

# Casefy: Uma Aplicação para Estudo de Casos Clínicos

João Victor R. Carvalho<sup>1</sup>, Alcemir Rodrigues Santos<sup>1</sup>, Joilane Alves Pereira Freire<sup>2</sup>

<sup>1</sup>Laboratório de Engenharia de Software (LES) – Universidade Estadual do Piauí (UESPI)  
Piripiri - PI - Brasil

<sup>2</sup>Departamento de Nutrição – Universidade Federal do Piauí (UFPI)  
Picos - PI - Brasil

jvrcarvalho@aluno.uespi.br, alcemir@prp.uespi.br, joilane@ufpi.edu.br

**Abstract.** *Teaching clinical cases is fundamental in the training of new professionals not only in the area of nutrition but also in other areas of health. Despite the importance of teaching cases, it often ends up being applied in a way that is unattractive to students, causing them to become unmotivated with the subject. In view of this problem, programs have been developed that attempt to apply a more dynamic and attractive form of teaching. Therefore, this work presents a multiplatform application for teaching clinical cases that is more practical, accessible and attractive for nutrition students. Through a lot of work and research into nutrition, it was possible to deliver a functional application that delivers everything that was proposed.*

**Resumo.** *O ensino de casos clínicos é fundamental na formação de novos profissionais não só da área de nutrição como também em outras áreas da saúde. Apesar da importância do ensino de casos, muitas vezes ela acaba sendo aplicada de uma forma nada atraente para os alunos, fazendo com que os eles fiquem desmotivados com a matéria. Tendo em vista esse problema, foram desenvolvidos programas que tentam aplicar uma forma de ensino mais dinâmica e atraente. Sendo assim, este trabalho apresenta um aplicativo multiplataforma para ensino de casos clínicos sendo mais prático, acessível e atraente para os estudantes de nutrição. Por meio de muito trabalho e pesquisa sobre nutrição foi possível entregar um aplicativo funcional que entrega tudo que foi proposto.*

## 1. Introdução

O ensino de casos clínicos é fundamental na formação de novos profissionais não só da área da nutrição, como também em todas as áreas da saúde. No entanto, apesar da importância do ensino de casos para os alunos de nutrição, muitas vezes a matéria é ministrada pelo professor de forma desinteressante fazendo com que os alunos percam o interesse pela disciplina. Isso ocorre não só pelo uso de técnicas tradicionais de ensino, mas como também a falta de uma inclusão tecnológica.

Segundo Galdeano [Galdeano et al. 2003] casos clínicos é uma das formas mais antigas e mais importantes no ensino de áreas da saúde. O estudo desses casos permite que os estudantes, profissionais e pesquisadores tenham uma visão geral sobre uma situação real ou fictícia de um paciente, assim facilitando que sejam tomadas decisões mais assertivas com relação à criação de dietas, planos alimentares e diagnósticos. Sendo assim, fica bem clara a importância do ensino de casos clínicos.

Apesar da importância do ensino da matéria de casos clínicos, muitos educadores utilizam métodos de ensino tradicionais para a transmissão do conteúdo, o que acaba por tornar a matéria nada atraente para os alunos, fazendo com que eles fiquem desmotivados ou desinteressados pelo conteúdo. Por se tratar de uma matéria tão fundamental é importante que ela seja ensinada e aplicada de forma prática.

Segundo Meira e Borges [Souza e Dourado 2015], a utilização de um método de ensino tradicional serve apenas para um aprendizado de conceitos teóricos. Nesse tipo de ensino os alunos são ensinados basicamente a decorar o conteúdo para depois responder alguma avaliação. Apesar da baixa eficiência de tal modelo de ensino o mesmo ainda é bastante usado pelos professores.

Com a finalidade de atacar este problema, foram desenvolvidos *softwares* e *Web sites* que fornecem estudos de casos clínicos de uma forma mais dinâmica. Cada aplicativo tem suas próprias características, alguns permitem criação de casos clínicos, outros fornecem um *feedback* ao final de cada atividade. Sendo assim, este trabalho busca juntar tanto a capacidade de criar casos clínicos como também estudar cada um deles de forma individual.

O aplicativo proposto tem o objetivo de proporcionar uma forma de estudo de casos clínicos mais dinâmica, prática e completa. O programa conta com casos clínicos e questionários referentes a cada caso pre armazenado, também dando a liberdade para os usuários criarem seus próprios casos clínicos e as questões referentes a eles. Também vale ressaltar que o aplicativo foi desenvolvido para funcionar em computadores e *smartphones*, permitindo assim que os usuários consigam usufruir das funcionalidades em qualquer lugar.

## 2. Conceito Básicos

Esta seção traz as definições de forma resumida das áreas relacionadas a este trabalho que são importantes para o entendimento da proposta.

**Problem Based-Learning (PBL)**, ou em português Aprendizagem Baseada em Problemas, Segundo Souza e Dourado [Souza e Dourado 2015] trata-se de uma técnica de ensino que promove a aquisição de conhecimento utilizando de problemas reais da área de ensino para ajudar no desenvolvimento de habilidades, competências e atitudes em todo processo de aprendizagem.

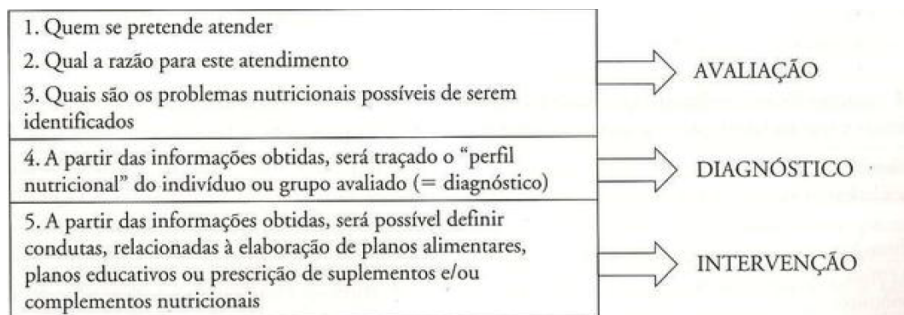
**Avaliação Nutricional** De acordo com Sampaio [Sampaio ], a avaliação nutricional é uma ferramenta essencial para monitorar a evolução de um paciente e acompanhar as respostas às intervenções nutricionais. Segundo o autor, a avaliação nutricional pode ser classificada em duas categorias: individual e coletiva. A avaliação nutricional coletiva é mais voltada para os profissionais que atuam na área da saúde coletiva, enquanto a individual é mais utilizada pelos nutricionistas em seu trabalho clínico domiciliar.

