

# Protótipo de Software para Auxílio no Controle da Doença Celíaca

Vanderluiz R. Oliveira<sup>1</sup>, Willen Leolatto Carneiro<sup>1</sup>, Renan O. Zapelini<sup>2</sup>

<sup>1</sup>Instituto Federal de Santa Catarina (IFSC) - Campus Lages  
Rua Heitor Villa Lobos, 225 – 88.506-400 – Lages – SC – Brasil

<sup>2</sup>Infosec Engineer – FULLTECH TECNOLOGIA  
Joinville - SC - Brasil

vanderluizsj@gmail.com, willen.leolatto@ifsc.edu.br,  
renanzapelini@hotmail.com

**Abstract.** *This course completion work addresses the creation of a responsive web application to assist individuals with celiac disease, a condition that requires a gluten-free diet. Given the scarcity of practical solutions on the market, the proposal aims to fill this gap, offering an application that provides clear information about gluten-free foods, making life easier for users. The methodology adopted involved identifying the needs of the target audience, analyzing existing applications, designing and implementing the application. The business model adopted is multilateral, connecting consumers, gluten-free food producers and advertisers. The front-end of the application was developed with the Angular framework, and the back-end follows the three-tier architecture. The revenue model includes user donations and advertising revenue. Despite the limitations, the application achieved most of its objectives, providing an effective solution for controlling celiac disease. The work recognizes the ongoing need for improvement and highlights possible future directions, such as artificial intelligence integration and expansion to mobile platforms.*

**Resumo.** *Este trabalho de conclusão de curso aborda a criação de uma aplicação web responsiva para auxiliar indivíduos com doença celíaca, uma condição que requer uma dieta isenta de glúten. Diante da escassez de soluções práticas no mercado, a proposta visa preencher essa lacuna, oferecendo uma aplicação que fornece informações claras sobre alimentos sem glúten, facilitando a vida dos usuários. A metodologia adotada envolveu a identificação das necessidades do público-alvo, análise de aplicativos existentes, projeto e implementação da aplicação. O modelo de negócios adotado é multilateral, conectando consumidores, produtores de alimentos sem glúten e anunciantes. O front-end da aplicação foi desenvolvido com o framework Angular, e o back-end segue a arquitetura de três camadas. O modelo de receita inclui doações dos usuários e receitas de anúncios. Apesar das limitações, a aplicação atingiu a maioria de seus objetivos, proporcionando uma solução eficaz para o controle da doença celíaca. O trabalho reconhece a necessidade contínua de aprimoramento e destaca possíveis direções futuras, como integração de inteligência artificial e expansão para plataformas móveis.*

## 1. Introdução

Segundo dados da Organização Mundial da Saúde (OMS), 1% da população mundial sofre com a intolerância permanente ao glúten, a chamada doença celíaca (Baptista, 2019). E de acordo com o Conselho Nacional de Saúde (CNS) a doença celíaca afeta em torno de 2 milhões de pessoas no Brasil, mas a maioria deles encontra-se sem diagnóstico (FENACELBRA, 2021). Mesmo após diagnosticado como mencionado na Biblioteca Virtual em Saúde (BVSMS) “O principal tratamento é a dieta com total ausência de glúten; quando a proteína é excluída da alimentação os sintomas desaparecem. A maior dificuldade para os pacientes é conviver com as restrições impostas pelos novos hábitos alimentares”(SAÚDE, 2020). Segundo a lei federal (Lei nº 10.674, de 16/05/2003) é obrigatório que todos os alimentos industrializados informem em seus rótulos a presença ou não de glúten para resguardar o direito à saúde dos portadores de doença celíaca (SAÚDE, 2020). E como relatado no site da Faculdade De Medicina De Minas Gerais (UFMG) CIÊNCIA (2019), o uso das informações nutricionais obrigatórias nos rótulos de alimentos é regulamentado desde 2001.

Mas, depois de 18 anos, os consumidores ainda não encontram informações claras nas embalagens. Entre os principais problemas estão a linguagem complicada e letras muito pequenas. Mesmo agora, em 2023, fazendo uma busca na web, ou nas lojas de aplicativo não se encontra uma solução capaz de atender a demanda, pois os aplicativos existentes, ou foram descontinuados, ou apresentam uma complexidade muito grande e muitos não apresentam as informações de forma clara e completa. Isso se deve, muito provavelmente, ao fato de ser um mercado com características bem específicas e ainda um pouco desconhecido, porém, esse mercado vem crescendo cada vez pela moda dos regimes sem glúten. Dados da consultoria internacional Euromonitor mostram que este setor deve ter uma expansão de 35% a 40% nos próximos três anos. Os produtos *gluten-free* existem para atender ao público celíaco, mas ganharam e conquistaram consumidores que não possuem a doença, mas que buscam uma alimentação natural e mais saudável. Há o entendimento de que a ausência de glúten ou lactose torna um produto *light* ou *fit* (Baptista, 2019).

Neste contexto, este projeto propõe a criação de uma aplicação capaz de auxiliar o celíaco, ou as pessoas interessadas em evitar o glúten, a encontrar de forma clara e fácil as informações necessárias para esse controle. Para alcançar esse resultado foram definidos os seguintes objetivos específicos:

- Identificar as necessidades específicas do público alvo;
- Analisar as aplicações e ferramentas já existentes como referência;
- Projetar a camada de armazenamento de dados e implementar a aplicação *front-end* e *back-end*;
- Convidar pessoas pertencentes ao público alvo a testarem a aplicação e avaliarem através de um questionário;
- Realizar melhorias de acordo com as avaliações.

Sob o aspecto metodológico, este trabalho foi dividido em cinco etapas macro. A primeira etapa será identificar as nuances da doença celíaca. Para isso foi realizada a leitura e estudo de artigos pesquisados *Internet*. A segunda etapa corresponde em fazer uma análise e comparação entre as aplicações já existentes para usar de referência. Para isso foi feita uma pesquisa sistemática das ferramentas e avaliação das mesmas. A terceira etapa será focada em projetar a camada de armazenamento de dados e criação de

*Application Programming Interface (API)*<sup>1</sup> utilizando *.Net*, para que o *front-end* consiga se comunicar com o banco de dados, será utilizado o sistema *SqlServer*, e como Ambiente de desenvolvimento integrado (*IDE*), será usado o *Visual Studio Code*. Na quarta etapa será feito o *front-end* seguindo padrões *UX*<sup>2</sup> e *UI*<sup>3</sup> utilizando *AngularJS*. A última etapa será a avaliação dos módulos desenvolvidos, onde será realizada uma pesquisa através de um questionário enviado a pessoas do público alvo dispostas a testar a ferramenta desenvolvida durante este trabalho.

Sob o aspecto metodológico, este trabalho se classifica de acordo com a natureza, sendo de pesquisa aplicada. No ponto de vista da abordagem do problema, é qualitativo. Da perspectiva dos objetivos, segue o caminho exploratório, visto que se aprofunda no problema e na solução existente, para agregar novas funcionalidades e possibilidades de melhorias. Referente ao ponto de vista dos procedimentos técnicos será uma pesquisa bibliográfica.

