | 1 |
| Óleos e gorduras | 1 | 2 | 2 |
| Açúcares e doces | 1 | 1,5 | 2 |

Os valores para cada porção leva em conta o tipo de alimento para ser quantificado.

Os nutricionistas utilizam software para montagem destas dietas, mas muitas vezes este não resolve de maneira satisfatória a forma como o nutricionista pensa que a dieta poderia ser gerada ou montada, abandonado assim estes software e indo para algo mais artesanal como planilhas.

2.2 Engenharia de Software

Esta seção contempla todas as referências utilizadas para confecção deste trabalho e desenvolvimento do algoritmo proposto para a sugestão de alimentos.

2.2.1 Requisitos de Software

Os requisitos são definidos como o que o sistema deve fazer e não como ele deva fazer (LUTOWSKI, 2005). E podem ser divididos em 2 grandes grupos, os funcionais e

não-funcionais.

Os requisitos funcionais são aqueles que descrevem as funcionalidades esperadas para o sistema forneça, dependendo somente "[...] do conhecimento passado pelos usuários sobre o negócio em si"(MACHADO, 2011). O autor ainda define que os requisitos não-funcionais não possuem uma ligação direta com as funcionalidade do sistema, e que estão ligados a padrões de qualidade como desempenho, manutenibilidade, robustez, entre outros.

Os requisitos são formalizados conforme o prosseguimento de alguma metodologia. Neste trabalho foi adotado o uso de metodologias ágeis, ou seja, os requisitos funcionais serão modelados como História de Usuários. (MACHADO, 2011) diz que as Histórias de Usuário são simples descrições feitas do ponto de vista do usuário, na qual representam uma funcionalidade na forma de uma expressão de intenção do que se deseja. São simples de serem entendidas sendo compreensíveis tanto para desenvolvedores quanto para usuários e são utilizadas como ferramenta para definição do comportamento de um sistema. Lembrando que é o trabalho do desenvolvedor falar a língua do usuário, não vice-versa.

Para os requisitos não-funcionais serão usados os QaSe (*Quality attributes Scenarios Essentials*), que segundo (BRITO, 2010, p. 76):

Quality Scenarios modelam os requisitos não-funcionais do sistema, contendo recomendações de uso, descrição do cenário abordado, interdependências, explicação de conflitos quando existirem, impactos causados no sistema quando um NFR não for devidamente atendido. A proposta fornece um tratamento de requisitos não-funcionais de forma leve, sem a grande quantidade de artefatos produzidos, com ênfase em atender as necessidades do stakeholders, sendo facilmente aplicável a projetos ágeis.

Assim como as Histórias de Usuários possuem um formulário padrão de preenchimento, vide seção ?? , os QaSe também o possuem, vide seção ??. E estes formulários serão utilizados como base de entendimento dos requisitos funcionais e não-funcionais.

2.2.2 Métricas

Métricas podem ser definidas como procedimentos de medições e escalas de medidas segundo (FILHO, 2013 apud ISO/IEC 9126-1,).

As métricas de software podem seguir as seguintes classificações segundo a iso (ISO/IEC 9126-1,):

- Métricas internas: Aplicáveis a um produto de software não executável;
- Métricas externas: Aplicáveis a um produto de software executável;
- Métricas de qualidade em uso: Aplicáveis a um produto de software que possa ser aferido as necessidades de usuários especificados;

A iso (ISO/IEC 9126-1,) e o autor (FILHO, 2013) ainda definem que cada métrica deve estar associado a uma escala, que tem como função expressar um significado a esta métrica, facilitando futuras comparações. As escalas podem ser divididas em:

- Nominal: Não existe ordem ou magnitude entre os valores. Servindo somente como separação de categorias;
- Ordinal: Existe uma comparação entre os valores mas não sendo claro quanto a magnitude entre eles;
- Intervalar: Uma ordem é existente e os tamanhos entre os intervalos dos valores são consistentes, porém proporção não são validas e não existe um zero absoluto ou unidade natural;
- Racional: "Possui ordem, magnitude, zero absoluto e unidade natural"(FILHO, 2013, p. 21);

A utilização de métricas é algum comum quando é necessário obter comparação de resultados. Do ponto de vista da tecnologia móvel, métricas relacionadas a tempo e consumo de memória ou bateria é algo completamente válido, uma vez que dispositivos móveis possuem uma limitação de hardware.

2.2.3 Algoritmos Gulosos

A estratégia "gulosa" consiste em fazer escolhas sequenciais, baseadas em escolhas individuais de acordo com o melhor menor critério que não seja tão caro para se avaliar (BAASE; GELDER, 2000). Ou seja, tentar desenvolver algoritmos utilizando-se de pequenos-passos⁹.

Uma vez que está escolha é feita, ela não poderá ser desfeita até que uma nova prova apareça, evidenciando assim, que a escolha anterior feita não foi uma boa escolha (BAASE; GELDER, 2000). Por este motivo, a estratégia "gulosa" não é a mais indicada para encontrar soluções ótimas (BAASE; GELDER, 2000).

Porém, o autor (KLEINBER; TARDOS, 2005, p.115) se refere ao sucesso de se encontrar a solução ótima através de algoritmos gulosos, dizendo que "Quando um algoritmo guloso consegue sucesso ao resolver um problema não-trivial, de forma ótima, isto implica em algo interessante e extremamente útil acerca da estrutura do problema, que esta regra de decisão local pode ser usada para construir o critério ótimo em algumas soluções."

⁹ *baby-steps*

2.2.4 Programação Dinâmica

Segundo o autor (TOSCANE; VELOSO, 2012), programação dinâmica pode ser definido como:

A ideia básica da programação dinâmica é construir por etapas uma resposta ótima combinando repostas já obtidas para partes menores. Inicialmente, a entrada é decomposta em partes mínimas, para as quais são obtidas respostas. Em cada passo, sub-resultados são combinados dando respostas para partes maiores, até que se obtenha uma resposta para o problema original. A decomposição é feita uma única vez e, além disso, os casos menores são tratados antes dos maiores.

Esses resultados obtidos são armazenados em uma tabela. A vantagem desta estratégia é o fato do problema ser resolvido e o seu resultado estar armazenado, assim caso algum sub-problema apareça o resultado para ele já está calculado (ZIVIANI, 2011).

KnapSack

O algoritmo é baseado na ideia de se ter uma mochila de capacidade W na qual os itens selecionados para entrar nessa mochila, que possuem um peso w e um valor v , devem ser selecionados de tal forma que o maior valor em itens seja levado e não ultrapasse o valor máximo do peso da mochila. Exemplo de itens com valores e pesos na tabela 6.

Tabela 6: Tabela de itens

Fonte: (DASGUPTA; PAPADIMITROU; VAZIRANI, 2009)

| Item | Peso(w) | Valor(v) |
|------|-------------|--------------|
| 1 | 6 | 30 |
| 2 | 3 | 14 |
| 3 | 4 | 16 |
| 4 | 2 | 9 |

O algoritmo *KnapSack*, também conhecido como "algoritmo da mochila" possui 2 tipos: com repetição e sem. A repetição é relacionada a como os itens serão analisados conforme o seu problema. Caso os itens possuam a restrição de ser usados somente uma vez, então a opção sem repetição é a mais viável. O ponto de vista da área de nutrição para este trabalho e a estratégia adotada para a solução do problema foram formulados para a utilização do modo *KnapSack* sem repetição. Isto será mais abordado no capítulo 5, relacionado a proposta.

O modo sem repetição consiste em preenchimento de uma tabela bidimensional com $W + 1$ linhas e $n + 1$ colunas, em que n é a quantidade de itens. Na qual um sub-problema pode ser expresso por $K(w, j)$, que significa: máximo valor alcançável usando uma mochila de capacidade w e itens $1, \dots, j$ (DASGUPTA; PAPADIMITROU; VAZIRANI, 2009). A fórmula seguinte mostra como a tabela deve ser preenchida.

$$K(w, j) = \begin{cases} 0 & \text{se } j = 0; \\ K(w, j - 1) & \text{se } w_j > W; \\ \max\{K(w - w_j, j - 1), K(w, j - 1)\} & \text{caso contrário;} \end{cases}$$

Em que a solução do problema é dada por $K(W, n)$.

2.2.5 Complexidade

A complexidade de um algoritmo reflete o esforço computacional requerido para executá-lo. Esse esforço computacional mede a quantidade de trabalho, em termos de tempo de execução ou da quantidade de memória requerida. (TOSCANE; VELOSO, 2012)

Este esforço computacional pode ser aferida através de medições empíricas, ou seja, medidas experimentais que podem ser usadas como parâmetros para futuras comparações. Porém existe uma grande dependência tanto do programa quanto da máquina, uma vez que cada máquina possui suas características e modo de operar. (TOSCANE; VELOSO, 2012)

Uma boa forma de se fazer estas aferências de esforço computacional é a utilização de análise matemática, que independe da implementação do algoritmo e pode ser aferida na fase de projeto de algoritmo. (TOSCANE; VELOSO, 2012)

Segundo (TOSCANE; VELOSO, 2012), o desempenho de um algoritmo pode ser dado através do esforço computacional de sua execução quando relacionamos isso a sua entrada de dados. O autor também alerta que, a complexidade de um algoritmo tem como principal diretriz a aferência do esforço computacional sobre um conjunto de entradas de dados a serem analisados e processados. Podem ser feitas 3 tipos diferentes de análise para um grupo de entradas: a pessimistas, a média e a otimista.

A complexidade vista do ponto pessimistas afere o pior valor, ou seja, o máximo esforço computacional que um algoritmo pode fazer. A complexidade média é uma média dos esforços no qual é levado em conta a probabilidade de ocorrência de cada entrada, ou seja, é um valor esperado para aferência. A complexidade otimista é o oposto do ponto de vista pessimista, ou seja, temos o melhor valor aferido sobre o algoritmo, no qual um esforço mínimo é feito. (TOSCANE; VELOSO, 2012)

São utilizadas estas análises para auxiliar nas comparações entre as funções de complexidade dos algoritmos. É recomendado utilizar da eficiência e da notação assintótica para se fazer análises mais corretas e seguras. Segundo (CORMEN et al., 2007), a eficiência assintótica pode ser observada como sendo o tempo de execução de um algoritmo mediante o crescimento do tamanho do grupo de entrada, e a notação assintótica

pode ser definida através do uso das notações $\Theta(g(n))$ ¹⁰, $O(g(n))$ ¹¹ e $\Omega(g(n))$ ¹².

Toda esta parte de análise, envolvendo notações assintóticas, pode ser chamada de análise assintóticas e segundo (FEOFILOF, 2015) a análise assintótica pode ser definida como:

Ao ver uma expressão como $n+10$ ou n^2+1 , a maioria das pessoas pensa automaticamente em valores pequenos de n . A análise de algoritmos faz exatamente o contrário: ignora os valores pequenos e concentra-se nos valores enormes de n . Para valores enormes de n , as funções

$$n^2, \frac{3}{2} \cdot n^2, 9999n^2, \frac{n^2}{1000}, n^2 + 100 \cdot n, \text{ etc.}$$

crescem todas com a mesma velocidade e portanto são todas equivalentes. Esse tipo de matemática, interessado somente em valores enormes de n , é chamado assintótico. Nessa matemática, as funções são classificadas em ordens (como as ordens religiosas da Idade Média); todas as funções de uma mesma ordem são equivalentes. As cinco funções acima, por exemplo, pertencem à mesma ordem.

Segundo (CORMEN et al., 2007) as notações citadas podem ser comparadas a:

$$\begin{aligned} f(n) &= O(g(n)) \approx a \leq b \\ f(n) &= \Omega(g(n)) \approx a \geq b \\ f(n) &= \Theta(g(n)) \approx a = b \end{aligned} \tag{2.4}$$

As métricas utilizadas para a aferência da complexidade de algoritmos são relacionadas a tempo de execução e quantidade de espaço em memória, que são comumente conhecidas como complexidade em tempo e complexidade em espaço.

2.2.6 Scrum

Segundo (SCHWABWER; SUTHERLAND, 2013), o Scrum é um *framework* no qual pessoas tem o poder de tratar e resolver problemas complexos e que podem se adaptar com o tempo, enquanto produtos são entregues com o mais alto valor possível, no qual os times são associados a papéis, eventos, artefatos e regras.

O Scrum é orientado a eventos *time-boxed*, ou seja, possuem uma duração fixa com tempo de início e fim. O primeiro evento que ocorre é o de planejamento da Sprint. Uma Sprint possui a duração de até 1 mês, não devendo ser reduzido ou aumentado o tempo assim que estabelecida a duração. Devem ser iniciadas uma imediatamente após o termino de outra.

¹⁰ Lê-se Big Theta de g de n

¹¹ Lê-se Big Oh de g de n

¹² Lê-se Big Omega de g de n

A metodologia possui caráter incremental, permitindo que uma versão inicial do trabalho seja feita e evoluída passo a passo- permite que uma versão inicial do trabalho seja feita e incrementada futuramente.”Sprints permitem previsibilidade que garante a inspeção e adaptação do progresso em direção à meta pelo menos a cada mês corrido. Sprints também limitam o risco ao custo de um mês corrido” (SCHWABWER; SUTHERLAND, 2013, p. 8).

2.3 Considerações Finais

Este capítulo teve o intuito de inserir o leitor ao domínio conexo de Nutrição e Engenharia de Software.

O domínio conexo de Nutrição traz noções básicas quanto a diferenciação entre macro e micro nutrientes, o processo e o passo-a-passo do estudo nutricional do paciente e como uma dieta pode ser fornecida com base no estudo nutricional do paciente.

O domínio conexo de Engenharia de Software traz noções quanto a diferença entre requisitos funcionais e não-funcionais e como ambos serão abordados neste trabalho. Também são fornecidos conhecimentos a cerca de métricas de software para comparação de resultados quanto a medição de tempo e consumo de memória. Outros pontos que foram ditos são relacionados a programação dinâmica, o qual podemos citar o *KnapSack*, e o conhecimento sobre complexidade e a utilização da mesma com métricas.

3 Suporte Tecnológico

Este capítulo aborda quais são as soluções tecnológicas utilizadas atualmente como suporte dos nutricionistas e quais ferramentas estarão sendo utilizadas como suporte técnico do autor deste Trabalho de Conclusão de Curso em relação ao domínio de Engenharia de Software.

A Seção 3.1 refere-se aos aplicativos utilizados atualmente para auxiliar os seus usuários em relação a sua alimentação. E graças a esta simbiose, entre paciente e aplicativo, os nutricionistas podem ser auxiliados a fazer um estudo mais aprofundado para corroborar com a saúde de seus pacientes.

A Seção 3.2 refere-se as ferramentas que serão utilizadas pelo o autor deste Trabalho de Conclusão de Curso, como forma de auxílio a mostrar o processo da metodologia, desenvolvimento do código-fonte e versionamento de código-fonte.

3.1 Aplicativo de Nutrição

Atualmente o maior aliado para a busca de uma dieta equilibrada e saudável é o uso de tecnologia. A equipe OnFit(ANDRADE; BERNARDES; COTRIM, 2015), antes de iniciar o projeto em si, fez uma grande busca de quais os aplicativos que atualmente ajudam pacientes, com ou sem, a presença de nutricionistas. Estes foram alguns dos aplicativos analisados:

- Dieta e Saúde¹;
- Lose it²;
- My Diet Diary³;
- My Fitness Pal⁴;
- TecnoNutri⁵;

Destes aplicativos somente os aplicativos Dieta e Saúde e TecnoNutri são voltados para a nutrição dos brasileiros.

¹ Link: <https://itunes.apple.com/br/app/dieta-e-saude/id400488936?mt=8>

² Link: <https://itunes.apple.com/us/app/lose-it!-weight-loss-program/id297368629?mt=8>

³ Link: <https://itunes.apple.com/br/app/my-diet-diary-calorie-counter/id414169919?mt=8>

⁴ Link: <https://itunes.apple.com/br/app/contador-calorias-e-monitor/id341232718?mt=8>

⁵ Link: <https://itunes.apple.com/br/app/tecnonutri-gestao-da-sua-dieta/id574794938?mt=8>

O objetivo desta pesquisa, para a equipe OnFit, é o de verificar qual o primor da arte para aplicativos de nutrição. Já para este Trabalho de Conclusão de Curso, os aplicativos Dieta e Saúde e TecnoNutri são os que mais se aproximam de um aplicativo completo que se faça a recomendação de quais alimentos e/ou porções de grupo alimentar devem ser ingeridos pelo usuário.

3.1.1 Dieta e Saúde

O aplicativo Dieta e Saúde é um dos mais completos atualmente. Se pensarmos na questão de todo o processo de geração de dieta, referenciado no capítulo anterior, o aplicativo consegue implementar boa parte do processo.

O primeiro ponto é o que atualmente o paciente, neste caso usuário de aplicativo, está ingerindo habitualmente seus alimentos, sendo possível fazer um recordatório dos dias anteriores.

O segundo ponto é a modularização das refeições em relação ao dia. Esta separação possibilita um maior controle sobre a alimentação do paciente, sendo possível adicionar alimentos ou gerar a dieta para as refeições do dia. O paciente pode aceitar ou não esta cardápio que pode ser gerado diariamente, sendo ele validado por uma equipe de nutricionistas associados ao aplicativo.

Os alimentos que podem ser inseridos em cada refeição pelo usuário são condicionados a utilização de pontos. Estes pontos são calculados apartir da ingestão dos alimentos escolhidos para aquela refeição e dos exercícios físicos cadastrados pelos paciente.

Quando o aplicativo é iniciado é perguntado qual é a altura, peso atual e peso futuro que o paciente deseja, e com base nisso uma quantidade máxima de pontos é calculada para o uso diário. Caso um alimento seja adicionado, pontos são retirados, caso algum exercício adicionado, pontos são adicionados. Fazendo assim um balanceio entre ingerir alimentos e exercitar-se.

Vale salientar que o aplicativo em si funciona somente com internet, ou seja, a geração de dieta, recordatório alimentar e validação de dieta com nutricionistas somente irão funcionar caso o dispositivo esteja conectado em rede.

3.1.2 TecnoNutri

O aplicativo se diz ser um gestor pessoal de alimentos, no qual estimula os usuários a procurar ajuda profissional. Sendo capaz de fazer o recordatório alimentar, adição das medidas antropométricas de peso atual, peso desejável e altura, e geração de dietas.

O aplicativo não faz a geração de dietas, porém auxilia o paciente a montar a sua própria dieta, indicando qual a quantidade de porções a serem usadas. Estas porções

são baseadas nas *DRI's* da organização mundial da saúde e no Guia alimentar para a população brasileira do Ministério da Saúde⁶.

O aplicativo mostra, após feito o aceite que está ciente sobre o risco de se ter uma dieta não ideal, a pirâmide alimentar brasileira, vide figura 3, em conjunto com as quantidades de cada porção referente a cada grupo de alimentos, sendo estas porções distribuídas entre todas as refeições diárias, como mostrada na figura 4.

Diferente do Dieta e Saúde, este aplicativo não possui a integração ou comunicação com nutricionistas devendo o paciente, no caso o usuário, manter esta comunicação por conta própria.

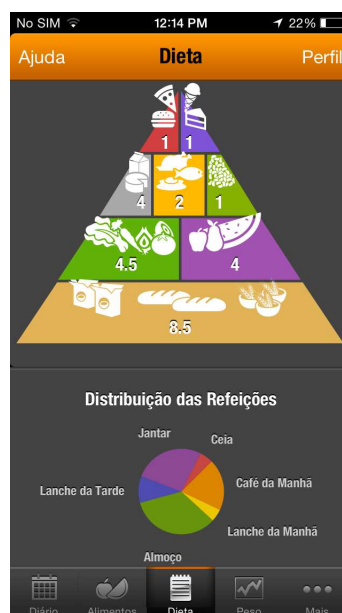


Figura 3: Pirâmide alimentar brasileira com porções alocadas pelo TecnoNutri

Fonte: (FREITAS, 2014)

3.2 Ferramentas de Engenharia de software

Nesta seção são descritas as ferramentas utilizadas como suporte para desenvolvimento do algoritmo proposto na seção de objetivos do capítulo Introdução.

O algoritmo proposto está sendo desenvolvido para uso em plataforma iOS, ou seja, iPhone 5C, 5S, 6 e 6 Plus. Para o desenvolvimento nesta plataforma é necessário o uso de ferramentas como o Xcode, Git, Draw.io.

