

CENTRO UNIVERSITÁRIO SAGRADO CORAÇÃO

DAVID CORRÊA GUEDES

YOUTHEALTH: DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA
ANÁLISE DO COMPORTAMENTO ALIMENTAR DE ESTUDANTES

BAURU

2021

DAVID CORRÊA GUEDES

YOUTHEALTH: DESENVOLVIMENTO DE UM APLICATIVO MÓVEL PARA
ANÁLISE DO COMPORTAMENTO ALIMENTAR DE ESTUDANTES

Monografia de Iniciação Científica
apresentada à Pró-Reitoria de Pesquisa e
Pós-Graduação como parte dos pré-
requisitos para aprovação do conselho,
sob orientação do Prof. Me. Patrick
Pedreira Silva.

BAURU

2021

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

G924y	<p>Guedes, david Corrêa</p> <p>YoutHealth: Desenvolvimento de um aplicativo móvel para análise do comportamento alimentar de estudantes / David Corrêa Guedes. -- 2021. 41f. : il.</p> <p>Orientador: Prof. M.e Patrick Pedreira Silva</p> <p>Monografia (Iniciação Científica em Ciência da Computação) - Centro Universitário Sagrado Coração - UNISAGRADO - Bauru - SP</p> <p>1. Estudantes universitários. 2. Comportamento Alimentar. 3. Engenharia de Software. 4. Aplicativo. I. Silva, Patrick Pedreira. II. Título.</p>
-------	--

RESUMO

Durante o curso de graduação os jovens universitários lidam com o stress e com a ansiedade, principalmente em períodos de avaliação. Há a hipótese de que nesse período a alimentação dos jovens podem sofrer alterações. Diante disso, este projeto pretendeu modelar e foi desenvolvido um aplicativo para auxílio na análise do comportamento alimentar dos acadêmicos do Unisagrado, por meio do levantamento de dados como os alimentos mais consumidos e registros da alimentação desses jovens. Para isso foi estudado uma linguagem de programação específica bem como técnicas para o desenvolvimento da aplicação. O aplicativo desenvolvido visou permitir o cadastro das refeições e emitir relatórios para que o usuário final obtivesse um feedback por meio da aplicação. Ao fim da pesquisa, foi verificada a importância de uma aplicação para auxílio e levantamento de dados acerca da alimentação dos estudantes, permitindo controlar as refeições realizadas e contribuindo com a possibilidade de gerar dados para possíveis estudos de outras áreas como por exemplo, a Nutrição.

Palavras-chave: Estudantes universitários. Comportamento Alimentar. Engenharia de Software. Aplicativo.

SUMÁRIO

<u>1</u>	<u>INTRODUÇÃO</u>	6
<u>2</u>	<u>OBJETIVOS</u>	8
<u>2.1</u>	<u>OBJETIVO GERAL</u>	8
<u>2.2</u>	<u>OBJETIVOS ESPECÍFICOS</u>	8
<u>3</u>	<u>REFERENCIAL TEÓRICO</u>	9
<u>3.1</u>	<u>ENGENHARIA DE SOFTWARE</u>	9
<u>3.1.1</u>	<u>Teste de software</u>	11
<u>3.1.2</u>	<u>Níveis de teste de software</u>	14
<u>3.1.3</u>	<u>Técnicas de teste de software</u>	17
<u>3.2</u>	<u>PADRÕES ALIMENTARES</u>	18
<u>3.3</u>	<u>TABELA BRASILEIRA DE COMPOSIÇÃO DE ALIMENTOS</u>	19
<u>4</u>	<u>METODOLOGIA</u>	20
<u>4.1</u>	<u>LEVANTAMENTO BIBLIOGRÁFICO</u>	20
<u>4.2</u>	<u>DEFINIÇÃO DA ARQUITETURA DO SISTEMA</u>	20
<u>4.3</u>	<u>LEVANTAMENTO DE RECURSOS TECNOLÓGICOS</u>	23
<u>4.3.1</u>	<u>Hardware</u>	23
<u>4.3.2</u>	<u>Software</u>	24
<u>4.4</u>	<u>DESENVOLVIMENTO DA FERRAMENTA</u>	27
<u>4.5</u>	<u>AVALIAÇÃO</u>	31
<u>5</u>	<u>RESULTADOS</u>	33
<u>5.1</u>	<u>BANCO DE DADOS</u>	33
<u>5.2</u>	<u>APLICAÇÃO</u>	34
<u>6</u>	<u>CONSIDERAÇÕES FINAIS</u>	39
	<u>REFERÊNCIAS</u>	40

1 INTRODUÇÃO

O comportamento alimentar do ser humano sofre mudanças desde o começo da humanidade, a princípio houve a necessidade de caçar e plantar para produzir seus alimentos até chegar ao que se tem nos dias de hoje, alimentos produzidos e processados de forma industrial. Tais mudanças se dão por diversos fatores, como cultural tendo em mente que cada região do globo tem seus costumes; regional, levando em conta os fatores como clima e vegetação e socioeconômicos, por questões como governo e economia.

A grande questão do comportamento alimentar da atualidade é a sua qualidade, que alimenta, mas não sustenta e vem acarretando diversos problemas quando o assunto é saúde.

Segundo dados do Ministério da Saúde, os índices de obesidade cresceram cerca de 67,8% nos últimos treze anos (INCA, 2019). Para a professora Jaime, este aumento está diretamente ligado à mudança nos hábitos alimentares (SANTOS, 2019). Dentre um dos fatores que estimula esta mudança de forma negativa é a falta de tempo, fazendo com que as pessoas se tornem adeptas dos famosos *fast-foods*, alimentos de preparo rápido que geralmente possuem altos níveis de açúcar, sódio, gordura e conservantes.

De acordo com uma pesquisa realizada pela Universidade de Boston em 2019, os estabelecimentos de *fast-foods* tiveram uma “piora” em seus cardápios nos últimos 30 anos. Os pesquisadores alegaram um aumento de 30 calorias nos pratos de entrada e 62 calorias nas sobremesas. Quanto ao teor de sódio, os pratos tiveram um aumento de pouco mais de 9% ao total do indicado para uma dieta diária (WELLS, 2019), números preocupantes tendo em vista que o Brasil é o que mais gasta neste setor na América do Sul, segundo dados levantados pela EAE Business School (ROMERO, 2016).

Além do mais, o número de consumidores de *fast-foods* tende a crescer no Brasil e no mundo, tendo em vista a facilidade de sua compra por meio de *apps* como iFood, Uber Eats e Rappi. Antes, era necessário ir ao estabelecimento para a compra e consumação destes. Hoje, com poucos cliques e em praticamente qualquer horário,

o consumidor compra e recebe o pedido em sua residência, hábito que os brasileiros estão aderindo e muitas vezes substituindo por suas refeições principais.

Outro fator relevante nessa problemática são os distúrbios alimentares como a anorexia e a bulimia. Suas causas são as mais diversas, indo desde os fatores biológicos aos socioculturais (STICE; SHAW, 2002). Segundo Silva *et al.* (2018) “os adolescentes, por estarem em processo de formação da identidade, ficam mais susceptíveis às influências de modelos físicos idealizados, veiculados pela mídia e pares”, assim, se submetem a dietas que colocam em risco a sua saúde.

Além desses, os jovens, com ênfase nos universitários ainda lidam com problemas, como stress e ansiedade por conta da graduação, principalmente em períodos avaliativos. De acordo com Lourenço (2016), as emoções afetam grandemente o comportamento alimentar tanto na escolha dos alimentos bem como na quantidade e frequência de sua ingestão.

Tendo em vista os aspectos mencionados, o projeto tem como finalidade se unir com a tecnologia para ajudar no levantamento dos dados acerca da alimentação dos jovens para uma melhor análise dos mesmos por meio dos resultados obtidos, como é descrito e proposto nos objetivos e metodologia a seguir.

2 OBJETIVOS

2.1 OBJETIVO GERAL

- a) Modelar e desenvolver um aplicativo para análise do comportamento alimentar de estudantes por meio de aplicação móvel.

2.2 OBJETIVOS ESPECÍFICOS

- a) Realizar um levantamento acerca dos alimentos mais consumidos pelos universitários;
- b) definir e estudar uma linguagem de programação específica tais como suas técnicas para o desenvolvimento da aplicação;
- c) planejar como será realizado o cadastro das refeições e datas de provas/trabalhos;
- d) definir o modo como será realizado os relatórios para apresentar um feedback ao usuário;
- e) implementar o aplicativo aos universitários do Centro Universitário Sagrado Coração (Unisagrado);
- f) analisar os resultados obtidos;
- g) promover a divulgação da experiência e resultados obtidos em eventos técnicos e científicos, publicações correlatas, e participar do Congresso Anual de Iniciação Científica e Desenvolvimento Tecnológico e Inovação do UNISAGRADO.

3 REFERENCIAL TEÓRICO

3.1 ENGENHARIA DE SOFTWARE

Engenharia de Software é o processo de estudar, criar e otimizar os processos de trabalho para os desenvolvedores de software, com isso, descartam-se as atividades de levantamento de requisitos, modelagem, design e codificação, pois não são consideradas típicas de um engenheiro de software, embora em muitas circunstâncias, é ele que executa tais atividades, e ao mesmo tempo, tem o foco em observar, avaliar, orientar e alterar os processos produtivos quando for preciso (WAZLAWICK, 2013).

A engenharia de software tem duas definições, uma delas é que a engenharia pode fazer qualquer produto funcionar, aplicando teorias, métodos e ferramentas quando for necessário, sempre procurando soluções para os problemas. Ela não se dedica apenas aos processos técnicos do desenvolvimento de software, mas também ao gerenciamento de projetos e o desenvolvimento de ferramentas, métodos e teorias para apoiar a produção do software. Os engenheiros de software se preocupam mais com os serviços, as interações e as restrições dentro das quais o sistema deve ser construído e operado (SOMMERVILLE, 2007)

As exigências para o desenvolvimento de um software são cada dia mais relevantes, pois empresas, órgãos e organizações precisam cada vez mais do mesmo para se manter em crescimento no mercado, isso conta com a enorme quantidade de informações indispensáveis para obter a competitividade, e tomar decisões prudentes no seu exato momento. Portanto, é preciso obter modificações do software para que ele possa moldar-se de acordo com o cenário proposto, ou seja, o software deve ser sujeito a qualquer tipo de manutenção (PRESSMAN, 2011).

Segundo Wazlawick (2013) a engenharia de software apresenta um conjunto de princípios que são usados no desenvolvimento de projetos, os quais são apresentados a seguir:

a) Decomposição: construção de softwares complexos a partir de partes cada vez mais simples.