**Avaliação Dietética** É um ferramenta muito importante para os nutricionista que permite criar um recordatório alimentar do paciente. As principais forma de avaliação dietética incluem: Recordatório Alimentar; Questionário de Frequência Alimentar (QFA); Diário Alimentar e Avaliação Antropométrica.

**Criação de Dietas** Segundo Julio e Sandra [Julio e Sandra 2009] A criação de uma dieta ou plano alimentar envolve muitas etapas que devem ser seguidas para conhecer

melhor o paciente, assim podendo criar uma dieta específica para cada caso. Ao observar a Figura 1 é possível ver algumas das etapas para a criação de uma dieta. É importante ressaltar que toda informação sobre o paciente é útil na criação da sua dieta, desde informações, pessoas como religião, região onde mora, medicamentos que usa com frequência, etc. Tudo isso influencia na criação de uma boa dieta ou plano alimentar.

**Casos Clínicos** Galdeano *et al.* [Galdeano et al. 2003] falam que um caso clínico pode ser definido como um conjunto de informações obtidas por meio de uma ampla coleta de dados de um paciente ou grupo de indivíduos, assim fornecendo informações relevantes para futuras tomadas de decisões. Galdeano *et al.* [Galdeano et al. 2003] menciona que esses casos clínicos são usados para realizar um estudo dos problemas e características do paciente a fim de entender suas necessidades, assim ajudando o profissional a criar estratégias para solucionar tais problemas. É importante ressaltar que o estudo de casos clínicos tem um maior destaque no ensino, visando favorecer a aprendizagem.



**Figura 1. Etapas para criação de uma dieta.**

*Fonte: adaptado de Julio e [Julio e Sandra 2009]*

### 3. Trabalhos Relacionados

Esta seção apresenta os trabalhos relacionados. Estes trabalhos foram obtidos através de pesquisas no Google Acadêmico, dos 9 artigos apresentados 5 foram utilizados como artigo base para o uso da técnica de *snowballing*. As pesquisas estão divididas em trabalhos que tratam da parte de formação acadêmica e trabalhos que são voltados mais para o meio nutricional.

#### 3.1. Aprendizagem e Técnicas de Ensino

Losekann *et al.* [Losekann et al. 2007] propuseram uma ferramenta chamada MEDCASE para estudos de casos clínicos na área da medicina, disponibilizada online. Os casos são organizados em macro-assuntos para facilitar a navegação do usuário em problemas específicos que ele deseja estudar. O sistema apresentado possui dois níveis de acesso: visitantes e administradores. Os administradores têm privilégios para gerenciar os casos clínicos, podendo publicar e excluir, enquanto os visitantes são usuários interessados em utilizar a ferramenta para fins acadêmicos. Além do estudo de casos clínicos, o *software* possui várias outras funcionalidades, incluindo núcleos para opiniões de especialistas, dicionário de termos, gerenciamento de usuários, referências bibliográficas e fórum de discussões.

De mesmo modo, Sebastiani [SEBASTIANI 2014] propõe o Sistema Interdisciplinar de Análise de Casos Clínicos (SIACC), que cria casos de estudo clínico para cursos na área da saúde. O SIACC fornece dados como anamnese do paciente, exames físicos, hipóteses diagnósticas, exames complementares, diagnósticos e desfechos. As informações são apresentadas aos alunos de maneiras diferentes, algumas como informações ilustrativas e outras como questionamentos para testar as escolhas dos alunos. Os casos clínicos no SIACC não seguem um padrão fixo e podem ter múltiplos desfechos. O sistema também possui um banco de imagens anatomopatológicas, histórico de execução dos casos clínicos, *feedback* instantâneo para as respostas dos alunos e um índice de desempenho para avaliar o resultado da resolução do caso.

Por outro lado, no trabalho de Costa *et al.* [da Costa et al. 2020] é apresentado uma ferramenta que tem como proposta, a criação de planos de aula. O *software* foi desenvolvido para *smartphone* visando implementar a metodologia de PBL de modo a incentivar mais o estudo entre os alunos. O *software* conta a função de criação de turmas, planos de turmas, problema a ser resolvido e inserção de referências para auxiliar nos estudos.

A pesquisa de Meira e Borges [Meira e Borges 2017] mostra a implementação da técnica de PBL através de um programa chamado Robocode. No trabalho é utilizado para estimular a aprendizagem de conceitos de programação de computadores e avaliar a competição científica com base nos conceitos das linguagens de programação Java e C-Sharp, sendo elas utilizadas dentro do programa Robocode.

### 3.2. Meio Nutricional

O trabalho de Steluti *et al.* [Steluti et al. 2020] apresenta o aplicativo GLOBALDIET um *software* nutricional desenvolvido pela A *International Agency for Research on Cancer-World Health Organization* (IARC-WHO) com o objetivo de padronizar a coleta de dados individuais de consumo de alimentos. O trabalho de Steluti *et al.* [Steluti et al. 2020] apresenta o desenvolvimento da adaptação do *software* para uma versão brasileira. A adaptação foi feita para trazer ao *software* uma tradução para o português como também adicionar informações referente a culinária brasileira. O aplicativo utiliza um recordatório alimentar de 24 horas padronizado, dividido em cinco etapas: informações gerais do paciente, lista rápida de alimentos e receitas, detalhamento dos alimentos, receitas e quantidades consumidas por meio de fotos, controle da quantidade de alimentos e nutrientes, e informações sobre o uso de suplementos dietéticos.

Gonçalves e Alves [Gonçalves e Alves 2010] apresentam um manual de treinamento do *software* NUTWIN. A ferramenta tem objetivo de auxiliar os profissionais que atuam na área de nutrição, além de facilitar a quantificação de nutrientes durante o desenvolvimento de planos alimentares. Diferente dos outros programas apresentados, esse é focado no mercado de trabalho, contando com ferramentas fundamentais para o nutricionista. NUTWIN conta com as seguintes ferramentas: gerar avaliações antropométricas; elaborar planos alimentares; trabalhar com alimentos quantificados em medidas caseiras; buscar, incluir, modificar e excluir alimentos, pacientes e tabelas auxiliares; incluir receitas com forma de preparo; gerar relatórios de todos os cadastros presentes no programa; cadastrar modelos para anamnese e exames bioquímicos; organizar em pastas os dados dos pacientes; exportar dados para Excel, Word e Acess; e Calculadora nutricional.