⁶ Estas informações podem ser encontradas no Termo de Uso do aplicativo

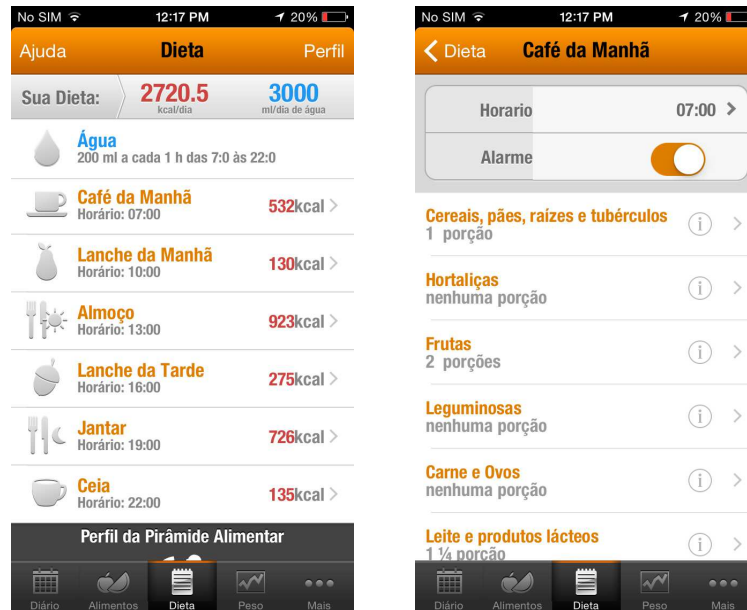


Figura 4: Distribuição das porções em relação as refeições

Fonte: (FREITAS, 2014)

3.2.1 Xcode

O Xcode é uma IDE poderosa que possui um ótimo ferramental para criação de interfaces gráficas, desenvolvimento de código-fonte, testes e debug de código-fonte. O debug de código-fonte através desta IDE tem um comportamento semelhantes a outras IDEs como o *NetBeans* e o *Eclipse*, pois ela permite ao desenvolvedor verificar em tempo de execução do programa os valores armazenados em variáveis e retornos de métodos e funções.

Em sua atual versão, esta IDE permite ao desenvolvedor escolher a linguagem e o paradigma usados para desenvolvimento. As linguagens suportadas são C, C++, Objective-C e Swift. As 3 últimas linguagens permitem o uso de programação multiparadigma - Estruturada ou Orientada a Objetos (OO).

O Xcode é uma IDE que possui a opção *Profile* que permite a análise do código fonte quanto a:

- Vazamento de memória⁷: Permite saber se algum elemento, previamente alocado, não foi desalocado quando não mais necessário;
- Gasto de energia da bateria⁸: Captura a informação sobre o use de energia no dispositivo;

⁷ No original Leaks

⁸ No original Energy Usage Instruments

- Mapeamento de Alocações de memória⁹: Faz o rastreamento de somente um processo quanto a alocação de elementos em memória;
- Perfil de tempo¹⁰: Permite saber qual o tempo de resposta de métodos e funções;

Estas são somente algumas das opções que o Xcode possibilita fazer a análise. Todos as opções de análise do Xcode mostram, de forma gráfica, o resultado, facilitando o entendimento para o usuário.

O Xcode também possibilita o uso de um banco de dados próprio, o *Core Data*. O Guia de Programação do *Core Data*, encontrado no Guia de Desenvolvimento para iOS da Apple([APPLE](#),), diz que este banco de dados permite total integração com a IDE quanto as ferramentas do *Profile*, automatiza a busca e persistência de dados, possui consultas de busca mais complexas, permite a criação, manutenção e deleção de relacionamentos de forma gráfica, e possui prevenção a carregamento de dados já carregados em memória¹¹ para evitar o desperdício de processamento.

3.2.2 Git

O Git será utilizado como versionador de código-fonte, para o desenvolvimento do software e para os arquivos de escrita deste Trabalho de Conclusão de Curso. Segundo ([CHACON; STRAUB, 2016](#), p. 27), um sistema de versionamento é : ” [...] *is a system that records changes to a file or set of files over time so that you can recall specific versions later*”.

Para se fazer o versionamento são utilizados *commits*. Segundo ([CHACON; STRAUB, 2016](#)), cada *commit* é um *snapshot*¹² do arquivo a ser versionado, e neste são guardados informações como o nome e o e-mail do autor, a mensagem escrita e o ponteiro para o *commit* anterior.

Com o auxílio do versionamento, será utilizado a técnica de *Branching* para organização das funcionalidades para o desenvolvimento do software proposto no Capítulo 5. A definição desta técnica é definido pelo ([CHACON; STRAUB, 2016](#)) como: ”A utilização desta técnica significa fazer a divergência da linha principal de desenvolvimento e continuar a fazer o trabalho sem bagunçar a linha principal de desenvolvimento¹³”. A linha principal é chamada de *master*.

⁹ No original Allocations

¹⁰ No original Time Profiler

¹¹ No original memory overhead

¹² O autor ([CHACON; STRAUB, 2016](#), p. 32) define com uma metáfora: ”[...] *it basically takes a picture of what all your files look like at that moment and stores a reference to that snapshot*.”

¹³ No original:”*Branching means you diverge from the main line of development and continue to do work without messing with that main line*.”

Após feito o trabalho em uma *branch*, diferente da linha principal, esta *branch* pode ser mesclada a linha principal, voltando assim todo o trabalho feito para a linha principal. A Figura 5 demonstra o esquemático da *Branching*.

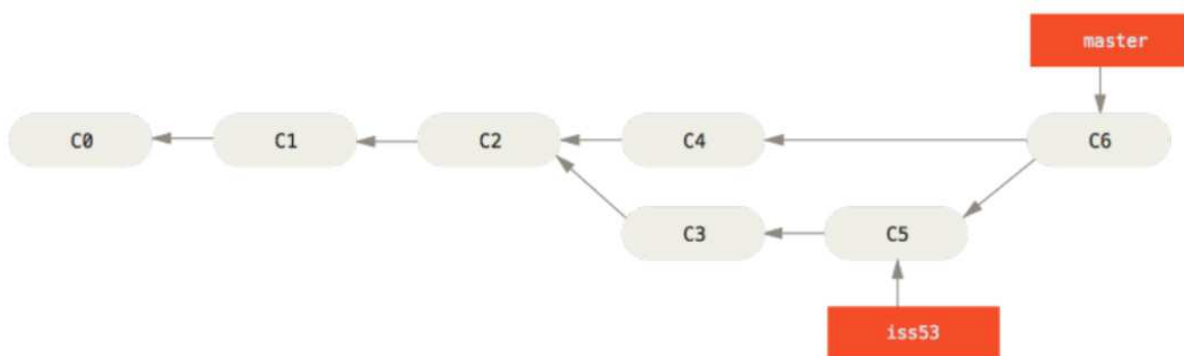


Figura 5: Fonte: (CHACON; STRAUB, 2016, p. 95) Figura 3-17

O uso do Git será feito para se guardar um histórico de evolução do software e dos arquivos de escrita deste Trabalho de Conclusão de Curso. Pois graças a isso, a manutenção e o entendimento de ambos pode ser feito de forma formal e facilitada.

3.2.3 Bonita

O Bonita será utilizado como forma de modelagem de processo através do *Business Process Model and Notation*(BPMN). A utilização desta ferramenta será para refletir a modelagem, que foi proposta, para a metodologia de: escrita, desenvolvimento e conclusão deste Trabalho de Conclusão de Curso.

3.2.4 Trello

Esta ferramenta será utilizada para organização das Histórias de Usuário. É um software de organização de *Boards online* que permite ao usuário alimentar estes como tarefas a serem feitas, desde tarefas simples do cotidiano (ex: lavar um carro ou fazer as compras de casa), a trabalhos mais complexos (ex: organização de um time em relação do que cada membro pode fazer). Este *Board* possui um rótulo, comumente usado para identificação do mesmo, e pode ser usado para gerar várias listas. As listas possuem somente um rótulo, comumente usado para identificação da mesma, sendo esta possível gerar vários *Cards* que podem ser observados na Figura 6. Por conseguinte, o *Card* possui um título, uma descrição, e vários *Checklists*. Cada *Checklist* possui um título e vários itens. Os itens são a menor unidade existente no Trello. A cada item marcado, a porcentagem do *Checklist* é atualizada, mostrando o desenvolvimento do mesmo.

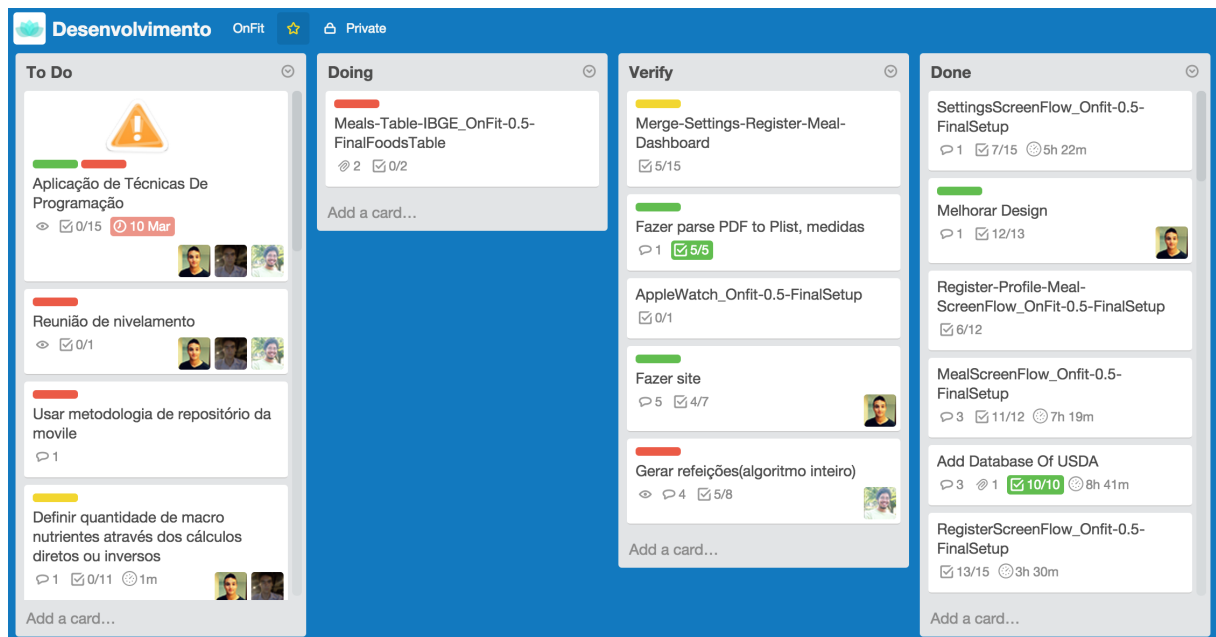


Figura 6: Trello usado pelos time de desenvolvimento OnFit

3.2.5 PunchTime

Uma outra ferramenta que será usada é o PunchTime, um software de medição do tempo para tarefas *online*. Este software possui um *plugin* que pode ser integrado ao Trello, adicionando aos *Cards* os componentes de:

- Estimativa de quando o *Card* estará pronto ou feito em horas;
- Tempo não contabilizado, caso esqueça de iniciar a medição de tempo; e
- Botão de iniciar relógio;

O funcionamento do *plugin* é simples. Ao se fazer a instalação do *plugin* no navegador, seguindo as instruções do *site*, as funcionalidades descritas anteriormente serão acessadas pelos *Card* do Trello. Para iniciar a medição de tempo da tarefa, basta clicar num *Card* e apertar o botão *Start*, então uma nova medição será iniciada e o registro será feito no PunchTime. Para finalizar a medição do tempo, o mesmo procedimento é feito, mas o botão a ser apertado será o *Stop*.

O tempo realizado por cada *Card* pode ser visualizado no próprio *site* do PunchTime.

3.3 Considerações Finais

Neste capítulo foi possível abordar sobre as tecnologias referente a área de nutrição e engenharia de software.

Na Seção 3.1 foi discutido sobre os melhores aplicativos referentes a área de nutrição, disponíveis atualmente, sendo que a minoria é voltada para os brasileiros. Destes aplicativos voltados para os brasileiros, foi dito quais seus pontos fortes e fracos em razão dos dados fornecidos para seus usuários.

Na Seção 3.2 foi discutido sobre as ferramentas que estão sendo utilizadas para auxiliar no desenvolvimento e suporte, tanto do software quanto para a escrita deste Trabalho de Conclusão de Curso.

4 Metodologia

Este capítulo tem como intuito mostrar as fases referentes a metodologia que será usada para condução de escrita deste Trabalho de Conclusão de Curso, desenvolvimento do algoritmo para sugestão de alimentos e análise dos resultados do algoritmo.

A primeira fase metodológica consiste em como este trabalho será desenvolvido, utilizando dos conceitos de Revisão Bibliográfica, Prova de Conceito e escrita em si deste trabalho, sendo mostrado na Seção 4.1. A segunda fase metodológica consiste no desenvolvimento do software utilizando-se uma adaptação do Scrum, sendo mostrado na Seção 4.2. A terceira fase metodológica consiste na análise dos dados obtidos e validação destes resultados com o profissional da área de nutrição. Uma vez que os dados são validados, uma nova rodada de ajuste de valores em equação é feita. Os detalhes desta análise estão na Seção 4.3.

Será utilizado como técnica metodológica Pesquisa-ação, que tem como objetivo dar ênfase à relação teoria e prática de toda a pesquisa (PINHEIRO, 2010). Esta técnica pode ser observada em torno de todo este capítulo, mais fortemente na parte do desenvolvimento do software e compreensão e conclusão dos dados analisados pelo profissional.

A Figura 7 mostra uma visão geral de como a metodologia é conduzida. As fases 2 e 3 possuem tarefas que são esclarecidas no decorrer de suas respectivas seções.

4.1 Metodologia para Desenvolvimento TCC

O presente trabalho pretende seguir uma metodologia cíclica, a qual será estruturada em fases. Cada ciclo será composto por três fases: Revisão Bibliográfica, Prova de conceito e Escrita do Trabalho de Conclusão de Curso (TCC). Esta metodologia será usada como um guia para condução da escrita do TCC.

4.1.1 Revisão Bibliográfica

A fase metodológica de Revisão Bibliográfica é a fase inicial da concepção do conhecimento. Esta fase é relacionada ao mapeamento do conhecimento e da síntese do conhecimento de trabalhos (CONFORTO; AMARAL; SILVA, 2011), já publicados, por outros pesquisadores em áreas afins deste Trabalho de Conclusão de Curso.

O mapeamento realizado na Revisão Bibliográfica serve de insumo para condução da próxima fase metodológica, no caso, Prova de Conceito.

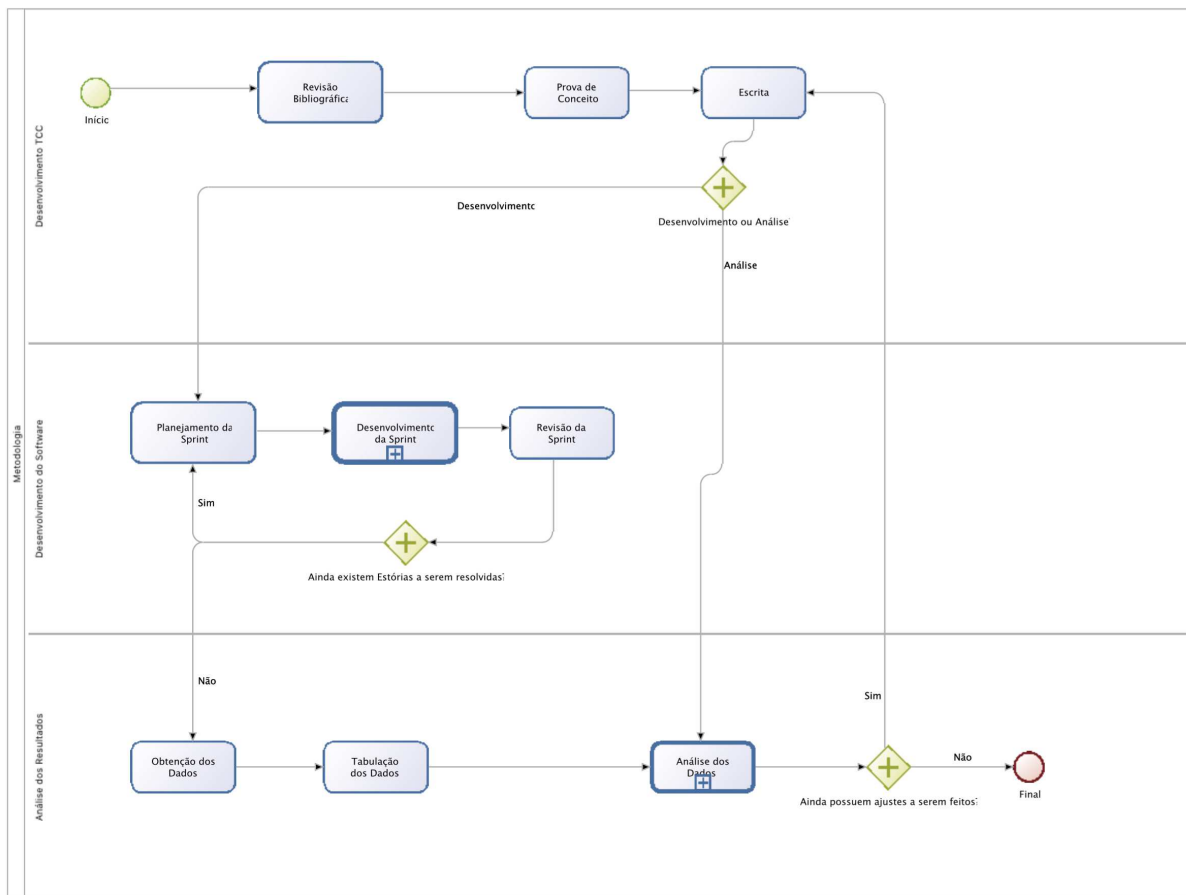


Figura 7: Visão geral da metodologia

4.1.2 Prova de Conceito

A segunda fase será a de prova de conceito, que consiste em “... denominar um modelo prático que possa provar o conceito (teórico) estabelecido por uma pesquisa ou artigo técnico”(SILVA, 2014). Uma prova de conceito permite a demonstração de uma ideia, fundamentada teoricamente, na prática. Ou seja, aplicar o que foi inicialmente idealizado na prática, e para isso é necessário uma formalização. A formalização da prova de conceito deste trabalho consiste em:

- Entendimento do escopo do problema analisado em cada ciclo;
- Análise de algoritmos que podem ser estendidos ao domínio conexo do problema;
- Implementação de algoritmos previamente analisados em contexto com o domínio conexo;
- Teste de algoritmo implementado como um entregável executável;
- Otimização de algoritmo;

O entendimento do escopo do problema é essencial, pois com base neste entendimento um estudo mais profundo pode ser feito e uma solução pode ser mais bem estudada.

Existem modelos de solução que podem ser implementados, resolvendo problemas previamente estudado. Estes modelos possuem classificações, como algoritmos: gulosos, dividir para conquistar e programação dinâmica. Um exemplo de algoritmo para programação dinâmica é o *Knapsack*, que permite calcular a melhor forma de se levar itens valiosos para uma viagem (KLEINBER; TARDOS, 2005).

O próximo passo é a implementação do algoritmo contextualizando o mesmo com o problema. Em seguida, alguns testes são feitos em relação ao todo. No momento de testes são analisados critérios como memória consumida e tempo de resposta do algoritmo em dispositivo móvel. Caso o algoritmo não apresente conformidade com os critérios, então uma otimização do mesmo é estudada e feita.

Esta fase possibilita uma via de retroalimentação com a fase de Revisão Bibliográfica, permitindo um maior refinamento da prova de conceito.

4.1.3 Escrita

A terceira fase será a de escrita do Trabalho de Conclusão de Curso(TCC), que consiste no:

- Apresentação do referencial teórico tanto para o domínio conexo da Nutrição quanto o de Engenharia de Software;
- Escrita formalizada das provas de conceito mediante análise de profissional da área de Nutrição;
- Escrita da síntese dos resultados das prova de conceito.

4.2 Metodologia para Desenvolvimento do Software

A metodologia utilizada para conduzir as atividades de desenvolvimento desse Trabalho de Conclusão de Curso será uma adaptação da metodologia ágil iterativa incremental embasada no Scrum.

