



Desenvolvimento de um sistema para criação de planos alimentares baseado em orçamento e preferência de refeições*

João Pedro Teixeira Dias¹

Octavio Augusto Pereira Martins²

Magali Rezende Gouvea Meireles³

Resumo

O presente trabalho apresenta o desenvolvimento de um Sistema para dispositivos móveis, voltado para o âmbito nutricional, dando a possibilidade de qualquer pessoa poder montar um plano alimentar. O objetivo do trabalho é facilitar o planejamento, economia e a praticidade ao seguir uma dieta. O usuário deverá informar para o sistema, sua idade, peso, altura, valor limite de gastos, e também se o objetivo é a perda de peso, manutenção ou ganho de massa. Com base nessas informações, será fornecida uma dieta que será calculada utilizando a taxa metabólica basal do usuário, e os macronutrientes de cada alimento. Também será possível fazer alterações no plano alimentar conforme a preferência, sendo apresentada uma lista com diversos alimentos e refeições possíveis para realização da troca. Em relação aos custos, cada item da lista, será exibido com seu preço médio, preço este que é calculado levando em conta o valor de cinco diferentes estabelecimentos.

Palavras-chave: Nutricional, Planejamento, Praticidade, Plano Alimentar, Gastos, Orçamento, Taxa Metabólica Basal, Preferência.

* Artigo apresentado à Revista Abakos

¹ Bacharelado em Sistemas de Informação pela PUC Minas, Brasil, 1272244@sga.pucminas.br

² Bacharelado em Sistemas de Informação pela PUC Minas, Brasil, oapmartins@sga.pucminas.br

³ Doutora em Ciência da Informação pela UFMG, Brasil, magali@pucminas.br

Abstract

The present work presents the development of a system for mobile devices, focused on the nutritional field, giving the possibility for anyone to create a dietary plan. The objective of the work is to facilitate planning, savings and practicality when following a diet. The user must inform the system of their age, weight, height, spending limit, and also whether the objective is weight loss, maintenance or mass gain. Based on this information, a diet will be provided that will be calculated using the user's basal metabolic rate and the macronutrients of each food. It will also be possible to make changes to the eating plan according to preference, with a list of different possible foods and meals to make the change. Regarding costs, each item on the list will be displayed with its average price, a price that is calculated taking into account the value of five different establishments.

Keywords: Nutritional, Planning, Practicality, Diet Plan, Expenses, Budget, Basal Metabolic Rate, Preference.

1 INTRODUÇÃO

Aplicações que buscam auxiliar o usuário em seus objetivos e necessidades de saúde, estão cada vez mais populares e numerosos. Além disso, a nutrição vem sendo cada vez mais exaltada e reconhecida por ser uma das maiores contribuintes no quesito prevenção a complicações de saúde. Este trabalho visa unir os conhecimentos de desenvolvimento de aplicações, e os métodos nutricionais de construção de plano alimentar, visando auxiliar o usuário final a encaixar um plano alimentar ao orçamento disponível para o mesmo.

Diante disso, abre-se uma oportunidade de inovar e trazer ao mercado, uma aplicação que irá fornecer um plano alimentar, onde o usuário poderá substituir as próprias refeições ou alimentos, por outras de seu próprio interesse. Caso um item do plano alimentar seja selecionado, irá ser exibido uma lista com possíveis alimentos disponíveis para troca. A troca não será aleatória por qualquer refeição, será realizada com base no horário, alimentos já inclusos na dieta e preço dos alimentos. O valor dos alimentos, e do plano alimentar completo, serão exibidos a todo momento no aplicativo, dando ao usuário a possibilidade de ter maior controle em relação aos gastos em sua dieta.

O objetivo deste trabalho é desenvolver um aplicativo que busca facilitar a vida de pessoas que pretendem fazer dieta, dando maior controle sobre os gastos com cada alimento e refeição, oferecendo planos alimentares que condizem com os gostos e vida financeira.

As próximas seções estão organizadas como descrito a seguir. A Seção 2 apresenta o referencial teórico, contendo informações sobre dieta e seus componentes, arquitetura e tecnologias adotadas no sistema, e artigos que foram relacionados com o presente trabalho. A Seção 3 apresenta a metodologia...

2 REFERENCIAL TEÓRICO

As próximas subseções apresentam conceitos relacionados a Dieta e seus componentes, Arquitetura de Sistemas, e Tecnologias e Ferramentas para desenvolvimento do aplicativo.