O trabalho de Moreno *et al.* [Moreno et al. 2011] apresenta um programa que tem como intuito a discussão de casos clínicos vinculado aos Sistema de Informações de Hospitalares. O *software* que foi nomeado como MEDCAST presa permitir que os pacientes e profissionais da saúde consigam acessar os dados remotamente de maneira fácil além de também prover segurança para os dados confidenciais dos pacientes e garantir qualidade na informação passada ao clínico. O programa MEDCAST ainda conta com funcionalidades como a possibilidade de realizar reuniões a distancia via *streaming* e compartilhamento de texto e documentos (podendo ser PDFs, casos clínicos, informações dos pacientes) o programa ainda permite a exibição de imagens estáticas e vídeos que poder sem exibidos em tempo de reunião.

### 3.3. Comparação

Abaixo na Tabela 1 é apresentado uma tabela de comparação entre as funcionalidades dos aplicativos apresentados a cima. Nela é possível notar que o aplicativo que está sido desenvolvido nesse trabalho tera o objetivo de superar os softwares estudados.

**Tabela 1. Tabela de características das plataformas.**

Características	Plataformas				
	NUTWIN	SIACC	MEDCASE	MEDCAST	CASEFY
É voltada para o meio acadêmico		✓	✓		✓
É voltado para o meio profissional	✓			✓	✓
É totalmente gratuito	✓	✓	✓	✓	✓
Permite criação de casos clínicos	✓	✓	✓	✓	✓
Possui atividades para praticar		✓			✓
Fornecer um histórico de atividades realizadas		✓	✓		✓
Multi-plataforma					✓

Os dois primeiros itens da Tabela 1 são responsáveis por avaliar se o aplicativo que esta sendo comparado consegue ser útil para um aluno ou um profissional atuante na área da nutrição. O terceiro item é alto explicativo, ele serve para comparar se os aplicativos estão disponível de forma gratuita ou paga. O quarto item da tabela é referente a capacidade de criação de casos clínicos, ou seja, a liberdade para o usuário criar seus próprios casos clínicos e salvar eles no *software*. A quinta funcionalidade da tabela se refere a capacidade de fornecimento ou apresentação de atividades praticas para cada caso clínico. Por fim o sexto e sétimo item servem para comparar se os aplicativos possuem um histórico que armazena as atividades respondidas pelos usuários e se o aplicativos possuem suporte para várias plataformas como *Mobile* e *Desktop*.

## 4. Casefy

Este trabalho tem como objetivo o desenvolvimento de um sistema para criação e estudo de casos clínicos nomeado de CASEFY. O aplicativo conta com versões para computadores e dispositivos moveis como *smartphone* e tablets o qual fornece casos clínicos

e atividades pre-armazenadas, também sendo possível a criação de novos casos clínicos completos. Ao final de cada atividade de caso os resultados são salvos em um histórico que serve para ver o desempenho do usuário em cada caso clínico.

#### 4.1. Metodologia

No desenvolvimento do trabalho foi adotada a metodologia ágil Scrum. Segundo Marco Tulio Valente [Valente 2020], Scrum é um método ágil, iterativo e incremental para gerenciamento de projetos. Tal método é bastante utilizado não só para desenvolvimento de aplicativos como também pode ser usado para escrita de livros, sendo assim no desenvolvimento do trabalho o Scrum foi usado tanto para a escrita do artigo como também para o processo de criação do aplicativo. Todo o trabalho estimado foi dividido em histórias de usuário, priorizado e dividido em *sprints* para execução.

Para o conteúdo do aplicativo, solicitou-se casos clínicos anonimizados e fictícios por e-mail a profissionais nutricionistas e professores de Nutrição em cursos superiores de Nutrição. Sendo assim, o material de estudo que é apresentado aos usuários do aplicativo foram validados por ao menos um profissional da área e são, portanto, consistentes com casos clínicos reais. Os casos clínicos foram, então, padronizados e transcritos em formato JSON. Quanto às questões dos questionários de cada caso clínico, alguns dos casos clínicos já possuíam questões vinculadas, para os demais que não possuíam, as questões foram elaboradas pelo primeiro autor com base na estrutura daquelas existentes e dados dos novos casos clínicos.

#### 4.2. Arquitetura do Sistema

CASEFY trata-se de um aplicativo multiplataforma construído utilizando-se preceitos de “arquitetura limpa de software”. Para isso a construção das telas do aplicativo foram feitas utilizando o *framework* Flutter<sup>1</sup> da linguagem Dart sendo ambos mantidos e desenvolvidos pelo Google.

A Figura 2 ilustra a arquitetura do sistema. Cada seta no diagrama representa os procedimentos realizados para ocorrer a comunicação entre o aplicativo, os dados locais, servidor e banco de dados. (A) ilustra a solicitação dos dados locais que estão salvos em um arquivo JSON dentro do código-fonte; (B) representa a solicitação HTTP a API; (C) A API faz a requisição ao banco de dados; (D) O banco de dados retorna os dados para a API; (E) O servidor envia os dados para o aplicativo. É importante mencionar que o *back-end* ainda vai ser desenvolvido.

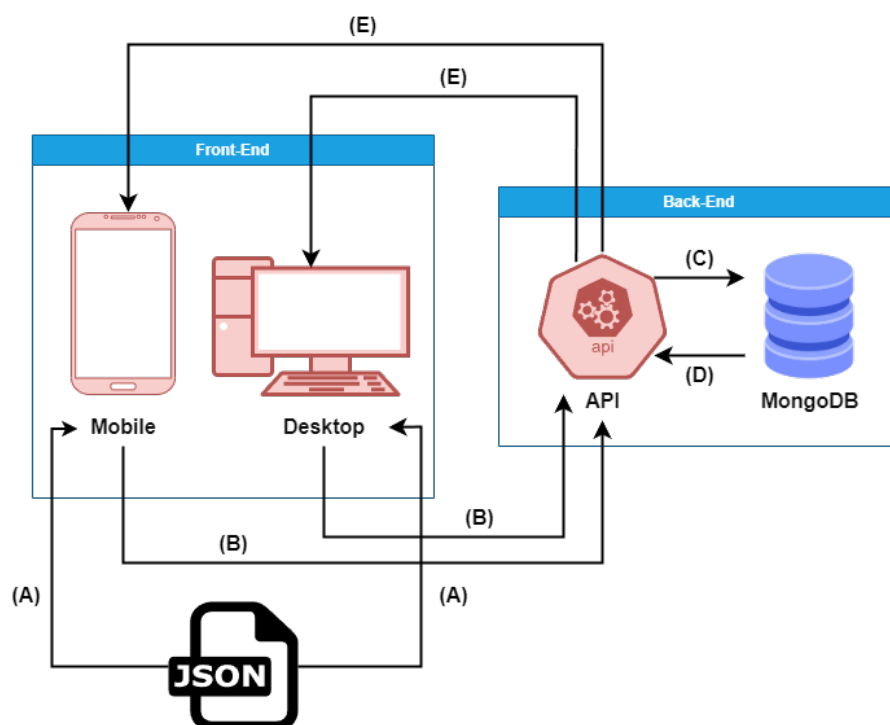
#### 4.3. Funcionalidades do Sistema

O *software* conta com 5 funcionalidades importantes sendo elas: apresentação dos casos clínicos, *quizz* referente a cada caso clínico, histórico de *quizzes* respondidos, criação de casos clínicos e configurações do aplicativo.