Este *framework* consiste em resolver problemas complexos, no qual o empirismo é a base de controle de processo. “O empirismo afirma que o conhecimento vem da experiência e de tomada de decisões baseadas no que é conhecido.”(SCHWABWER; SUTHERLAND, 2013). Basicamente, são três os pilares que sustentam este processo empírico:

- Transparência: São aspectos do processo que ficam visíveis aos envolvidos com os resultados, ou seja, um padrão comum deve ser adotado pelos envolvidos para que

uma visão comum seja compartilhada. Alguns exemplos são: utilização de uma linguagem mais comum, definição comum de “pronto” é compartilhada por aqueles que realizam o trabalho e por aqueles que aceitam o resultado do mesmo;

- Inspeção: Devem ser feitas, preferencialmente, por pessoal especializado, conferindo maior benefício. As inspeções devem ocorrer frequentemente, mas não ao ponto de atrapalhar a própria execução das tarefas;
- Adaptação: Caso o produto ou processo inspecionado não esteja dentro dos limites aceitáveis, os mesmos devem ser ajustados assim que possível, minimizando o impacto do retrabalho. “O Scrum prescreve quatro Eventos formais, contidos dentro dos limites da Sprint, para inspeção e adaptação...” (SCHWABWER; SUTHERLAND, 2013).
 - Reunião de planejamento da Sprint;
 - Reunião diária;
 - Reunião de revisão da Sprint;
 - Retrospectiva da Sprint.

A reunião de planejamento de Sprint contempla todo o trabalho a ser planejado naquela Sprint. Esta reunião responde a algumas perguntas:

1. O que pode ser entregue como resultado do incremento da próxima Sprint?
2. Como o trabalho necessário para entregar o incremento será realizado?

O objetivo deste planejamento é obter uma meta para a Sprint, dando ao time de desenvolvimento o motivo de estar sendo construído este incremento. O incremento é a meta a ser satisfeita através da implementação do Backlog do Produto do Backlog da Sprint.

O Backlog do Produto dá ao time de desenvolvimento uma estimativa de qual é o tamanho do produto a ser desenvolvido. Já o Backlog da Sprint dá ao time de desenvolvimento um norte sobre o que deve ser feito naquela Sprint.

O segundo evento ocorrido é a Reunião Diária, que é basicamente uma reunião que ocorre em 15 minutos provida para o planejamento das atividades das próximas 24 horas. O processo de inspeção ocorre naturalmente neste evento, uma vez que os participantes respondem as perguntas de: o que fiz ontem, o que farei hoje e o que está me impedindo de fazer alguma futura para finalizar a sprint. Somente o time de desenvolvimento participa desta reunião. “Reuniões Diárias melhoram as comunicações, eliminam outras reuniões, identificam e removem impedimentos para o desenvolvimento, destacam e promovem rápidas tomadas de decisão, e melhoram o nível de conhecimento do time de desenvolvimento.

Esta é uma reunião chave para inspeção e adaptação.”(SCHWABWER; SUTHERLAND, 2013, p. 11)

O terceiro evento é a Revisão de Sprint, que sempre ocorre ao final de cada Sprint com o objetivo de inspecionar o que foi incrementado e adaptar tanto o Backlog do Produto quanto o Backlog da próxima Sprint caso precise. A reunião deve incluir todos os envolvidos no Scrum, sendo possível esclarecer: quais itens do Backlog da Sprint ficaram prontos e quais não ficaram, mostrar quais problemas aconteceram, estão acontecendo e como foram resolvidos, demonstração do incremento pronto. Todo o grupo colabora sobre o que fazer em seguida. As saídas geradas neste momento poderão ser usadas como insumos de entrada para a reunião de Planejamento da próxima Sprint.

O quarto e último evento é a Retrospectiva da Sprint, que é a oportunidade de melhoria do Time Scrum, adotando novos planos de melhoria a serem aplicados na próxima Sprint. Este evento ocorre depois da Revisão de Sprint e antes do planejamento da próxima Sprint, e o propósito desse é: saber como a última Sprint foi em relação às pessoas, relacionamentos, processos e ferramentas, identificar e ordenar potenciais melhorias de forma que todo o Time Scrum melhore enquanto trabalha. “A implementação destas melhorias na próxima Sprint é a forma de adaptação à inspeção que o Time Scrum faz a si próprio. A Retrospectiva da Sprint fornece um evento dedicado e focado na inspeção e adaptação, no entanto, as melhorias podem ser adotadas a qualquer momento.”(SCHWABWER; SUTHERLAND, 2013, p. 13).

4.2.1 Adaptação Adotada

A adaptação para o desenvolvimento do software ocorrerá da mesma forma que o processo do Scrum, vide Figura 8. As exceções são relatadas no decorrer desta seção.

Papéis

Os papéis adotados para este trabalho serão:

- O Product Owner (PO) será o grupo composto pelos desenvolvedores do aplicativo OnFit;
- O Scrum Master será o grupo composto pelos Orientador e Coorientador deste trabalho;
- O time de desenvolvimento será composto somente pelo autor deste trabalho.

Sprints

Pretende-se utilizar iterações curtas, i.e. Sprints de duas semanas (em média), facilitando a modularização do que deve ser feito. A escolha deste duração foi baseada

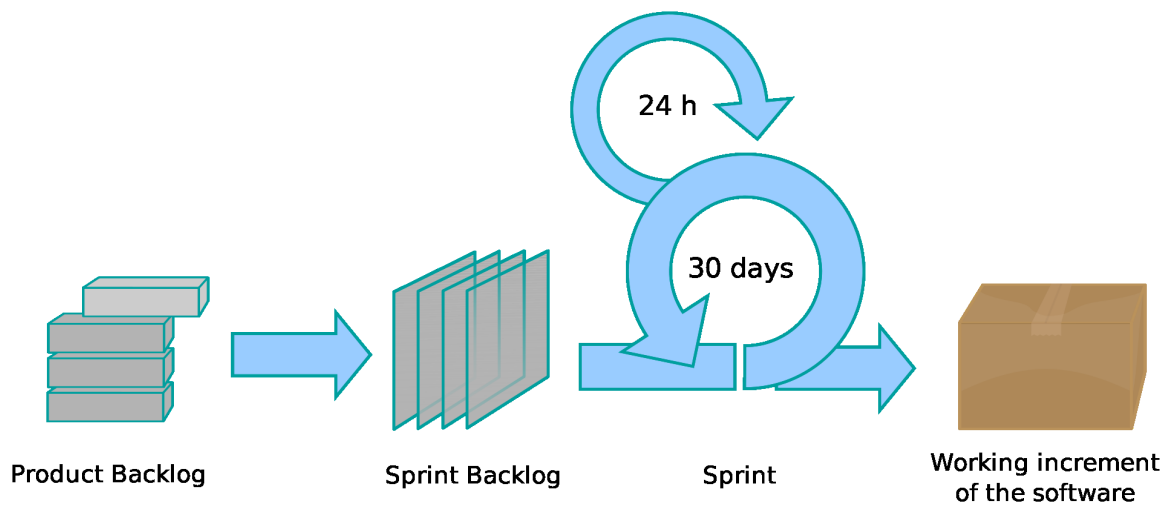


Figura 8: Metodologia Scrum segundo (SCHWABWER; SUTHERLAND, 2013)

em experiências do time de desenvolvimento em projetos passados, não sendo um valor escolhido arbitrariamente.

Reuniões

As reuniões possuem uma estreita relação no que diz respeito a comunicação do time como um todo. A adaptação destas em relação a este TCC pode ser observado em consonância com a seguinte lista.

- **Diárias:** As reuniões diárias irão ocorrer de forma a manter uma cadência de desenvolvimento do produto, respondendo as perguntas base sobre o que fiz ontem, o que farei hoje e o que é impeditivo para finalizar a Sprint em relação ao que estou fazendo;
- **Planejamento de Sprint:** Estas reuniões vão manter o padrão para o Scrum;
- **Revião de Sprint:** Estas reuniões vão manter o padrão para o Scrum;
- **Retrospectiva:** Estas serão adaptadas para serem feitas juntamente com o Scrum Master e o Product Owner.

4.2.2 Suporte Ferramental de Metodologia

Serão utilizadas as ferramentas Trello e PunchTime, vide Seção 3.2, para gerenciar as atividades do TCC, bem como manter o rastro das ações realizadas ao longo do trabalho.

O Trello será usado como um Kanban e o PunchTime para pontuação das Histórias de Usuário. Fazendo-se um paralelo temos:

1. Cada *Board* do Trello será um quadro Kanban. Caso seja necessário mais de um quadro é possível utilizar mais de um;
2. Cada Lista do Trello será uma coluna do quadro KanBam, sendo utilizado no mínimo 4 colunas: *To Do*, *In Progress*, *Verify* e *Done*. Caso seja necessário, será adicionado mais colunas. Estas listam podem ser vistas na Figura 9;

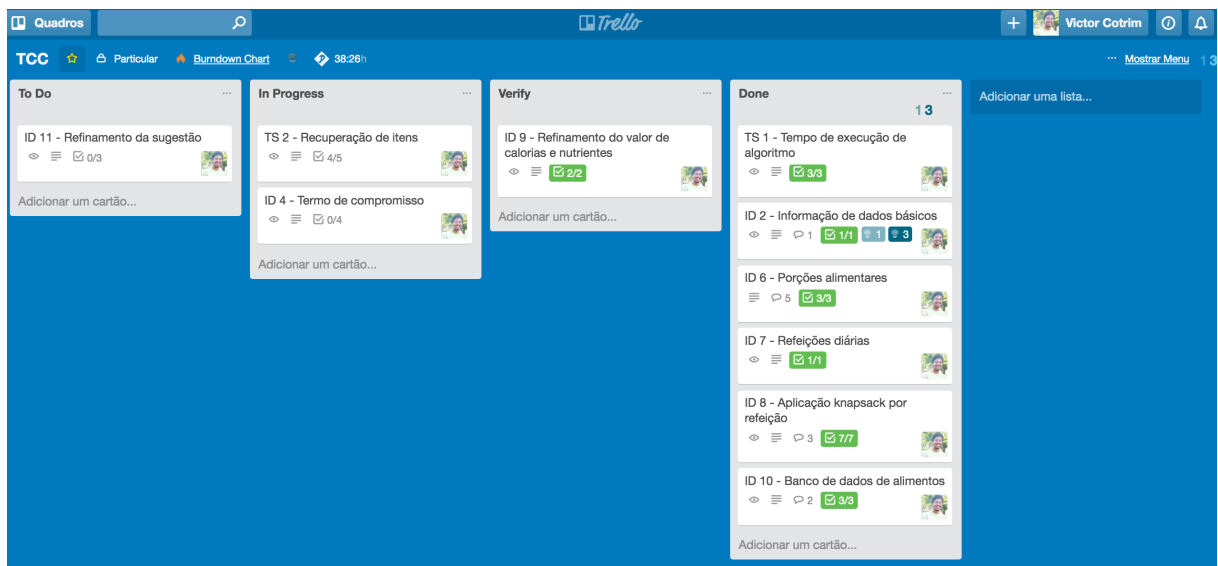


Figura 9: Trello sendo utilizado como Kanban

3. Cada *Card* do Trello será uma História de Usuário, no qual o título será a História, a descrição do *Card* será a descrição da História. Detalhes das Histórias de Usuário podem ser visto na Figura 10;
4. Cada *Checklists* do Trello será dado como tarefas a serem feitas relacionadas àquela História de Usuário.

Com a utilização do *plugin* do PunchTime, a pontuação referente a cada História de Usuário será alocada em horas. Uma conversão de pontos em horas será feita. Serão considerados ainda os critérios de dificuldade a serem completados pela tarefa e o tamanho da tarefa a ser feita. Todas as Histórias podem ser visualizadas em sua completude no apêndice B.

4.3 Metodologia de Análise dos Resultados

Esta seção refere-se a como os dados obtidos serão analisados e sintetizados, com o intuito de evidenciar os resultados obtidos. A análise dos resultados, assim como na

ID 2 - Informação de dados básicos
na lista [Done](#)

Membros **Punchtime** **0:14h**

Descrição [Editar](#)
Eu como desenvolvedor quero informar os dados do usuário para saber a quantidade de calorias, carboidratos, lipídios e proteínas que ele deve ingerir diariamente.
Dados: Gênero, Idade, Altura, Peso

[Histórico]
V 0: Eu como usuário quero que o aplicativo me sugira alimentos conforme o meu objetivo, para saber o que devo comer em cada refeição diária

☒ **Checklist** [Ocultar itens concluídos](#) [Excluir...](#)
100%
☒ ~~Fazer implementação da fórmula em G~~
Adicionar item...

Adicionar Comentário
 Escrever um comentário...

Enviar

Adicionar
 Membros
 Etiquetas
☒ **Checklist**
 Data Entrega
 Anexo

Punchtime
+ Add log
 View logs

Ações
 Mover
 Copiar
 Assinar
 Arquivar

Figura 10: Utilização de *Card* como História de Usuário

seção anterior, será realizado utilizando uma abordagem iterativa incremental, na qual os resultados de uma iteração será utilizado como entrada para a próxima iteração.

4.3.1 Obtenção e Tabulação dos Dados

Os dados a serem obtidos formarão o resultado do algoritmo proposto no Capítulo 5.

Uma lista com os alimentos sugeridos para cada refeição diária será feita. Cada alimento sugerido irá mostrar os valores de seus respectivos macro nutrientes, conforme Tabela 8.

A lista de alimentos sempre informará para qual situação a mesma foi gerada, levando em conta as informações do paciente, conforme Tabela 7.

Tabela 7: Exemplo de perfil de paciente para geração da lista de alimentos sugeridos

| Sexo | Idade | Altura (m) | Peso (kg) | Dias atividade física | Objetivo |
|-----------|-------|------------|-----------|-----------------------|-------------|
| Masculino | 25 | 1,72 | 85 | 1-2 dias/semana | Manter peso |

Tabela 8: Exemplo de lista sugerida de alimentos conforme perfil de paciente

| Refeição Diária | Alimento | Quantidade | Carboidrato (g) | Lipídios (g) | Proteínas (g) |
|-----------------|-----------------|---------------------|-----------------|--------------|---------------|
| Café da Manhã | Pão integral | 1 fatia | 10 | 0 | 3 |
| | Presunto | 1 fatia | 0 | 2 | 3 |
| | Queijo prato | 1 fatia | 0 | 3 | 3 |
| Colação | Maçã | 1 unidade | 12 | 0 | 0 |
| | Banana | 1 unidade | 17 | 0 | 0 |
| Almoço | Arroz | 2 colheres | 25 | 2 | 3 |
| | Feijão | 1 concha | 21 | 2 | 8 |
| | Carne moída | 2 colheres | 1 | 10 | 25 |
| | Tomate | 2 rodela | 1 | 0 | 0 |
| Lanche da tarde | Pão comum | 1 unidade | 29 | 1 | 4 |
| | Presunto | 1 fatia | 0 | 2 | 3 |
| | Queijo prato | 1 fatia | 0 | 3 | 3 |
| Jantar | Arroz | 2 colheres | 25 | 2 | 3 |
| | Feijoada | 2 conchas | 76 | 35 | 49 |
| | Laranja | 2 rodela | 4 | 0 | 0 |
| Ceia | Iogurte Natural | 1 copo de requeijão | 11 | 7 | 8 |

4.3.2 Estratégia da Análise de Dados

Os dados obtidos serão submetidos para um profissional da área de nutrição. Este profissional irá analisar se a lista de alimentos gerada está conforme o objetivo do paciente. Caso não esteja conforme, os devidos ajustes serão feitos e uma nova submissão de lista será feita. Esta estratégia pode ser vista na Figura 11.

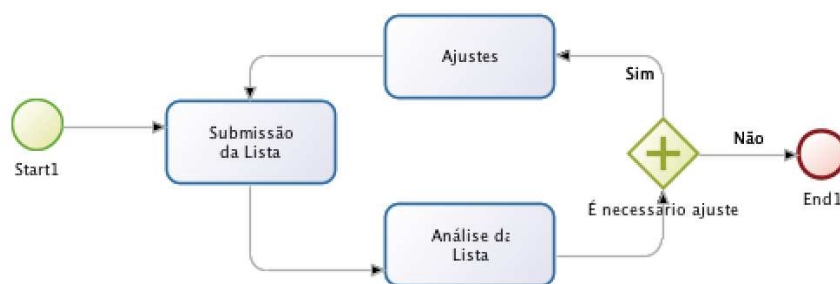


Figura 11: Estratégia para análise dos dados

4.3.3 Resultados da Análise de Dados

Os resultados referentes a Seção 4.3.2 serão discutidos, em razão dos ajustes a serem feitos tanto nas equações de pontuação geral e específica¹, quanto no algoritmo.

Cada ciclo de ajuste será documentado no Capítulo ??, contendo uma visão do estado atual e uma visão do que foi ajustado.

¹ Mais detalhes sobre as equações de pontuação geral no Capítulo 5

4.3.4 Conclusão da Análise de Dados

Em relação aos dados da sugestão de alimentos, mudanças podem ser observadas entre cada entrega feita para validação pelo profissional, uma vez que os dados antropométricos do usuário foram analisados², uma lista com os alimentos sugeridos irá surgir. Entre cada entrega os pontos de mudança serão relatados, ou seja, um rastreo entre o que será pedido e o que será feito existirá, deixando assim claro as mudanças decorrentes de cada entregue.

Em relação ao desenvolvimento do software, os dados podem ser observados conforme o tempo de execução e espaço em memória ocupado pelo algoritmo. Os dados de tempos de execução e espaço em memória sempre serão atualizados entre as entregas, pois a mudança a ser requisitada pelo profissional irá afetar os mesmos.

4.4 Considerações Finais

Este capítulo teve como intuito mostrar a metodologia para desenvolvimento deste Trabalho de Conclusão de Curso em relação a sua organização e desenvolvimento do software.

Foi mostrado que na primeira fase, foi feito um levantamento de todo o referencial teórico a ser utilizado e referenciado durante a fase de desenvolvimento de software e escrita do TCC.

A segunda fase, na qual o desenvolvimento do software ocorre, acontece de forma iterativa incremental utilizando-se de adaptação de métodos ágeis.

A terceira, e última fase, ocorre a validação da lista de alimentos sugeridos para o usuário. Esta lista será validada pelo profissional da área de nutrição, com o intuito de refinar os valores referentes às equações utilizadas pelo algoritmo, vide Capítulo 5. Este refinamento ocorrerá em ciclos de ajuste, no qual um ponto ótimo poderá ser alcançado. Todos a documentação feita, referente ao algoritmo e a como o mesmo é ajustado, será mantida no Capítulo ??, Estado Atual.

² Mais detalhes na Tabela 7 e no Capítulo 2

5 Algoritmo para Sugestão de Alimentos

Este capítulo inicia com uma explanação sobre os problemas relacionados a aplicações móveis no contexto de nutrição expondo uma solução para o principal problema encontrado: a sugestão de alimentos. A Seção 5.1 se refere a explanação dos problemas. A Seção 5.2 se refere a solução, sendo demonstrado seu processo passo a passo.

5.1 Problema

Foi identificado uma necessidade de melhoria dos aplicativos utilizados atualmente na área de nutrição(ANDRADE; BERNARDES; COTRIM, 2015). Um dos problemas encontrados foi em relação a interface dos aplicativos: a maior parte deles tinham muitas informações em somente uma tela. Ou seja, confundiam o usuário deixando-o sem saber o que fazer, e é sabido que aplicativos com excesso de informação, e com massiva quantidade de dados a serem inseridos pelo usuário, atrapalham tanto na experiência que o usuário irá ter quanto na usabilidade do mesmo. Esse problema de excesso de informações, para o aplicativo *OnFit*, foi resolvido depois de vários estudos e mudanças do design.

Outro ponto a ser enfatizado é quanto inserção de alimentos na dieta do usuário e a utilização da funcionalidade da sugestão de alimentos.

Para a inserção dos alimentos, pode ser dito que em alguns casos é feita de forma não facilitada, em que o usuário é obrigado a navegar em várias telas até fazer a inserção do alimento em sua refeição.

Para a sugestão de alimentos, não é mostrado ao usuário como ou quem está por trás desta sugestão. Ou seja, não se tem total transparência dos dados calculados¹. Este problema está sendo enfrentado pela equipe do OnFit, e depois de 2 tentativas sem sucesso, o autor deste TCC iniciou o estudo de algoritmos que possam solucionar este problema.

A partir destes problemas uma solução foi obtida.