Além da primeira seção, o restante do trabalho está organizado da seguinte forma: a seção dois apresenta o referencial teórico, ou seja, descreve os assuntos necessários para o entendimento do trabalho como um todo e a análise de trabalhos similares; a seção três registra como será realizado o desenvolvimento do projeto; a seção quatro apresenta a avaliação do projeto e a seção cinco finaliza o trabalho com as considerações finais.

## **2. Referencial Teórico**

A seção 2.1 do referencial teórico aborda a relação entre a Internet das Coisas (IoT) e smartphones, destacando a definição da IoT como a integração de tecnologias de comunicação e objetos no ambiente físico ao virtual. O texto enfatiza a proliferação de smartphones no Brasil e sua utilidade na leitura de códigos de barras, ressaltando a praticidade e portabilidade desses dispositivos. Na seção 2.2, são explorados aplicativos para intolerâncias alimentares, evidenciando estudos que atestam sua eficácia na melhoria da adesão à dieta e na gestão de condições como a doença celíaca. A personalização desses aplicativos é destacada como essencial. A seção 2.3 discute as funcionalidades de aplicativos para intolerância ao glúten, salientando estudos que avaliam sua eficácia na identificação de alimentos seguros. A necessidade de verificar a confiabilidade das informações é enfatizada, apontando para a variedade de aplicativos disponíveis. Ao final existe uma menção a aplicativos específicos, como o Loomos e o Ilergic, onde se destaca a proposta de criar uma aplicação com um banco de dados alimentado por idealizadores e usuários, enfatizando a importância da troca de informações entre diferentes grupos de usuários.

### **2.1. Internet das Coisas e os Smartphones**

”De acordo com Santos et al. (2016), a IoT pode ser entendida como a combinação de diversas tecnologias de comunicação e a integração de objetos em um ambiente físico ao mundo virtual”(Maschietto e et al., 2021).

---

<sup>1</sup>API: É um conjunto de definições e protocolos para criar e integrar softwares.

<sup>2</sup>User Experience: Trata-se de como o cliente interage com o seu produto ou serviço a partir dos elementos disponibilizados.

<sup>3</sup>User Interface: Trata-se de uma área voltada para criação de interfaces mais fáceis e amigáveis.

O Brasil tem mais de um smartphone por habitante, segundo levantamento da FGV (2021) são 242 milhões de celulares inteligentes em uso no país, que tem pouco mais de 214 milhões de habitantes nesse mesmo período, de acordo com o IBGE.

E a câmera de smartphones tem se tornado uma ferramenta bastante útil para diversos tipos de aplicações, e uma delas é a leitura de códigos de barras. A leitura de códigos de barras pode ser utilizada em diversas situações, como no controle de estoques, em vendas e até mesmo em pagamentos.

Uma das principais vantagens de utilizar a câmera de smartphones para a leitura de códigos de barras é a praticidade. Diferentemente de outros dispositivos, como scanners, os smartphones são portáteis e sempre estão à mão, permitindo que a leitura seja feita de forma rápida e simples.

Além disso, muitos smartphones possuem tecnologia de reconhecimento de códigos de barras integrada, o que torna a leitura ainda mais rápida e eficiente. Ao utilizar um aplicativo de leitura de códigos de barras, o usuário pode simplesmente apontar a câmera para o código e obter as informações necessárias em questão de segundos.

## **2.2. Uso da tecnologia no combate a intolerâncias alimentares**

Os estudos relacionados a aplicativos de controle de intolerância alimentar têm ganhado cada vez mais relevância na área da saúde. Esses aplicativos têm como objetivo auxiliar as pessoas que possuem restrições alimentares a identificar quais alimentos podem ser consumidos sem riscos à saúde, além de ajudar a monitorar e controlar a ingestão de alimentos.

Segundo um estudo realizado por Magrini e et al. (2020), o uso de aplicativos móveis pode ser uma ferramenta útil para melhorar a adesão à dieta em pessoas com intolerância alimentar. Os autores relataram que a utilização desses aplicativos foi capaz de aumentar o conhecimento nutricional dos usuários, além de fornecer informações precisas e atualizadas sobre os alimentos.

Outro estudo interessante foi conduzido por Borghi e et al. (2021), que avaliaram a eficácia de um aplicativo móvel de controle alimentar em pacientes com doença celíaca. Os resultados indicaram que o uso do aplicativo foi capaz de melhorar a adesão à dieta sem glúten, além de reduzir o risco de contaminação acidental.

Além disso, um estudo mais antigo de 2015, realizado por Chomutare e et al. (2015) destacou a importância da personalização dos aplicativos de controle de intolerância alimentar. Segundo os autores, a adaptação do aplicativo às necessidades individuais do usuário é essencial para garantir a eficácia do tratamento.

## **2.3. Funcionalidades Disponibilizadas**

A intolerância ao glúten é uma condição que afeta muitas pessoas em todo o mundo, e a tecnologia tem se mostrado uma ferramenta útil para ajudar aqueles que precisam controlar sua dieta livre de glúten. Existem vários aplicativos disponíveis que ajudam a rastrear a ingestão de alimentos, encontrar alimentos seguros para consumo e até mesmo localizar restaurantes que oferecem opções sem glúten.

Um estudo realizado por Chinnici et al. (2020) avaliou a eficácia de três aplicativos populares de controle de dieta sem glúten: "*Gluten Free Scanner*", "*Find Me Gluten*

*Free*” e *”Gluten Free on the Go*”. Os resultados mostraram que esses aplicativos podem ser úteis para encontrar alimentos sem glúten, mas também destacaram a importância de verificar a confiabilidade das informações fornecidas pelos aplicativos, pois nem sempre são precisas.

Outro estudo realizado por Comino et al. (2016) avaliou a precisão de um aplicativo chamado *”Gluten Free*” na detecção de glúten em alimentos processados. Os resultados mostraram que o aplicativo teve uma taxa de precisão de 84,3%, o que sugere que pode ser uma ferramenta útil para pessoas com intolerância ao glúten.

Além disso, um estudo de revisão realizado por Silvester e Duerksen (2016) avaliou vários aplicativos disponíveis para pessoas com intolerância ao glúten. Os autores concluíram que muitos desses aplicativos são úteis para encontrar alimentos seguros para consumo, mas destacaram que é importante verificar a precisão das informações fornecidas pelos aplicativos antes de usá-los.

O Loomos de Licks e Busata (2020) faz, de forma simples, o que diz na descrição que consta em seu site *”Abra o aplicativo e aponte a câmera para o rótulo de qualquer produto, e veja se há ingredientes que você não quer, ou não pode, consumir”*. O que é incrivelmente útil se você tiver alguma limitação visual, mas se não for esse o caso, no final das contas o resultado é o mesmo se você simplesmente ler o rótulo.

E por último, o Ilergic que sem dúvida foi o mais completo, mas por ser um pouco mais genérico e não ser direcionado a intolerância ao glúten, ele acaba se mostrando confuso às vezes, ao mostrar produtos que pode e que não podem ser consumidos na mesma aba. E acaba não sendo tão prático ao depender exclusivamente da colaboração de seus usuários na identificação dos produtos.

Em resumo, a ideia é criar uma aplicação que contará com um banco de dados alimentado tanto por seus idealizadores, quanto pelas consultas e avaliações dos usuários e que futuramente permitirá a troca de informações pelos diversos grupos de usuários. Hoje já existem várias aplicações disponíveis para ajudar pessoas com intolerância ao glúten a controlar sua dieta. No entanto, é importante verificar a confiabilidade das informações fornecidas pelos aplicativos antes de usá-los como fonte única de informação.