Inicialmente, o usuário tem acesso a uma lista que contém todos os casos clínicos registrados no aplicativo junto a uma lista que mostra os casos clínicos fornecidos por outros usuários. Quando o usuário selecionar o caso clínico que deseja estudar ele é enviado a uma nova tela onde serão apresentadas todas as informações necessárias para ter um

---

<sup>1</sup>Disponível em: flutter.dev, Acessado em: 13 de junho de 2023



**Figura 2. Arquitetura do Sistema.**

*Fonte: Autoria Própria*

entendimento sobre o “paciente fictício”. A tela de apresentação de casos clínicos conta com informações como: introdução; anamnese clínico nutricional; hábitos; condições alimentares; avaliação antropométrica; exames laboratoriais e recordatório alimentar.

Cada caso clínico possui um *quizz* vinculado ao mesmo. Sendo assim, quando o usuário concluir o estudo do caso, ele tem a opção de iniciar um *quizz* para testar seu conhecimento. Esses questionários são construídos com base nas informações que foram obtidas previamente na leitura do caso clínico. Quando o questionário for finalizado, é apresentado um resumo onde mostra a quantidade de acertos e erros com base nas respostas fornecidas e por fim será salvo no histórico o desempenho do usuário. Vale ressaltar que o usuário consegue criar seus próprios casos clínicos que ficam salvos de forma online possibilitando serem acessados por outras pessoas.

Para deixar o aplicativo mais customizado ao gosto do usuário o aplicativo conta com uma seção para configurações do aplicativo. Nessa parte de configurações existem opções que vão dificultar ou facilitar a experiência do usuário com relação aos casos clínicos e os questionários referentes a eles. O aplicativo conta com opções para desabilitar o *timer* de cada questão do *quizz*, ocultar dados não fornecidos nos casos clínicos e limpar o histórico de questionários respondidos.

#### 4.4. Back-End

O *back-end* é referente a parte responsável por fazer a conexão entre o que o usuário tem acesso e o que o *software* necessita para funcionar de forma correta. De forma resumida o *back-end* será responsável por fornecer os dados para o *front-end* e ainda receber os dados que foram fornecidos pelo usuário na interface e armazenar no servidor.

#### 4.4.1. Banco de Dados

O banco de dados utilizado para armazenamento dos casos clínicos criados pelos usuários foi o MONGODB. É importante ressaltar que trata-se de um banco de dados não-relacional. Sendo assim, ele não usa esquema de tabelas com linhas e colunas encontrado em banco de dados relacionais como MySQL. O motivo da escolha de um banco de dados NoSQL foi por conta da praticidade em lidar com uma grande quantidade de dados, rápida escalabilidade, velocidade e sistema de consultas mais simples já que sua estrutura é similar a um JSON.

#### 4.4.2. API

Abaixo na Tabela 2 é apresentado os *endpoint* da API. As rotas presentes na Tabela 2 são divididas em rotas que utilizam o método GET e o método POST, esses protocolos foram utilizados para que a API consiga receber os dados fornecidos pelo usuário como também retornar dados que forem solicitados pelo mesmo. Basicamente com essas rotas é possível tanto registrar um caso clínico com um *quizz*, como também retornar uma lista de casos clínicos ou um caso clínico em específico.

**Tabela 2. Descrição das rotas da API.**

Endpoint	Descrição
URLBASE/register/	Rota para registrar novo caso clínico
URLBASE/createquizz/:id	Rota para criar quizz
URLBASE/createoptions/:id	Rota para criar opções do quizz
URLBASE/clinicalcases	Rota para retornar lista de casos clínicos registrado
URLBASE/clinicalcase/:id	Rota para retornar caso clínico específico
URLBASE/quizz/:id	Rota para retornar quizz de um caso clínico
URLBASE/quizz/questions/:id	Rota para retornar questões de um quizz

#### 4.5. Front-End

O *front-end* é a parte do desenvolvimento de uma interface gráfica para os usuários do aplicativo. Essa interface gráfica é feita para que o usuário possa visualizar e interagir com o aplicativo de uma forma mais atraente. A interface gráfica vai exibir para o usuário todas as informações e funcionalidades contidas no *software* como textos, imagens, botões, etc. Para o desenvolvimento *front-end* desse trabalho será usado o *framework* Flutter da linguagem Dart.

A interface do aplicativo já está completa contando com todas as telas necessárias para que o usuário consiga usufruir do *software*. É válido ressaltar que tanto as telas para versão *mobile* quanto para a versão *desktop* estão prontas, porém é importante avisar que elas podem sofrer algum tipo de mudança dependendo do *feedback* dos usuários.

A Figura 3 mostra as telas principais do aplicativo. A Figura 3.a mostra a tela inicial do aplicativo que basicamente apresenta o *software* e dá liberdade para o usuário escolher se quer começar o estudo de algum caso clínico logo no início. A tela da Figura 3.b apresenta a lista de todos os casos clínicos registrados no aplicativo, nessa tela o