2.1 Dieta e seus componentes

Uma dieta é definida por tudo aquilo que um indivíduo se alimenta. É feita uma distribuição de macro e micronutrientes, de modo que idealmente, cada pessoa terá uma distribuição específica visando atingir um objetivo, levando em conta a sua taxa metabólica basal.

A taxa metabólica basal (TMB) é a quantidade de energia necessária para a manutenção das funções vitais do organismo quando em repouso. Ela é medida em calorias, que é a energia extraída pelo nosso corpo a partir dos macronutrientes (carboidratos, proteínas e gorduras totais) (ANTUNES et al., 2015).

Os carboidratos, proteínas e lipídios, por serem os nutrientes de maior contribuição energética para uma dieta, se enquadram na classe dos macronutrientes. Uma distribuição adequada desses nutrientes é de extrema importância para a nutrição de um paciente. Essa distribuição é feita em relação ao percentual de calorias que o indivíduo necessita, portanto, indivíduos diferentes, com necessidades/objetivos diferentes, terão diferentes distribuições de macronutrientes em seu plano alimentar.

As vitaminas e minerais são considerados os micronutrientes. Isso se dá pois esses compostos não apresentam relevante potencial calórico e energético para uma dieta, além disso são necessários para as reações químicas que ocorrem no corpo. Os micronutrientes podem ser encontrados em diversos alimentos, sendo mais abundantes em frutas legumes e verduras (MAGNO et al., 2018).

Uma dieta pode ser definida por tudo que é ingerido por uma pessoa, independente dos objetivos: ganhar massa, perder peso ou realizar a manutenção do peso. Existem diversos tipos de dietas com propósitos diferentes, mas para o presente trabalho, foram abordadas os três tipos.

O ganho de massa muscular, de maneira simples, é a necessidade de um indivíduo adquirir mais calorias do que consegue gastar durante o dia. As calorias adquiridas, precisam estar acima de sua taxa metabólica basal (MENON; SANTOS, 2012). Para auxiliar na construção da massa muscular, alimentos ricos em carboidratos e proteínas são aliados. A média de ingestão de proteína em pessoas regularmente ativas, podem variar entre 1,5 ou 2,0 por quilo corporal (MENON; SANTOS, 2012).

A dieta de perda de peso segue no sentido contrário de pessoas que estão buscando ganho de massa muscular. Para reduzir o peso corporal é necessário adquirir menos calorias do que é possível gastar durante o dia. As calorias adquiridas, precisam estar abaixo de sua taxa metabólica basal (JOB; OLIVEIRA, 2019).

Uma dieta de manutenção de peso é quando não se busca ganho e nem perda de peso, apenas o controle. O consumo de calorias durante o dia, precisa ser próximo a taxa metabólica basal. Neste tipo de dieta, não é muito comum uma restrição muito grande de alimentos, nem o consumo elevado (JOB; OLIVEIRA, 2019).

2.2 Arquitetura de Sistemas

2.2.1 MVC

O MVC (*Model-View-Controller*) é um padrão de arquitetura criado por Trygve Reenskaug, em 1979. Esse padrão possibilita a divisão do projeto em camadas, viabilizando a independência e reutilização de componentes, e o desenvolvimento de um código organizado e enxuto (SILVA, 2012).

A composição do MVC é feita pelas camadas de Modelo, Visão e Controle. O Modelo

é responsável pelo acesso e manipulação dos dados dentro da aplicação. Ela se comunica diretamente com o controlador. A camada de Visão é a parte responsável pela interface do usuário. Nessa camada, os dados são requisitados ao Controlador que por sua vez requisitada ao modelo. O Controlador é a camada responsável intermediar a conexão entre o Modelo e a Visão.

2.3 Tecnologias e Ferramentas

As próximas subseções apresentam conceitos relacionados às tecnologias e às plataformas utilizadas para desenvolvimento do sistema.

2.3.1 Dart

Dart é uma linguagem desenvolvida pela Google, para criação de *softwares* multiplataforma. Seu lançamento ocorreu na GOTO Conference 2011, Dinamarca (RISSI; DALLILO, 2022).

Sua missão inicial era substituir o JavaScript como principal tecnologia presente nos navegadores, a ideia era oferecer uma alternativa mais moderna e robusta para o desenvolvimento *Web*. Um dos pontos que dificultaram sua adoção, foi o JavaScript já ser uma linguagem consolidada no mercado, e também a má fama do Google em descontinuar projetos.