- b) Abstração: executar um elemento em uma linguagem de nível mais alto do que for preciso para sua construção.
- c) Generalização: é usada na classificação de dados. Ou seja, objetos podem ser classificados em várias classes simultaneamente.
- d) Padronização: elaboração de padrões de programações ajuda no desenvolvimento de produtos com uma qualidade significativa.
- e) Flexibilidade: facilidade para acomodar as mudanças nos requisitos que não foram previstas.
- f) Formalidade: desenvolver um produto com tal formalidade que tenha que ser atingida.
- g) Rastreabilidade: produzir especificações para que saiba quando ocorre alguma alteração.
- h) Desenvolvimento iterativo: realizar vários ciclos distintos para o desenvolvimento de um processo para o amadurecimento do produto final.
- i) Gerenciamento de Requisitos: sabendo que os requisitos mudam frequentemente, é necessário gerenciar todas essas mudanças para o desenvolvimento e evolução do software.
- j) Arquiteturas baseadas em componentes: melhor forma de se adquirir reusabilidade e para saber lidar com a complexidade.
- k) Modelagem visual: é possível produzir sistemas completos sem passar nas fases mais clássicas da programação.
- l) Verificação contínua da qualidade: a qualidade tem que obter total atenção do começo ao fim de qualquer projeto de desenvolvimento.
- m) Controle de mudanças: permite que volte atrás em um caminho de desenvolvimento ou até produzir em paralelo de diferentes versões do mesmo componente.
- n) Gerenciamento de riscos: realiza atividades preventivas para riscos ou até mesmo a probabilidade do mesmo.

3.1.1 TESTE DE SOFTWARE

Antes de iniciar a discussão sobre teste de software é preciso esclarecer alguns conceitos relacionados a essa atividade. Inicialmente, faz-se necessário conhecer a diferença entre Defeitos, Erros e Falhas. As definições que serão usadas aqui seguem a terminologia padrão para Engenharia de Software do IEEE – *Institute of Electrical and Electronics Engineers* – (IEEE) de 1990 (DIAS NETO, 2010).

a) Defeito é um ato inconsistente cometido por um indivíduo ao tentar entender uma determinada informação, resolver um problema ou utilizar um método ou uma ferramenta. Por exemplo, uma instrução ou comando incorreto.

b) Erro é uma manifestação concreta de um defeito num artefato de software. Diferença entre o valor obtido e o valor esperado, ou seja, qualquer estado intermediário incorreto ou resultado inesperado na execução de um programa constitui um erro.

c) Falha é o comportamento operacional do software diferente do esperado pelo usuário. Uma falha pode ter sido causada por diversos erros e alguns erros podem nunca causar uma falha.

A Figura 1 expressa a diferença entre esses conceitos.

Figura 1 – Defeito x erro x falha



Fonte: Dias Neto (2010).

Como pode ser observado na Figura 2, defeitos fazem parte do universo físico (a aplicação propriamente dita) e são causados por pessoas, por exemplo, através do mal uso de uma tecnologia, podendo ocasionar a manifestação de erros em um produto, ou seja, a construção de um software de forma diferente ao que foi especificado (universo de informação). Por fim, os erros geram falhas, que são comportamentos inesperados em um software, que afetam diretamente o usuário final da aplicação (universo do usuário), e podem inviabilizar a utilização de um software.

Teste de software pode ser definido como sendo o processo de execução de um produto para determinar se ele atingiu suas especificações e funcionou corretamente no ambiente para o qual foi projetado, ao mesmo tempo em que faz parte de todo o processo de engenharia de software. O seu objetivo é revelar falhas em um produto, para que as causas dessas falhas sejam identificadas e possam ser corrigidas pela equipe de desenvolvimento antes da entrega final. Por conta dessa característica das atividades de teste, é dito que sua natureza é “destrutiva”, e não “construtiva”, pois visa ao aumento da confiança de um produto através da exposição de seus problemas, porém antes de sua entrega ao usuário final (DIAS NETO, 2010).

Ainda segundo o autor, o conceito de teste de software pode ser compreendido através de uma visão intuitiva ou mesmo de uma maneira formal. Existem atualmente várias definições para esse conceito. De uma forma simples, testar um software significa verificar através de uma execução controlada se o seu comportamento ocorre de acordo com o especificado. O objetivo principal desta tarefa é revelar o número máximo de falhas dispondo do mínimo de esforço, ou seja, mostrar aos que desenvolvem se os resultados estão ou não de acordo com os padrões estabelecidos (DIAS NETO, 2010).

Para Pressman (2011), o processo de testes visa encontrar falhas no sistema para corrigi-las antes de distribuir o sistema, ou de atualizar o software com novos recursos, portanto o sistema deve ser projetado e implementado pensando na facilidade da realização dos testes. Por sua vez os testes devem ser feitos levando em consideração certas características para que nada seja deixado para trás.

Em conformidade com o autor, o conceito de testabilidade consiste em se medir o quão simples é o ato de testar um software. As características apresentadas a seguir caracterizam um software a ser testável:

- a) Operabilidade: um sistema projetado e implementado tendo em mente a qualidade, terá poucas falhas quando os testes forem realizados.
- b) Observabilidade: quando é possível ver com clareza as entradas, saídas e variáveis do sistema fica mais fácil de detectar possíveis falhas.
- c) Controlabilidade: entradas geram saídas específicas, e para cada tipo de saída existirá um tipo de entrada específica. Se o engenheiro puder controlar essas entradas ficará mais fácil realizar os testes.
- d) Decomponibilidade: o sistema é construído a partir de módulos e pode ser testado em partes.
- e) Simplicidade: Quanto mais simples um sistema for e ainda sim atingindo o objetivo, mais simples serão os testes.
- f) Estabilidade: Quanto menos alterações o software tiver, menos testes precisarão ser feitos.
- g) Compreensibilidade: Quanto mais informações estiverem disponíveis para o entendimento do software, mais eficazes serão os testes, isso inclui manuais organizados, detalhados e especificados.

A atividade de teste é composta por alguns elementos essenciais que auxiliam na formalização desta atividade, os quais serão apresentados a seguir.

- a) Caso de Teste: descreve uma condição particular a ser testada e é composto por valores de entrada, restrições para a sua execução e um resultado ou comportamento esperado (CRAIG; JASKIEL, 2002).
- b) Procedimento de Teste: é uma descrição dos passos necessários para executar um caso (ou um grupo de casos) de teste (CRAIG; JASKIEL, 2002).
- c) Critério de Teste: serve para selecionar e avaliar casos de teste de forma a aumentar as possibilidades de provocar falhas ou, quando isso não ocorre, estabelecer um nível elevado de confiança na correção do produto (ROCHA, A. R.; MALDONADO, J. C.; WEBER, K. C., 2001). Os critérios de teste podem ser utilizados como:
 - i. Critério de Cobertura dos Testes: permite a identificação de partes do programa que devem ser executadas para garantir a qualidade do software e indicar quando o mesmo foi suficientemente testado (RAPPS; WEYUKER, 1982). Ou seja, determinar o percentual de elementos necessários por um critério de teste que foram executados pelo conjunto de casos de teste.

ii. Critério de Adequação de Casos de Teste: Quando, a partir de um conjunto de casos de teste T qualquer, ele é utilizado para verificar se T satisfaz os requisitos de teste estabelecidos pelo critério. Ou seja, este critério avalia se os casos de teste definidos são suficientes ou não para avaliação de um produto ou uma função (ROCHA, A. R.; MALDONADO, J. C.; WEBER, K. C., 2001).

iii. Critério de Geração de Casos de Teste: quando o critério é utilizado para gerar um conjunto de casos de teste T adequado para um produto ou função, ou seja, este critério define as regras e diretrizes para geração dos casos de teste de um produto que esteja de acordo com o critério de adequação definido anteriormente (ROCHA, A. R.; MALDONADO, J. C.; WEBER, K. C., 2001).

3.1.2 NÍVEIS DE TESTE DE SOFTWARE

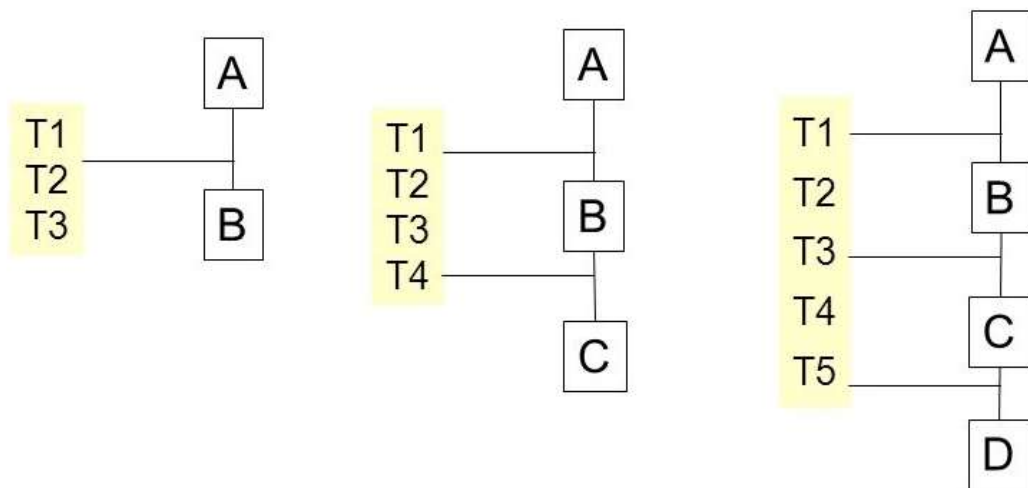
O planejamento dos testes deve ocorrer em diferentes níveis e em paralelo ao desenvolvimento do software. Segundo Rocha, Maldonado e Weber (2001) os principais níveis de teste de software são:

Teste de Unidade: também conhecido como teste unitário. Tem por objetivo explorar a menor unidade do projeto, procurando provocar falhas ocasionadas por defeitos de lógica e de implementação em cada módulo, separadamente. O universo alvo desse tipo de teste são os métodos dos objetos ou mesmo pequenos trechos de código.

Teste de Integração: O processo de integração do sistema engloba a criação de um sistema com base em seus componentes e o teste do sistema derivado da interação de diferentes componentes. Os componentes podem ser comerciais, reusáveis adequados a certo sistema ou novos componentes desenvolvidos. Neste teste é verificado se todos os componentes são chamados corretamente, se transmitem os dados corretamente e no seu devido tempo e se funcionam “como um relógio”, ou seja, trabalhando em conjunto e ou um dependendo do outro (SOMMERVILLE, 2007).

Na Figura 2, pode-se entender melhor o funcionamento do teste de integração.

Figura 2 - Testes de integração incremental.