### **3. Vida Sem Glúten**

Este trabalho apresenta o prototipo da aplicação Vida Celíaca, que é baseada no modelo de arquitetura de soluções IoT de 5 camadas. Sua concepção nasce da necessidade de criar uma solução mais focada em uma dieta com restrição de glúten e que contemple tanto a colaboração tão bem desenvolvida pelo ilergic, como também tenha um pouco da praticidade do Loomos.

#### **3.1. Modelo de Negócio**

O modelo de negócios da aplicação será voltada para auxiliar pessoas com intolerância ao glúten e irá se basear no conceito de plataforma multilateral. Nesse modelo, a aplicação atua como um intermediário entre diferentes grupos de usuários, como consumidores em uma dieta sem glúten, produtores de alimentos sem glúten e anunciantes. A aplicação oferecerá uma série de serviços para cada um desses grupos de usuários, criando valor para todos os envolvidos.

Para os consumidores com intolerância ao glúten, a aplicação irá fornecer informações sobre alimentos sem glúten, incluindo receitas, sugestões de restaurantes e lojas que vendem produtos sem glúten, além de permitir que os usuários compartilhem informações e experiências.

Para os produtores de alimentos sem glúten, a plataforma vai ser um canal de vendas, permitindo que eles vendam diretamente aos consumidores ou anunciem seus produtos na aplicação. Além disso, a plataforma pode ajudar os produtores a entender melhor as necessidades dos consumidores, oferecendo feedback e análises de dados.

Finalmente, para os anunciantes, a plataforma oferece uma audiência altamente segmentada, permitindo que eles alcancem consumidores com intolerância ao glúten de forma eficaz e personalizada.

Para gerar receita, a aplicação usará inicialmente uma combinação de doações de usuários e anúncios. As doações vão ser solicitadas diretamente aos usuários por meio de um modelo de "pague o que puder", onde os usuários escolhem quanto desejam doar. Isso permite que a aplicação mantenha-se financeiramente sustentável enquanto oferece serviços gratuitos para os usuários.

Além disso, a plataforma pode gerar receita por meio de anúncios segmentados, que são exibidos apenas para os usuários relevantes, como aqueles que estão procurando por alimentos sem glúten ou que têm interesse em produtos relacionados à saúde e bem-estar.

De acordo com Amit e Zott (2012), plataformas multi-laterais têm se tornado uma forma popular de negócio nos últimos anos. Esses autores afirmam que, ao criar valor para múltiplos grupos de usuários, as plataformas podem gerar uma vantagem competitiva sustentável e, ao mesmo tempo, maximizar o valor para todas as partes envolvidas. A abordagem multi-lateral também é descrita por Parker et al. (2016), como um modelo de negócio bem-sucedido para empresas digitais, permitindo que elas capturem valor por meio da coordenação e conexão de diferentes grupos de usuários.

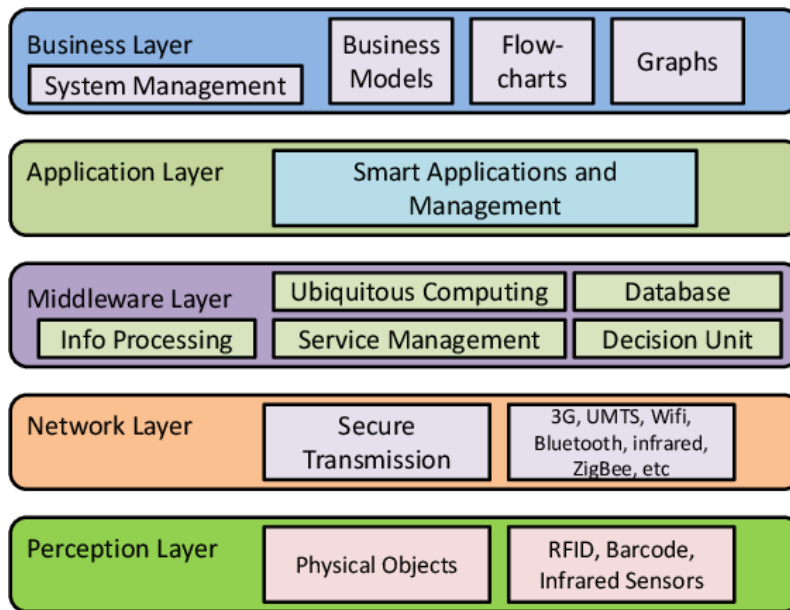
## **3.2. Arquitetura IoT**

Nesta seção será apresentada a arquitetura da solução proposta, baseada em um modelo de arquitetura de 5 camadas.

Antes destas, várias propostas haviam sido realizadas na comunidade científica, sendo as principais baseadas 3, 5 e 7 camadas. Dentre estas, a arquitetura de 5 camadas é o modelo mais aplicável devido a sua simplicidade Al-Fuqaha et al. (2015). Segundo Khan et al. (2012), a arquitetura de 5 camadas utiliza a estrutura apresentada na Figura 1:

### **3.2.1. Camada de Percepção**

Essa camada utiliza tecnologias como leitura de códigos de barras e integração com dispositivos para proporcionar uma experiência mais eficiente e personalizada aos usuários com doença celíaca. Ela se concentra na entrada de dados dos usuários relacionados às suas intolerâncias alimentares e aos produtos que eles consomem e onde é possível encontrar esses produtos. A Figura 2 apresenta o *hardware* utilizado na construção da solução proposta.



**Figura 1. Figura do modelo de 5 camadas. Khan et al. (2012)**



**Figura 2. Figura de um Smartphone e um carrinho de compras**

Até o momento, observamos que o uso do celular tornou-se uma parte integrante do nosso cotidiano, sendo praticamente onipresente. A maioria das pessoas carrega um consigo, portanto, iremos incorporar essa realidade à nossa aplicação, buscando otimizar o uso desse recurso amplamente disponível.

### **3.2.2. Camada de Rede**

A camada de rede é um componente essencial em uma aplicação IoT, pois é responsável por permitir a comunicação entre dispositivos e sistemas. No caso da aplicação em

questão, ela utilizará o celular do usuário como sensor de localização e a câmera como leitor, a camada de rede será responsável por permitir que esses dados sejam enviados e recebidos de forma segura e eficiente.

De acordo com Li et al. (2020), a camada de rede em uma aplicação IoT é responsável por garantir a conectividade e a interoperabilidade entre os dispositivos. Para isso, ela utiliza diferentes protocolos e tecnologias de rede, como o TCP/IP, Bluetooth, Wi-Fi e 5G, entre outros.

No caso específico de uma aplicação que utiliza o celular do usuário como sensor de localização e câmera, é importante que a camada de rede permita a transmissão dos dados de forma segura e eficiente. Para isso, é necessário utilizar protocolos de segurança, como o SSL/TLS e o IPSec, que garantem a confidencialidade e a integridade dos dados transmitidos.

Além disso, é importante considerar a qualidade da rede utilizada para a transmissão dos dados. Segundo Zhang, Luo e Hu (2018), a qualidade da rede pode afetar significativamente o desempenho de uma aplicação IoT. Por isso, é importante utilizar tecnologias de rede que permitam a transmissão dos dados de forma rápida e confiável, como o 5G, que possui uma alta taxa de transferência de dados e baixa latência.