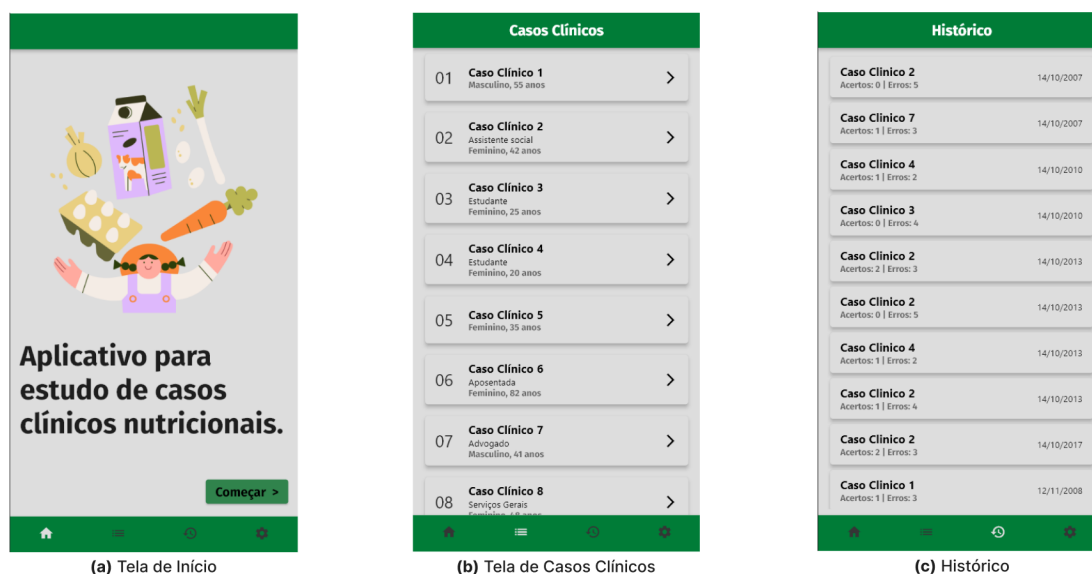


usuário pode escolher qualquer item da lista para estudar. A Figura 3.c mostra o histórico de quizzes respondidos.

Quando o usuário seleciona um caso clínico para estudar é aberta as telas da Figura 4. A Figura 4.a mostra a tela de carregamento do caso clínico, nela são mostradas dicas que podem auxiliar no estudo do caso. Já as telas apresentadas nas Figuras 4.b e 4.c são basicamente iguais à única diferença é que uma mostra o caso clínico com as abas fechadas e outra mostra um exemplo com uma das abas abertas. Nas Figuras 4.b e 4.c é possível notar a presença de um botão de *start*, ele serve para inicializar o *quizz* do caso clínico.

A Figura 5 mostra a tela do *quizz* e suas variações. Na tela da Figura 5.a é apresentado o *quizz* com sua pergunta no topo da tela, logo em seguida uma lista com todas as opções onde apenas uma esta correta e um *timer* que mostra o tempo para responder à questão. Se o tempo do *timer* acabar o *quizz* seguirá para a próxima questão e a anterior e dada como incorreta. As telas da Figura 5.b e 4c mostram quando o usuário responde alguma questão de forma certa ou errada.

Por fim as telas da Figura 6 mostram a tela inicial com a lista de casos clínicos aberta (Figura 6.a) e Figura 6.b mostra a tela que apresenta o caso clinico para estudo. Ambas as telas mostram como fica o aplicativo quando executado na sua versão *desktop*.

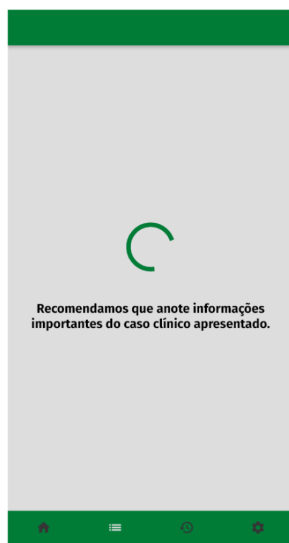


**Figura 3. Telas principais.**  
*Fonte: Autoria Própria*

#### 4.6. Armazenamento

Formam utilizadas 3 formas diferentes de armazenamento no aplicativo, cada forma de armazenamento corresponde a um tipo de dado que é salvo no sistema. Para o armazenamento dos casos clínicos foi utilizado o formato JavaScript Object Notation (JSON). Esses arquivos json estão contidos dentro do código-fonte, e são carregados no front-end para que as informações contidas neles possam ser exibidas para o usuário.

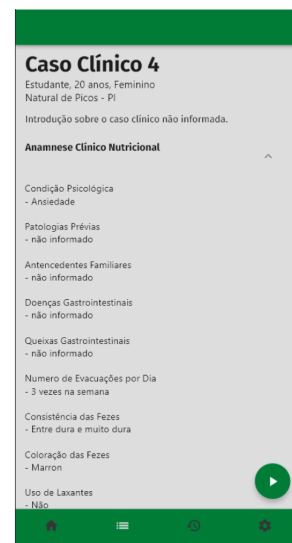
A segunda forma de armazenamento é utilizada para guardar as configurações do aplicativo que são modificadas pelo usuário. Para esse tipo de armazenamento foi utili-