5.2 Solução

A solução proposta para os problemas encontrados na seção anterior, é o de produzir um algoritmo que sugira os alimentos de forma coerente e rápida. Ou seja, coerente em razão de estar alinhado com o objetivo do usuário, e rápida em relação ao desempenho que o algoritmo tem ao analisar os alimentos que são sugeridos ao usuário.

¹ Vide Capítulo 3 caso deseje mais informações.

O algoritmo leva em conta a refeição que está sendo gerada, ou seja, a cada refeição existe uma quantidade de carboidratos, fibras, proteínas e lipídios a serem supridas. Ao se conseguir suprir as necessidades das refeições, consequentemente a necessidade diária é suprida. Mas é necessário ficar atento ao fato da não perfeição naturalmente inclusa no processo de alimentação: essa necessidade raramente será suprida com a quantidade necessária correta, sempre haverá um *gap*, positivo ou negativo, em relação a necessidade ideal (ONFIT, 2014).

Para entender o algoritmo, esta seção foi separada em 4 subseções:

- Preparação dos Dados: os dados antropométricos são utilizados para calcular os valores de GEB², FAF³, EER⁴ e quantidade de macro nutrientes em gramas;
- Formulação dos Itens Conforme Alimentos Escolhidos: os alimentos utilizados no algoritmo são formulados utilizando seus nomes e quantidade de porção;
- Cálculo do Valor dos Itens: os itens formulados na subseção anterior possuem seus valores calculados por equações pré-definidas;
- Uso e Análise do *Knapsack*: utilização do *knapsack* para a sugestão de alimentos, juntamente de uma análise em relação a sua complexidade e estratégia de implementação.

Para este trabalho vale salientar que esta solução foi produzida primeiramente utilizando a linguagem C em um computador com a intenção de se fazer o máximo de otimizações, medições e análises possíveis. Posteriormente a solução foi portada para a plataforma iOS, e a partir deste porte foi aproveitado todo o código feito em C para *Objective-C* com algumas adaptações. Uma destas adaptações é a utilizações de *Arrays* elásticos e genéricos com o intuito de adicionar os itens escolhidos pelo *knapsack* e recuperar os itens do banco de dados.

no qual o porte para iOS foi feito a partir de todo este estudo.

5.2.1 Preparação dos Dados

Os dados de entrada utilizados para se iniciar o algoritmo são os dados antropométricos: sexo, altura, peso atual e idade. Com base nestes dados é calculado o nível de GEB. A equação 2.1 é utilizada para homens e a 2.2 para mulheres, para saber o quanto o corpo gasta de energia diariamente.

² Gasto Energético Basal

³ Fator de Atividade Física

⁴ Estimativa de Energia Recomendada

O próximo passo é o cálculo do FAF, que pode ser aferido na Tabela 1. O penúltimo passo é o cálculo da EER através da equação 2.3. E o último passo, para conclusão do levantamento dos dados do paciente a serem utilizados no algoritmo, é o alinhamento do EER com o seu objetivo, que pode ser aferido na Tabela 2.

O intuito do cálculo de alinhamento da EER é saber o quanto de macro nutrientes será necessário para o usuário ingerir em cada refeição diária. Os valores de alinhamento de EER, encontrados na Tabela 2, foram averiguados mediante dados levantados com profissional da área (ONFIT, 2015, Entrevista nutricionista Silvia). Estes valores previamente definidos podem variar conforme Tabela 3.

Tabela 9: Total de macro nutrientes a serem consumidos em relação ao EER

| Objetivo | Carboidrato (%) | Lipídeo (%) | Proteína (%) | Fibras (%) |
|----------------|-----------------|-------------|--------------|------------|
| Perder Peso | 45 | 20 | 35 | 1,4 |
| Ganhar Músculo | 45 | 20 | 35 | 1,4 |
| Manter Peso | 45 | 20 | 35 | 1,4 |

O nível de EER calculado é fracionado conforme a Tabela 9 e estes valores são utilizados como referência para o cálculo da quantidade de carboidratos, lipídeos e proteínas.

As refeições possuem quantidade diferentes de calorias a serem supridas. A Tabela 10 mostra as recomendações diárias de cada refeição, em porcentagem.

Tabela 10: Fracionamento das gramas em cada refeições

| Refeição | Faixa de valor aceitável (%) |
|----------|------------------------------|
| Desjejum | 15 |
| Colação | 5 a 10 |
| Almoço | 30 |
| Lanche | 15 |
| Jantar | 25 |
| Ceia | 5 a 10 |

Mediante troca de informações com nutricionista, foi informado que para se calcular as quantidades de macro nutrientes as equações 5.1, 5.2 e 5.3 são utilizadas. A equivalência energética, tanto do carboidrato quanto da proteína, é dita como 4 calorias é igual a 1 grama, para os lipídios é dita como 9 calorias equivalem a 1 grama.

$$Carboidrato(g) = \frac{EER_{macronutriente}}{4} \quad (5.1)$$

$$Proteína(g) = \frac{EER_{macronutriente}}{4} \quad (5.2)$$

$$Lipídios(g) = \frac{EER_{macronutriente}}{9} \quad (5.3)$$

Um exemplo prático disso pode ser feito. Imagine que após todos os cálculos baseados nos dados antropométricos, uma pessoa tenha seu EER sendo igual a 2000 calorias e o seu objetivo é manter o peso. Desse total de calorias para o carboidrato, 45% será utilizado, ou seja, 900 calorias. Fazendo-se a conversão temos 225 gramas de carboidrato a serem consumidas diariamente. Esses carboidratos diários são separados conforme a refeição, ou seja, para o almoço esta pessoa terá que ingerir 67,5 gramas. Esse processo é aplicado aos outros macronutrientes também.

5.2.2 Formulação dos Itens Conforme Alimentos Escolhidos

Uma solução encontrada para análise dos alimentos para se fazer a sugestão dos mesmos é a utilização de modelos, como programação dinâmica. O modelo que melhor se enquadra à situação observada para este Trabalho de Conclusão de Curso é o modelo *Knapsack*, ou comumente conhecido como problema da mochila⁵.

Os itens utilizados no *Knapsack* serão os alimentos provenientes dos grupos alimentares escolhidos para cada refeição, de tal forma que será utilizada uma combinação entre quantidade do alimento e o alimento propriamente dito, tendo como formato n itens para somente um alimento, ou seja, para cada alimento será combinado com n quantidades de porções que o alimento pode possuir. Isto é exemplificado na Tabela 11. Os itens são provenientes do banco de dados do aplicativo OnFit.

Tabela 11: Exemplo da formação dos itens para o *knapsack* conforme alimentos

| Alimento | Porção | Item |
|----------|---------------------|------------------------------|
| Arroz | 1 colher de arroz | [Arroz, 1 colher de arroz] |
| Arroz | 2 colheres de arroz | [Arroz, 2 colheres de arroz] |

Não foi utilizado porções infinitas para a combinação dos vários itens. Inicialmente não foi usado mais do que 3 porções para qualquer alimento. Estes dados são empíricos e refinados conforme ciclo de ajustes com a nutricionista⁶.

5.2.3 Cálculo do Valor dos Itens

A intenção de se fazer a formulação dos itens é facilitar o cálculo de uma pontuação global para cada item a ser usado pelo *knapsack*. Foi utilizado a quantidade de gramas de cada macronutriente encontrado no item para o cálculo de uma pontuação.

⁵ Para mais informações, vide 2.2.4.

⁶ Vide Seção 4.3 caso deseje mais informações.

Algumas características que podemos notar para que alguma equação $f(x)$ resulte em uma pontuação são:

- Não existem valores negativos de entrada, pois não existem itens com valores de macronutrientes negativos. Isso nos revela que a equação deva possuir um x positivo;
- Existe um valor limite de ingestão para cada macronutriente previamente calculado, então o cenário ideal é do valor calculado não receber a maior pontuação quando ultrapassado este valor limite. Isso nos revela que a equação deva possuir um valor máximo para seu $f(x)$.

A estratégia para calcular a pontuação dos itens foi utilizar algumas equações de diferente tipos, porém ainda mantendo as características observadas para se ter uma pontuação, aplicando assim a equação para cada macronutriente e somando os resultados na pontuação global do item. O propósito é verificar qual a equação que melhor informe os itens mais valiosos para o algoritmo, tendo em vista o objetivo do usuário.

Algumas possíveis equações que conseguem se ater a esse perfil são: aquelas que naturalmente possuem uma concavidade virada para baixo ou aquelas que possuem um bico. Porém algumas adequações são necessárias para o seu uso.

Um exemplo de equação que possui bico pode ser feita utilizando a seguinte equação:

$$f(x) = \begin{cases} x & \text{para } 0 \leq x \leq V_l; \\ -x + 2 \cdot V_l & \text{para } V_l \leq x \leq 2 \cdot V_l; \end{cases}$$

Em que x representa o valor do macronutriente do alimento, em gramas, V_l representa o valor limite de macronutriente a ser ingerido, em gramas, e $f(x)$ o valor da pontuação para aquele macronutriente.

Uma equação que foi utilizado para se calcular a pontuação global foi a equação:

$$f(x) = \begin{cases} \sin\left(\frac{x \cdot \pi}{V_l \cdot 2}\right) \cdot P_{max} & \text{se } x \leq 2 \cdot V_l; \\ 0 & \text{caso contrário;} \end{cases}$$

Em que x , V_l e $f(x)$ representam valor do macronutriente do item, valor limite de macronutriente a ser ingerido e valor de pontuação, respectivamente. P_{max} representa a quantidade máxima de pontos que o ponto $(V_l, f(x))$ pode obter, ou seja, a quantidade máxima de pontos que aquele macronutriente pode receber quanto mais próximo estiver da necessidade do usuário. Ainda para esta equação foi utilizado o limite para 2 vezes o valor limite que o usuário pode ingerir por convenção.

A tabela 12 mostra o valor de 2 itens calculados com a equação anterior. Utilizou-se neste caso um $V_{l-carbo} = 40$, $V_{l-proteina} = 30$ e um $V_{l-lipidio} = 7$. Para o valor de pontuação máxima foi usado o $P_{max} = 100$ como valor base, definido por convenção, para o cálculo de todos os macronutrientes.

Tabela 12: Exemplo de valoração dos itens

| Alimento | Porção | Item | Carboidratos (g) | Proteínas (g) | Lipídios (g) | Valor |
|----------|---------------------|------------------------------|------------------|---------------|--------------|--------|
| Arroz | 1 colher de arroz | [Arroz, 1 colher de arroz] | 3 | 1,5 | 1 | 41,852 |
| Arroz | 2 colheres de arroz | [Arroz, 2 colheres de arroz] | 6 | 3 | 2 | 82,376 |

Fazendo-se as devidas substituições, temos que o valor de pontuação global para o primeiro item é de 41,852 e para o segundo item 82,376.

Tabela 13: Campos e tamanhos da estrutura para cada item

| Nome do Campo | Tipo do Campo | Tamanho(bytes) |
|---------------|---------------|----------------|
| id | int | 4 |
| name | char[150] | 150 |
| protein | float | 4 |
| carbo | float | 4 |
| fat | float | 4 |
| cal | float | 4 |
| value | float | 4 |
| servingType | char[100] | 100 |
| Total | | 274 |

A quantidade de itens recuperados do banco de dados do *OnFit* é de aproximadamente 17 mil, e todos foram alocados utilizando da estrutura referenciada na tabela 13. A quantidade de memória alocada foi de $17.000 \cdot 274 = 4.658.000$ bytes, ou seja, aproximadamente 4,7 Megabytes.

5.2.4 Uso e Análise do *knapsack*

Nesta seção é explorado como o *knapsack* foi utilizado, definindo os limites em relação a quantidade de colunas e linhas usadas para montar a tabela, como o algoritmo funciona, e ao final da seção a exemplificação do mesmo é feita.

A estratégia de uso do *knapsack* é aplicar o algoritmo nos alimentos de cada refeição, sendo esses escolhidos conforme o seu valor de pontuação global previamente calculado. Esta estratégia foi feita inicialmente utilizando a linguagem C para uma melhor performance e otimização de código. Após feita todas as otimizações necessárias o código foi portado para o aplicativo OnFit e utilizado para análise dos resultados.

A ideia de se adaptar o *knapsack*, foi inspirada pela sua condição de resolução de problemas de forma ótima - utilizando-se de sub-problemas ótimos-, e combinação entre os itens para a formação destes sub-problemas.

Definição dos limites

Para se iniciar o algoritmo, o primeiro passo é fazer o levantamento da quantidade de linhas e colunas a serem utilizadas pela tabela do *knapsack*.

A contagem das colunas é feita através do total de macronutrientes a serem ingeridos naquela refeição. Caso a quantidade de proteínas a serem ingeridas for igual a 30, de carboidratos ingerida for igual a 40 e de lipídios for igual a 10, então um total de $T_{colunas} = 31 \cdot 41 \cdot 11 = 13.981$ colunas existirá, pois para a contagem de zero proteínas, zero carboidratos e zero lipídios, existirá uma coluna, e para a contagem de zero proteínas, zero carboidratos e 1 lipídio uma outra coluna é formada.

Isso nos remete semelhança a um sistema de contagem mas com limite móvel, pois assim como o sistema decimal ao alcançar o décimo elemento recomeça a contagem das casas das unidades em zero e dezenas em 1, esta contagem de colunas possui este mesmo padrão, sendo que os lipídios podem ser comparados a casa das unidades, os carboidratos das dezenas e as proteínas das centenas.

Já para a quantidade de linhas a contagem dos n itens a serem utilizados no *knapsack* foi feita. Este valor é proveniente da quantidade n de itens mais uma unidade, ou seja, $n + 1$.

Para este trabalho a quantidade de itens analisados é de aproximadamente 17 mil para a possível maior entrada, e mesmo utilizando a atual estratégia de se dividir estes itens para cada refeição, não é viável manter esta tabela armazenada no sistema.

Fazendo-se o cálculo da quantidade de memória alocada no momento da execução para 6 mil itens temos $n+1 = 6.001$ linhas, e utilizando do caso já dito de $T_{colunas} = 13.981$, temos 83.899.981 células. Por padrão as células da tabela do *knapsack* utilizado para este trabalho são do tipo inteiro, ou seja, utilizam 4 bytes de alocação de memória no momento da execução do algoritmo. Finalizando o cálculo temos por volta de 336 Megabytes de memória alocada para esta tabela.

Manter esta tabela é muito oneroso para o sistema, e uma forma feita para contornar isto foi implementar o *knapsack* utilizando somente 2 linhas da tabela em vez de $n + 1$ linhas.

Utilizando esta abordagem a quantidade de memória alocada pelo sistema é calculado em $T_{colunas} * 2 = 27.962 * 4 = 111.848$ bytes. Ou seja, uma mudança de 336 Megabytes para 111 Kilobytes de alocação de memória.

Funcionamento do knapsack

Foi dito na seção anterior que a tabela a ser utilizada pelo *knapsack* foi implementada utilizando-se somente 2 linhas, e eis aqui o processo esclarecido de como irá funcionar esta otimização:

1. Toda a tabela é inicializada com zeros;
2. Para a primeira linha, nada é feito. Logo os valores de zero são mantidos;
3. Para a segunda linha os valores são calculados utilizando a primeira linha como referência;
4. É feita a troca para a primeira linha e os valores dela são calculados utilizando a segunda linha como referência;
5. O processo de troca de linha de referência é feito até o término dos itens analisados.

Ao mesmo tempo que a tabela vem sendo construída evolutivamente nas 2 linhas, temos o processo de recuperação de itens acontecendo simultaneamente. Pois em função desta otimização de espaço, uma nova forma de recuperação de itens teve que ser criada. Esta tabela de 2 linhas foi produzida a partir da quantidade máxima de proteínas, carboidratos e lipídios ingeridos a serem ingeridos pelo usuário, e sua configuração é mostrada no Código 5.1.

Listing 5.1: Configuração inicial tabela *knapsack*

```

1
2 typedef struct MacroNutrient{
3     int actualIndex;
4     int maxIndex;
5 }MacroNutrient;
6
7 int** setupNewMatrix(int maxLine, MacroNutrient protein,
8     MacroNutrient carbo, MacroNutrient fat, int value){
9
10    int** dp= (int**)malloc(maxLine*sizeof(int*));
11
12    int totalColumns = protein.maxIndex*carbo.maxIndex*fat.
13        maxIndex;
14
15    int i = 0;
16    int j = 0;
17
18    for (i = 0; i < maxLine; i++) {
19        dp[i] = (int*)malloc(sizeof(int)*totalColumns);
20        for (j=0; j<totalColumns; j++) {
21            dp[i][j] = value;
22        }
23    }
24 }
```

```
21 |
22 |     return dp;
23 | }
```

A forma comum de recuperação de itens de uma tabela do *knapsack* é feita após o término do cálculo de todas as células, no qual a solução é inferida da célula localizada na última linha e última coluna desta tabela. A recuperação comum pode ser observada no código 5.2.

Listing 5.2: Recuperação de itens do *knapsack* comumente usada

```
1 Item* recuperaItens (Tabela tabela , Item* itens) {
2     int line=0;
3     int column = tabela.column;
4     Item bag[itens.size];
5
6     for (line=tabela.lines; line > 0; line--) {
7
8         int actualValue = tabela[line][column];
9         int lineAboveValue = tabela[line-1][column];
10
11         if actualValue != lineAboveValue {
12             Item currentItem = itens[line];
13             bag.add(currentItem);
14             column = column - currentItem.weight;
15         }
16     }
17
18     return bag;
19 }
```

Pode ser observado que esta solução de recuperação utiliza toda a tabela para saber quais são os itens escolhidos pelo algoritmo. Como a atual tabela deste trabalho possui somente 2 linhas, então não é possível fazer o rastreamento utilizando do método comum.

Como dito anteriormente, ao passo que os itens são analisados pelo *knapsack* o rastreamento é feito. O método utilizado para se fazer isto é baseado no fato de marcar se o item foi escolhido ou não durante o cálculo de cada célula da tabela, tornando possível recuperar todos os itens previamente escolhidos na última célula da tabela.

Este método de guardar o estado de recuperação de cada célula foi feito utilizando

da lógica de 1 byte conseguir representar até 8 itens analisados, ou seja, a cada um bit marcado como 1 neste byte, significa dizer que o item foi escolhido, caso o bit esteja marcado como 0 então o item não foi escolhido.

Para se calcular a quantide de bytes a serem usados em cada células é feito o cálculo $T_b = \lceil \frac{T_{itens}}{8} \rceil$, em que T_b é o total de bytes existente em cada célula da tabela e T_{itens} é o total de itens a serem utilizados para esta tabela. Finalizando este cálculo, um total $total_{bytes}$ pode ser encontrado se considerando o exemplo anterior com $T_{itens} = 6$ mil itens:

$$\begin{aligned}
 T_{colunas} &= 13.981 \\
 n_{linhas} &= 2 \\
 T_b &= 750 \\
 total_{bytes} &= T_{colunas} \cdot n_{linhas} \cdot T_b \\
 total_{bytes} &= 13.981 \cdot 2 \cdot 750 \\
 total_{bytes} &= 20.971.500 \text{ bytes} \\
 total_{bytes} &\cong 21 \text{ Mega bytes}
 \end{aligned} \tag{5.4}$$

No momento de criação da tabela de valores do *knapsack* também é criada uma tabela idêntica, porém com um vetor de bytes a serem utilizados em cada célula. Então este vetor de bytes são zerados utilizando uma máscara de bit proveniente do cálculo $0 \text{ AND } 255$, ou seja, um bit com zeros $0000 \ 0000$ AND um bit com $1111 \ 1111$. Isso garante que os bytes a serem utilizados para o monitoramento do estado de recuperação estejam sempre zerados no início do processo.

O próximo passo é calcular uma máscara de bits para cada bit a ser setado, porém para se fazer isto é necessário ter todos os ID de itens iguais a sua respectiva linha de cálculo na tabela do *knapsack*, pois ao se fazer o cálculo $index_{bit} \equiv ID_{item} \pmod{8}$, é obtido o index do bit a ser setado no byte de rastreamento. Utilizando-se o valor do $index_{bit}$ em uma potência de 2, a máscara é calculada. A Tabela 14 mostra todos os valores possíveis para esta máscara. Para encontrar o byte no qual o bit precisa ser setado é feito o cálculo $index_{byte} = \frac{ID_{item}}{8}$, em que $index_{byte}$ se refere ao index do byte.

O passo seguinte é seguir o mesmo padrão de escolha do *knapsack* para setar o melhor valor, isto pode ser visto no Código 5.3.