O Dart é orientada a objetos, mas pode ser considerada flexível, pois permite que os programadores definam variáveis que aceitam todos os tipos de valores, chamadas de variáveis dinâmicas. A linguagem possui sintaxe com estilo baseado no C. Isso faz com que seja muito similar à linguagens como Java e C#.

2.3.2 Flutter

Flutter foi criada pelo Google como uma plataforma de desenvolvimento móvel de código aberto, também destinada a auxiliar os desenvolvedores na criação de interfaces para múltiplas plataformas. É construída na linguagem Dart, que é uma linguagem orientada a objetos que se compila em código nativo. É de fácil aprendizado para desenvolvedores provenientes de outras linguagens como Java e JavaScript, e oferece recursos como forte tipagem, genéricos e programação assíncrona (WU, 2018).

O Flutter é baseado no conceito de *widgets*, que são basicamente componentes *UI* que podem ser usados para criar interfaces de usuário complexas com apenas algumas linhas de código. Esse *framework* fornece um conjunto de *widgets* pré-construídos, mas também possibilita que os desenvolvedores façam seus próprios. Os *widgets* são classificados como *Stateless* e *Stateful widgets*, dependendo se eles gerenciam ou não algum estado. *Widgets Stateless* são aqueles que simplesmente exibem informações e não exigem nenhum dado mutável. Os *State-*

ful widgets, por outro lado, gerenciam o estado. Este estado pode ser interno ao próprio *widget*, ou pode vir de uma fonte externa. Para que um *Stateful Widget* possa mudar sua aparência em resposta a eventos, ele precisa ter acesso a um objeto imutável conhecido como uma instância *State* (WU, 2018; FENTAW, 2020).

Várias camadas compõem a estrutura Flutter (SHAH et al., 2019). A renderização da interface do usuário é de responsabilidade da camada *Widgets*. Ela usa uma abordagem declarativa para definir a interface de usuário. Isto significa que não é necessário escrever código imperativo para atualizar a interface de usuário. A camada *Engine*, por sua vez, está encarregada de coordenar a comunicação entre o *framework* e o código nativo da plataforma. Por fim, a camada de fundação fornece serviços fundamentais, incluindo rede, E/S e suporte de acessibilidade.

2.3.3 Node.js

Node.js (Node) é ambiente de execução em uma máquina virtual própria para interpretar e executar os scripts de forma autônoma. Esse ambiente utiliza a linguagem JavaScript e permite o desenvolvimento de aplicações web. Essa plataforma foi criada a partir do JavaScript Engine V8 do Google, visando viabilizar e facilitar a construção de aplicações (SOUSA, 2015).

Desenvolvido em 2009 pelo estadunidense Ryan Dahl, o Node visa alto desempenho, baixo consumo de memória e suportar processos de servidor de longa duração (SOUSA, 2015).

O Node opera em apenas uma única *thread*, diferente de outras plataformas de desenvolvimento. Para isso, é utilizada uma abordagem baseada em eventos de *input/output* não bloqueante (SOUSA, 2015).

2.3.4 Firebase

Firebase é uma plataforma de gerenciamento e desenvolvimento de dispositivos móveis, desenvolvida pela Firebase, Inc. em 2011 e adquirida pelo Google em 2014. A plataforma surgiu de uma necessidade dos desenvolvedores em gerenciar várias infraestruturas que faziam parte de um mesmo aplicativo. Com ela os desenvolvedores podem focar na criação de sua própria aplicação, poupando tempo e recursos (LI et al., 2018).

Atualmente o Firebase é dividido em 3 partes nomeadas de : *Analytics*; *Develop* e *Grow*.

O *Analytics* é a seção onde são disponibilizadas informações sobre o aplicativo em geral. É possível obter informações como: localidade, idade e sexo dos usuários, eventos que acontecem dentro do aplicativos e erros de sintaxe. O firebase oferece uma função de exportar todos os dados coletados para um *data warehouse* hospedado em nuvem, permitindo a criação de relatórios detalhados de dados de forma personalizada.