(a) Tela de Dica



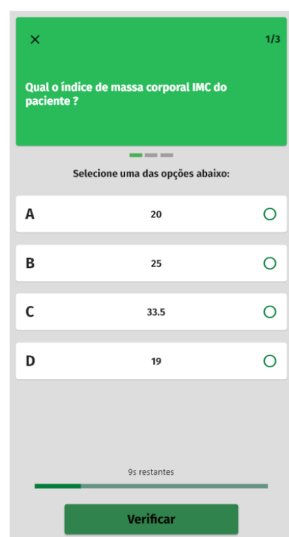
(b) Tela De Caso Clínico



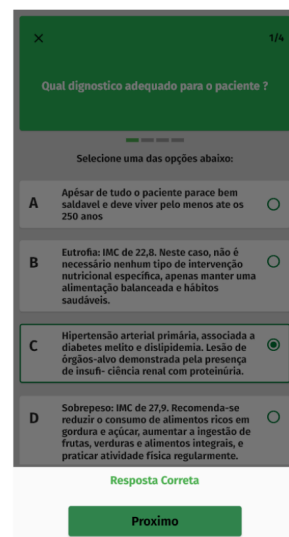
(c) Tela De Caso Clínico 2

**Figura 4. Telas Casos Clínicos.**

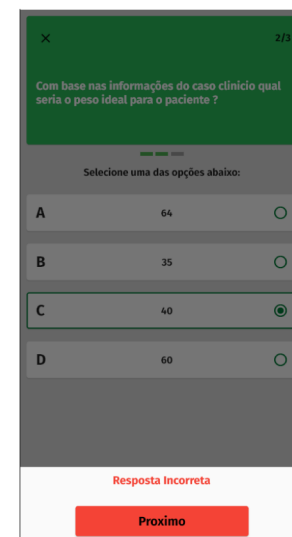
*Fonte: Autoria Própria*



(a) Tela do Quiz



(b) Tela do Quiz Respostas Certa



(b) Tela do Quiz Respostas Errada

**Figura 5. Telas Quiz.**

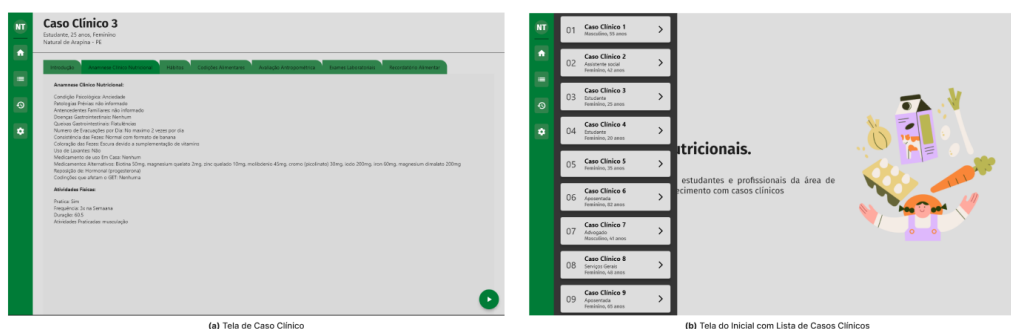
*Fonte: Autoria Própria*

zando o *Shared Preferences*<sup>2</sup> uma biblioteca do *Flutter* utilizada para gravação e leitura de pares e valores de forma simples. Tal biblioteca foi escolhida devido a sua facilidade de implementação e compatibilidade com diversos tipos de sistemas operacionais.

Por fim a terceira forma de armazenamento é utilizada para guardar o histórico das respostas dos *quiz* de cada caso clínico. Para esse tipo de armazenamento foi utilizado o *Isar Database*<sup>3</sup> sendo também um biblioteca do *Flutter*. *Isar* é um banco de dados rápido e fácil feito para ser usado junto a *Flutter* e assim como o *Shared Preferences*

<sup>2</sup>Disponível em: [https://pub.dev/packages/shared\\_preferences](https://pub.dev/packages/shared_preferences), Acessador em: 13 de junho de 2023

<sup>3</sup>Disponível em: <https://pub.dev/packages/isar>, Acessador em: 13 de junho de 2023



**Figura 6. Telas Desktop.**

*Fonte: Autoria Própria*

ele foi escolhido por sua facilidade em implementar e ainda compatibilidade com vários sistemas operacionais.

#### 4.7. Levantamento de Requisitos de Software

Durante o processo de desenvolvimento foi feito o levantamento de alguns requisitos necessários para que o *software* possua-se todas as características e funcionalidades necessárias para suprir as necessidades dos usuários. Os requisitos foram criados com base na leitura de artigos e livros da área de nutrição, além de conversas com alunos e profissionais da área. Para uma melhor apresentação e especificação dos requisitos foi utilizado o modelo de história de usuário apresentado na Tabela 3. É importante ressaltar, que cada requisito foi separado por ordem de prioridade e por dependências, visto que algumas funcionalidades do sistema dependem de outras.

#### 4.8. Desenvolvimento

O desenvolvimento deste projeto foi dividido em 5 partes, sendo elas: (i) estudo sobre a área de nutrição; (ii) estudo sobre casos clínicos; (iii) esboço do aplicativo e por fim, (iv) desenvolvimento do aplicativo, (v) Desenvolvimento da API.

O estudo da área de nutrição e casos clínicos foi feita por meio de livros e artigos que falam sobre a área. Além disso, foram feitas reuniões com uma professora da UFPI. As reuniões com a professora foram úteis para ter uma visão sobre a área de nutrição e ainda entender melhor como funciona um caso clínico, além disso, a mesma fez indicação de livros e aplicativos para material de estudo.

Foram feitas duas versões de esboço de telas sendo um *wireframe* que esboça a estrutura da interface com poucos detalhes servindo para dar uma noção básica sobre a localização de botões, textos e imagens, os *wireframes* não foram incluídos em favor das telas reais. A segunda versão é mais detalhada contendo cores, textos, ícones e imagens, essa versão respeita a primeira, servindo para mostrar como deve ficar o resultado das telas do aplicativo. Para a criação do *wireframe* foi utilizado o *software* Balsamiq Wireframe da empresa Balsamiq e para as telas detalhadas o Figma<sup>4</sup>.

As ferramentas utilizadas para o desenvolvimento do aplicativo foram o editor de código-fonte Visual Studio Code, tal ferramenta é desenvolvida e mantida pela Microsoft. Para controle de versões do aplicativo foi utilizado o Git que nada mais é que um

<sup>4</sup>Disponível em: <https://www.figma.com>, Acessador em: 13 de junho de 2023

**Tabela 3. Estórias de usuários definidas.**

#	Ator	Estória	D	P	C
1	Usuário	ES01: Eu como usuário, preciso ter acesso a lista de casos clínicos.	-	Alta	3
2	Usuário	ES02: Eu como usuário, preciso ter acesso a todas as informações do caso clínico.	1	Alta	5
3	Usuário	ES03: Eu como usuário, preciso ter a opção de iniciar um questionário a respeito do caso clínico selecionado.	1, 2	Alta	8
4	Usuário	ES04: Eu como usuário, preciso saber quando acertei ou errei alguma questão do <i>quizz</i> .	1, 3	Baixa	3
5	Usuário	ES05: Eu como usuário, preciso ver o total de acerto e erros do <i>quizz</i> respondido.	1, 3	Baixa	3
6	Usuário	ES06: Eu como usuário, preciso ter a opção de desabilitar o <i>timer</i> das questões do <i>quizz</i> .	-	Alta	3
7	Usuário	ES07: Eu como usuário, preciso ter a opção de ocultar dados não fornecidos pelo caso clínico.	1	Baixa	3
8	Usuário	ES08: Eu como usuário, preciso ter acesso a um histórico com os resultados de todos os <i>quizz</i> já respondidos.	1	Baixa	3
9	Usuário	ES09: Eu como usuário, preciso criar meus próprios casos clínicos	-	Alta	5
10	Usuário	ES10: Eu como usuário, preciso criar meu próprio <i>quizz</i> para o caso clínico que estou criando	-	Alta	5

D: Dependência; P: Prioridade; C:Complexidade.

sistema de controle de versão distribuído e para armazenamento do código foi utilizado o GitHub que é uma plataforma para hospedagem de código.

O desenvolvimento do aplicativo foi feito utilizando o *framework* Flutter e a linguagem Dart. Segundo o próprio site do Flutter, ele se trata de um *framework* de código aberto para desenvolvimento multiplataforma utilizando um único código Dart como base, ou seja, um único código para criar telas para aplicativos de dispositivos móveis, *Web* e *desktop*.

O desenvolvimento da API foi feita utilizando Node.JS<sup>5</sup> com Express.JS<sup>6</sup> e o Prisma ORM<sup>7</sup> ambos presentes na linguagem JavaScript. De forma resumida o Node.JS serve para construir aplicativos de redes escalonáveis, o Express.JS por sua vez é um *framework web* do Node.js que fornece recursos para criação de servidores *web* e por fim Prisma ORM trata-se de um *Object Relational Mapper* (ORM) de código aberto que funciona em conjunto com Node.JS e Typescript.