Listing 5.3: Método de recuperação de itens

```

1      if ((p-(int)actualItem.protein >=0) &&
2          (c-(int)actualItem.carbo >=0) &&
3          (f-(int)actualItem.fat >=0)) {
4

```

Tabela 14: Máscara de bit possíveis

| id_{bit} | Potência de 2 | Máscara de bit |
|------------|---------------|----------------|
| 0 | 2^0 | 0000 0001 |
| 1 | 2^1 | 0000 0010 |
| 2 | 2^2 | 0000 0100 |
| 3 | 2^3 | 0000 1000 |
| 4 | 2^4 | 0001 0000 |
| 5 | 2^5 | 0010 0000 |
| 6 | 2^6 | 0100 0000 |
| 7 | 2^7 | 1000 0000 |

```

5      int indexP = p-(int)actualItem.protein;
6      int indexC = c-(int)actualItem.carbo;
7      int indexF = f-(int)actualItem.fat;
8
9      int previousLineValue = (*matrix)[previousLine][p][c][f];
10     int previousLineWithItemValue = (*matrix)[previousLine][
        indexP][indexC][indexF]+ actualItem.value;
11
12     float maxValue = max(previousLineValue ,
        previousLineWithItemValue);
13
14     (*matrix)[line][p][c][f] = maxValue;
15
16     // set up byte for bag
17     // discover byte
18     idByte = actualItem.id/8;
19     // discover bit
20     idBit = actualItem.id%8;
21     // get bitmask
22     bitMask = bitMaskForByte(idBit);
23     if (previousLineValue > previousLineWithItemValue) {
24         int countByte = 0;
25         for (countByte=0; countByte<sizeBytes; countByte++) {
26             bagTrack[line][p][c][f][countByte] = bagTrack[
                previousLine][p][c][f][countByte];
27         }
28
29     }else{
30

```

```

31     int countByte = 0;
32     for (countByte=0; countByte<sizeBytes; countByte++) {
33         bagTrack[line][p][c][f][countByte] = bagTrack[
34             previousLine][indexP][indexC][indexF][countByte];
35     }
36     // set bit in correct byte
37     bagTrack[line][p][c][f][idByte] = bagTrack[previousLine
38         ][indexP][indexC][indexF][idByte] | bitMask;
39 }
40 }else{
41     (*matrix)[line][p][c][f] = (*matrix)[previousLine][p][c][
42         f];
43     int countByte = 0;
44     for (countByte=0; countByte<sizeBytes; countByte++) {
45         bagTrack[line][p][c][f][countByte] = bagTrack[
46             previousLine][p][c][f][countByte];
47     }

```

E como último passo é feita a recuperação dos itens a partir da última célula da tabela de rastreamento. O Código 5.4 mostra como ocorre a recuperação dos itens.

Listing 5.4: Recuperação de itens a partir dos bytes da última célula

```

1  // retrieve data from bagTrack
2  if (lines%2 != 0) {
3      line = 0;
4  }else {
5      line = 1;
6  }
7  char* trackCell = bagTrack[line][protein.maxIndex-1][carbo.
8      maxIndex-1][fat.maxIndex-1];
9
10 Item bag[numFoods];
11
12 int j;
13 int itemCount=0;
14 for (i=0; i<sizeBytes; i++) {
15     for (j=0; j<8; j++) {
16         int itemIndex = i*8+j;

```

```
17     int trackValue = (int)(trackCell[i] & (char)pow(2,j));
18     int bitMaskNumber = (int)pow(2,j);
19
20     if (trackValue == bitMaskNumber) {
21         bag[itemCount] = itens[itemIndex];
22         itemCount++;
23     }
24 }
25 }
26
27 Item empty;
28 empty.id = -1;
29 bag[itemCount] = empty;
30
31 Bag newBag;
32 newBag.items = bag;
33 newBag.size = itemCount;
34 return newBag;
35 }
```

Após feita a recuperação dos itens, os mesmos são salvos em arquivo CSV (.csv, *comma separated value*) e repassados para profissional de nutrição para avaliação.

Abordagem Iterativa

A execução do *knapsack* possui duas abordagens: iterativa ou recursiva. A abordagem utilizada para o algoritmo confeccionado neste Trabalho de Conclusão de Curso foi a iterativa.

A abordagem recursiva, por sua natureza de utilização, consome grande quantidade de memória RAM quando utilizada. Ou seja, pelo fato do algoritmo estar sendo executado em um dispositivo móvel, a natureza deste dispositivo não permite que uma grande quantidade de memória RAM seja consumida, uma vez que a massa de itens analisada é muito grande, pois é utilizada uma lista de alimentos com n tipos de medições. Por este motivo foi utilizada a abordagem iterativa.

5.3 Considerações Finais

Este capítulo teve como intuito mostrar o problema a ser resolvido, qual a solução para o mesmo e como este problema foi abordado e desenvolvido. Na primeira seção é feita a definição dos problemas.

Na segunda seção é mostrado qual é a solução, como os dados do usuários são preparados, como os itens a serem utilizados são formulados, como os dados de entrada são preparados, armazenados e contabilizados para serem utilizados pelo algoritmo, e como é feito e utilizado o *knapsack* em função das restrições de espaço e tempo já citadas.

6 Resultados Obtidos

Este capítulo tem o intuito de mostrar as Histórias de Usuário levantadas e finalizadas, como o código do algoritmo foi modularizado e testado, desempenho em espaço e tempo, análise de complexidade e as evoluções em ciclos do balanceamento de valores para a pontuação global dos itens.

Para o ciclo de evoluções foram utilizados o conceito de cenários. Os cenários são as informações básicas dos usuários que foram utilizadas para se fazer a recomendação de alimentos.

6.1 Análise dos Requisitos

Nesta seção os requisitos deste trabalho são mostrados conforme o problema encontrado em relação a sugestão de alimentos que é feita atualmente em outros aplicativos. Ainda pode ser visto qual a total completude de resolução de cada requisito levantado.

6.1.1 Requisitos Funcionais

Os requisitos funcionais foram formalizados como Histórias de Usuário e podem ser observados na Tabela 15.

Tabela 15: Histórias de Usuário levantadas

| ID | Título | Descrição |
|----|-----------------------------|--|
| 2 | Informação de dados básicos | Eu como desenvolvedor quero informar os dados do usuário para saber a quantidade de calorias, carboidratos, lipídios e proteínas que ele deve ingerir diariamente. Dados: Gênero, Idade, Altura, Peso |
| 4 | Termo de compromisso | Eu como integrante do OnFit gostaria que o usuário assinasse um termo de uso relacionado ao uso da sugestão de alimentos para não ter problemas legais. |
| 6 | Porções alimentares | Eu como desenvolvedor quero integrar aos alimentos as possíveis porções existentes de cada um para formatar os itens a serem usados no knapsack. |

| | | |
|-------|---|--|
| 7 | Refeições diárias | Eu como desenvolvedor quero separar a quantidade de nutrientes a serem ingeridos diariamente em 6 refeições diárias para melhor aplicar o algoritmo de recomendação de alimentos. |
| 8 | Aplicação knapsack por refeição | Eu como desenvolver quero aplicar o algoritmo knapsack nos alimentos de cada refeição diária para saber quais são os melhores alimentos a serem utilizados pelo usuário. |
| 9 | Refinamento do valor de calorias e nutrientes | Eu como desenvolvedor quero utilizar do objetivo e da quantidade de dias de exercícios na semana do usuário para refinar a quantidade de nutrientes e calorias diários. Objetivos: Perder peso, ganhar massa muscular e manter peso Exercícios na semana: 0, 1-2, 3-4, 4-5,6-7 dias na semana. |
| 10 | Banco de dados de alimentos | Eu como desenvolvedor desejo integrar o banco de dados para utilizar os alimentos no algoritmo de recomendação alimentar |
| TS-01 | Tempo de execução do algoritmo | Eu como Engenheiro de Software quero fazer medições do tempo de execução do software e plotar o gráfico para estas medições com o intuito de documentar a análise e implementar melhorias. |
| TS-02 | Recuperação de itens | Eu como Engenheiro de Software quero recuperar todos os itens analisados pelo knapsack utilizando a metodologia imposta pelo TCC para melhorar a implementação e otimização do algoritmo. |

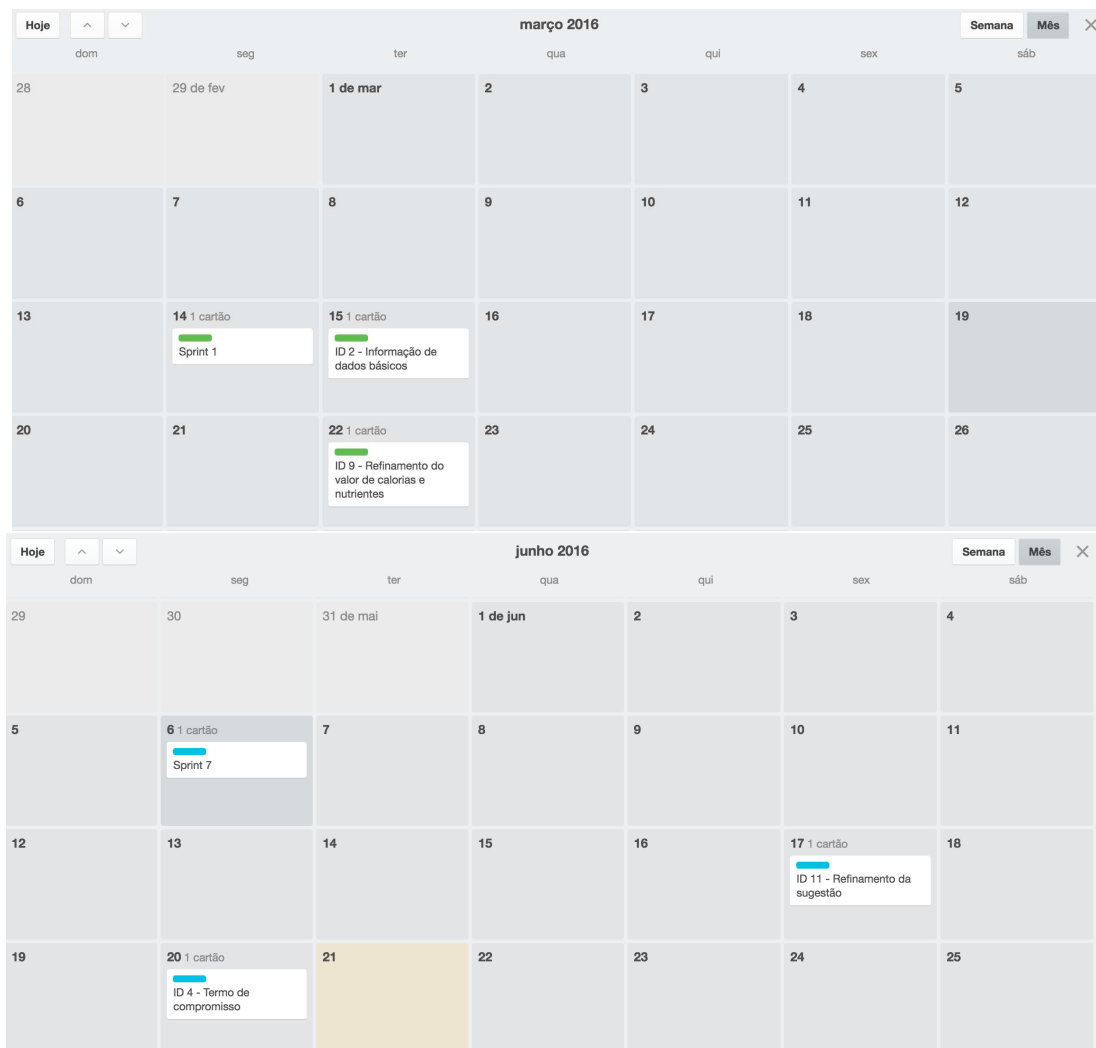
Tarefas foram adicionadas para se ter a total completude das Histórias de Usuários. Foi utilizado o *Trello* e o *PunchTime* como ferramentas para auxiliar no gerenciamento destas Histórias¹. A Tabela 16 mostra qual a porcentagem de completude e a qual *Sprint* cada uma foi iniciada e finalizada.

A implementação do TCC teve o seu começo no dia 14 de Março de 2016, cada *Sprint* teve duração de 2 semanas, com um total de 7 *Sprints*, finalizando a implementação no dia 20 de Junho de 2016. O *Trello* também foi utilizado como ferramenta de controle das Histórias de Usuário em cada *Sprint*. Esta integração pode ser vista na Figura 12.

¹ Vide Subseção 4.2.2 para maiores detalhes de como foi feita a utilização destas ferramentas.

Tabela 16: Completude das Histórias de Usuário. Tarefas feitas por tarefas levantadas.

| ID | Tarefas | Completude | Sprint Inicial | Sprint Final |
|-------|---------|------------|----------------|--------------|
| 2 | 1/1 | 100% | Sprint 1 | Sprint 1 |
| 9 | 2/2 | 100% | Sprint 1 | Sprint 1 |
| 7 | 1/1 | 100% | Sprint 2 | Sprint 2 |
| 6 | 3/3 | 100% | Sprint 2 | Sprint 2 |
| 10 | 3/3 | 100% | Sprint 3 | Sprint 4 |
| 8 | 7/7 | 100% | Sprint 3 | Sprint 4 |
| TS-01 | 3/3 | 100% | Sprint 3 | Sprint 5 |
| TS-02 | 5/5 | 100% | Sprint 4 | Sprint 6 |
| 4 | 0/4 | 0% | Sprint 7 | Sprint 7 |
| 11 | 0/3 | 0% | Sprint 7 | Sprint 7 |
| | Total | 80% | | |

Figura 12: Trello sendo utilizado para controle de *Sprint*

Todas as Histórias de Usuário e detalhes das suas tarefas podem ser observadas no apêndice B.

6.1.2 Requisitos Não-Funcionais

O contexto utilizado para este trabalho engloba as necessidades envolvidas do contexto de dispositivos móveis, ou seja, existem restrições a serem seguidas, e uma das restrições existentes é a configuração física do dispositivo móvel em uso.

O dispositivo móvel utilizado é o iPhone 5C, estas são as configurações mínimas utilizáveis para o projeto. As configurações podem ser vistas na Tabela 17.

Tabela 17: Configurações iPhone 5C

| Nome configuração | Valor da configuração |
|------------------------|-----------------------|
| Modelo | iPhone 5C |
| Processador | Chip A6 |
| Sistema Operacional | iOS 9.3.2 |
| Capacidade Interna | 32 Gb |
| Capacidade Memória RAM | 1Gb |

Relacionando estas configurações com o projeto, alguns cenários de requisitos não funcionais foram levantados conforme configurações de modelo do formulário *QaSE*².

Nas Tabelas 18 e 19, pode se observar como os requisitos não funcionais influenciam no projeto como um todo.

Tabela 18: Requisitos não funcionais: Título e Descrição

| ID | Autor | Título | Descrição |
|----|---------------|--|---|
| 1 | Victor Cotrim | Baixo tempo de execução do algoritmo | <p>Este cenário indica que o algoritmo feito para sugerir os alimentos para o usuário, deve ter um alto desempenho em relação a sua execução.</p> <p>A situação que pode ser delimitada para este cenário é o momento de execução do algoritmo.</p> <p>Os limites do tempo de execução deverá ser estipulado mediante observações de uso do usuário.</p> <p>O aplicativo, OnFit, irá prover a iteração necessária entre o algoritmo de sugestão de alimentos e o usuário.</p> |
| 2 | Victor Cotrim | Baixa utilização de memória RAM do algoritmo | <p>Este cenário indica que o algoritmo deve utilizar uma quantidade de memória RAM suficiente para não atrapalhar na execução do sistema.</p> |

² vide Seção 2.2 para mais informações sobre os formulários *QaSE*.

| | | | |
|--|--|--|--|
| | | | <p>A situação que pode ser delimitada para este cenário é o momento da execução do algoritmo.</p> <p>Já é sabido pelo autor deste cenário, que o algoritmo em sua execução deve evitar passar os 300 Megabytes de espaço e nunca ultrapassar os 500 Megabytes.</p> |
|--|--|--|--|

Tabela 19: Requisitos não funcionais: Interdependências, Conflitos e Impactos

| ID | Interdependências entre NFR | Conflitos entre NFR | Impactos do não atendimento aos NFR |
|----|--|--------------------------|--|
| 1 | Dependência direta entre o cenário 2. Pois quanto maior a quantidade de RAM utilizada, maior será o tempo de execução do algoritmo | Sem conflitos detectados | <p>O primeiro fator é o roubo de ociosidade do sistema, pois quanto mais tempo um processo permanece ativo, maior é a dificuldade de se manter os outros processos³ ininterruptos.</p> <p>O segundo fator faz com que o aplicativo <i>OnFit</i> tenha sua fluidez e simplicidade comprometida, pois quanto maior o tempo de espera para concluir uma ação, menor é a disposição do usuário em utilizar aquele aplicativo⁴.</p> |
| 2 | Sem dependência entre outro NFR | Sem conflitos detectados | Caso este requisito não seja atendido, existe a possibilidade de onerar o sistema operacional do aparelho utilizado. Outros aplicativos, ou até mesmo o sistema operacional, do iPhone não irá funcionar de maneira correta. |

³ Entenda-se outros processos como outros aplicativos abertos em segundo plano ativos.

⁴ Observações feitas pela equipe *OnFit*([ANDRADE](#); [BERNARDES](#); [COTRIM](#), 2015).

6.2 Módulos do Código

Desde o início da implementação da terceira versão do algoritmo, suites de testes foram implementados com o intuito de manter a qualidade e integridade do código. Desta forma, sempre que alguma inserção ou refatoração de código era feita, se tinha controle de quais funções continuavam funcionando sem erros.

Toda a suite de testes foi feita utilizando o XCTest. Este *framework* permite a utilização de testes unitários, desempenho e outros tipos de teste, em módulos separados do código a ser testado. Sendo assim possível fazer a execução do módulo de testes personalizado de forma independente do módulo nativo de testes provido pelo XCode.

O código foi dividido em 7 módulos, e baseado nesses módulos as suites de testes puderam ser melhor dividida e organizadas, corroborando assim com o objetivo específico 3 relacionado a documentação e boas práticas. Estes módulos foram divididos em:

- *Database Manager*: Responsável por gerenciar as recuperações dos itens a serem utilizados no *knapsack* no banco de dados;
- *File Manager*: Responsável por gerenciar a geração dos arquivos em .CSV contendo informações do usuário e da recomendação gerada;
- *Food Recommendation Equations*: Responsável por gerenciar qual a equação utilizada para calcular os itens recuperados no banco de dados;
- *Knapsack Manager*: Responsável por gerenciar a geração da tabela utilizada pelo *knapsack* e seu respectivo preenchimento;
- *Patient Manager*: Responsável por recuperar as informações básicas do paciente do *OnFit*, calcular a sua EER e informar a quantidade de macro nutriente a ser ingerida em cada refeição;
- *Tracking Tables*: Responsável por preencher a tabela de rastreamento de qual alimento será utilizado ou não;
- *Main*: Responsável por utilizar dos módulos anteriores e rodar o algoritmo na ordem correta.

A Figura 13 mostra como os módulos se relacionam entre si.