Em resumo, a camada de rede é um componente essencial em uma aplicação IoT que utiliza o celular do usuário como sensor de localização e câmera. É importante garantir a segurança e a eficiência na transmissão dos dados, bem como utilizar tecnologias de rede que permitam uma alta qualidade na transmissão dos dados.

### **3.2.3. Camada de Middleware**

A camada de middleware de uma aplicação IoT é responsável por gerenciar a comunicação entre os dispositivos IoT e a nuvem. No caso em que a aplicação utilizará o celular do usuário como sensor, a camada de middleware terá um papel fundamental na integração dos dados coletados pelos dispositivos com a plataforma de gerenciamento e análise da aplicação.

De acordo com Li et al. (2017), "o middleware é uma camada intermediária entre os dispositivos IoT e a nuvem, que fornece serviços como gerenciamento de dados, comunicação, segurança e escalabilidade". A camada de middleware atua como um facilitador da comunicação entre os dispositivos IoT e a plataforma de gerenciamento, permitindo que os dados coletados sejam processados e analisados de forma eficiente.

Para a aplicação proposta, a camada de middleware precisa ser capaz de coletar e integrar os dados provenientes de diferentes fontes. Segundo Díaz et al. (2020), "os dados coletados pelos dispositivos IoT precisam ser integrados e processados de forma eficiente para fornecer informações úteis para os usuários e aplicativos".

Além disso, a camada de middleware também desempenha um papel importante na garantia da segurança e privacidade dos dados coletados pelos dispositivos IoT. De acordo com Zhang et al. (2018), "o middleware deve implementar medidas de segurança robustas para proteger os dados confidenciais coletados pelos dispositivos IoT".

Em resumo, a camada de middleware é fundamental para o funcionamento de

uma aplicação IoT que utiliza o celular do usuário como sensor de localização e a câmera como leitor. É responsabilidade da camada de middleware garantir a integração eficiente dos dados coletados pelos dispositivos IoT e a segurança dos dados transmitidos.

#### **3.2.4. Camada de Aplicação**

A camada de aplicação será responsável por coordenar a interação entre as diferentes camadas de rede e os dispositivos envolvidos na comunicação. Essa camada é responsável por garantir a integração e interoperabilidade dos diferentes dispositivos e protocolos envolvidos na comunicação, permitindo que a aplicação IoT funcione de forma harmoniosa e eficiente.

De acordo com Lloret et al. (2019), a camada de aplicação é responsável por fornecer serviços específicos para as aplicações IoT, incluindo gerenciamento de recursos, processamento de dados, controle de acesso, gerenciamento de dispositivos e interface de usuário. Para nossa aplicação IoT a camada de aplicação será responsável por coordenar a transmissão dos dados de localização e imagem para o servidor ou dispositivo de processamento.

Além disso, segundo Sun et al. (2019), a camada de aplicação deve garantir a segurança e privacidade dos dados transmitidos entre os dispositivos IoT e o servidor, especialmente quando se trata de dados pessoais do usuário, como a localização e a imagem capturada pela câmera. Nesse sentido, é importante que a aplicação implemente técnicas de criptografia e autenticação para proteger os dados transmitidos contra ataques cibernéticos.

Por fim, é importante que a camada de aplicação seja projetada de forma a garantir a escalabilidade e flexibilidade da aplicação, permitindo que novos dispositivos e serviços sejam integrados à medida que a aplicação evolui.

#### **3.2.5. Camada de Negócios**

A Camada de Negócios será responsável por gerenciar os dados coletados e processá-los de maneira a fornecer informações úteis para o usuário final. Segundo Dargie e Poellabauer (2010), a Camada de Negócios é responsável por "coordenar as atividades dos diferentes componentes de software da aplicação e garantir que os dados coletados sejam processados e transformados em informações úteis para os usuários".

Além disso, a camada de negócios deverá garantir a segurança dos dados coletados e a privacidade dos usuários. Segundo Lee et al. (2015), "a proteção dos dados é um dos principais desafios na IoT, já que os dados coletados podem ser muito sensíveis e vulneráveis a ataques". Nesse sentido, a camada de negócios deve implementar mecanismos de segurança, como criptografia de dados e autenticação de usuários, para garantir a integridade dos dados coletados.

Outra importante função da camada de negócios é a análise dos dados coletados. Segundo Gubbi et al. (2013), "a análise dos dados é uma etapa crítica na IoT, já que os dados coletados podem ser muito volumosos e não estruturados". A camada de negócios

deverá ser capaz de processar os dados de maneira a extrair informações úteis para os usuários, como padrões de comportamento ou análises estatísticas.

Por fim, a camada de negócios deverá estar integrada com as outras camadas da aplicação IoT, como a camada de acesso à rede e a camada de interface com o usuário. Segundo Atzori et al. (2010), "a integração das diferentes camadas da aplicação é fundamental para garantir o funcionamento adequado da IoT e proporcionar uma experiência de usuário satisfatória".

Em suma, a Camada de Negócios será fundamental para o funcionamento adequado da aplicação IoT que utilizará o celular do usuário como sensor de localização e a câmera como leitor. Ela será responsável por coordenar as atividades dos diferentes componentes da aplicação, garantir a segurança dos dados coletados, processar e analisar os dados para fornecer informações úteis para os usuários, e integrar as diferentes camadas da aplicação para proporcionar uma experiência de usuário satisfatória.

### **3.3. Requisitos Funcionais**

Requisitos funcionais são especificações detalhadas que descrevem as funcionalidades específicas que um sistema ou aplicação deve oferecer. Eles delineiam o que o sistema deve fazer em termos de operações, serviços ou interações com seus usuários. A seguir, são apresentados 12 requisitos funcionais, selecionados com base em aplicações similares já disponíveis no mercado, para a criação de um Minimum Viable Product (MVP):

- 01 - Cadastro de usuário;
- 02 - Edição de usuário;
- 03 - Exclusão de usuário
- 04 - Busca de usuário
- 05 - Autenticação de usuário
- 06 - Cadastro de produto
- 07 - Edição de produto
- 08 - Busca de produto
- 09 - Exclusão de produto
- 10 - Leitura de código de barras
- 11 - Indicação de Produtos Seguros
- 12 - Mostrar estabelecimentos que vendem produtos sem glúten

### **3.4. Requisitos mínimos recomendados**

Como observado por Patterson e Hennessy (2017) "Ter os requisitos mínimos para usar uma aplicação é fundamental para garantir que os usuários tenham uma experiência satisfatória e sem problemas."

Requisitos mínimos para um computador:

- Sistema operacional: Windows 7 SP1 ou superior, macOS 10.13 ou superior, ou uma distribuição Linux compatível.
- Navegador: Os navegadores modernos geralmente são compatíveis com o Angular. Recomenda-se utilizar a versão mais recente do Google Chrome, Mozilla Firefox, Safari ou Microsoft Edge.
- Hardware:

- Processador: Processador dual-core ou superior.
- Memória RAM: Pelo menos 4 GB de RAM.
- Armazenamento: Espaço livre suficiente para instalar o sistema operacional, o navegador e a aplicação.