<sup>5</sup>Disponível em: <https://nodejs.org/en>, Acessado em: 30 de outubro de 2023

<sup>6</sup>Disponível em: <https://expressjs.com>, Acessado em: 30 de outubro de 2023

<sup>7</sup>Disponível em: <https://www.prisma.io>, Acessado em: 30 de outubro de 2023

## 5. Avaliação

Foi realizada a avaliação de usabilidade do sistema para verificar a qualidade da interface de usuário. Para fazer a avaliação de usabilidade foi utilizado 3 métodos distintos, sendo a análise ad-hoc de usabilidade, *checklist* de usabilidade e por fim foi feita a avaliação heurística que é bastante conhecida pelo seu potencial de identificação de problemas de usabilidade fazendo com que ela seja uma ótima escolha para avaliação de sistemas. Mercês Salvador e Araújo [das Mercês Salvador e de Araújo 2023].

### 5.1. Análise Ad-hoc

A avaliação de ad-hoc é uma avaliação ou análise informal, não planejada e não estruturada de um determinado assunto, situação, problema ou objeto. Durante o desenvolvimento da interface foi utilizado a forma de avaliação ad-hoc para a descrição de possíveis problemas de usabilidade. A Tabela 4 apresenta os pontos que são abordados ao descrever um problema.

**Tabela 4. Descrição do problema.**

<b>Categoria</b>	<b>Descrição</b>
Contexto	Situação de uso em que o problema pode ser verificado
Causa	Refere-se a funcionalidade do sistema que leva a certo problema
Efeito sobre o usuário	Impacto do problema na experiência do usuário final
Efeitos sobre a tarefa	Isso se relaciona com o resultado ou consequência da ação realizada na tarefa
Possível correção	Aponta uma possível forma de alteração para correção no sistema

### 5.2. Checklist de Usabilidade

O *checklist* de usabilidade consiste em uma série de pontos baseados em uma lista de verificação, tal lista ajuda a verificar se o sistema que está sendo desenvolvido cumpre com todos os requisitos e se possui pequenos erros que muitas vezes passam de forma despercebida. O *checklist* foi feito através do site UX Sapo<sup>8</sup> que prove diferentes tipos de *checklist*, como o Casefy se trata de um aplicativo multiplataforma foi utilizado o *checklist* web e mobile. Vale ressaltar que tal método é bastante usado para fazer uma avaliação rápida de um app ou página web.

### 5.3. Heurística de Usabilidade

Heurística de Nielsen ou heurística de usabilidade é um conjunto que ajudam a avaliar a usabilidade de uma interface de usuário. Para a avaliação de heurística de usabilidade foi feito o uso das 10 heurísticas propostas por Nielsen em 1995 [Nielsen 1994]. Abaixo a Tabela 5 lista as 10 heurísticas de Nielsen e a Tabela 6 lista as escalas de severidade. Os itens da Tabela são ordenados de acordo com o grau de impacto que um problema específico pode causar no público consumidor do produto. Vale ressaltar que a Tabela 6 irá ajudar na compreensão da Tabela

<sup>8</sup>Disponível em: <https://ux.sapo.pt>, Acessado em: 30 de outubro de 2023

**Tabela 5. Heurísticas de Nielsen.**

Identificador	Heurística	Descrição
H01	Visibilidade do estado do sistema	O sistema mantém o usuário sempre informado sobre o que está acontecendo no mesmo.
H02	Consistência de padrões	Evitar que o usuário tenha que pensar se ações ou situações diferentes significam a mesma coisa.
H03	Flexibilidade e eficiência de uso	Fornece opções que otimizam a experiência de usuário (atalhos, teclas de funções, abreviações).
H04	Estética e design minimalista	Evita o uso de informações irrelevantes.
H05	Compatibilidade do sistema e o mundo real	O sistema utiliza uma linguagem aos usuários, em vez de termos técnicos e específicos.
H06	Reconhecimento de erros	Utilizar linguagem simples para apresentar os erros e mostrar como contorná-los.
H07	Controle do usuário e liberdade	Oferece saída de emergência, permitindo que os usuários saiam facilmente de situações inesperadas.
H08	Reconhecimento e memorização	Fazer com que as ações e opções, sejam visíveis, o usuário não tem a obrigação de lembrar de tudo.
H09	Prevenção de erros	Prevenir a ocorrência de erros.
H10	Ajuda e documentação	Fornece informações que podem ser facilmente encontradas e orienta os usuários.

## 6. Resultados

Ainda é esperado a construção e implementação da API e interfaces para inclusão de casos clínicos e *quizzes* assim permitindo que o aplicativo funcione de forma mais dinâmica e permita que ele seja populado por alunos e professores da área da nutrição.

### 6.1. Análise Ad-hoc

A Tabela 7 apresenta os problemas identificados durante a avaliação ad-hoc de usabilidade. Vale ressaltar que problemas de baixa severidade que não afetam diretamente o sistema ou que podem ser facilmente contornados serão corrigidos em trabalhos futuros, por outro lado problemas que interferiam diretamente na experiência do usuário com o aplicativo e problemas que atrapalhavam diretamente uma funcionalidade essencial do sistema tiveram um nível de prioridade maior na hora da correção.

Tabela 6. Escala de severidade.

Identificador	Severidade	Descrição
S01	Problema cosmético ou superficial	Problema normalmente estético, corrigido quando se tem tempo sobrando.
S02	Problema de usabilidade pequeno	Problema que pode ser contornado pelo usuário e possui baixa severidade para ser solucionado.
S03	Problema de usabilidade grande	Problema com impacto considerado alto na funcionalidade do sistema e por isso deve ser solucionado rápido.
S04	Problema de usabilidade crítico	Problema crítico que deve ser solucionado antes do lançamento do sistema.

Tabela 7. Análise ad-hoc.

Problema	Contexto	Severidade
Data errada no histórico	Ocorre ao salvar o resultado de um quizz no histórico	S01
Entrada de texto pouco intuitiva	Ocorre quando cria um casos clínico	S02
Ausência de unidades de medidas	Ocorre quando se estuda um caso clínico	S01
Tempo curto para responde quizz	Ocorre quando se responde um quizz com o timer ativo	S02

## 6.2. Checklist de Usabilidade

Através do *Checklist* de usabilidade disponibilizado pelo UX Sapo, foi possível identificar tanto as características sólidas de usabilidade quanto as que precisam ser aprimoradas. O checklist específico para a versão *desktop* indicou que o software satisfaz eficientemente requisitos fundamentais, como navegação, *feedback* ao usuário, layout, legibilidade de textos, responsividade e desempenho. No entanto, o aplicativo demonstrou deficiências em aspectos como a apresentação de formulários e nas opções disponíveis para auxiliar o usuário.