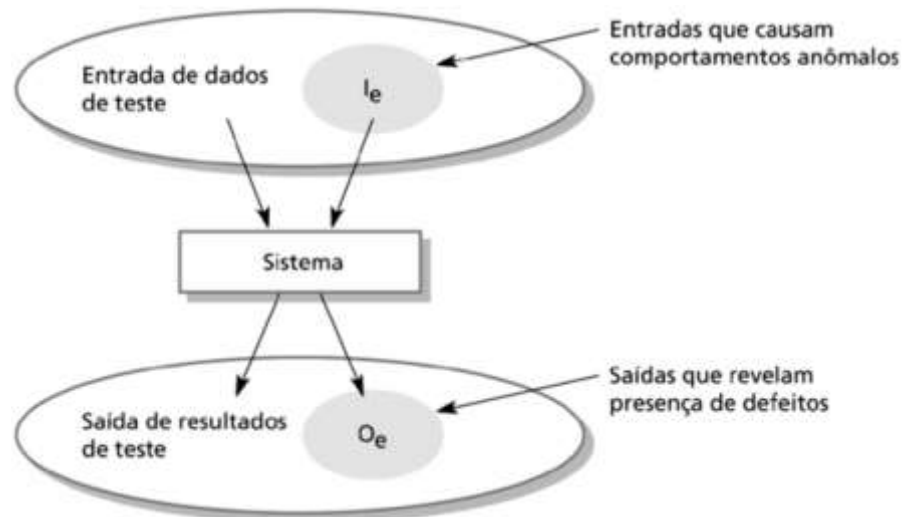


Fonte: Elaborada pelo autor.

[...] A, B, C e D são componentes e T1 a T5 referem – se aos testes de características incorporadas ao sistema. T1, T2 e T3 são executados inicialmente no sistema composto dos componentes A e B (sistema mínimo). Se esses testes revelarem defeitos, estes serão corrigidos. Os componentes C é integrado e T1, T2 e T3 são repetidos para assegurar que não há interação inesperadas entre A e B. Se surgirem problemas nesses testes, isso provavelmente significa que eles são decorrentes da interação com o novo componente. A origem do problema é localizada, simplificando a identificação e correção do defeito. O conjunto de testes T4 é também executado no sistema. Finalmente, o componente D é integrado e testado por meio dos testes existentes e dos novos testes (T5) (SOMMERVILLE, 2007, p. 358).

Teste de Releases - também chamado como teste caixa-preta e teste funcional, nessa etapa é testado o release do sistema que será entregue ao usuário. Tem como objetivo aumentar a confiança do desenvolvedor, no qual saberá se o sistema atende a todos os requisitos, ou seja, se ele atender a todos os requisitos ele poderá ser entregue ao cliente ou ao seu destino final. Mas para isso o sistema deve fornecer funcionalidade, desempenho, confiabilidade e que não forneça nenhum tipo de falhas durante sua execução (SOMMERVILLE, 2007). Ainda segundo o autor, no teste de releases, a equipe de teste mune entradas no sistema para verificar se as saídas serão ou não as esperadas, se a resposta for não, algum problema foi detectado. Como especificado na Figura 3.

Figura 3 – Teste de caixa preta.



Fonte: Sommerville (2007, p.359).

Pode-se afirmar que na área de teste de software, o teste em si não comprova que o produto está desprovido de defeitos (WAZLAWICK, 1967).

Teste de Sistema - avalia o software em busca de falhas por meio da utilização do mesmo, como se fosse um usuário final. Dessa maneira, os testes são executados nos mesmos ambientes, com as mesmas condições e com os mesmos dados de entrada que um usuário utilizaria no seu dia a dia de manipulação do software. Verifica se o produto satisfaz seus requisitos.

Teste de Aceitação - são realizados geralmente por um restrito grupo de usuários finais do sistema. Esses simulam operações de rotina do sistema de modo a verificar se seu comportamento está de acordo com o solicitado.

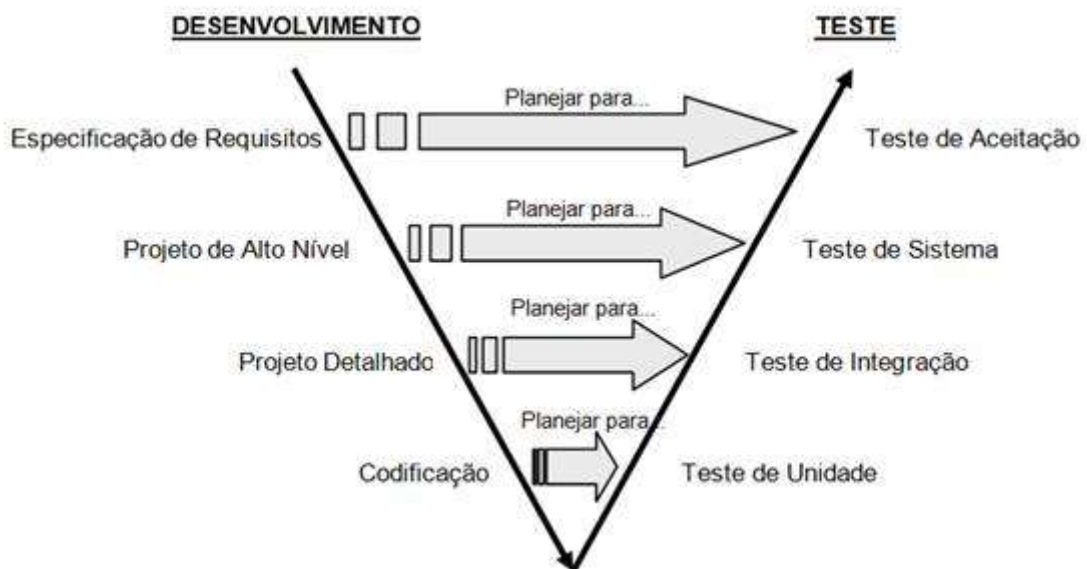
Teste de Regressão - não corresponde a um nível de teste, mas é uma estratégia importante para redução de “efeitos colaterais”. Consiste em se aplicar, a cada nova versão do software ou a cada ciclo, todos os testes que já foram aplicados nas versões ou ciclos de teste anteriores do sistema. Pode ser aplicado em qualquer nível de teste.

Dessa forma, conforme traz a Figura 4, o planejamento e projeto dos testes devem ocorrer de cima para baixo, ou seja:

Inicialmente é planejado o teste de aceitação a partir do documento de requisitos. Após isso é planejado o teste de sistema a partir do projeto de alto nível do

software. Em seguida ocorre o planejamento dos testes de integração a partir o projeto detalhado. E por fim, o planejamento dos testes a partir da codificação.

Figura 4 - Modelo V descrevendo o paralelismo entre as atividades de desenvolvimento e teste de software.



Fonte: Craig e Jaskiel (2002).

3.1.3 TÉCNICAS DE TESTE DE SOFTWARE

Atualmente existem muitas maneiras de se testar um software. Mesmo assim, existem as técnicas que sempre foram muito utilizadas em sistemas desenvolvidos sobre linguagens estruturadas que ainda hoje têm grande valia para os sistemas orientados a objeto. Apesar dos paradigmas de desenvolvimento serem diferentes, o objetivo principal destas técnicas continua a ser o mesmo: encontrar falhas no software.

De acordo com Dias Neto (2010) as técnicas de teste são classificadas de acordo com a origem das informações utilizadas para estabelecer os requisitos de teste. Elas contemplam diferentes perspectivas do software, e impõe-se a necessidade de se estabelecer uma estratégia de teste que contemple as vantagens e os aspectos complementares dessas técnicas. As técnicas existentes são: técnica funcional e estrutural.

Outras técnicas de teste que podem e devem ser utilizadas de acordo com necessidades de negócio ou restrições tecnológicas são: teste de desempenho, teste de usabilidade, teste de carga, teste de stress, teste de confiabilidade e teste de recuperação.

Segundo Dias Neto (2010) alguns autores chegam a definir uma técnica de teste caixa cinza, que seria um mesclado do uso das técnicas de caixa preta e caixa branca, mas, como toda execução de trabalho relacionado à atividade de teste utilizará simultaneamente mais de uma técnica de teste, é recomendável que se fixe os conceitos primários de cada técnica.

3.2 PADRÕES ALIMENTARES

Em preludio ao papel desempenhado pelas dietas na saúde nutricional dos indivíduos, analisar e identificar padrões alimentares se faz de grande importância. Informações como quantidade e combinações entre os alimentos abre um leque de possibilidades para estudos referentes ao papel que estes desempenham em determinados fatores de risco à saúde (SILVA; LYRA; LIMA, 2016), levando em consideração que os alimentos e nutrientes não são consumidos separadamente, mas sim, em conjunto (TAVARES *et al.* 2014).

No Brasil, os hábitos alimentares dos jovens são preocupantes. Segundo os resultados de uma pesquisa publicada pela Universidade de Cambridge em 2014, a alimentação dos adolescentes era marcada pelo consumo regular de doces e refrigerantes e menos de 30% consumiam vegetais. O estudo ainda apontou que a maior parte dos expostos à alimentação incorreta eram os adolescentes da região sul e mais velhos (AZEREDO *et al.* 2015).

A grande problemática acerca dos hábitos alimentares não saudáveis são suas consequências, tendo em vista seu favorecimento para as chamadas *DCNT* – Doenças Crônicas Não Transmissíveis como diabetes, câncer e hipertensão (TOCANTINS, 2017). De acordo com dados publicados pelo Ministério da Saúde, 72,4% dos óbitos registrados em 2009 foram causadas por alguma DCNT (SAÚDE, 2011, p. 121).

Devido a essa ligação entre alimentação-doença, desenvolver ferramentas que auxiliem no levantamento de dados acerca da alimentação dos jovens se faz de

grande importância para estudos. Levando em consideração a era atual, a digital, uma aplicação mobile seria a ferramenta ideal devido a sua facilidade e abrangência no meio jovem, além disso, os dados obtidos digitalmente são mais fáceis de serem trabalhados, mediante ao grande número de ferramentas que as plataformas e linguagens de desenvolvimento oferecem.

3.3 TABELA BRASILEIRA DE COMPOSIÇÃO DE ALIMENTOS

A considerar as questões acerca dos alimentos, seus nutrientes e suas diversas características, levantar e armazenar seus dados se faz de muito útil para diversos fins, com destaque aos nutricionais. Sendo assim, o NEPA (Núcleo de Estudos e Pesquisas em Alimentação) – UNICAMP desenvolveu o projeto TACO, uma “*tabela*” contendo as características nutricionais da maioria dos alimentos contidos no território brasileiro.

O projeto em questão contou com o financiamento do Ministério do Desenvolvimento Social e Combate à Fome e do Ministério da Saúde. Seu desenvolvimento foi baseado em fases envolvendo diversos pesquisadores renomados na área e os estudos realizados foram analisados por laboratórios altamente capacitados (NEPA). O resultado foi um trabalho rico em informações e de grande utilidade pública, disponibilizado pelo site do NEPA.