Requisitos mínimos para um celular:

- Sistema operacional: iOS 12 ou superior para dispositivos Apple (iPhone, iPad, iPod touch) ou Android 7.0 Nougat ou superior para dispositivos Android.
- Navegador: Recomenda-se utilizar a versão mais recente do Safari para dispositivos Apple ou do Google Chrome para dispositivos Android.
  - Hardware: Os requisitos de hardware podem variar dependendo do dispositivo específico, mas, em geral, é recomendável ter um processador de pelo menos 1,5 GHz, 2 GB de RAM e espaço livre suficiente para instalar o sistema operacional, o navegador e a aplicação.

### **3.5. Banco de Dados**

O SQL Server é um sistema de gerenciamento de banco de dados relacional (SGBDR) desenvolvido pela Microsoft. É amplamente utilizado para armazenar, gerenciar e recuperar dados em aplicativos empresariais e, portanto, é uma escolha popular para aplicações web.

### **3.6. Casos de Uso**

O diagrama de casos de uso é uma ferramenta fundamental na engenharia de requisitos de software, pois permite uma representação visual dos atores envolvidos em um sistema e das interações entre eles. Ele descreve como os usuários (atores) interagem com o sistema, identificando as principais funcionalidades e os fluxos de eventos que ocorrem durante a utilização do software.

A importância do diagrama de casos de uso reside na sua capacidade de capturar os requisitos funcionais do sistema de forma clara e compreensível. Ele fornece uma visão geral das funcionalidades do sistema e ajuda a estabelecer uma base comum de entendimento entre os stakeholders, incluindo desenvolvedores, clientes e usuários finais.

De acordo com Larman (2004), renomado autor e especialista em engenharia de software, "O caso de uso é uma ferramenta poderosa para capturar requisitos do usuário. Ele descreve o comportamento do sistema de um ponto de vista externo."

Para esse trabalho foram criados três diagramas sendo eles o de administrador do sistema representado na Figura 3, o de usuário comum representado na Figura 4 e o de usuário parceiro representado na Figura 5.

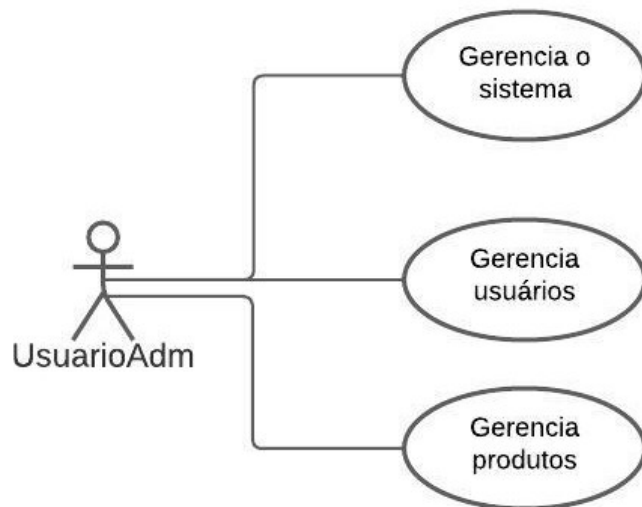


Figura 3. Figura do diagrama de casos de uso do administrador do sistema

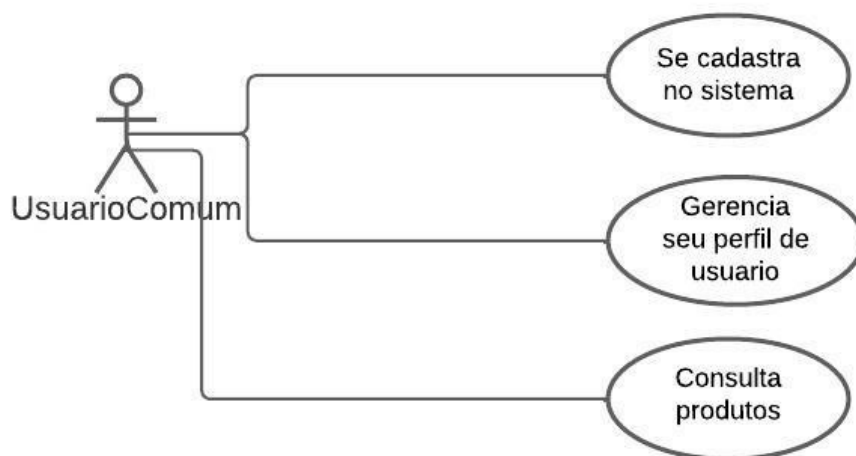


Figura 4. Figura do diagrama de casos de uso do usuário comum

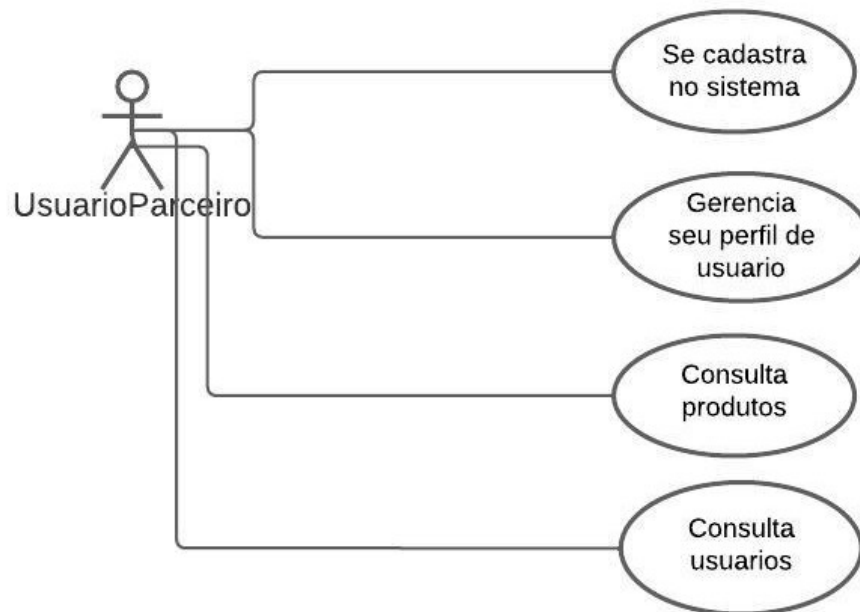


Figura 5. Figura do diagrama de casos de uso do usuário parceiro

### 3.7. Front-End

O *frontend* da aplicação web desempenha um papel crucial na experiência do usuário e na usabilidade da plataforma. No contexto da doença celíaca, uma aplicação web bem projetada pode ajudar os pacientes a gerenciar sua dieta e estilo de vida de forma mais eficaz, proporcionando informações claras e acessíveis sobre os alimentos que podem consumir com segurança.

A escolha das tecnologias é uma parte fundamental do desenvolvimento do *frontend*. No caso descrito, o uso do *framework Angular* foi uma excelente escolha. O *Angular* é um *framework JavaScript* mantido pelo Google e isso permitiu a criação de uma aplicação web dinâmica e de alta performance. Sua arquitetura modular e recursos avançados fizeram dele uma opção sólida e confiável.

O uso do SCSS (Sass) para estilização ofereceu maior flexibilidade e organização ao escrever estilos para a aplicação. Isso permitiu a criação de folhas de estilo mais eficientes e reutilizáveis, tornando a manutenção e o desenvolvimento mais fáceis no longo prazo.