A seção de *Develop* é onde são localizadas as ferramentas de desenvolvimento que aux-

ilham o desenvolvedor. As informações são armazenadas no *Realtime Database*, que é uma base de dados hospedada em núvens que utiliza a linguagem *NoSql*. Para manter a lógica e regra de negócio do *backend* centralizada e segura, foi criado o *Cloud Functions*, que exclui a necessidade de gerenciar o próprio servidor. Para a criação e gerenciamento de domínios existe o *Firebase Hosting*. Para *download* e *upload* de arquivos é utilizado o *Firebase Storage*. Os erros que acontecem nos aplicativos podem ser capturados pelo *Crash Reporting*, onde o desenvolvedor consegue diagnosticar a causa das possíveis falhas. E por fim, opções de gerenciamento de desempenho e performance também são oferecidas pelo *Performance Monitoring* (CHATTERJEE et al., 2018).

A seção de *Grow* é a parte relacionada ao crescimento e engajamento. São fornecidas funções que ajudam na realização de testes dentro do aplicativo, enviar mensagens para os usuários e visualizar padrões de comportamento dentro do aplicativo.

2.4 Categorização de alimentos e produtos

A categorização de produtos é feita com base nas 5 principais categorias de alimentos, de modo que cada alimento é colocado na categoria que tem seu macronutriente mais abundante, exceto frutas e vegetais. Sendo elas:

1. Fontes de Carboidratos

- (a) Grãos e cereais (pão, arroz, macarrão, aveia)
- (b) Leguminosas (feijões, lentilhas, grão-de-bico)
- (c) Vegetais ricos em amido (batatas, milho)
- (d) Produtos à base de amido (massas, tortilhas)

2. Fontes de Proteínas

- (a) Carnes magras (frango, peito de peru, carne magra)
- (b) Peixes (salmão, atum, tilápia)
- (c) Ovos
- (d) Tofu e outras alternativas à carne

3. Fontes de Gordura

- (a) Óleos (azeite de oliva, óleo de coco, óleo de canola)
- (b) Nozes e sementes (amêndoas, nozes, sementes de chia)
- (c) Abacate
- (d) Manteiga

4. Frutas

- (a) Maçãs
- (b) Bananas
- (c) Morango
- (d) Uva

5. Vegetais

- (a) Brócolis
- (b) Cenoura
- (c) Espinafre
- (d) Tomate

Além dessas categorias, existe também divisões por faixas de horário do dia. Desse modo, os alimentos são categorizados nas faixas de horário mais propícias para o consumo. Sendo essas faixas:

1. Manhã

- (a) 06:00 - 09:00
- (b) 09:00 - 12:00

2. Tarde

- (a) 12:00 - 15:00
- (b) 15:00 - 18:00

3. Noite

- (a) 18:00 - 21:00
- (b) 21:00 - 00:00

[ADICIONAR TRATATIVA PARA ALERGIAS A ALIMENTOS]

2.5 Trabalhos Relacionados

O primeiro artigo analisado é Tecnologia em Saude: versão brasileira do software GloboDiet para avaliação do consumo alimentar em estudos epidemiológicos (STELUTI et al., 2020). O software não apresenta análise de produtos e a recomendação de produtos se baseia no consumo energético e nos macronutrientes dos alimentos.

O segundo artigo apresenta o CardNutri: Um Software de planejamento de cardápios nutricionais semanais para alimentação escolar aplicando inteligência artificial (MOREIRA et al., 2017). A análise de produtos é feita por comparações de tabela de nutrientes, preço e

localidade. A recomendação de alimentos é realizada por faixa etária, variedade de alimentos, harmonia de preparações, e um valor máximo a ser gasto para fazer cada refeição.

Para quesito de comparação, o artigo atual tem análise dos produtos baseada nas comparações das tabelas nutricionais, localidade do usuário, e o preço dos produtos, usando a localidade para aferir de forma mais precisa os preços. Além disso, a recomendação de alimentos é feita com base no plano alimentar, na localidade, no orçamento do paciente, e nas preferências do usuário, que pode selecionar quais tipos de alimentos deseja consumir em quais faixas de horários do dia.

2.5.1 Análise e Recomendação de Produtos

Observando o primeiro artigo, foi definido que não foram encontradas informações relacionadas a análise de produtos. E sobre a recomendação de produtos, foi realizada com base no consumo energético dos macronutrientes (STELUTI et al., 2020).

O último artigo analisado foi o CardNutri, no qual a análise de produtos é feita por meio de inteligência artificial, e a recomendação de alimentação é realizada por faixa etária, variedade, harmonia das preparações e um valor máximo a ser pago por refeição de estudantes (MOREIRA et al., 2017).