4 METODOLOGIA

Esta seção apresenta a metodologia para a investigação e projeção de um aplicativo para auxílio na análise do comportamento alimentar de estudantes por meio de uma aplicação móvel. A metodologia adotada envolveu as etapas de levantamento de requisitos, estruturação e representação do conhecimento, modelagem e implementação do aplicativo e testes. Para cumprir essas etapas foram desenvolvidas as seguintes atividades:

- a) Levantamento bibliográfico;
- b) definição da arquitetura do aplicativo proposto;
- c) levantamento de recursos tecnológicos a serem utilizados para implementação do aplicativo;
- d) desenvolvimento e implementação da ferramenta;
- e) avaliação do aplicativo.

4.1 LEVANTAMENTO BIBLIOGRÁFICO

Tendo em vista que o trabalho teve como produto uma aplicação móvel, os estudos foram direcionados para seleção de uma linguagem de programação que melhor se adequasse ao desenvolvimento de aplicações móveis, modelagem de banco de dados e de acordo com o tema proposto, levantamento bibliográfico sobre o hábito alimentar de estudantes. Primeiramente foi realizado um levantamento bibliográfico tendo como base livros, sites com fontes verificadas e artigos científicos para auxiliar no desenvolvimento do projeto.

De acordo com os dados levantados, foram postos em prática as técnicas para o modelamento e posteriormente o desenvolvimento de um aplicativo móvel para ser implementado e testado.

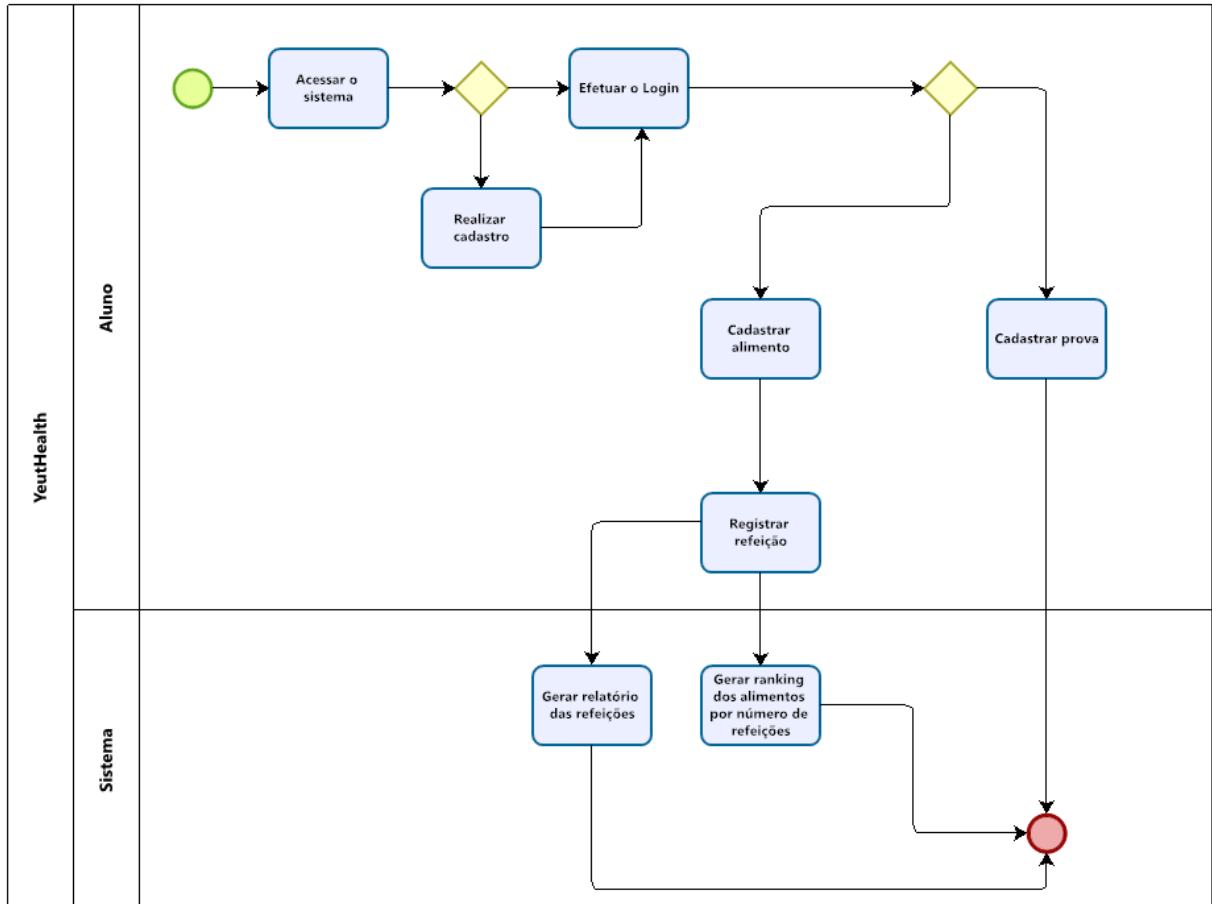
4.2 DEFINIÇÃO DA ARQUITETURA DO SISTEMA

O sistema é responsável pelas seguintes atividades:

- a) **Atividade 1:** Cadastrar e acessar a aplicação por meio de login autenticado por token de acesso.
- b) **Atividade 2:** Gerenciar os alimentos para serem utilizados no cadastro das refeições.
- c) **Atividade 3:** Gerenciar as refeições por meio de registro, possível alteração e deleção do mesmo.
- d) **Atividade 4:** Registrar datas em que os usuários irão realizar provas e/ou trabalhos, sendo possível descrever a matéria e conteúdo a ser estudado.
- e) **Atividade 5:** Gerar um relatório a respeito das alimentações dos alunos de forma individual, no perfil de cada aluno.
- f) **Atividade 6:** Exibir de forma estruturada os alimentos mais consumidos pelos estudantes.

Para elucidar, foi elaborado um diagrama de processos que pode ser visualizado na Figura 5.

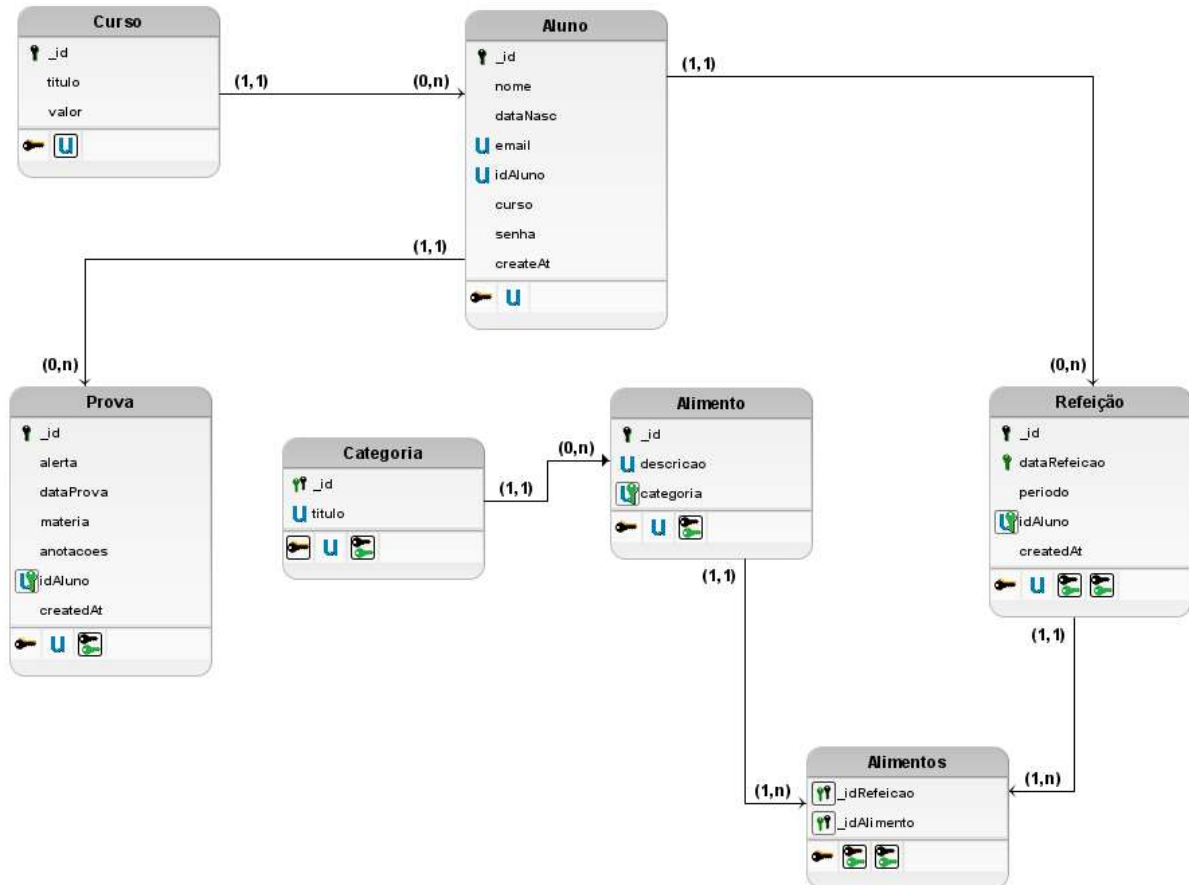
Figura 5 – Diagrama de processos.



Fonte: Elaborado pelo autor.

A partir do diagrama de processos, foi possível visualizar o projeto de outro ângulo, elucidando o fluxo do sistema, tendo como o próximo passo o levantamento de requisitos e sua representação por meio da modelagem UML - Unified Modeling Language. Segundo Guedes (2011), a UML tem como objetivo auxiliar no desenvolvimento do projeto sendo responsável por definir requisitos, comportamentos, estrutura lógica e a dinâmica dos processos, essas por sua vez, podem ser definidas antes do início do desenvolvimento do projeto, sendo a abordagem adotada no projeto em questão. O resultado do diagrama desenvolvido pode ser conferido na Figura 6.

Figura 6 – UML.



Fonte: Elaborado pelo autor.

4.3 LEVANTAMENTO DE RECURSOS TECNOLÓGICOS

Os recursos tecnológicos utilizados para o desenvolvimento do projeto foram divididos em duas categorias: Hardware e Software.

4.3.1 HARDWARE

Para o desenvolvimento do aplicativo, foi utilizado como principal ferramenta um notebook contendo Windows 10 Home Single Language (64 bits), processador Intel(R) Core(TM) i3-4005U CPU @ 1.70GHz 1.70 GHz e 6 GB de memória RAM.

Os eventuais testes foram realizados em um smartphone, utilizando como sistema operacional Android 10 Samsung One UI 2.0, processador 4x 2.8 GHz Kryo 385 + 4x 1.7 GHz Kryo 385 e 6 GB de memória RAM.