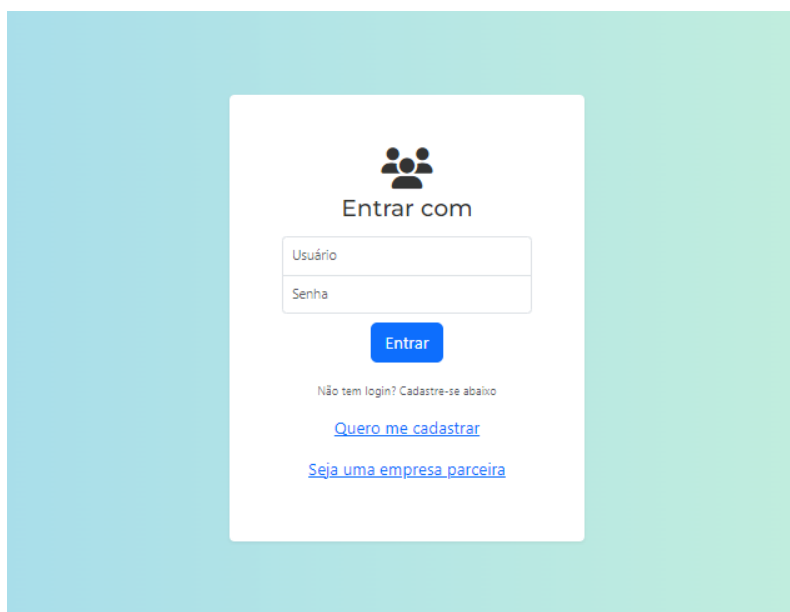
O HTML é a linguagem fundamental para estruturar o conteúdo da aplicação web. A utilização correta de marcação semântica, como cabeçalhos, parágrafos, listas e tabelas, foi crucial para garantir a acessibilidade e usabilidade da aplicação.

O *ngx-bootstrap* é uma biblioteca de componentes Angular baseada no *Bootstrap*, um dos frameworks mais populares para desenvolvimento de interfaces de usuário responsivas e estilizadas. A utilização do *ngx-bootstrap* simplificou a criação de compo-

nentes interativos, como o modal e carousel, economizando tempo de desenvolvimento e melhorando a experiência do usuário.

Por fim, para gerenciar as dependências da aplicação, fora utilizado o gerenciador de pacotes *Node Package Manager (NPM)*, e o sistema de controle de versões *Git*, que possibilita o versionamento do código, além de facilitar o trabalho e a conferência das etapas realizadas pela equipe através do *Github*.

Tela de *login*: Permitir que os usuários acessem suas contas na plataforma. Figura 6:



**Figura 6. Tela de login**

Tela de cadastro de usuário: Permitir o cadastro de novos usuários. Figura 7:

**Cadastro de Usuário**

Nome:

Email:

Data de nascimento:

Telefone:

Senha:

Confirmar Senha:

Cep:

Rua:

Numero:

Complemento:

Bairro:

Cidade:

Estado:

Selecione suas intolerâncias:

- Intolerante ao glúten
- Intolerante à lactose
- Intolerante à aveia

Eu concordo com os [Termos de Uso](#).

[Registrar](#)

[Já sou Cadastrado](#)

SOBRE PRODUTOS CONTATO

Se você está pronto para abraçar uma vida mais saudável e livre de glúten, seremos seu parceiro. Não deixe o glúten controlar sua dieta, controle você mesmo. Siga nossas redes sociais e nos ajude nessa luta por uma vida mais saudável!

f @ i

© 2023 Vida Sem Glúten

**Figura 7. Tela de cadastro de usuário**

Tela de cadastro de usuário parceiro: Permitir o cadastro de novos parceiros ao projeto. Figura 8:

**Cadastro de Empresa Parceira**

Nome:

Data de fundação da empresa:

WebSite:

Email:

Senha:

Confirmar Senha:

Descrição:

Cep:

Rua:

Numero:

Complemento:

Bairro:

Cidade:

Estado:

Eu concordo com os [Termos de Uso](#).

[Registrar](#)

[Já sou Cadastrado](#)

SOBRE PRODUTOS CONTATO

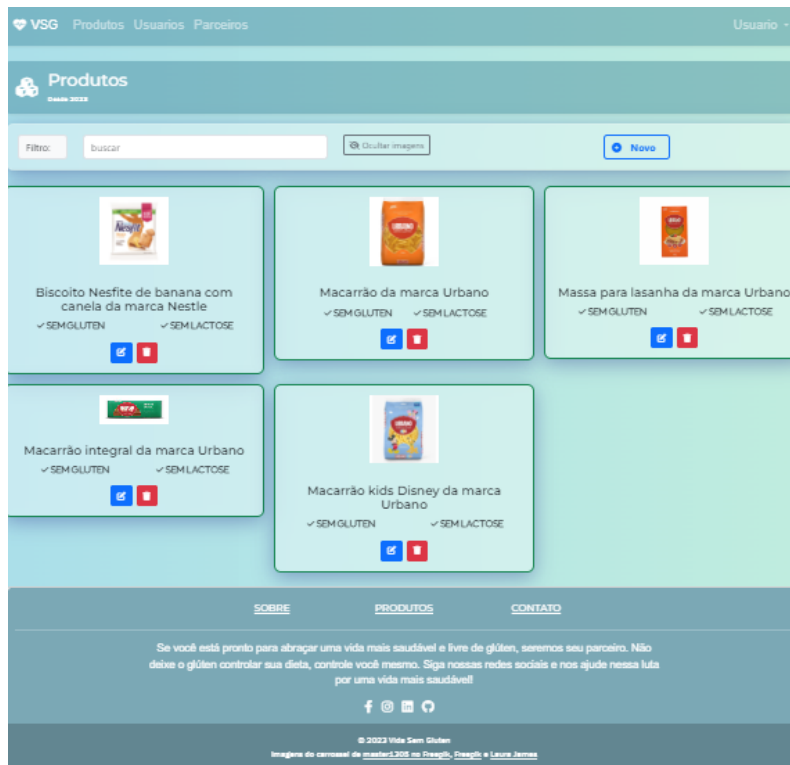
Se você está pronto para abraçar uma vida mais saudável e livre de glúten, seremos seu parceiro. Não deixe o glúten controlar sua dieta, controle você mesmo. Siga nossas redes sociais e nos ajude nessa luta por uma vida mais saudável!

f @ i

© 2023 Vida Sem Glúten  
Imagens do carrossel de: [maior1305.no.freepik](#), [freepik](#) e [Laura James](#)