6.3 Testes Unitários

Toda a suite de testes unitários foi utilizada sempre que uma mudança substancial ocorria ao código. Esta suite foi dividida em módulos assim como o código em si, faci-

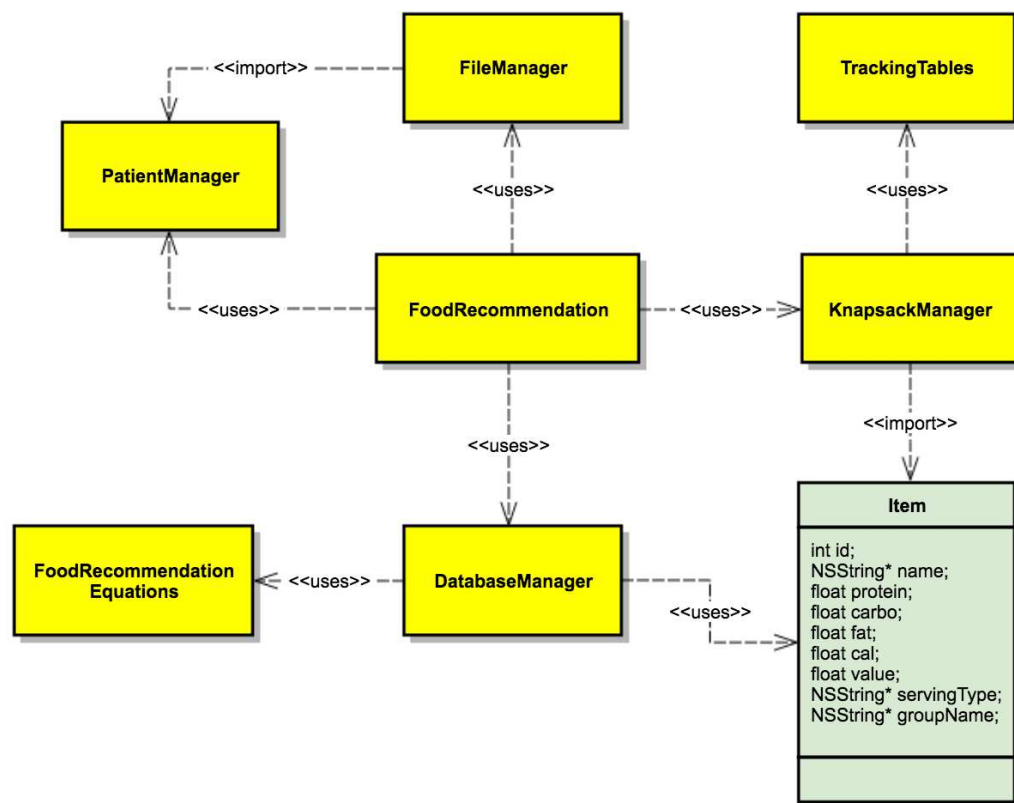


Figura 13: Esquemático de módulos e como se relacionam

litando o teste de cada função e corroboração com o objetivo específico 3 relacionado a documentação e boas práticas.

Com esta separação ocorrida, tanto de código quanto de testes, foi então facilmente percebido como a cobertura de código pode ser feita. Para a cobertura o próprio XCode permite ser facilmente consultado sempre que o código era testado. A Figura 14 mostra a cobertura de cada módulo.

| Name | Coverage ▾ |
|-------------------------------------|-------------|
| ▼ tccTests.xctest | <div></div> |
| ▶ TestFoodRecommendation.m | <div></div> |
| ▶ TestTrackingTable.m | <div></div> |
| ▶ TestDatabaseManager.m | <div></div> |
| ▶ TestFileManager.m | <div></div> |
| ▶ TestPatientManager.m | <div></div> |
| ▶ TestKnapsackManager.m | <div></div> |
| ▶ TestFoodRecommendationEquations.m | <div></div> |
| ▶ tccTests.m | <div></div> |

Figura 14: Cobertura de código provida pelo XCode e XCTest *framework*

6.4 Testes de Desempenho

Os testes de desempenho foram utilizados como uma forma de sempre estar alerta em relação a qual o tempo que o algoritmo utiliza para se fazer a recomendação alimentar do usuário.

Para o XCode, não existe uma diferenciação de se rodar testes unitários e testes de desempenho no mesmo arquivo de teste. Sendo assim, pode ser observado na aba de testes de gerenciador de testes qual o tempo medido daquele teste de desempenho. Isto pode ser observado na Figura ??.



| All | | Passed | Failed | All | Performance | Set Baselines... | Filter |
|---|--------|----------|--------|-----|-------------|------------------|--------|
| Tests | Status | Time | | | | | |
| TestFoodRecommendation > Selected tests | | | | | | | |
| testPerformanceKnapsack | Passed | 821.30 s | | | | | |

Figura 15: Testes de desempenho para o knapsack realizado ao final do ciclo 2

Vale lembrar que o *knapsack* adaptado utiliza da otimização da tabela de 2 linhas. E pode ser observado a mudança entre tempos de execução do algoritmo que utiliza desta otimização e o que não utiliza esta otimização em relação a quantidade de entradas.

Pode ser percebido que a otimização para menos de 10 mil itens possui uma diferença pequena entre tempos, sendo isto compensatório, pois enquanto o algoritmo que não se utiliza da otimização aloca uma quantidade grandiosa de memória em um menor tempo, aquele que utiliza da otimização aloca uma quantidade mínima de memória em tempo um pouco maior com o porém de realizar todo o processamento da tabela em um menor tempo para todos os itens do banco de dados. Estas informações podem ser inferidas do gráfico na Figura 16.

Ao se utilizar do algoritmo para gerar a recomendação alimentar a duração estimada para a recomendação de alimentos é de aproximadamente 2 horas e não aloca mais que 400 Megabytes de memória RAM. Em uma relação com as primeiras versões implementadas, este tempo é mais do que bom, pois anteriormente o algoritmo não conseguia ser executado em sua completude e ainda alocava mais de 4 GigaBytes de memória RAM, resultando assim em uma não geração da lista de recomendação alimentar.

6.5 Análise de Complexidade

O *knapsack* possui complexidade $O(nC)$, em que n é a quantidade de vezes em que um item foi analisado e C a quantidade de itens utilizados no *knapsack* (SKIENA, 2008, p. 428). O autor ainda cita que esta complexidade é ótima uma vez que $C \leq 1000$ e não tão boa para $C \geq 10.000.000$, ou seja, pode ser inferido que quanto maior a quantidade de itens a serem utilizados pior é o desempenho em relação a tempo e espaço no momento da

execução. Ainda pode ser inferido que grandes valores de n na situação mediana corrobora também para uma piora em relação ao desempenho em tempo e espaço.

O algoritmo desenvolvido neste TCC e adaptado do *knapsack* é executado 6 vezes - uma por refeição -, para a geração da lista de alimentos recomendados e cada uma destas execuções possuem quantidades diferentes de itens a serem analisados. O valor de n pode ser dito como a quantidade de carboidrato, proteína e lipídios calculados para a refeição e o valor de C a quantidade de itens utilizados para a refeição. Logo pode ser inferido que a complexidade do algoritmo é $O(nC)$ em que $n = c \cdot p \cdot l$ e c = quantidade de carboidrato calculada, p = quantidade de proteínas calculada e l = quantidade de lipídios calculada.

Quando a execução das refeições com quantidade menores de 1000 itens acontece, o tempo utilizado não ultrapassa os 15 segundos de execução, menos de 5 segundos para $n \leq 5000$ e aproximadamente 10 para $n \geq 15.000$. Estas refeições são o café da manhã, colação, lanche da tarde e ceia. Já para as refeições com mais de 2000 itens o tempo de execução fica superior a 8 minutos, tempo este que foi agravado para valor de $n \geq 60.000$, mas para $n \leq 25.000$ o tempo é aproximado a 80 segundos. Estas refeições são o almoço, com até 8000 itens e o jantar, com até 2500 itens.

Em relação ao espaço utilizado o *knapsack* naturalmente utiliza $O(nC) \cdot S_{bytes}$ bytes, em que S_{bytes} é o tamanho do tipo de byte a ser utilizado em sua tabela, se tipo inteiro então será utilizado 4 bytes, se tipo tipo carácter então será utilizado 1 byte. Para este TCC uma otimização foi feita e este valor de C passou a ser igual a 2, resultando em uma complexidade em espaço de $2 \cdot O(n)$. Porém com esta otimização a possibilidade de

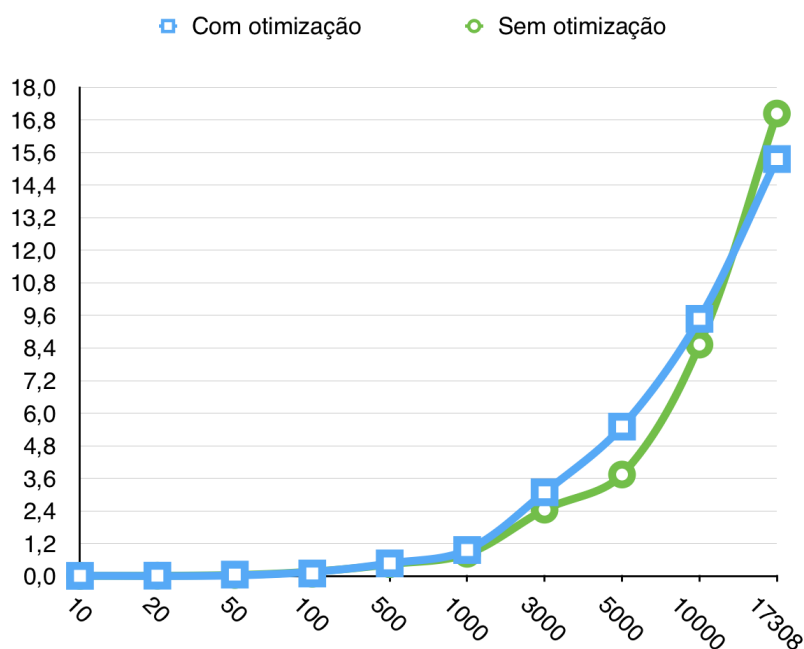


Figura 16: Gráfico tempo de execução X entradas do algoritmo, com e sem otimização

recuperar os itens através da tabela foi removida, pois não haveria uma forma de detectar qual item foi escolhido ou não, e uma nova forma de rastreamento foi desenvolvida. Esta nova forma utiliza de 1 byte para rastrear até 8 itens⁵ e uma tabela de 2 linhas por n colunas, então sua complexidade seria de $2 \cdot O(nB)$ em que $B \equiv C \bmod 8$. A complexidade total para o espaço utilizado fica em $2 \cdot O(n)$ para a tabela do *knapsack* e $2 \cdot O(nB)$ para a tabela de rastreamento, resultando em $2 \cdot (O(n) + O(nb))$.

6.6 Ciclos de Ajuste do Algoritmo

O algoritmo sempre gera um arquivo CSV ao final do processo, esse arquivo é enviado ao nutricionista e analisado pelo mesmo com o intuito de saber a qualidade da recomendação gerada para um cenário.

Caso o cenário não esteja conforme os itens recomendados, então um novo balanceamento é feito e uma nova recomendação é gerada. O balanceamento de valores sempre depende da equação utilizada assim como seu cenário. Foram utilizadas para este processo 1 equação:

$$f(x) = \begin{cases} \text{sen}\left(\frac{x \cdot \pi}{V_i \cdot 2}\right) \cdot P_{max} & \text{se } x \leq 2 \cdot V_i; \\ 0 & \text{caso contrário;} \end{cases}$$

O balanceamento dos valores começam com valores padrões de $P_{maxCarbo} = 50$, $P_{maxProtein} = 40$ e $P_{maxLip} = 10$, então para a primeira equação utilizada para o cálculo da pontuação de cada macronutriente é feita a substituição do P_{max} . Estes valores representam para o algoritmo uma dieta rica em carboidratos. Se o valor de $P_{maxProtein}$ for maior do que o $P_{maxCarbo}$, então teríamos, para o algoritmo, uma análise de alimentos para uma dieta rica em proteínas.

Estes valores foram escolhidos como representativo da quantidade de cada macro nutriente existente no prato do brasileiro comum. Estes valores são mudados conforme consulta com nutricionista.

Os cenários escolhidos para fazer estes ciclos de avaliação podem ser vistos na Tabela 20.

Tabela 20: Cenários utilizados para balanceamento de valores

| | Cenário 1 | Cenário 2 | Cenário 3 | Cenário 4 |
|------|-----------|-----------|-----------|-----------|
| Sexo | Masculino | Feminino | Masculino | Feminino |

⁵ Detalhes do funcionamento vide Seção 5.2.4

| | | | | |
|-------------------|--------------|-------------|-------------|-------------|
| Idade(Anos) | 26 | 26 | 42 | 42 |
| Altura(m) | 1.72 | 1.60 | 1.72 | 1.72 |
| Peso(Kg) | 85 | 76 | 90 | 70 |
| Dias de exercício | 1-2 dias | 1-2 dias | 3-4 dias | 0 |
| Objetivo | Ganhar massa | Perder peso | Perder peso | Manter peso |

Foram feitos 4 ciclos de avaliação para o balanceamento destes valores, com o intuito de comparar se os valores devem ser aumentados ou diminuídos. Os ciclos foram comparados entre si para uma definição de quais valores seriam utilizados para se fazer a recomendação alimentar.

6.6.1 Ciclo 1

O primeiro ciclo foi proposto como primeiro parâmetro de comparação entre as sugestões geradas, pois este é utilizado como caso de comparação central. Os valores utilizados para todos os cenários foram: $P_{maxCarbo} = 50$, $P_{maxProtein} = 40$ e $P_{maxLip} = 10$.

Em todos os casos foram alavancados as seguintes mudanças:

1. Restrição das unidades de medidas a serem utilizadas para cada grupo alimentar, ou seja, aliás de se utilizar para o grupo do Ovos as unidades de colher de chá, gramas e outras unidades de medida, somente a utilização da unidade de Unidades. Para o grupo das Frutas somente será utilizado as unidades de medida Unidades, Xícara, Pedacos e Tigela. Estas mudanças foram refletidas em todos os grupos; e
2. Remoção de alimentos que não são utilizados para recomendação alimentar, como por exemplo, carnes cruas, ovos crus, vegetais refogados.

Após a implementação da remoção das quantidades de unidades de medida e dos alimentos que não são utilizados para recomendação alimentar foi percebido que esta mudança reflete diretamente no tempo de execução do algoritmo. O tempo de execução do algoritmo caiu de 2 horas para pouco mais de 1 hora e a quantidade de memória RAM alocada para pouco menos de 400 Megabytes.

Ao final de cada análise, novos valores de cada cenário foram propostos e uma nova recomendação foi feita.

6.6.2 Ciclo 2

O segundo ciclo foi proposto como segundo parâmetro de comparação entre as sugestões geradas, sendo que cada para cada cenário foram propostos novos valores. Neste

ciclo os valores para cada cenários foram propostos individualmente. Os valores foram propostos pelo nutricionista.

- Cenário 1: $P_{maxCarbo} = 30$, $P_{maxProtein} = 40$ e $P_{maxLip} = 30$;
- Cenário 2: $P_{maxCarbo} = 40$, $P_{maxProtein} = 30$ e $P_{maxLip} = 30$;
- Cenário 3: $P_{maxCarbo} = 40$, $P_{maxProtein} = 30$ e $P_{maxLip} = 30$;
- Cenário 4: $P_{maxCarbo} = 30$, $P_{maxProtein} = 40$ e $P_{maxLip} = 30$;

Os resultados da análise para o cenário 1 foram:

1. Alta quantidade de alimentos ricos em proteínas, sendo necessário baixar o valor base;
2. Quantidade ponderada de alimentos ricos em carboidrato, porém ainda alto, sendo necessário baixar o valor base; e
3. Quantidade baixa de alimentos ricos em lipídios, sendo necessário aumentar o valor base.

Os resultados da análise para os cenários 2, 3 e 4 foram:

1. Quantidade de alimentos ricos em proteína está bem balanceado, manter valor base para proteínas;
2. Quantidade de alimentos ricos em carboidrato esta adequado, porém ainda alto, sendo necessário baixar valor base; e
3. Quantidade de alimentos ricos em lipídios está baixo, sendo necessário aumentar o valor base.

Os valores para os cenários foram modificados para os seguintes valores conforme observações da profissional de nutrição, e uma nova rodada de recomendações foi feita.

- Cenário 1: $P_{maxCarbo} = 30$, $P_{maxProtein} = 20$ e $P_{maxLip} = 40$;
- Cenário 2: $P_{maxCarbo} = 25$, $P_{maxProtein} = 40$ e $P_{maxLip} = 35$;
- Cenário 3: $P_{maxCarbo} = 25$, $P_{maxProtein} = 40$ e $P_{maxLip} = 35$;
- Cenário 4: $P_{maxCarbo} = 30$, $P_{maxProtein} = 30$ e $P_{maxLip} = 40$;

E ainda em tempo de análise, profissional de nutrição pediu a retirada dos alimentos preparados a milanesa.

Após a implementação dos novos valores, não houve mudança no tempo do processamento e na quantidade de memória RAM alocada.

6.6.3 Ciclo 3

O terceiro ciclo foi proposto como terceiro parâmetro de comparação entre as sugestões geradas, sendo que para cada cenário foram propostos novos valores. Neste ciclo os valores para cada cenários foram propostos individualmente. Os valores foram propostos pelo nutricionista.

- Cenário 1: $P_{maxCarbo} = 30$, $P_{maxProtein} = 20$ e $P_{maxLip} = 40$;
- Cenário 2: $P_{maxCarbo} = 25$, $P_{maxProtein} = 40$ e $P_{maxLip} = 35$;
- Cenário 3: $P_{maxCarbo} = 25$, $P_{maxProtein} = 40$ e $P_{maxLip} = 35$;
- Cenário 4: $P_{maxCarbo} = 30$, $P_{maxProtein} = 30$ e $P_{maxLip} = 40$;

Os resultados da análise para todos os cenários foram:

1. Quantidade de alimentos ricos em lipídios está baixa, portanto o valor base para todos os cenários devam ser aumentados;
2. Os valores base para os outros macros é mantido estável.

Foi percebido que os alimentos escolhidos estão sendo repetido bastante entre um ciclo e outro. Portanto algumas suspeitas foram notados:

1. Os valores estão estabilizando, uma vez que não existem mais tantas mudanças entre os alimentos porém a quantidade de alimentos ricos em lipídios é baixa, o que contradiz esta suspeita;
2. Pelo fato dos alimentos estarem se repetindo existe a suspeita de que o banco de dados está pobre em relação ao que pode ser recomendado, porém este banco de dados foi feito pelo IBGE, que o fez em relação ao alimentos mais consumidos pelo brasileiro.
3. Os valores não estão balanceados o suficiente e devem ser aumentados ou diminuídos para se ter uma maior diferença entre os valores calculados para cada macro.

Levando em conta a suspeita 3, os valores de lipídios de todos os cenário foram dobrados.

6.6.4 Ciclo 4

O quarto ciclo foi proposto como ciclo de análise como resultado das comparações entre as sugestões geradas nos ciclos anteriores. Neste ciclo os valores para cada cenários foram pensados e analisados conforme objetivo nutricional de cada cenário.

- Cenário 1: $P_{maxCarbo} = 30$, $P_{maxProtein} = 20$ e $P_{maxLip} = 80$;
- Cenário 2: $P_{maxCarbo} = 25$, $P_{maxProtein} = 40$ e $P_{maxLip} = 70$;
- Cenário 3: $P_{maxCarbo} = 25$, $P_{maxProtein} = 40$ e $P_{maxLip} = 70$;
- Cenário 4: $P_{maxCarbo} = 30$, $P_{maxProtein} = 30$ e $P_{maxLip} = 80$;

Os resultados da análise para todos os cenários foram:

1. Quantidade de alimentos ricos em lipídios está baixa;

Esta análise trouxe a suspeita de que as quantidades padrões de 1, 2 e 3 porções para cada alimento no momento da análise do algoritmo não são o suficiente para alguns alimentos. Esta modificação da quantidades direcionadas merecem um estudo um pouco mais aprofundado, não sendo abordadas neste TCC.

Ainda com base no resultado da análise, os valores base de lipídios foram mantidos e os valores de carboidrato foram baixados para todos os cenários. Somente para o cenário 1 o valor de proteína foi baixado.

6.6.5 Conclusão dos Ciclos

O algoritmo se mostra estabilizar em relação a lista de alimentos gerada ao final de cada cenário. Esta lista vem sendo repetida para cada cenário, não havendo mudanças significativas para uma recomendação para o usuário, mas muito expressivas para uso do profissional de nutrição.

Profissional de nutrição sugere algumas modificações em relação a pontuação, relacionando o IMC do usuário no caso dos cenários 2, 3 e 4, com novas pontuações. Um estudo sobre isto merece ser aprofundado, mas para este TCC foge ao escopo em relação ao tempo de entrega.

6.7 Considerações Finais

Este capítulo teve o intuito de mostrar todos os resultados obtidos com o desenvolvimento deste algoritmo de recomendação de alimentos.

A primeira seção se refere aos requisitos elicitados para este TCC. A segunda, terceira e quarta seção se referem a como o código foi modularizado e testado.

A quinta seção mostra a análise de complexidade realizada tanto para tempo quanto para espaço em tempo de execução do algoritmo desenvolvido.

A última seção se refere a como o algoritmo desenvolvida teve os seus valores de pontuação de macro nutrientes de cada item balanceado em ciclos de evolução com a profissional da área de nutrição.

A parte relacionada aos ciclos de ajustes dos valores de pontuação corroboram diretamente com o objetivo 2 relacionado ao que tange a solução de um algoritmo para recomendação de alimentos, no quesito de validação e ajustes do algoritmo por profissional da nutrição.