3 METODOLOGIA

A metodologia escolhida será a Pesquisa Aplicada, com o objetivo de adquirir novos conhecimentos para o desenvolvimento ou aprimoramento de produtos, processos e sistemas.

Para desenvolvimento da aplicação, vai ser adotado um modelo sequencial. Esse processo é controlado por um cronograma, e é composto por 3 (três) etapas: Planejamento, Execução e Validação, nas quais são detalhadas a seguir.

3.1 Planejamento

Esta etapa, neste projeto, é a de maior importância, pois será um planejamento coerente com o tempo de 3,5 meses que é disponibilizado para a construção desta aplicação. Esta etapa será subdividida em três etapas: Produção de documentos técnicos, produção de documento gerenciais e o levantamento das API's disponíveis para consulta.

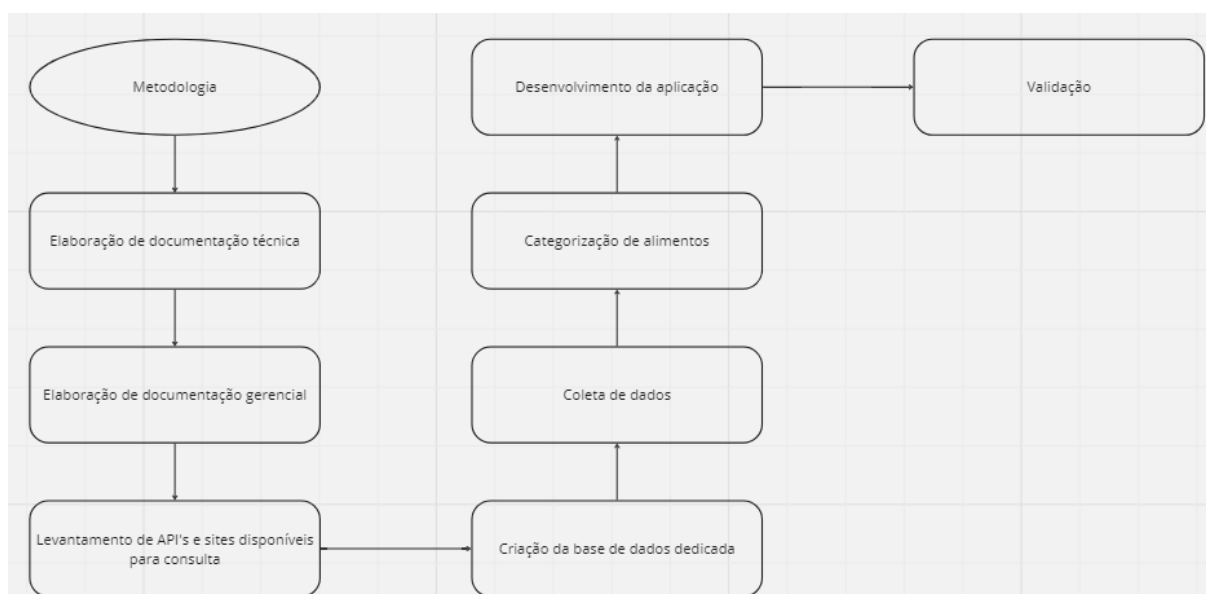


Figure 1 – Planejamento de execução por etapas

3.1.1 Documentos técnicos

A documentação técnica será de extrema importância para nortear a construção deste sistema. Dentre eles estão:

- Levantamento de requisitos funcionais e não funcionais. Serão os nossos guias para a elaboração das funcionalidades do sistema. Tudo o que estiver contido nos documentos possibilitará que o software seja criado, atualizado e reparado sempre que necessário de acordo com o que foi inicialmente estipulado.
- Casos de uso baseados em histórias de usuários. É importante trazer situações reais para a aplicação, portanto a utilização de historias de usuários é indispensável.
- Restrição e mecanismos arquiteturais. Será feito um levantamento das tecnologias a serem utilizadas para atender a todas as demandas funcionais e não funcionais do sistema.

3.1.2 Documentos gerenciais

A documentação gerencial será essencial para ser avaliado a viabilidade deste projeto. E dentre eles estão:

- Modelo de projeto Canvas. Este modelo condensa as principais informações do sistema, de modo a ser facilmente visualizado e alterado, para que em alguma eventual mudança, tenhamos essa flexibilidade.