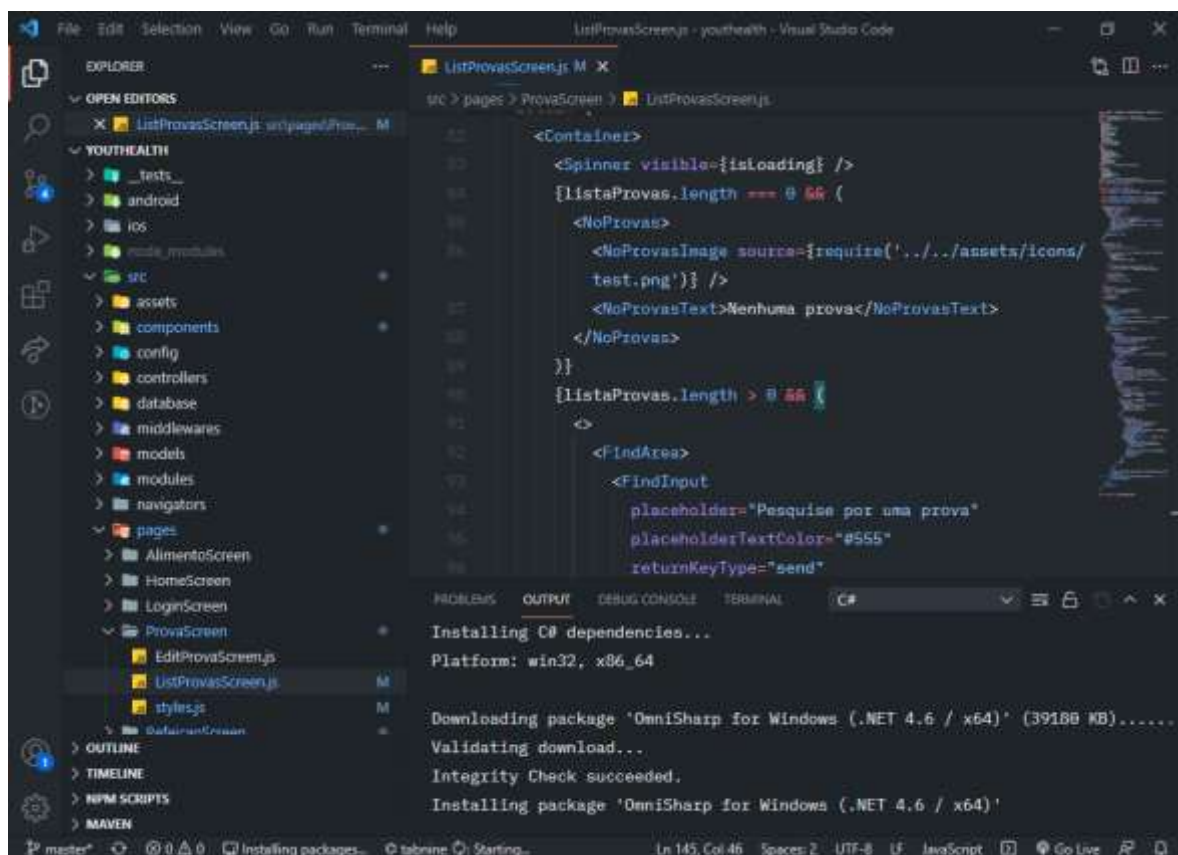
O projeto contou com um logotipo desenvolvido por meio de um iPad de 7ª geração com processador A10 Fusion.

A escolha dos dispositivos foi baseada nos equipamentos que o pesquisador já possuía, os mesmos foram suficientes para suprir as necessidades do projeto.

4.3.2 SOFTWARE

O desenvolvimento do código fonte da aplicação foi realizado utilizando o Visual Studio Code ilustrado por meio da Figura 7, sendo escolhido por ser amplamente conhecido e reconhecido por ser leve e intuitivo. A ferramenta apresenta a possibilidade de instalação de *plugins*, facilitando e agilizando o desenvolvimento do código, tornando-o mais limpo, organizado, semântico e menos suscetível a erros.

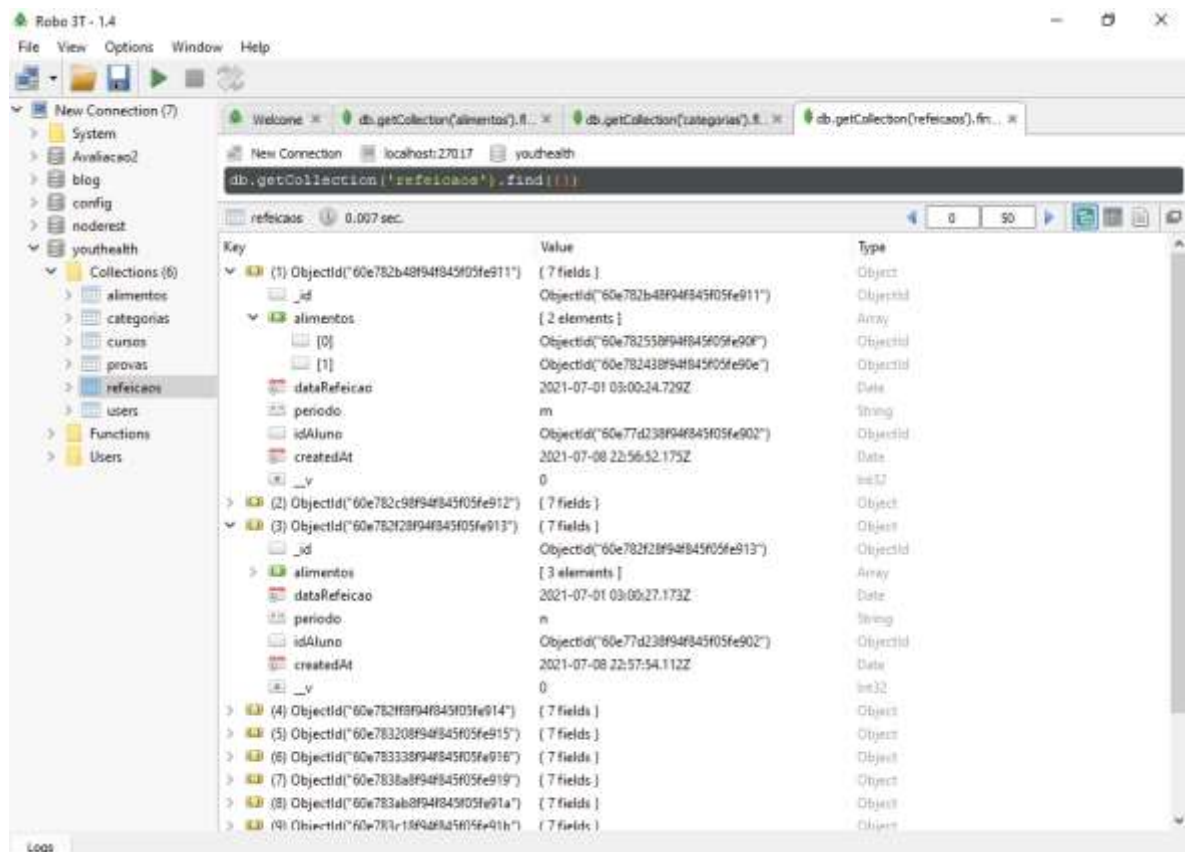
Figura 7 – VISUAL STUDIO CODE.



Fonte: Elaborado pelo autor.

Para manutenção e testes no banco de dados, foi utilizada a versão gratuita do software Robo 3T (Figura 8). A aplicação permite visualizar o banco de dados de forma completa, possibilitando realizar os testes durante o desenvolvimento do projeto.

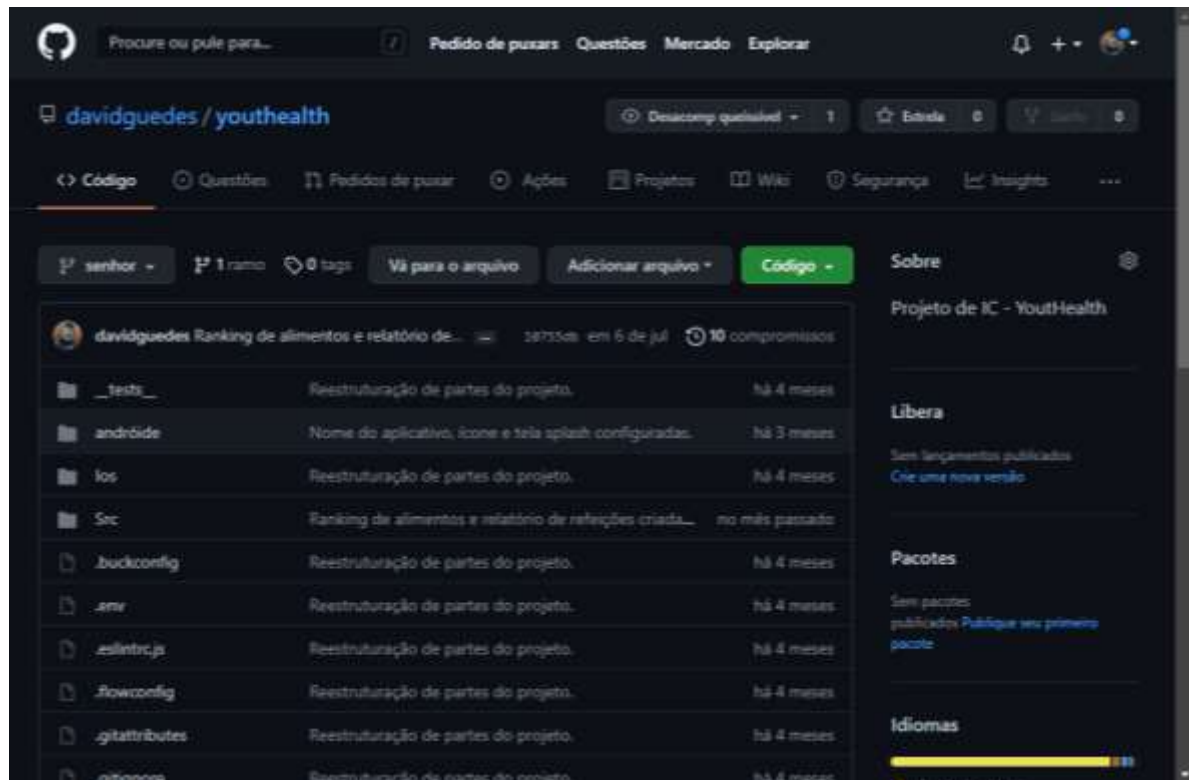
Figura 8 – Robo 3T.



Fonte: Elaborado pelo autor.

Durante o desenvolvimento, foi utilizado o sistema de controle de versões Git e a plataforma GitHub (Figura 9) para fazer o gerenciamento de versões e backup de segurança do projeto, dessa forma, foi registrado todo o processo de desenvolvimento.