Toda a parte de módulos do código, testes unitários e de desempenho, análise de requisitos e ciclos de ajustes dos valores de pontuação corroboram com o objetivo específico 3 relacionado a documentação, boas práticas e testes.

7 Conclusão

Neste Trabalho de Conclusão de Curso foi possível perceber como um algoritmo para recomendação de alimentos foi desenvolvido utilizando técnicas proveniente da Engenharia de Software.

Os problemas relacionados a área de Nutrição foram devidamente estudados pelo o autor deste TCC, sendo orientado por profissional de nutrição em momentos de grande *gap* de conhecimento, e o mesmo pode ser dito para os problemas relacionadas a área de Engenharia de Software, mas com orientação de professores com grande expertise em análise de algoritmos. Estes problemas foram devidamente entendidos e transformados em requisitos, para um melhor planejamento e documentação do algoritmo desenvolvido neste TCC. Para o desenvolvimento dos requisitos foi feita uma adaptação metodológica tanto para desenvolvimento de software quanto para o avaliação da lista de alimentos resultante do algoritmo desenvolvido. Para o desenvolvimento de software houve a adaptação do *Scrum* para a realidade desta TCC na ferramenta *Trello*.

Para este TCC foram estudadas formas de adaptação do algoritmo *knapsack* para o contexto da recomendação de alimentos, desde a adaptação dos macro nutrientes de cada alimento para um peso e pontuação únicos, até a presente adaptação da transformação dos macro nutrientes como o peso dos itens e pontuação global inferida do somatório do valor de cada macro nutriente.

Também pode ser observado que mesmo com o algoritmo devidamente adaptado, ainda foi necessário a consulta de profissional de nutrição para averiguar se as listas com os alimentos recomendados estavam corretas. Averiguação esta que foi devidamente feita e documentada em ciclos de ajustes.

Foi notado que este algoritmo desenvolvido para plataforma móvel necessita de mais lapidamento. Mesmo com grande estudo e otimização do algoritmo, não foi possível melhorar o desempenho para esta área. Mas com relação ao desempenho num fator geral houve grande evolução desde a sua concepção até o atual estado.

O primeiro objetivo específico que cita a investigação de uma solução que envolve o perfil do interessado, questão de desempenho e não comprometimento da usabilidade em dispositivos móveis foi completa parcialmente, pois com a investigação foi desenvolvida uma solução que consegue utilizar do perfil do interessado com um bom desempenho, porém não bom o suficiente da ótica para dispositivos móveis.

O algoritmo parece ser muito promissor em relação a utilização para o profissional de nutrição em seus pacientes. Esta abordagem deve ser estudada e adaptada para um

novo contexto de desenvolvimento.

O segundo objetivo específico está relacionado ao que tange a solução de um algoritmo para recomendação de alimentos, este objetivo foi completo e pode ser observado no capítulo anterior.

O terceiro objetivo específico relaciona-se a questão de documentação e boas práticas para a Engenharia de Software, este objetivo foi completo e isso pode ser observado em relação à documentação levantada durante todo este TCC. No quesito requisitos foram elicited Histórias de Usuário. No quesito código e tests foram feitos testes unitários e de desempenho assim como a cobertura de código.

7.1 Sugestão de Trabalhos Futuros

1. Estudar mais a fundo a questão da quantidade a ser ingerida de cada alimento;
2. Fazer novas rodadas de pesquisas de balanceamento de valores, utilizando X cenários por objetivo nutricional do usuário. Baseado nisso os valores de cada item seriam melhor estabelecidos;
3. Melhorar separação de alimentos no banco de dados. Com isso a recomendação alimentar ficaria mais inteligente;
4. Propor melhor forma de recuperar os itens do knapsack, isso melhoraria o desempenho como um todo, tanto em relação ao espaço quanto em tempo;
5. Estudar novas equações a serem utilizadas para calcular o valor dos macro nutrientes, e com base nisso fazer Y ciclos de avaliação em X cenários;
6. Atrelar a pontuação do usuário ao IMC e informações da quantidade de macro nutriente calculada.

Referências

- ANDRADE, L.; BERNARDES, T.; COTRIM, V. *OnFit (OnFit, app)*. V 1.0. 2015. App Store. Disponível em loja virtual de aplicativos da Apple. Objective C. iPhone 5C, iPhone 5S, iPhone 6 e iPhone 6 plus, iOS. Citado 4 vezes nas páginas 30, 47, 65 e 83.
- APPLE. *iOS Developer Library*. [S.l.]. Acessado em 12 de Junho de 2015. O guia pode ser encontrado em formato digital. Disponível em: <<https://developer.apple.com/library/ios/navigation/>>. Citado na página 51.
- AUTORES, v. t. V. *Nutrição Moderna na saúde e na doença*. 10 edição. ed. [S.l.]: Manole, 2009. Volume Único. Citado na página 107.
- BAASE, S.; GELDER, A. V. *Computer Algorithms*. 3. ed. [S.l.]: Addison Wesley Longman, 2000. Citado na página 42.
- BRITO, R. S. de. *Uma Proposta para Modelagem de Requisitos Não-Funcionais em Projetos Ágeis*. Dissertação (Mestrado) — UFPE - UNIVERSIDADE FEDERAL DE PERNAMBUCO, 2010. Citado na página 41.
- CHACON, S.; STRAUB, B. *Pro Git*. 2. ed. Appres, 2016. Disponível em: <<https://progit2.s3.amazonaws.com/en/2016-01-18-98093/progit-en.997.pdf>>. Citado 3 vezes nas páginas 15, 51 e 52.
- CONFORTO, E. C.; AMARAL, D. C.; SILVA, S. L. da. Roteiro para revisão bibliográfica sistematizada: aplicação no desenvolvimento de produtos e gerenciamento de projetos. *Oitavo Congresso Brasileiro de Gestão de Desenvolvimento do Produto*, 2011. Citado na página 55.
- CORMEN, T. H. et al. *Introduction to Algorithms, second edition*. 8. ed. [S.l.]: McGraw-Hill, 2007. Citado 2 vezes nas páginas 44 e 45.
- DASGUPTA, S.; PAPADIMITROU, C.; VAZIRANI, U. *Algoritmos*. [S.l.]: McGraw-Hill, 2009. Citado na página 43.
- FEOFILOF, P. *Análise assintótica: ordens O, Omega e Theta*. 2015. Site IME-USP. Site acessado em 13/01/2016 às 23:00 horas. Disponível em: <http://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/Oh.html>. Citado na página 44.
- FILHO, C. M. de O. *Kalibro: interpretação de métricas de código-fonte*. Dissertação (Mestrado) — Instituto de Matemática e Estatística da Universidade de São Paulo, 2013. Citado 2 vezes nas páginas 41 e 42.
- FREITAS, M. *Tecnonutri - Gestão da sua dieta, contador de calorias, registro de alimentação, cálculo do IMC, tabela nutricional de alimentos, emagreça com saúde*. 2014. AppStore Apple. Citado 2 vezes nas páginas 49 e 50.
- HIRSCHBRUCH, M. D. *Nutrição esportiva: Uma visão prática*. 3 edição. ed. [S.l.]: Manole, 2014. Volume único. Citado 2 vezes nas páginas 105 e 107.

- ISO/IEC 9126-1. *ISO/IEC 9126-1 - Software engineering - Product quality - Part 1: Quality model*. Citado na página 41.
- KLEINBER, J.; TARDOS Éva. *Algorithm Design*. [S.l.]: Pearson, 2005. Citado 2 vezes nas páginas 42 e 57.
- LUTOWSKI, R. *Software requirements : encapsulation, quality, and reuse*. [S.l.]: Auerbach Publications, 2005. Citado na página 40.
- MACHADO, F. N. R. *Análise e Gestão de requisitos de software: onde nascem os sistemas*. [S.l.]: Editora Érica, 2011. Citado 2 vezes nas páginas 40 e 41.
- MAHAN, L. K.; ESCOTT-STUMP, S. *Krause, Alimentos, Nutrição e Dietoterapia*. 12. ed. [S.l.]: Elsevier, 2010. Citado 10 vezes nas páginas 33, 35, 38, 103, 105, 106, 107, 108, 109 e 110.
- ONFIT, E. SOUSA, Aline Queiroz de. *Entrevista com nutricionista Aline*. 2014. Brasília. Entrevista concedida a Lucas Andrade, Thiago Bernardes e Victor Cotrim. Citado 2 vezes nas páginas 30 e 66.
- ONFIT, E. FARIA, Silvia Leite CM. *Entrevista com nutricionista Silvia*. 2015. Brasília. Entrevista concedida a Thiago Bernardes e Victor Cotrim. Citado 5 vezes nas páginas 38, 39, 67, 110 e 111.
- PEREIRA, T. F. C. Obesidade: A epidemia do século xxi? *Psicologia.pt. O portal dos psicólogos*, 2007. Universidade de Trás-os-Montes e Alto Douro, Portugal. Disponível em: <<http://www.psicologia.pt/artigos/textos/TL0208.pdf>>. Citado na página 31.
- PHILIPPI, S. T. *Pirâmide dos alimentos: Fundamentos básicos da nutrição*. [S.l.]: Manole, 2008. Citado 2 vezes nas páginas 33 e 34.
- PINHEIRO, J. M. dos S. *Da Iniciação Científica ao TCC: Uma abordagem Para os Cursos de Tecnologia*. [S.l.]: Editora Ciência Moderna, 2010. Citado na página 55.
- SAÚDE, M. da; BÁSICA, S. de Atenção à Saúde Departamento de A. *Guia alimentar para a população brasileira*. Editora MS, 2014. Disponível em : <<http://portalsaude.saude.gov.br/images/pdf/2014/novembro/05/Guia-Alimentar-para-a-pop-brasiliera-Miolo-PDF-Internet.pdf>>. Data de acesso: 1 de Abril de 2015. Disponível em: <<http://portalsaude.saude.gov.br/images/pdf/2014/novembro/05/Guia-Alimentar-para-a-pop-brasiliera-Miolo-PDF-Internet.pdf>>. Citado na página 31.
- SCHWABWER, K.; SUTHERLAND, J. *Um guia definitivo para o Scrum : As regras do jogo*. 2013. Data de acesso: 18 de Abril de 2015. Citado 5 vezes nas páginas 15, 45, 57, 58 e 59.
- SICHIERI, R. et al. Recomendações de alimentação e nutrição saudável para a população brasileira. *Arquivo Brasileiro Endocrinologia Metab [online]*, v. 44, 2000. Citado na página 31.
- SILVA, M. A. da. *Prova de conceito (PoC) em projetos*. 2014. Data de acesso: 15 de Abril de 2015. Disponível em: <<http://pmkb.com.br/artigo/prova-de-conceito-poc-em-projetos/>>. Citado na página 56.

SILVA, S. M. C. S. da; MURA, J. D. P. *Tratado de alimentação, nutrição e dietoterapia*. [S.l.]: Roca, 2007. Citado 14 vezes nas páginas 35, 36, 37, 38, 39, 40, 103, 104, 105, 106, 107, 110, 111 e 112.

SKIENA, S. S. *The Algorithm Design Manual*. Second edition. [S.l.]: Springer, 2008. Citado na página 86.

TOSCANE, L. V.; VELOSO, P. A. S. *Complexidade de algoritmos: análise, projeto e métodos*. [S.l.]: Bookman, 2012. Citado 2 vezes nas páginas 42 e 44.

Universidade Católica de Brasília. *Brazilian Education Programm for iOS Development*. 2015. Distrito Federal. Citado na página 30.

VOLP, A. C. P. et al. Energy expenditure: components and evaluation methods. *Nutrición Hospitalaria*, 2011. Citado 2 vezes nas páginas 38 e 110.

ZIVIANI, N. *Projeto de algoritmos com implementações em Pascal e C*. [S.l.]: Cengage Learning, 2011. Citado na página 43.

Apêndices

APÊNDICE A – Referencial Teórico

Apêndice voltado para o capítulo de referencial teórico.

A.1 Nutrição

Esta seção apresenta todo estudo relacionado ao domínio conexo de nutrição.

A.1.1 Métodos Retrospectivos

Os métodos retrospectivos mais utilizados para retenção e obtenção desses dados alimentares são o recordatório alimentar de 24h, QFCA¹ e o Histórico Dietético.

Segundo (SILVA; MURA, 2007) o recordatório alimentar é rápido e fácil de ser administrado, uma vez que o entrevistador pede ao paciente para que se recorde o que comeu nas últimas 24h. O entrevistado terá que fornecer informações sobre quando comeu/bebeu; o que comeu/bebeu; como foi o preparado o alimento/bebida; quanto foi comido/bebido, ficando a cargo do entrevistador a identificação destas informações. Algumas regras para a utilização desta método:

- Não avisar o paciente antecipadamente, pois o mesmo pode ter a intenção de mudar os alimentos já consumidos;
- Realizar este recordatório mais de uma vez, pois existe a possibilidade do paciente recordar um dia atípico;
- Questionar o paciente quanto a ingestão de bebidas alcoólicas e lanches extra ao longo do dia, como por exemplo: balas, chicletes, suplementos nutricionais;

(MAHAN; ESCOTT-STUMP, 2010) diz que este método não é indicada para além das 24h consecutivas, como por exemplo para 3 dias consecutivos, pois vale ressaltar o viés da capacidade de recordação do paciente. O uso concomitante deste método é mais válido do que o uso corrente, pois assim o viés da memória será amenizado melhorando a precisão dos dados. Porém é necessário tomar cuidado ao se usar concomitantemente pois a regra de não avisar o paciente pode ser violada sem acidentalmente, uma vez que a entrevista ocorra regularmente.

Segundo (SILVA; MURA, 2007) o QFCA consiste basicamente em uma lista de alimentos ou grupo de alimentos para quais o entrevistado (em caso auto-administrativo)

¹ Questionário de Frequência de Consumo Alimentar

ou o entrevistador deva anotar a frequência na qual os alimentos são consumidos em unidades de tempo. As perguntas podem ser abertas ou fechadas. A intenção do QFCA é o de saber o que foi ingerido e o quando foi ingerido a cerca daquele alimento, e para a obtenção de dados fidedignos é necessário explicar ao entrevistado deva proceder ao responder o questionário. É um método relativamente rápido, uma vez que leva entre 20 a 60 minutos², contra 10 a 30 minutos do recordatório alimentar de 24h. Porém as informações providas de múltiplos recordatórios é muita mais difícil de obtenção, uma vez que o tempo e disponibilidade do entrevistado devem ser levado em conta, fazendo o QFCA ser a melhor opção.

E o último método é o Histórico Dietético, que consiste em uma entrevista feita com o paciente com o intuito de prover informações sobre seus hábitos alimentares passados e presentes de forma mais detalhada possível, permitindo assim a obtenção de informações alimentares sobre: preferências, aversões, hábitos, intolerâncias, crenças e tabu, apetite, padrões de refeições - desde horários a locais - e hábitos de atividades físicas. Este método ainda pode ser usado em conjunto com o recordatório de 24h e o QFCA. Pelo fato desta entrevista ser muito extensa "... o entrevistado pode fazer julgamentos sobre a sua alimentação. Nesse caso as respostas poderão refletir o que pensam consumir (ou que gostariam que o entrevistador pensasse que consumiam) e não o que realmente ingerem."(SILVA; MURA, 2007).

As vantagens e desvantagens entre estes métodos podem ser observadas na tabela 21.

Juntamente desses métodos pode ser utilizado algumas técnicas de Estimativa do Tamanho das porções, que é o maior desafio na coleta de dados para a avaliação dietética.

Uma das técnicas mais usadas é o Registro Fotográfico, que consiste em manuais contendo fotos de diferentes porções do mesmo alimento, sendo usados no momento da entrevista com paciente para classifica-los. Com esta técnica as porções maiores são subestimadas e as menores superestimadas pelos entrevistados³, mas com a vantagem de saber aproximadamente a quantidade ingerida de cada alimento.

A.1.2 Avaliação Antropométrica

A Avaliação Dietética tem o objetivo de descobrir e analisar o que o paciente do profissional da saúde está consumindo. Consoante a (SILVA; MURA, 2007) a avaliação Antropométrica tem como ímpeto a análise do corpo deste paciente. O resultado desta análise possibilita a construção de índices e indicadores antropométricos para a avaliação do indivíduo em relação a população na qual este está inserido, exemplo disso é a com-

² Em estudo de Yanek et al, citado em (SILVA; MURA, 2007)

³ Em experimento de Nelson et al investigando mulheres e homens de 18 a 90 anos que se serviram de 4 a 6 alimentos por refeição, segundo (SILVA; MURA, 2007, p. 151)

Tabela 21: Vantagens e Desvantagens entre métodos retrospectivos

| Método | Vantagem | Desvantagem |
|---------------------------------|---|---|
| Recordatório de 24h | Rápido Prático | Não reflete a ingestão usual ou entre finais de semana. É uma método que depende da memória do paciente Paciente precisa saber o tamanho das porções de cada alimento |
| Frequência de Consumo Alimentar | Método de baixo custo Relativamente rápido Excelente para obtenção de padrões de ingestão | Não fornece informações detalhadas sobre a quantidade consumida |
| Histórico Dietético | Obtenção de dados mais precisos como: preferências, hábitos e padrões de refeição | Extensa entrevista Entrevistado pode se levar a fazer julgamentos sobre sua alimentação |

paração dos índices antropométricos de uma criança em relação a população para saber se a criança possui crescimento satisfatório

A definição de antropometria, segundo ([HIRSCHBRUCH, 2014](#)) é:

A antropometria é o emprego da relação das mensurações físicas de um indivíduo com um padrão de referência que determina seu crescimento e desenvolvimento, e é um componente da avaliação nutricional. [...]. As medidas antropométricas mais comumente utilizadas são: massa corporal, estatura, dobras cutâneas, circunferências corporais e bioimpedância.

Conforme ([MAHAN; ESCOTT-STUMP, 2010](#)) estas medidas antropométricas podem ser usadas para avaliar o excesso nutricional ou a desnutrição. Elas podem ser usadas para monitorar os efeitos da intervenção alimentar, uma vez que os indivíduos que realizam estas medições devem ser treinados apropriadamente, pelo menos 20 sessões práticas podem ser necessárias para se tornar proeficiente⁴. Estes dados providos das medidas são mais valiosos se forem precisos e registrados ao longo do tempo.

O autor ainda afirma que os padrões de referências de peso e altura usados hoje são baseados numa amostra estatística da população dos Estados Unidos. Uma vez que as medidas de altura são valiosas quando usadas conjuntamente com outras técnicas de medição antropométrica, podemos diferenciar 2 formas de se fazer a aferência da altura: direto e indireto. No método direto o indivíduo deve conseguir ficar em pé com a coluna

⁴ Referência a (Lee e Nieman,2000)

ereta enquanto a sua altura é aferida utilizando alguma haste de medida. No método indireto o indivíduo não consegue ficar em pé e a aferência da altura deva ser com o mesmo deitado ou sentado utilizando uma fita métrica. Já para o peso 2 conceitos podem ser aplicados, o PU (Peso corporal Usual) que é o peso do indivíduo no dia-a-dia que somente depende da memória do indivíduo, e o peso atual que é a medida real obtida em exame.

Para determinar se o peso de um adulto é apropriado para sua altura, o cálculo do IMC (Índice de Massa Corporal) deve ser feito e avaliado.

Também conhecido como índice de Quetelet o IMC é uma medida válida para o estado nutricional no qual depende somente das medidas de altura e peso. O resultado do cálculo do IMC pode indicar desnutrição ou sobrepeso e pode ser calculado com a formula [A.1](#).

$$\begin{aligned} P &= \text{Peso}(kg) \\ A &= \text{Altura}(m) \\ IMC &= \frac{P}{A^2} \end{aligned} \tag{A.1}$$

Em consonância com ([SILVA; MURA, 2007](#)) os valores do IMC, ou classificação do estado nutricional, são padronizados pela OMS (Organização Mundial da Saúde) e podem ser consultados em uma tabela para saber qual é o nível de desnutrição ou sobrepeso de uma pessoa. A tabela [22](#) mostra os valores padronizados para cada nível de classificação do estado nutricional.