Como foi mencionado na seção de avaliação, foi realizado um *checklist* de usabilidade para a versão *mobile*, também fornecido pelo UX Sapo. Assim como o resultado obtido na versão *desktop* o aplicativo possui um desempenho em questões que envolvem *feedback* do usuário, layouts, legibilidade de textos, desempenho e claro responsividade para diversos tipos de celulares. Porém, vale lembrar que assim como na versão *desktop* a versão *mobile* falha em quesito de formulários e ajuda ao usuário.

Com base nos problemas identificados, pode ser realizado as seguintes correções para melhorar a usabilidade do sistema: Melhorar a apresentação de erros ao usuário, facilitar com que o usuário identifique o tipo de dado aceito por um campo do formulário, por mensagens de erro do formulário de forma mais clara e adicionar mais textos de ajuda para sanar dúvidas.

### 6.3. Heurística de Usabilidade

Os problemas encontrados durante a análise de heurística do sistema, são listados na Tabela 8. Dentre os problemas encontrados foram corrigidos: Requisito de *quizz*, registro de *quizz*, desabilitar *timer* e *quizz* não existente, o problema de tipo de entrada sera corrigido em trabalhos futuros, devido a pouca deposição de tempo para corrigi-lo.

**Tabela 8. Análise heurística do sistema.**

Problema	Severidade	Violação	Descrição
Requisito de quizz	S04	H09	Ao iniciar um <i>quizz</i> online o mesmo entrar em <i>loop</i> infinito
Registro de quizz	S04	H09	Ao registrar um novo <i>quizz</i> é enviado apenas a ultima questão
Desabilitar <i>timer</i>	S02	H01, H02	Ao desabilitar o <i>timer</i> do <i>quizz</i> o contador ainda funcionava porem a barra de contagem não era exibida
<i>Quizz</i> não existente	S02	H01, H02, H09	Possibilidade de iniciar um <i>quizz</i> de um casos clínico que não possui <i>quizz</i>
Tipo de entrada	S02	H01, H02, H06, H08	Na versão <i>desktop</i> o usuário não possui limitação no tipo de dados que pode ser inserido em um campo do formulário

## 7. Considerações Finais e Trabalhos Futuros

Dado os estudos de casos clínicos foi possível notar a importância de tal material para a área de nutrição. Sendo assim, o presente trabalho teve como finalidade, entregar um sistema completo com apresentação e criação *quizzes* e casos clínicos para auxiliar no estudo de nutrição. O CASEFY é uma aplicativo multiplataforma que tem o objetivo de deixar o estudo de casos clínicos mais atraente e divertido para alunos de nutrição utilizando de *quizzes* interativos para por em pratica o que o usuário aprendeu durante a leitura do caso clínico.

O sistema desenvolvido busca motivar os usuários a estudar casos clínicos em qualquer lugar e qualquer hora, permitindo assim uma forma de passar o tempo enquanto põem em pratica sua analise de casos. Com base nos resultados da avaliação, é possível concluir que o aplicativo cumpre com sua proposta, porem vale lembrar que os resultados também apontam a necessidade de algumas melhorias no aplicativo principalmente na parte de criação de novos casos clínicos.

Quanto aos trabalhos futuros, pode ser feita a correção de problemas que foram identificados durantes as avaliações do aplicativo e novas funcionalidades como: Corrigir o formulário de criação de casos clínicos e criar uma versão *desktop* melhorando o mesmo,



criar um sistema de autenticação de usuário para que seja possível filtra melhor quem registra casos clínicos e criar uma versão *dark mode* para o aplicativo.

## Referências

- da Costa, A. J. S., de Almeida, L. M., Rolim, M. P. S., and da Silva, R. O. (2020). Ferramenta de apoio à metodologia de aprendizado baseado em problemas-e-PBL. *Brazilian Journal of Development*, 6(12):93695–93713.
- das Mercês Salvador, L. and de Araújo, T. M. U. (2023). Técnicas para avaliação de usabilidade em aplicações de dispositivos móveis: Uma revisão sistemática qualitativa da literatura. *Revista GEMInIS*, 14(1):71–84.
- Galdeano, L. E., Rossi, L. A., and Zago, M. M. F. (2003). Roteiro instrucional para a elaboração de um estudo de caso clínico. *Revista Latino-Americana de Enfermagem*, 11:371–375.
- Gonçalves, K. and Alves, J. (2010). *Manual de treinamento em software de apoio á nutrição*. 1 edition.
- Julio, T. and Sandra, R. (2009). *Avaliação Nutricional Teoria e Prática*. Guanabara Koogan, 1 edition.
- Losekann, A., Hertzog, H. M., Basso, L. d. O., Cardoso, M., Bitencourt, R., Camozzato Filho, P., and Ballester, D. P. (2007). Medcase-uma ferramenta online para estudo de casos clínicos. *Scientia Medica*, 17(4):184–91.
- Meira, M. C. and Borges, M. A. (2017). Aprendizagem de linguagem de programação com metodologia pbl em competições científicas com robocode. In *Anais do XXIII Workshop de Informática na Escola*, pages 195–204. SBC.
- Moreno, R. A., Lima, V., Lopes, I., and Gutierrez, M. A. (2011). Medcast-sistema colaborativo para discussão de casos clínicos. *Journal of Health Informatics*, 3(3).
- Nielsen, J. (1994). *Usability engineering*. Morgan Kaufmann.
- Sampaio, L. R. *Avaliação nutricional*, volume 1. EDUFBA.
- SEBASTIANI, R. L. (2014). Software para criação de casos clínicos interativos para práticas de ensino de medicina. *Revista Tecnologia e Tendências*, 9(1):71–85.
- Souza, S. C. and Dourado, L. G. P. (2015). Aprendizagem baseada em problemas (ABP): um método de aprendizagem inovador para o ensino educativo.
- Steluti, J., Crispim, S. P., Araujo, M. C., Peralta, A. M., Pereira, R. A., Sichieri, R., Yokoo, E. M., and Marchioni, D. M. (2020). Tecnologia em saúde: versão brasileira do software globodiet para avaliação do consumo alimentar em estudos epidemiológicos. *Revista Brasileira de Epidemiologia*, 23:e200013.
- Valente, M. T. (2020). Engenharia de software moderna. *Princípios e Práticas para Desenvolvimento de Software com Produtividade*, 1:24.