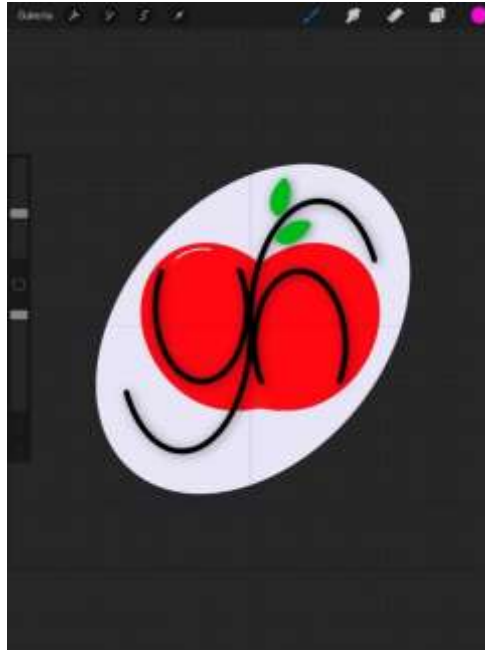
Figura 9 – GitHub.



Fonte: Elaborado pelo autor.

O logotipo do projeto foi desenvolvido fazendo uso do aplicativo Procreate (Figura 10). O aplicativo em questão foi obtido por meio da loja do dispositivo.

Figura 10 – Procreate.



Fonte: Elaborado pelo autor.

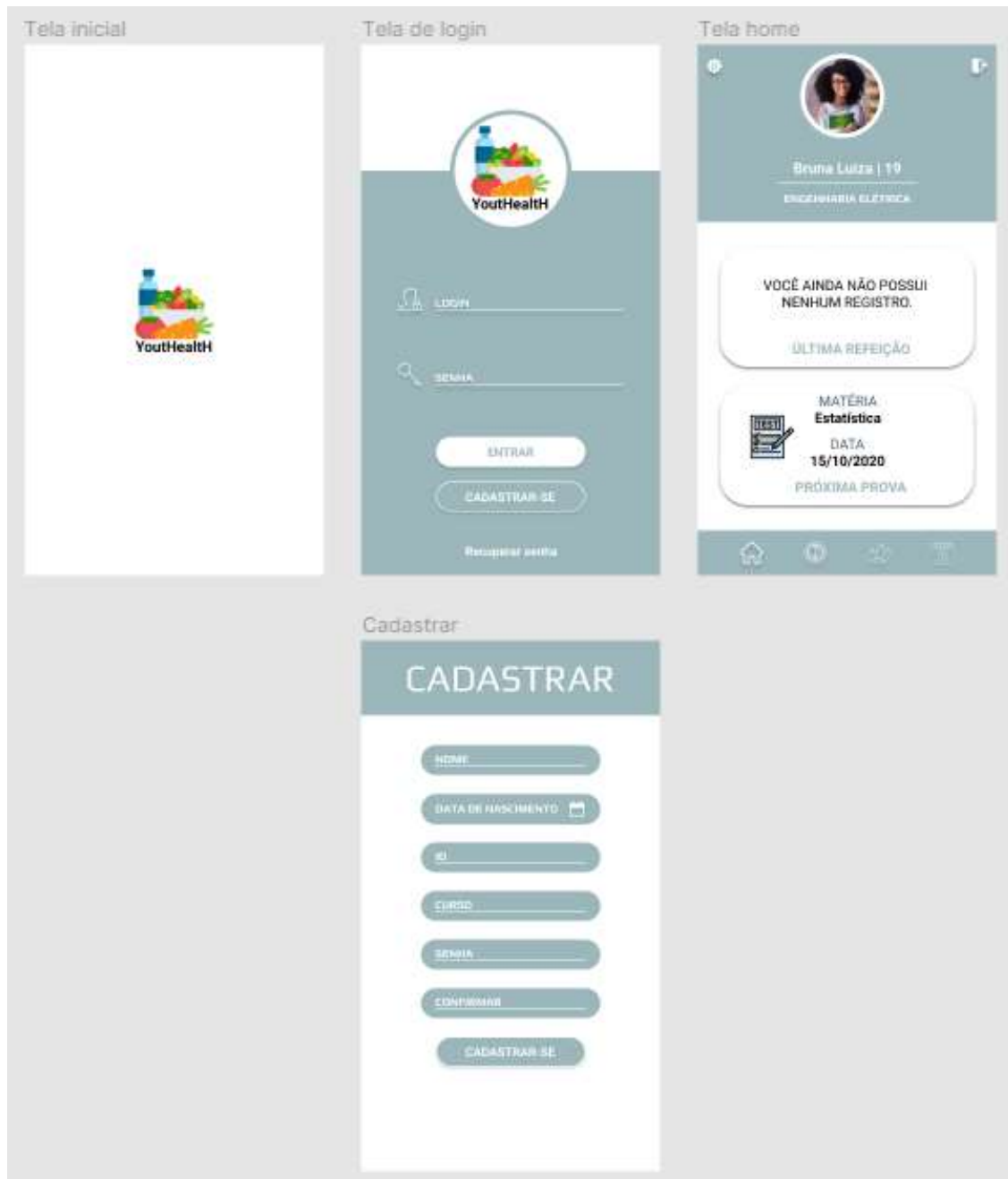
4.4 DESENVOLVIMENTO DA FERRAMENTA

O processo de desenvolvimento da ferramenta foi dividido em três etapas: a começar pela modelagem e construção do banco de dados, que foi realizado utilizando o MongoDB, banco de dados classificado como NoSQL e orientado a arquivos. Sua escolha se deu por sua grande popularidade, conteúdo abrangente quanto a documentação e bom desempenho, sendo uma ótima escolha para aplicativos mobile com pouca complexidade nos relacionamentos das bases de dados, como no caso do projeto proposto. Para essa etapa, foi utilizado os dados levantados com a UML. Foram definidos as classes, os atributos e os devidos relacionamentos, resultando em um banco de dados estruturado e definido conforme as necessidades do projeto.

Posteriormente, foi pensada a parte gráfica do sistema, levando em conta o público-alvo – estudantes. Para essa etapa, foi realizada a prototipagem do projeto, considerando a parte gráfica e o fluxo do sistema, sendo utilizado como base o diagrama de processos elaborado anteriormente. O protótipo foi desenvolvido por meio da ferramenta online, Figma, e obteve como resultado um esquema para facilitar no desenvolvimento da aplicação.

A Figura 11 ilustra uma das partes do protótipo. Inicialmente, ao lado esquerdo, é apresentada a tela inicial que é exibida ao abrir a aplicação. Ao centro é mostrada a tela de login por onde o usuário pode efetuar a entrada ao sistema ou ir para a tela de cadastro, exibida abaixo, onde se faz necessário preencher o formulário. Ambas as telas redirecionam para a tela *home*, ilustrada ao lado esquerdo, na qual o usuário possui o menu de acesso às várias áreas do sistema bem como a opção de alterar o cadastrado (ícone de engrenagem ao lado superior esquerdo da tela) ou fazer o *logout* (ícone de porta aberta ao lado direito da tela).

Figura 11 – Protótipo desenvolvido no Figma – Tela inicial, login, cadastro e home

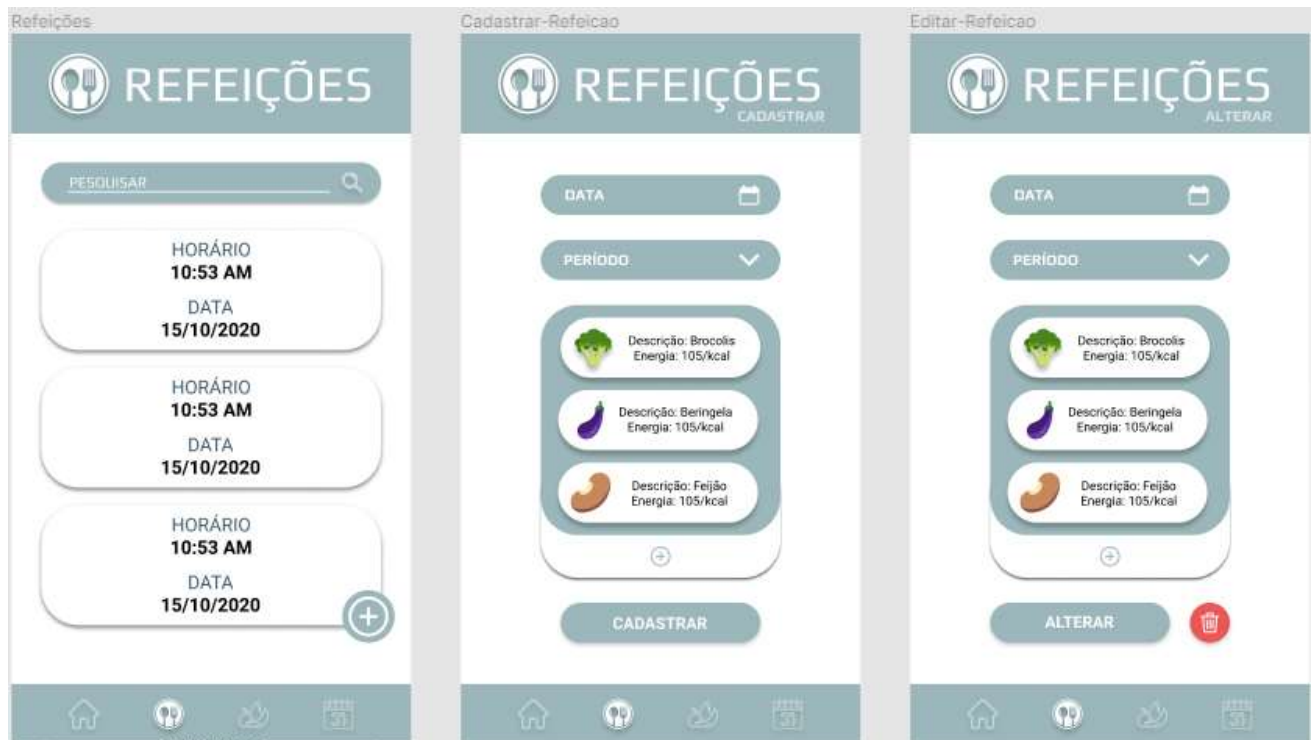


Fonte: Elaborado pelo autor.

Outras partes do protótipo são ilustradas na Figura 12. Inicialmente é apresentada a tela onde é possível visualizar as refeições feitas pelo usuário. As refeições são exibidas em pequenos blocos com algumas informações referentes ao horário e data da refeição. A parte inferior direita da tela apresenta um botão com o símbolo "+", responsável por redirecionar o usuário para a segunda tela ilustrada ao centro (tela de cadastro da refeição). Por meio dela, o usuário informa os dados a

respeito da refeição e faz o seu registro. A última tela apresentada ao lado direito é responsável por editar uma refeição já cadastrada.