Tabela 22: Classificação Estado nutricional de acordo com IMC

Fonte: ([SILVA; MURA, 2007](#))

| Classificação Estado Nutricional | IMC |
|----------------------------------|----------------------------|
| Desnutrição Grave | < 16 |
| Desnutrição Moderada | ≥ 16 e $\leq 16,99$ |
| Desnutrição Leve | ≥ 17 e $\leq 18,49$ |
| Eutrófico | $\geq 18,5$ e $\leq 24,99$ |
| Sobrepeso | ≥ 25 e $\leq 29,99$ |
| Obesidade grau 1 | ≥ 30 e $\leq 34,99$ |
| Obesidade grau 2 | ≥ 35 e $\leq 39,99$ |
| Obesidade grau 3 | ≥ 40 |

O IMC serve para saber qual o peso corporal apropriado em relação a faixa de valores na qual o IMC foi calculado, e segundo ([MAHAN; ESCOTT-STUMP, 2010](#)) isto pode ser calculado em 2 etapas:

- Etapa 1 - Cálculo do IMC atual Exemplo: Mulher com 1,72 m de altura e peso de 84 kg

$$IMC = \frac{P}{A^2} \quad (A.2)$$

$$IMC = \frac{84}{1,72^2} \quad (A.3)$$

$$IMC = 28,94 = \textit{Sobrepeso} \quad (A.4)$$

- Etapa 2 - Cálculo do limite de peso apropriado de acordo com o IMC entre 18,5 e 24,9

$$(18,5) \cdot (1,72^2) = 54,8kg \quad (A.5)$$

$$(24,9) \cdot (1,72^2) = 73,8kg \quad (A.6)$$

Limites de peso adequados: 54,8 kg a 73,8 kg

O IMC também pode ser usado para triagem do excesso de peso em crianças e adolescentes utilizando-se algoritmos adequados, segundo (AUTORES, 2009). O autor ainda cita que atualmente o IMC é aceito internacionalmente como meio de identificação de sobrepeso e obesidade independente do sexo ou idade, e ainda pode ser usado como uma ferramenta para identificar indivíduos que possuem um maior risco de desenvolver doenças: diabetes tipo 2, cardíacas e câncer, uma vez que o IMC aumenta o risco aumenta de forma gradual.

Existem algumas ressalvas que (HIRSCHBRUCH, 2014) quanto ao uso do IMC:

O uso do peso(massa corporal) e IMC como únicos parâmetros deve ser visto com ressalvas, uma vez que indivíduos fisicamente ativos estão sujeitos ao aumento de massa magra, que pode resultar em aumento ou manutenção do peso. Um IMC acima do recomendado não significa excesso de gordura, por isso o peso e o IMC devem ser utilizados em conjunto com métodos de avaliação de gordura corporal, para classificação dos indivíduos em relação ao percentual de gordura.

Além das medidas de peso, altura e IMC, ainda existe a aferência das dobras cutâneas e circunferência. As dobras cutâneas e circunferências são consideradas, dentro das medidas antropométricas, como composição corporal.

As dobras cutâneas "Expressam a quantidade de tecido adiposo corporal e, conseqüentemente, as reservas de energia e o estado nutricional atual. Porém, refletem apenas a disposição da gordura localizada na região subcutânea.", definição de (SILVA; MURA, 2007). Além disso as dobras cutâneas são utilizadas com a suposição de que 50% da gordura corporal seja subcutânea, segundo (MAHAN; ESCOTT-STUMP, 2010) em seu livro.

Para (SILVA; MURA, 2007) as medidas de circunferência,

Não fornecem medidas específicas de composição corporal, mas são úteis para quantificar diferenças interindividuais, permitindo identificar, dentro de uma população, indivíduos com maior risco para desnutrição e diferenças intra-individuais durante o acompanhamento nutricional.

O autor ainda enfatiza que a relação da circunferência abdominal e do quadril fornecem informações que dizem a respeito ao risco de saúde. Segundo (MAHAN; ESCOTT-STUMP, 2010), estas medidas quando fora de proporção são indicativo de risco a doenças associadas à obesidade. As medidas da circunferência da cintura eram usadas com frequência, e esta medida juntamente com a medida da circunferência do quadril se chama RCQ⁵, e eram utilizada para detectar possíveis sinais de excesso de deposição de gordura em indivíduos com infecção por vírus da imunodeficiência humana.

A.1.3 Avaliação Bioquímica

A utilização de dados laboratoriais para avaliação de bioquímica do paciente é fundada no fato de saber qualquer alteração clínica ou antropométrica antes que as mesmas se manifeste no paciente (MAHAN; ESCOTT-STUMP, 2010). Os dados colhidos trazem informações acerca da disponibilidade de nutrientes existente em fluidos e tecidos biológicos, sendo possível analisar os níveis destes nutrientes quanto a deficiência, quando níveis de nutrientes estão baixos, e toxicidade, quando os níveis de nutrientes estão altos.

O autor ainda cita:

Os testes laboratoriais são solicitados para diagnosticar doenças. avaliar os planos de tratamento, monitorar a eficiência das medicações e avaliar a terapia clínica nutricional(TNC). Lesões ou doenças crônicas agudas podem desencadear alterações dramáticas nos resultados dos testes laboratoriais, incluindo rápida deterioração do estado nutricional. Entretanto, doenças crônicas que se desenvolvem lentamente ao longo do tempo também influenciam esses resultados, mas não está claro o quanto as escolhas de estilo de vida, a genética ou combinação de fatores adicionais contribuem para a condição atual.

⁵ Razão entre a circunferência de cintura e quadril

A.1.4 Avaliação Nutricional

A avaliação nutricional, para este TCC, é composta pelos 7 passos seguintes:

Passo 1: Estudo das Patologias e Seu Estado Nutricional

Estudar o estado atual do paciente é essencial para saber qual a sua melhor adaptação quanto a dieta que será gerada.

Ao se saber qual patologia o paciente possui é possível fazer um estudo da mesma com o intuito de saber quais as restrições alimentares possíveis de serem feitas. No livro do autor ([MAHAN; ESCOTT-STUMP, 2010](#)) existem apêndices relacionando patologia com restrições alimentares que pode ser usado como material de suporte para este estudo. Melhorias da dieta também podem ser realizadas, uma vez que o conhecimento do patologia pode ajudar em identificar quais podem ser os possíveis nutrientes em falta ou excesso no organismo do paciente.

Geralmente no estudo das patologias existem certezas que podem ser afirmadas na maioria dos casos: existe a desnutrição ou excesso de algum nutriente no sistema do paciente. E isso nos leva ao passo seguinte.

Passo 2: Análise dos Resultados das Avaliações Dietéticas

A avaliação dietética refere-se aos métodos retrospectivos e prospectivos que são feitos para saber qual é a alimentação atual do paciente.

Um estudo do resultado desta avaliação é feito levando em conta critérios quantitativos e qualitativos. Uma vez que as medidas caseiras dos alimentos citados nos resultados desta avaliação são transformados para gramas⁶, a quantidade de macros e micros nutrientes que são ingeridas pelo paciente podem ser calculadas, inferindo daí se o paciente possui uma desnutrição ou excesso de nutrientes.

Com base nestes dados algumas patologias podem, inclusive, ser explicadas utilizando da análise dos critérios quantitativos⁷. Pois a baixa quantidade de nutrientes⁸ podem causar a deficiência na formação de outros nutrientes, que consequentemente influenciam na formação de substâncias essenciais para o corpo. A alta quantidade de nutrientes⁹ em alguns casos podem causar intoxicação do organismo, vide ([MAHAN; ESCOTT-STUMP, 2010](#), Cap.15, figura 15.2).

Os critérios qualitativos¹⁰ são utilizados para saber se a alimentação atual do paciente está ajudando ou não na saúde do paciente.

Passo 3: Cálculo das Necessidades

⁶ Medida usual feita pelos nutricionistas

⁷ Contagem da quantidade macros e micros

⁸ Desnutrição de nutriente

⁹ Excesso de nutriente

¹⁰ Saber se a alimentação atual é rica ou pobre em questão de nutrientes

Neste passo são feitos os cálculos, baseados nas medidas antropométricas, das necessidades calóricas e de macro e micro nutrientes. O objetivo deste passo é tentar manter aquilo que o paciente ingere habitualmente com a nova dieta que será montada, assim esta adaptação de dieta será melhor aceita.

O nutricionista pode manter alguns alimentos que o paciente habitualmente ingere, mas se o mesmo faz mal não é indicado fazer a retirada completa do alimento e sim fazer a diminuição do mesmo para não haver um impacto tão grande em relação a dieta.

Os nutricionistas usam as chamadas DRIs¹¹ como referência de valores. A concepção de DRI é "um termo global concebido para englobar os quatro tipos específicos de nutrientes (ingestão adequada [IA], necessidade média estimada [EAR], cota dietética recomendada [EAR] e grau de ingestão máxima tolerável[IM])" (MAHAN; ESCOTT-STUMP, 2010).

Para (SILVA; MURA, 2007) ainda existe mais uma referência de valor chamada de estimativa das necessidades de energia (EER). Ainda é dito que destas 5 medidas, 4 são utilizadas para prescrição das dietas : EER, RDA, EAR e AI. A medida de EAR somente deve ser usada se conhecido o desvio-padrão do grupo (SILVA; MURA, 2007, p. 385).

Para (VOLP et al., 2011) o cálculo da EER é baseada no Gasto Energético Basal (GEB). Uma vez feito o cálculo do GEB, o fator de atividade física é aferido na tabela 1 conforme a quantidade de dias na semana especificados pelo paciente e em seguida o valor de EER é ajustado conforme tabela de objetivos (ONFIT, 2015, Entrevista nutricionista Silvia).

Para dar início ao passo seguinte é necessário o cálculo de algumas variáveis como o valor de GEB e EER e as quantidades diárias de carboidratos, proteínas, fibras e lipídeos.

O primeiro cálculo realizado é o de gasto calórico do paciente. O cálculo pode ser feito usando a fórmula A.7 para homens e a fórmula A.8 para mulheres (VOLP et al., 2011, p. 437 Tabela 2).

$$\begin{aligned} P &= \text{Peso}(kg) \\ h &= \text{Altura}(cm) \\ t &= \text{idade}(Anos) \end{aligned} \tag{A.7}$$

$$GEB = 66.5 + (13.75 \cdot P) + (5.003 \cdot h) - (6.775 \cdot t)$$

$$GEB = 655.1 + (9.563 \cdot P) + (1.85 \cdot h) - (4.676 \cdot t) \tag{A.8}$$

¹¹ Ingestão dietética de referência, ou no original *Dietary Reference Intake*

O nível de atividade física pode ser calculado com base na tabela 23 utilizando o Fator Atividade Física (FAF), sendo os valores adquiridos de forma empírica conforme o nível de atividade física executado durante a semana (ONFIT, 2015, Entrevista nutricionista Silvia).

Tabela 23: Valor de Atividade física em relação ao nível de atividade na semana

Fonte: (ONFIT, 2015, Entrevista nutricionista Silvia)

| Nível atividade física(dias/semana) | Valor FAF | Classificação Paciente |
|-------------------------------------|-----------|------------------------|
| 0 | 1 | Sedentário |
| 1-2 | 1.2 | Pouco Sedentário |
| 3-4 | 1.375 | Pouco Ativo |
| 4-5 | 1.55 | Moderado |
| 6-7 | 1.725 | Ativo |

Utilizando-se das equações A.7 e A.8 e tabela 23, a necessidade energética pode ser calculada mediante equação A.9.

$$EER = GEB * FAF \quad (A.9)$$

Conforme tabela 24, o ajuste da EER é feito em consonância com os objetivo do paciente.

Tabela 24: Cálculo de ajuste empirico mediante objetivos dos pacientes

Fonte: (ONFIT, 2015, Entrevista nutricionista Silvia)

| Objetivos | Fórmula de Ajuste |
|-----------------|-------------------|
| Perder Peso | EER - 500 |
| Ganhar Músculos | EER + 500 |
| Manter Peso | EER |

Ao que diz respeito aos cálculos, (SILVA; MURA, 2007) relacionou o Institute of Medicine quanto as referências relacionadas aos macronutrientes em uma faixa de distribuição aceitável. A tabela 25 relaciona os valores adequados para cada macro, em porcentagem, utilizando como referência o EER.

Tabela 25: Faixa de valor aceitável diário de macronutrientes

| Macronutriente | Faixa de valor aceitável |
|----------------|--------------------------|
| Proteína | 10 a 35% |
| Lípides | 20 a 35% |
| Carboidratos | 45 a 65% |

Para as fibras uma tabela foi concebida, levando em conta estudos epidemiológicos prospectivos (SILVA; MURA, 2007, p. 386). A tabela 26 mostra os valores correspondentes a ingestão diária, dividindo quanto ao gênero e idade.

Tabela 26: Ingestão Adequada diária de fibras

| Categoria | Ingestão Adequada (g/dia) |
|-----------------|---------------------------|
| Homens | |
| 19 - 50 anos | 38 |
| 51 anos ou mais | 30 |
| Mulheres | |
| 19 - 50 anos | 25 |
| 51 anos ou mais | 21 |

Sabendo-se as quantidade de cada macro nutriente que é recomendado para o paciente ingerir, agora é possível realizar o próximo passo, a montagem da dieta.

Passo 4: Montagem da Dieta

A montagem da dieta do paciente respeita o estudo das patologias feita no passo 1, quanto a melhorias e restrições.

Para montagem da dieta são utilizados guias alimentares e normas de saúde, levando em conta as preferências, restrições e melhorias da dieta do paciente. (SILVA; MURA, 2007, p. 174) diz,

No intuito de garantir uma alimentação equilibrada e saudável, a pirâmide foi a representação gráfica escolhida para dispor os alimentos separados em níveis e agrupados por tipos de nutrientes.

A pirâmide alimentar brasileira reflete os grupos alimentares que foram dispostos em ordem de quantidade de porções. A tabela 27 mostra a distribuição da quantidade de porções em relação ao calor calórico cálculo da dieta, o valor do EER.

Tabela 27: Valores em porções

| Grupos | Dieta 1600kcal | Dieta 2200kcal | Dieta 2800kcal |
|--------------------------------------|----------------|----------------|----------------|
| Arroz, massa, pão, batata e mandioca | 5 | 7 | 9 |
| Verduras e legumes | 4 | 4,5 | 5 |
| Frutas | 3 | 4 | 5 |
| Leite, queijo e iogurte | 3 | 3 | 3 |
| Carnes e ovos | 1 | 1,5 | 2 |
| Feijões | 1 | 1 | 1 |
| Óleos e gorduras | 1 | 2 | 2 |
| Açúcares e doces | 1 | 1,5 | 2 |

Os valores para cada porção leva em conta o tipo de alimento para ser quantificado.

Os nutricionistas utilizam software para montagem destas dietas, mas muitas vezes este não resolve de maneira satisfatória a forma como o nutricionista pensa que a dieta

poderia ser gerada ou montada, abandonado assim estes software e indo para algo mais artesanal como planilhas.

Passo 5: Montagem da Lista de Substituição

Uma lista de substituição é formada por vários alimentos que podem ser usados como substitutos para os alimentos originais da dieta original.

Uma lista de substituição é fornecida quando o paciente possui somente 1 dieta montada. O valor perfeito para a quantidade de dietas montadas para o paciente é 3. A dieta é montada com o tempo de vida de 1 semana. Uma vez que o paciente possua somente 1 dieta, então a necessidade de uma lista de substituição é grande pois a mesma possibilita a variabilidade de alimentos que garante uma maior probabilidade de ingestão de nutrientes variados. Uma vez que o paciente possua 3 dietas, então a necessidade de uma lista de substituição não é tão grande, pois a com 3 dietas a mão existe a garantia de 3 semana de diferentes tipos de ingestão de alimentos variados.

Passo 6: Conversão para Medidas Caseiras

Os alimentos aqui são convertidos de porções para medidas caseiras, facilitando assim o manuseio das dietas para o paciente são mais condizentes com a sua realidade.

Passo 7: Entrega de dieta e Educação Nutricional

A entrega da dieta é o último passo, uma vez que é explicado o porque da mudança dos alimentos, quais seus objetivos em relação a saúde do paciente e a importância desta dieta para a saúde.

APÊNDICE B – Histórias de Usuário

Este capítulo tem como intuito mostrar as tarefas de todas as Histórias de Usuários utilizadas pelo o autor deste TCC.

Tabela 28: Histórias de Usuário e suas respectivas tarefas

| ID | Descrição | Tarefas |
|----|--|--|
| 2 | Eu como desenvolvedor quero informar os dados do usuário para saber a quantidade de calorias, carboidratos, lipídios e proteínas que ele deve ingerir diariamente.Dados: Gênero, Idade, Altura, Peso | Implementar fórmula em C. |
| 4 | Eu como integrante do OnFit gostaria que o usuário assine um termo de uso relacionado ao uso da sugestão de alimentos para não ter problemas legais | Utilizar termos de compromisso do OnFit como base. Pegar informações básicas da Nutricionista e inserir no documento. Entregar para assinatura. Anexar em TCC. |
| 6 | Eu como desenvolvedor quero integrar aos alimentos as possíveis porções existentes de cada um para formatar os itens a serem usados no knapsack. | Pegar informações referentes as porções de cada alimento na base dados do OnFit. Adicionar valor de caloria de cada alimento. 1g Ptn = 4 calorias, 1g Carbo = 4 calorias e 1g Fat = 9 calorias. |

| | | |
|---|---|---|
| | | Implementar correta quantidade de cada alimento conforme fator de conversão. |
| 7 | Eu como desenvolvedor quero separar a quantidade de nutrientes a serem ingeridos diariamente em 6 refeições diárias para melhor aplicar o algoritmo de recomendação de alimentos. | Implementar separação de nutrientes pela quantidade de refeições conforme anotações da nutricionista. |
| 8 | Eu como desenvolver quero aplicar o algoritmo knapsack nos alimentos de cada refeição diária para saber quais são os melhores alimentos a serem utilizados pelo usuário. | <p>Montar knapsack conforme foi conversado com professor Edson.</p> <p>Otimizar KnapSack para utilizar com 2 linhas.</p> <p>Conversar com Edson a melhor forma de utilizar a liberação da primeira linha. Obs: Conversa e dúvidas foram tiradas com o Mauricio.</p> <p>Fazer contagem da quantidade de bytes alocados por linha.</p> <p>Conversar com Maurício a respeito do valor de cada item, falando sobre aquilo que o Edson comentou sobre as múltiplas formas de comparação de fórmula..</p> <p>Conversar com o Edson, melhor forma de recuperar os itens do knapsack.</p> <p>Separar itens por grupos para cada refeição.</p> |

| | | |
|-------|---|---|
| 9 | Eu como desenvolvedor quero utilizar do objetivo e da quantidade de dias de exercícios na semana do usuário para refinar a quantidade de nutrientes e calorias diários. Objetivos: Perder peso, ganhar massa muscular e manter peso. Exercícios na semana: 0, 1-2, 3-4, 4-5,6-7 dias na semana. | <p>Inserir Exercícios conforme fórmula explicada pela nutricionista.</p> <p>Inserir Objetivos conforme fórmula explicada pela nutricionista.</p> |
| 10 | Eu como desenvolvedor desejo integrar o banco de dados para utilizar os alimentos no algoritmo de recomendação alimentar | <p>Estudar como acessar banco de dados em C, caso não funcione, estudar outra forma de pegar essas informações.</p> <p>Implementar integração com SQLite BD.</p> <p>Fazer nova listagem de itens com informação de grupo alimentar.</p> |
| 11 | Eu como desenvolvedor quero entregar a lista de alimentos para profissional da nutrição para refinar a busca de itens do knapsack | <p>Levantamento dos cenários de uso.</p> <p>Complementar estratégia de busca com novas pontuações.</p> <p>Envio de itens encontrados para análise.</p> |
| TS-01 | Eu como Engenheiro de Software quero fazer medições do tempo de execução do software e plotar o gráfico para estas medições com o intuito de documentar a análise e implementar melhorias | <p>Medir tempos de execução do knapsack sem otimização em 2 linhas.</p> |

| | | |
|-------|--|--|
| | | <p>Medir tempos de execução do knapsack com otimização em 2 linhas.</p> <p>Fazer análise como base de comparação, documentar análise em TCC e implementar melhorias.</p> |
| TS-02 | Eu como Engenheiro de Software quero recuperar todos os itens analisados pelo knapsack utilizando a metodologia imposta pelo TCC para melhorar a implementação e otimização do algoritmo | <p>Pesquisa de como recuperar os itens de knapsack utilizando otimização de 2 linhas..</p> <p>Estudo de viabilidade de solução.</p> <p>Implementação.</p> <p>Salvar lista de alimentos em .CSV.</p> <p>Integração OnFit.</p> |