- Estrutura analítica do projeto (EAP). Esta técnica nos permite organizar uma entrega grande em diversas entregas menores, sendo assim adotando uma metodologia mais ágil e mais fácil de gerenciar.
- Kanban. Esse quadro permite controlar o fluxo de tarefas de todo o projeto. Sendo assim, é possível saber sempre onde se encontra cada etapa.
- Lista de riscos. Essa técnica auxilia a pensar e antecipar algumas situações, com o intuito de planejar o que fazer caso cada uma delas aconteça. Levando em consideração um tempo de 3,5 meses, é importante que a maioria dos casos já estejam antecipados, para que seja possível um tempo de resposta menor em relação a qualquer adversidade.

3.1.3 Levantamento de API's disponíveis para consulta

Para categorização dos produtos em relação ao preço e as informações nutricionais, é necessário ter acesso à localidade do usuário. Além disso, é importante ter acesso a informações de mercados perto do usuário. Para isso, vai ser preciso de pelo menos uma API de algum mercado que esteja próximo ao usuário. Visando isso, essa etapa será focada em pesquisa e estudos sobre a maior quantidade possível de API's de mercados, de forma que seja atendido o objetivo de cobrir uma relevante área em uma cidade.

3.2 Execução

A etapa de execução será a maior e mais demorada do projeto. Esta etapa será também subdividida entre: Elaboração de diagramas, Categorização de produtos e Desenvolvimento da aplicação .

3.2.1 Elaboração de diagramas

Ainda na área da documentação técnica, a elaboração de alguns diagramas é a primeira etapa da execução do projeto. Esses diagramas são facilitadores e fontes de consulta para o desenvolvimento da aplicação. Dentre eles estão:

- Diagrama de Classe
- Diagrama de Entidade e Relacionamento
- Diagrama de Contexto
- Diagrama de Contêiner
- Modelo C4

3.2.2 Categorização de produtos

A categorização de produtos é uma parte complexa do projeto. Para referência de cálculos, uma parte da aplicação rodará no servidor, e terá como responsabilidade preencher as tabelas de um banco de dados relacional com as informações nutricionais dos produtos, e além disso, o preço atual de cada um desses produtos, em diversos estabelecimentos. Para auxiliar esse processo, o *backend* da aplicação também fará consultas pontuais em fontes confiáveis e ajudará a manter o banco de dados o mais atualizado possível. Isso ocorrerá da seguinte forma: ao paciente inserir o plano alimentar no aplicativo, os alimentos descritos nesse plano alimentar serão categorizados de acordo com suas informações nutricionais e os preços disponíveis nas API's dos mercados mais próximos do paciente, garantindo assim uma atualização mais recente dos produtos.

3.2.3 Desenvolvimento da aplicação

O desenvolvimento da aplicação será baseado em uma arquitetura MVC, sendo o componente *Model* feito em MySQL, o componente *View* feito em Flutter e por fim, o componente *Controller* feito em Node.js. Essa escolha foi feita devido à facilidade que essas tecnologias nos proporcionam em cada um dos casos:

- *Model* de persistência, inicialmente será feito em MySQL, por ser simples de aplicar, manipular e ser capaz de lidar com a quantidade de dados estimada para uma versão beta.
- *View*, foi escolhido o Flutter por ser uma linguagem eficaz, que gera aplicações nativas em diferentes Sistemas Operacionais, e por ser uma linguagem bem documentada.
- *Controller* será feito em Node.js, e essa escolha foi feita por ser uma linguagem para lidar com testes automatizados, pode ser utilizado para requisições em API's, e de robustez e confiabilidade importantes, uma vez que permite a execução de códigos JavaScript fora de um navegador web.

Além disso, a aplicação conterá uma lógica baseada em um algoritmo de *Machine Learning* para fazer uma clusterização de produtos de modo que a composição desses produtos se encaixe no plano alimentar do paciente, e também se encaixe no orçamento proposto. O algoritmo a ser utilizado será o K-means para a clusterização de produtos semelhantes, agrupando esses produtos e atribuindo entre eles um grau de similaridade para com todos os outros.

3.3 Validação

Para esta última etapa, a validação será feita na prática, pessoalmente, com a ida a alguns estabelecimentos. Após treinamento do sistema, alguns testes serão feitos. Os resultados desses

testes serão levados para os estabelecimentos que a aplicação abrange, e será colocado à prova se a aplicação realmente foi capaz de fazer um novo plano alimentar, com as mesmas condições nutricionais, mas com o preço final menor que a original, de modo a caber no orçamento do paciente.