Figura 12 – Protótipo desenvolvido no Figma – Tela de refeições



Fonte: Elaborado pelo autor.

A Figura 13 apresenta ao lado esquerdo a tela de alimentos, onde é feita a listagem de todos os alimentos cadastrados e possíveis para serem inseridos nos registros das refeições. Ao centro e ao lado direito são mostradas as telas de provas. A figura ao meio apresenta a tela responsável pelo cadastro das provas, sendo necessário informar os dados no formulário em questão. Ao lado direito é mostrada a tela de edição da prova, responsável por alterar as informações cadastradas anteriormente.

Figura 13 – Protótipo desenvolvido no Figma – Tela de alimentos e provas



Fonte: Elaborado pelo autor.

Por fim, foi realizado o estudo e colocado em prática os aprendizados acerca das tecnologias de desenvolvimento de software, fazendo uso da ferramenta Visual Studio Code e das bibliotecas React Native e Node JS. As bibliotecas foram escolhidas com base na linguagem de programação, levando em consideração que ambas utilizam o JavaScript, facilitando no desenvolvimento do código e minimizando o tempo despendido para aprendizado da sintaxe da linguagem, sendo possível dedicar mais tempo a escrita do código e testes, processo que demandou mais tempo no projeto.

4.5 AVALIAÇÃO

No decorrer do desenvolvimento do projeto, a aplicação foi avaliada pelo autor durante todo o período, do ponto de vista de boas práticas de programação, funcionalidade e interfaces, levando em consideração que o mesmo é um estudante universitário do curso de Ciência da Computação. Durante a avaliação foi possível

verificar possibilidades de melhoria e mudanças tanto na parte visual quanto funcional do projeto. Os resultados podem ser visualizados no capítulo seguinte.

5 RESULTADOS

A seguir irão ser apresentados os resultados da pesquisa (o software, em si, desenvolvido), com ilustrações e explicações sobre a aplicação.

5.1 BANCO DE DADOS

Por meio da UML, o banco de dados foi projetado e utilizado na aplicação. A seguir, é ilustrado por meio da Figura 14 um registro feito no MongoDB Compass de duas refeições cadastradas no sistema.

Figura 14 – PrintSreen de registros da collection responsável pelas refeições.

```

    _id: ObjectId("60e782b48f94f845f05fe911")
  ▾ alimentos: Array
      0: ObjectId("60e782558f94f845f05fe90f")
      1: ObjectId("60e782438f94f845f05fe90e")
    dataRefeicao: 2021-07-01T03:00:24.729+00:00
    periodo: "m"
    idAluno: ObjectId("60e77d238f94f845f05fe902")
    createdAt: 2021-07-08T22:56:52.175+00:00
    __v: 0

    _id: ObjectId("60e782c98f94f845f05fe912")
  ▾ alimentos: Array
      0: ObjectId("60e782118f94f845f05fe90c")
      1: ObjectId("60dde5514a726e335018db04")
      2: ObjectId("60e782038f94f845f05fe90b")
      3: ObjectId("60dde5584a726e335018db05")
    dataRefeicao: 2021-07-01T03:00:51.852+00:00
    periodo: "t"
    idAluno: ObjectId("60e77d238f94f845f05fe902")
    createdAt: 2021-07-08T22:57:13.078+00:00
    __v: 0

```

Fonte: Elaborado pelo autor.

5.2 APLICAÇÃO

A seguir, são ilustradas e descritas as telas do sistema desenvolvido em React Native. Por meio dele, o aluno pode realizar o cadastro de seu perfil, de alimentos, das refeições e de suas provas.

Por meio da Figura 15, é possível verificar ao lado esquerdo a tela de login do sistema. O botão “Cadastrar-se” é responsável por levar o usuário a tela de cadastro, ilustrado na segunda parte da imagem. Para realizar o cadastro, o usuário precisa informar todos os campos presentes no formulário, não sendo possível o cadastro de mais de um perfil com o mesmo ID de aluno ou e-mail.

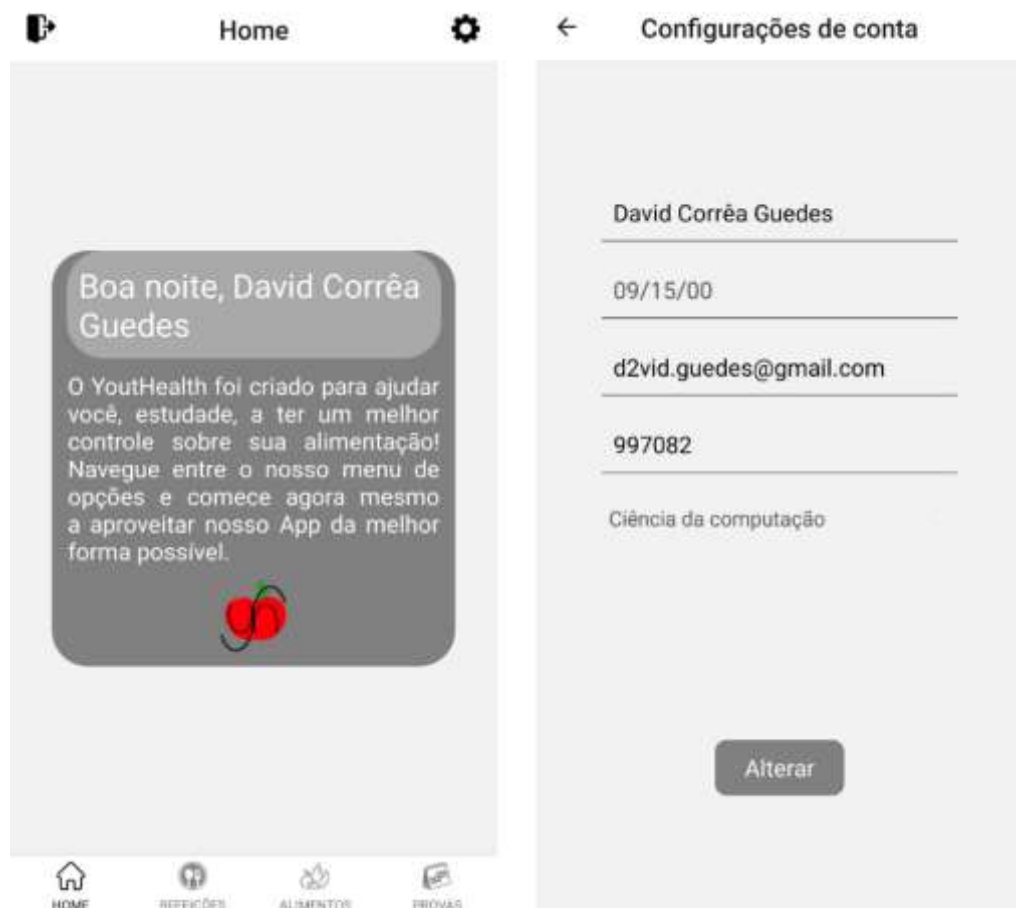
Figura 15 – Tela de login e de registro.

The figure displays two mobile application screens side-by-side. The left screen is the login page, featuring a logo of two red apples with a black swirl. Below the logo are two input fields labeled 'Digite seu login' and 'Digite sua senha'. At the bottom of the login screen are two buttons: 'Entrar' and 'Cadastrar-se'. The right screen is the registration page, titled 'Cadastrar-se' with a back arrow. It contains seven input fields: 'Nome completo', '07/08/21', 'Email', 'ID de aluno', 'Ciência da computação', 'Senha', and 'Confirmar senha'. A 'Cadastrar-se' button is located at the bottom of the registration form.

Fonte: Elaborado pelo autor.

Ao fazer o login ou se registrar, é gerado um token único da sessão que possui duração de 60 minutos e o usuário é redirecionado para a tela inicial, mostrada na Figura 16, ao lado esquerdo. É possível perceber na parte inferior o menu de navegação onde o usuário pode acessar de fato as funções do sistema. O primeiro ícone é referente a tela home, o segundo redireciona a página das refeições, o seguinte leva o usuário aos alimentos e o último leva ao gerenciamento das provas. Na parte superior da tela é possível visualizar o botão responsável pelo *logout* (lado esquerdo) e o botão responsável por redirecionar o usuário a tela de configurações de seu perfil (lado direito), tela mostrada na segunda parte da imagem, onde o usuário possui acesso as configurações de seu perfil, sendo possível alterá-las.

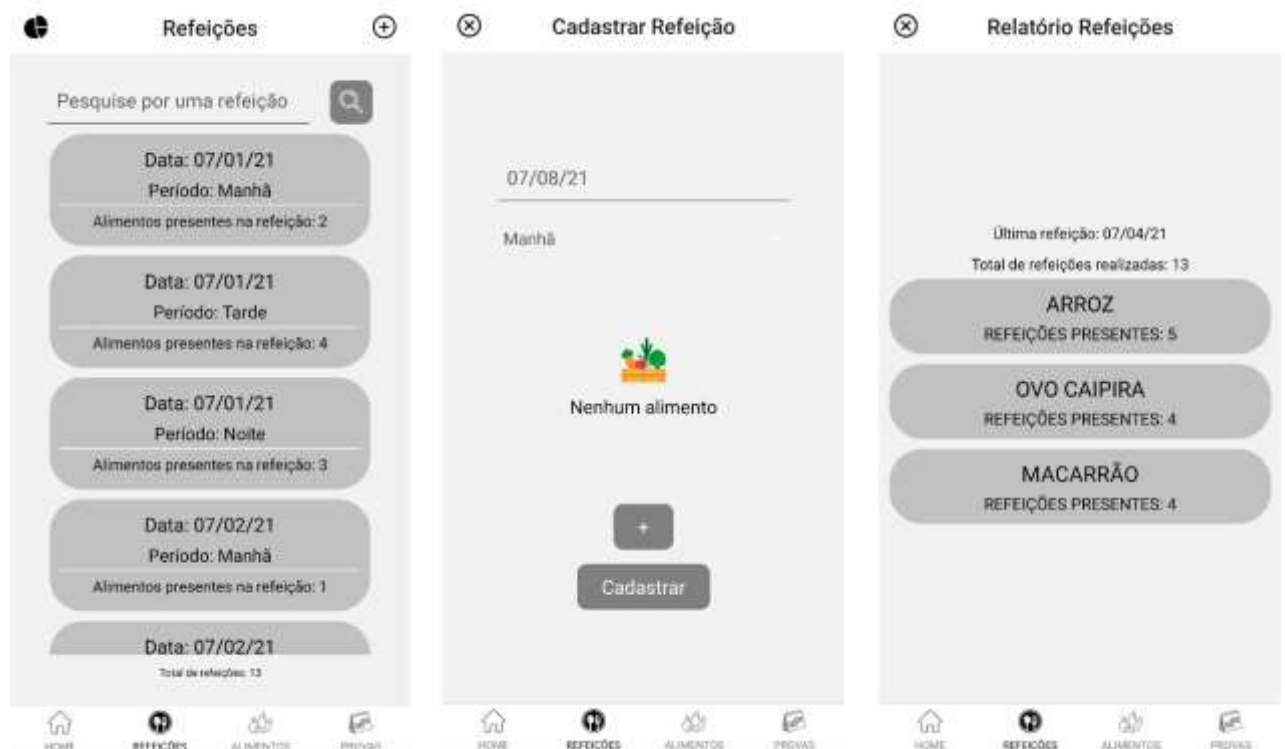
Figura 16 – Tela inicial e tela de configurações da conta.



Fonte: Elaborado pelo autor.

A Figura 17 é dividida em três imagens, ilustrando a tela de refeições e suas divisões, responsáveis por todo o fluxo desses registros. A primeira parte da imagem ilustra a tela em que o usuário se depara ao acessar a aba Refeições e, por ela, é possível visualizar todas as refeições que o usuário realizou, apresentando o registro de data, período e alimentos. Ao clicar sobre algum dos registros, o usuário é redirecionado a página de edição, onde pode alterar alguma das informações ou até mesmo deletar o registro. A segunda parte da imagem representa a tela de cadastro das refeições que pode ser acessado pelo ícone “+” na parte superior direita da primeira imagem. Ao acessar a página, o usuário necessita informar a data da refeição, o período, sendo dividido em manhã, tarde ou noite e os alimentos presentes na refeição, necessitando ao menos um alimento para efetuar o registro. A terceira e última parte da imagem representa o relatório das refeições, acessado por meio do ícone de gráfico presente na parte superior esquerda da primeira imagem. O relatório é individual por aluno e apresenta a data da última refeição, o total de refeições realizadas e os três alimentos mais consumidos pelo usuário.

Figura 17 – Telas referentes aos registros das refeições.



Fonte: Elaborado pelo autor.

Em sequência, na Figura 18 são apresentadas as telas referentes às ações realizadas acerca dos alimentos no sistema. Na primeira parte da imagem, é possível visualizar a listagem de todos os alimentos cadastrados no sistema descritos por descrição e categoria. Ao clicar sobre algum dos registros, o usuário é redirecionado para a aba de edição do alimento, onde é possível alterar a descrição, categoria ou até mesmo excluir o registro, exceto se ele estiver presente em alguma refeição, incluindo os registros dos demais usuários.

Figura 18 – Telas referentes aos alimentos.



Fonte: Elaborado pelo autor.

Por último, pode-se visualizar por meio da Figura 19 as telas referentes ao fluxo do cadastro de provas do usuário. A primeira parte da imagem ilustra os registros de provas, podendo ser visualizado a data da prova, a matéria, uma descrição se houver e uma marcação de alerta, se este estiver marcado no cadastro da prova em questão. Os registros podem conter duas cores diferentes, sendo a cor cinza provas com datas

futuras e as verdes como provas com data igual ou inferior a data atual. Ao lado direito é possível visualizar a tela de cadastro das provas, acessada por meio do ícone “+” na parte superior direito da primeira imagem. Ao acessar a essa tela, o usuário precisa informar uma data de quando a prova irá ocorrer, a matéria também se faz obrigatória e os campos de descrição e alerta sendo opcionais, encerrando as funções do sistema.

Figura 19 – Telas referentes as provas.



Fonte: Elaborado pelo autor.

6 CONSIDERAÇÕES FINAIS

O presente projeto foi elaborado tendo em vista os hábitos alimentares dos estudantes, levando em consideração as demandas do dia a dia aliado as tarefas de um curso acadêmico, que podem influenciar nas mudanças dos hábitos alimentares por questões variadas como, por exemplo, a falta de tempo ou estresse. Tendo isso em vista esse cenário o sistema tem como finalidade levantar dados a respeito das refeições sendo necessário que o usuário utilize o sistema de forma recorrente, para gerar dados precisos e de grande utilidade.

Para possíveis trabalhos futuros, o sistema possui a coleta de vários dados que podem ser relevantes para novos formatos de relatórios e bons resultados podem surgir a partir destes, levando em conta a vasta exploração que o projeto em si ainda pode vir a ter, tendo como objetivo auxiliar toda a comunidade.

O desenvolvimento do projeto foi de grande avalia, sendo necessário empenho e estudo acerca de diversas ferramentas, elaboração de diagramas, levantamento de requisitos, aprendizado de uma linguagem de programação (JavaScript) e bibliotecas como o React Native e Node JS, além do estudo sobre o banco NoSQL MongoDB e sua implementação com a linguagem escolhida. De uma forma geral, o projeto colocou em prática diversos dos assuntos tratados em sala de aula durante a graduação, sendo extremamente valioso para a fixação e busca do aprendizado.

REFERÊNCIAS

AZEREDO, C. M. *et al.* Dietary intake of Brazilian adolescents. **Public Health Nutrition**, [s.l.], v.18, n.7, p. 1215-1224, May. 2015. Disponível em: <https://doi.org/10.1017/S1368980014001463>. Acesso em: 07 mar. 2020.

BRASIL. Ministério da Saúde. **SAÚDE BRASIL 2010**. Brasília, DF: Ministério da Saúde, 2011. Disponível em: http://bvsms.saude.gov.br/bvs/publicacoes/saude_brasil_2010.pdf. Acesso em: 07 mar. 2020.

CRAIG, R. D.; JASKIEL, S. P. **Systematic Software Testing**. Norwood: Artech House Publishers, 2002.

DIAS NETO, A. C. Introdução a Teste de Software. **Revista Engenharia de Software Magazine**, [s.l.], n. 1, p. 54-59, jun. 2010.

GUEDES, G. **UML 2: uma abordagem prática**. 2 ed. São Paulo: Novatec Editora, 2011.

INCA. Brasileiros atingem maior índice de obesidade nos últimos treze anos. *In: inca.gov.*, [s.l.], 25 jul. 2019. Disponível em: <https://www.inca.gov.br/noticias/brasileiros-atingem-maior-indice-de-obesidade-dos-ultimos-treze-anos-de-acordo-com-pesquisa>. Acesso em: 06 fev. 2020.

LOURENÇO, A. S. A. **Ingestão de alimentos como mecanismo de regulação da ansiedade**. 2016. Dissertação (Mestrado em Psicologia Clínica e da Saúde) - Escola de Psicologia e Ciências da Vida, Universidade Lusófona de Humanidades e Tecnologias, Lisboa, 2016. Disponível em: <http://recil.grupolusofona.pt/bitstream/handle/10437/7230/TESE%20Ana%20Louren%C3%A7o.pdf?sequence=1>. Acesso em: 17 fev. 2020.

NÚCLEO DE ESTUDOS E PESQUISAS EM ALIMENTAÇÃO. Tabela Brasileira de Composição de Alimentos. Projeto taco. *In: nepa.unicamp.*, Campinas, [2011]. Disponível em: <http://www.nepa.unicamp.br/taco/tabela.php?ativo=tabela>. Acesso em: 08 mar. 2020.

PRESSMAN, R. **Engenharia de Software: uma abordagem profissional**. 7 ed. São Paulo: McGraw-Hill, 2011.

RAPPS, S.; WEYUKER, E. J. **Data Flow Analysis Techniques for Program Test Data Selection**. In: 6th International Conference on Software Engineering, Tokio, Japan, 1982, p. 272-78.

ROCHA, A. R.; MALDONADO, J. C.; WEBER, K. C. (Orgs.). **Qualidade de Software: teoria e prática**. São Paulo: Prentice Hall, 2001.

ROMERO, D. L. Brasileiros estão entre os maiores consumidores de 'fast food' do mundo. **EL PAÍS**, Madri; São Paulo, jan. 2016. Disponível em: https://brasil.elpais.com/brasil/2016/01/21/economia/1453403379_213071.html. Acesso em: 16 fev. 2020.

SANTOS, C. Novos hábitos alimentares são a causa do aumento da obesidade. **Jornal da USP**, São Paulo, mai. 2017. Disponível em: <https://jornal.usp.br/atualidades/novos-habitos-alimentares-sao-a-causa-do-aumento-da-obesidade/>. Acesso em: 06 fev. 2020.

SILVA, A. M. B. *et al.* Jovens Insatisfeitos com a Imagem Corporal: Estresse, Autoestima e Problemas Alimentares. **Psico-USF**, Campinas, v. 23, n. 3, p. 483-495, jul. 2018. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-82712018000300483&lng=en&nrm=iso. Acesso em: 17 fev. 2020.

SILVA, D. F. O.; LYRA, C. O.; LIMA, S. C. V. C. Padrões alimentares de adolescentes e associação com fatores de risco cardiovascular: uma revisão sistemática. **Ciência & Saúde Coletiva**, Rio de Janeiro, v. 21, n. 4, p. 1181-1196, abr. 2016. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1413-81232016000401181&lng=pt&nrm=iso. Acesso em: 07 mar. 2020.

SOMMERVILLE, Ian. **Engenharia de Software**. 8 ed. São Paulo: Pearson AddisonWesley, 2007.

STICE, E; SHAW, H. E. Role of body dissatisfaction in the onset and maintenance of eating pathology: a synthesis of research findings. **Journal of psychosomatic research**, Oxford, v. 53, n. 5, p. 985-993, Nov. 2002. Disponível em: <https://www.ncbi.nlm.nih.gov/pubmed/12445588>. Acesso em: 16 fev. 2020.

TAVARES, L. F. *et al.* Padrões alimentares de adolescentes brasileiros: resultados da Pesquisa Nacional de Saúde do Escolar (PeNSE). **Cadernos de Saúde Pública**, Rio de Janeiro, v. 30, n. 12, p. 2679-2690, dec. 2014. Disponível em: http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-311X2014001202679&lng=en&nrm=iso. Acesso em: 07 mar. 2020.

TOCANTINS. Secretaria da Saúde. **Brasileiros atingem maior índice de obesidade nos últimos treze anos**. [s.l.], [2017]. Disponível em: <https://saude.to.gov.br/vigilancia-em-saude/doencas-transmissiveis-e-nao-transmissiveis/-dant/doencas-cronicas-nao-transmissiveis>. Acesso em: 07 de mar. de 2020.

WAZLAWICK, R. S. **Engenharia de Software: Conceitos e Práticas**. 1 ed. Rio de Janeiro: Elsevier, 2013.

WELLS, S. Fast Food Restaurants Have Expanded More Than Their Menus. **The Brink**, [s.l.], Apr. 2019. Disponível em: <http://www.bu.edu/articles/2019/fast-food-expansion/>. Acesso em: 15 fev 2020.