3.4 Cronograma

Atividades	Agosto	Setembro	Outubro	Novembro	Dezembro
Planejamento	x				
Finalizamento da proposta	x	x			
Execução		x	x		
Validação			x	x	x

Para criação do cronograma, foi levado em conta os estágios dos produtos em 4 (quatro) meses de produção. A etapa de Planejamento, vai ocupar apenas o mês de agosto. A etapa de execução vai ocupar o mês de agosto, setembro e outubro. E por fim, a etapa de validação, vai ocupar os meses de outubro e novembro. Pode-se observar que algumas etapas podem começar quando outras ainda estão em execução.

References

- ANTUNES, Hanna K.M. et al. Análise de taxa metabólica basal e composição corporal de idosos do sexo masculino antes e seis meses após exercícios de resistência. In: UNIVERSIDADE FEDERAL DE SÃO PAULO-UNIFESP/EPM. São Paulo - SP, BR: Scielo, 2015.
- CHATTERJEE, Nilanjan et al. Real-time communication application based on android using google firebase. **IJARCSMS**, v. 6, 04 2018.
- FENTAW, Awel Eshetu. Cross platform mobile application development: a comparison study of react native vs flutter. 2020.
- JOB, Alessandra Rocha; OLIVEIRA, Ana Cristina Soares de. Percepção dos acadêmicos de nutrição sobre a efetividade a longo prazo dos métodos de dietas restritivas para a perda e o controle de peso. **Revista Brasileira de Obesidade, Nutrição e Emagrecimento**, São Paulo, v. 13, n. 78, p. 291–298, mar./abril. 2019.
- LI, Wu-Jeng et al. Justiot internet of things based on the firebase real-time database. 2018. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8353979>. Acesso em: 04 de set. 2023.
- MAGNO, Fernanda Cristina Carvalho Mattos et al. **Macro e micronutrientes na orientação nutricional para obesidade**. 44. ed. Juiz de Fora, MG, BR: HU Revista, 2018. 251-259 p.
- MENON, Daiane; SANTOS, Jacqueline Schaurich dos. Consumo de proteína por praticantes de musculação que objetivam hipertrofia muscular. In: UNIVERSIDADE DE CAXIAS DO SUL. Caxias do Sul - RS, BR: Scielo, 2012.
- MOREIRA, Rafaela Priscila Cruz; MARTINS, Flávio Vinícius Cruzeiro; WANNER, Elizabeth Fialho. Cardnutri: Um software de planejamento de cardápios nutricionais semanais para alimentação escolar aplicando inteligência artificial. Rio de Janeiro, RJ, BR, 2017. Disponível em: <https://www.reciis.icict.fiocruz.br/index.php/reciis/article/view/1272>. Acesso em: 24 de nov. 2022.
- RISSI, Matheus; DALLILO, Felipe Diniz. Flutter um framework para desenvolvimento mobile. 2022. Disponível em: <https://recima21.com.br/index.php/recima21/article/view/2230>. Acesso em: 04 de set. 2023.
- SHAH, Kewal; SINHA, Harsh; MISHRA, Payal. Analysis of cross-platform mobile app development tools. In: IEEE. **2019 IEEE 5th International Conference for Convergence in Technology (I2CT)**. [S.l.], 2019. p. 1–7.
- SILVA, Valéria Martins da. Revisão sistemática da evolução mvc na base acm. 2012. Disponível em: https://41jaiio.sadio.org.ar/sites/default/files/31_EST2012.pdf. Acesso em: 04 de set. 2023.
- SOUSA, Filipe Perdigão de. Criação de framework rest/hateoas open source para desenvolvimento de apis em node.js. Porto, PT, 2015. Disponível em: <https://core.ac.uk/download/pdf/302972302.pdf>. Acesso em: 24 de nov. 2022.
- STELUTI, Josiane et al. Tecnologia em saúde: versão brasileira do software globodiet para avaliação do consumo alimentar em estudos epidemiológicos. São Paulo, SP, BR, 2020. Disponível em: <https://www.scielo.br/j/rbepid/a/DDF5kZKPwsbzBP3ZDDdTh8F/?lang=pt>. Acesso em: 24 de nov. 2022.

WU, Wenhao. React native vs flutter, cross-platforms mobile application frameworks. Metropolia Ammattikorkeakoulu, 2018.