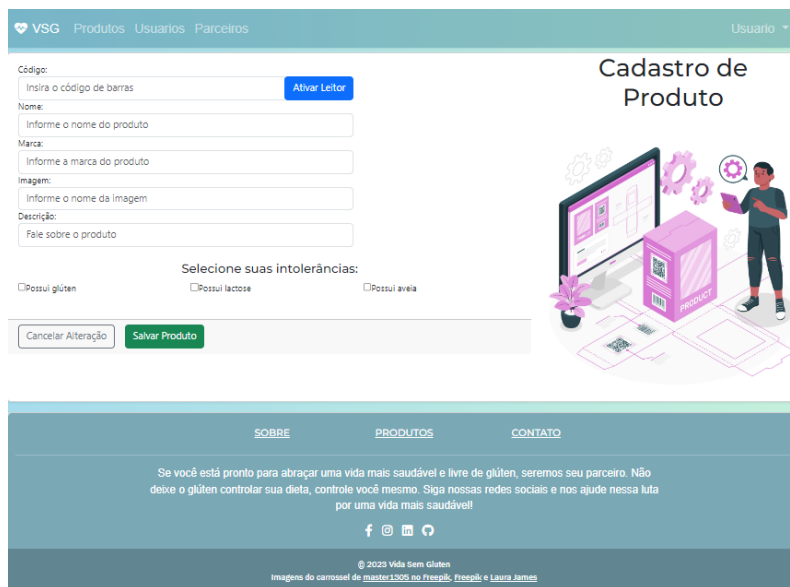
**Figura 8. Tela de cadastro de usuário parceiro**

Pesquisa de Alimentos: Uma função de pesquisa avançada que permite aos usuários procurar alimentos específicos e obter informações sobre sua adequação para pessoas com doença celíaca. Figura 9:



**Figura 9. Tela de pesquisa de produtos**

Tela de cadastro de produtos: Permite o cadastro de novos produtos na aplicação.  
 Figura 10:



**Figura 10. Tela de pesquisa de produtos**

Tela inicial: Apresenta a pagina e sua proposta aos usuarios. Figura 11:



Figura 11. Tela inicial

### 3.8. Back-End

No lado do servidor a aplicação foi criada no modelo de 3 camadas, que é uma arquitetura comum para desenvolver aplicativos no mundo do desenvolvimento de *software*. Ele é usado para separar as preocupações e organizar o código de uma maneira que torna o desenvolvimento, a manutenção e a escalabilidade mais fáceis. Quando se trata de uma aplicação em C#, o *backend* geralmente segue o modelo de 3 camadas, que consiste em três camadas principais:

- Camada de apresentação: Para conectar a camada de backend à camada de apresentação, foi construído uma API que permite à camada de apresentação interagir com a lógica de negócios. Isso pode ser feito usando protocolos como HTTP (*APIs RESTful*) ou outras tecnologias de comunicação, dependendo dos requisitos do projeto. Figura 12:

```

1 using Microsoft.AspNetCore.Mvc;
2 using VidasemGluten.Domain;
3 using VidasemGluten.Repositories;
4
5 namespace VidasemGluten.API.Controllers
6 {
7     [ApiController]
8     [Route("api")]
9     public class ProdutoController : ControllerBase
10    {
11        private IProdutoRepository _repository { get; set; }
12
13        /// <summary>
14        /// Efetua injeção de dependência que já está registrada no sistema de injeção (IServiceCollection)
15        /// </summary>
16        /// <param name="repository"></param>
17        public ProdutoController(IProdutoRepository repository)
18        {
19            _repository = repository;
20        }
21
22        [HttpPost]
23        [Route("produtos")]
24        public async Task<ActionResult> PostProduto([FromBody] Produto novoProduto)
25        {
26            _repository.Criar(novoProduto);
27        }
28
29        [HttpGet]
30        [Route("produtos")]
31        public async Task<ActionResult> GetProdutos()
32        {
33            return Ok(_repository.Listar());
34        }
35
36        [HttpGet]
37        [Route("codigo")]
38        public async Task<ActionResult> GetProduto(string codigo)
39        {
40            return Ok(_repository.Buscar(codigo));
41        }
42
43        [Authorize(Roles = "admin")]
44        [HttpPut]
45        [Route("produtos")]
46        public async Task<ActionResult> Put([FromBody] Produto produtoEditado)
47        {
48            _repository.Atualizar(produtoEditado);
49        }
50
51        [Authorize(Roles = "admin")]
52        [HttpDelete]
53        [Route("produtos/{codigo}")]
54        public async Task<ActionResult> Delete(string codigo)
55        {
56            _repository.Excluir(codigo);
57        }
58    }
59 }

```

Figura 12. Código da controladora produtos da api

- Camada de lógica de negócios : A camada de lógica de negócios é o coração da aplicação e é responsável por processar e manipular os dados de acordo com as regras de negócios da aplicação. Figura 13:

```

4 using System.Threading.Tasks;
5
6 namespace VidasemGluten.Domain
7 {
8     public class Produto
9     {
10        public Guid Id { get; set; }
11        public string Codigo { get; set; }
12        public string? ImagemUrl { get; set; }
13        public string? Marca { get; set; }
14        public string? Nome { get; set; }
15        public string? Descricao { get; set; }
16        public bool? PossuiGluten { get; set; }
17        public bool? PossuiLactose { get; set; }
18        public bool? PossuiAveia { get; set; }
19
20        public List<Usuario> Usuarios { get; set; }
21
22        public override string ToString()
23        {
24            return $" Id:{Codigo}\n Nome: {Nome}\n Descrição: {Descricao}\n";
25        }
26    }
27 }

```

Figura 13. Código da classe produto

- Camada de acesso a dados: A camada de acesso a dados é responsável por interagir com o banco de dados ou qualquer outra fonte de dados utilizada pela aplicação. Nessa aplicação foi usado o padrão *repository* para uma maior abstração e foi criado o banco de dados usando o *Entity Framework Core* como é melhor explicado abaixo:

- Configurações de entidades:

O código na imagem faz parte de uma configuração de mapeamento de entidades usando o *Entity Framework Core*. Ele está configurando como a entidade *Produto* deve ser mapeada para o banco de dados. Figura 14:

```

1  using Microsoft.EntityFrameworkCore;
2  using Microsoft.EntityFrameworkCore.Metadata.Builders;
3  using vidaSemGluten.Domain;
4
5  namespace vidaSemGluten.Infra.Data.Configuracoes.Entidades;
6
7  public class ProdutoConfiguration: IEntityTypeConfiguration<Produto>
8  {
9      public void Configure(EntityTypeBuilder<Produto> builder)
10     {
11         builder.HasKey(prod => prod.Id);
12
13         builder.Property(prod => prod.Codigo)
14             .HasMaxLength(150)
15             .IsRequired(true);
16
17         builder.Property(prod => prod.Descricao).HasMaxLength(200).IsRequired(false);
18
19         builder.Property(prod => prod.Nome).HasMaxLength(100).IsRequired(true);
20
21         builder.Property(prod => prod.Marca).HasMaxLength(100).IsRequired(true);
22
23         builder.Property(prod => prod.ImageUrl).HasMaxLength(255).IsRequired(true);
24
25         builder.Property(prod => prod.PossuiAveia).IsRequired(true);
26         builder.Property(prod => prod.PossuiGluten).IsRequired(true);
27         builder.Property(prod => prod.PossuiLactose).IsRequired(true);
28
29         builder.HasMany(prod => prod.Usuarios).WithMany(usr => usr.Produtos);
30     }
31 }

```

Figura 14. Código de configuração da entidade Produto

– Migrations:

A imagem a seguir é uma migração para um banco de dados usando o *Entity Framework Core*. Ele está escrito em C# e utiliza o *Microsoft.EntityFrameworkCore.Migrations* para criar tabelas no banco de dados. Figura 15:

```

1  using System;
2  using Microsoft.EntityFrameworkCore.Migrations;
3
4  #nullable disable
5
6  namespace vidaSemGluten.Infra.Data.Migrations
7  {
8      public partial class InitialMigrations : Migration
9      {
10         protected override void up(MigrationBuilder migrationBuilder)
11         {
12             migrationBuilder.CreateTable(
13                 name: "Enderecos",
14                 columns: table => new
15                 {
16                     Id = table.Column<Guid>(type: "uniqueidentifier", nullable: false),
17                     Rua = table.Column<string>(type: "nvarchar(max)", nullable: true),
18                     Numero = table.Column<int>(type: "int", nullable: true),
19                     Bairro = table.Column<string>(type: "nvarchar(max)", nullable: true),
20                     Cidade = table.Column<string>(type: "nvarchar(max)", nullable: true),
21                     Estado = table.Column<string>(type: "nvarchar(max)", nullable: true),
22                     Cep = table.Column<string>(type: "nvarchar(max)", nullable: true),
23                     Complemento = table.Column<string>(type: "nvarchar(max)", nullable: true)
24                 },
25                 constraints: table =>
26                 {
27                     table.PrimaryKey("PK_Enderecos", x => x.Id);
28                 });
29
30             migrationBuilder.CreateTable(
31                 name: "Produtos",
32                 columns: table => new
33                 {
34                     Id = table.Column<Guid>(type: "uniqueidentifier", nullable: false),
35                     Codigo = table.Column<string>(type: "nvarchar(150)", maxLength: 150, nullable: false),
36                     ImageUrl = table.Column<string>(type: "nvarchar(255)", maxLength: 255, nullable: false),
37                     Marca = table.Column<string>(type: "nvarchar(100)", maxLength: 100, nullable: false),
38                     Nome = table.Column<string>(type: "nvarchar(100)", maxLength: 100, nullable: false),
39                     Descricao = table.Column<string>(type: "nvarchar(200)", maxLength: 200, nullable: true),
40                     PossuiGluten = table.Column<bool>(type: "bit", nullable: false),
41                     PossuiLactose = table.Column<bool>(type: "bit", nullable: false),
42                     PossuiAveia = table.Column<bool>(type: "bit", nullable: false)
43                 },
44                 constraints: table =>
45                 {
46                     table.PrimaryKey("PK_Produtos", x => x.Id);
47                 });
48         }
49     }
50 }

```

Figura 15. Código da classe ProdutoRepository

– Padrão Repository:

Este código é um exemplo de implementação de um repositório em C# utilizando o *Entity Framework Core* para operações CRUD (*Create, Read, Update, Delete*) em entidades do tipo *Produto*. Figura 16:

```

1 using Microsoft.EntityFrameworkCore;
2 using vidaSemGluten.Domain;
3 using vidaSemGluten.Domain.Repositories;
4
5
6 namespace vidaSemGluten.Infra.Data
7 {
8     public class ProdutoRepository : IProdutoRepository
9     {
10         private readonly VidaSemGlutenContext _context;
11
12         public ProdutoRepository(VidaSemGlutenContext context)
13         {
14             _context = context;
15         }
16
17         public async Task InserirProduto(Produto novoProduto)
18         {
19             _context.Produtos.Add(novoProduto);
20             await _context.SaveChangesAsync();
21         }
22
23         public async Task<IList<Produto>> BuscaTodosProdutos()
24         {
25             return await _context.Produtos.ToListAsync();
26         }
27
28         public async Task<Produto> BuscaProduto(string codigo)
29         {
30             return await _context.Produtos.FirstOrDefaultAsync(producao => producao.Codigo == codigo);
31         }
32
33         public async Task EditarProduto(Produto produtoEditado)
34         {
35             _context.Produtos.Update(produtoEditado);
36             await _context.SaveChangesAsync();
37         }
38
39         public async Task DeletarProduto(string codigo)
40         {
41             var produto = await _context.Produtos.FirstOrDefaultAsync(producao => producao.Codigo == codigo);
42             if (produto is not null)
43             {
44                 _context.Produtos.Remove(produto);
45                 await _context.SaveChangesAsync();
46             }
47         }
48     }
49 }

```

Figura 16. Código da classe ProdutoRepository

Este código é um repositório genérico chamado BaseRepository<T> que implementa a interface IBaseRepository<T> para operações básicas de CRUD em entidades do tipo T, onde T deve ser um tipo que herda da classe *Usuario*. Figura 17:

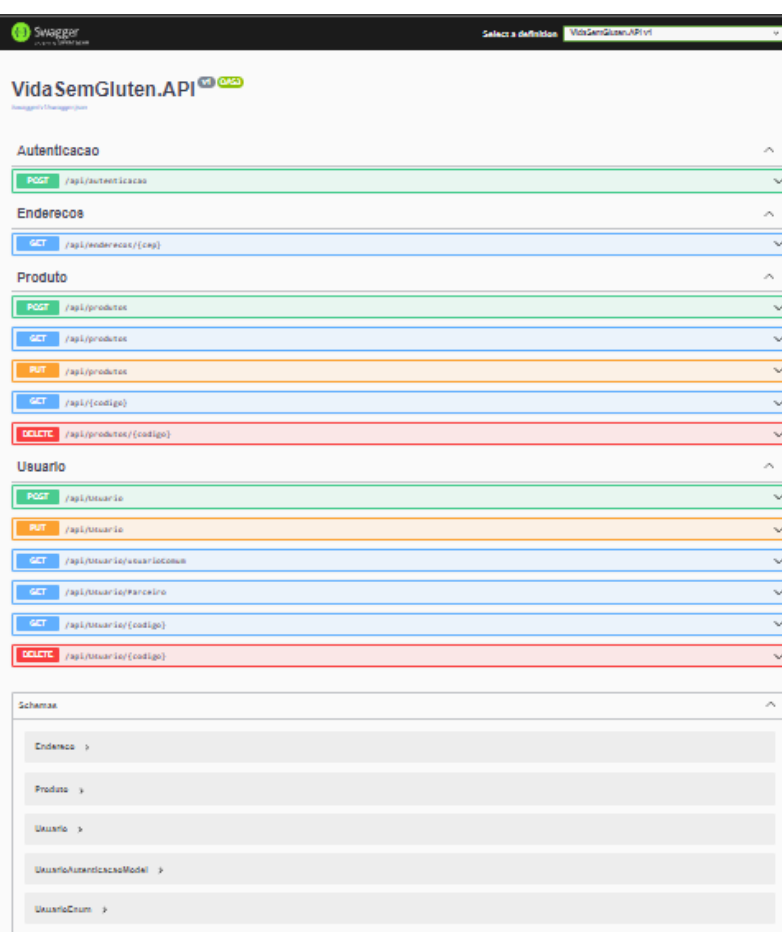
```

1 using Microsoft.EntityFrameworkCore;
2 using vidaSemGluten.Domain;
3 using vidaSemGluten.Domain.Repositories;
4
5 namespace vidaSemGluten.Infra.Data;
6
7 public class BaseRepository<T> : IBaseRepository<T> where T : Usuario
8 {
9     protected VidaSemGlutenContext _context;
10    protected DbSet<T> _entities;
11
12    public BaseRepository(VidaSemGlutenContext context)
13    {
14        ...
15    }
16
17    public async Task Inserir(T entidade)
18    {
19        ...
20    }
21
22    public async Task<IList<T>> BuscarTodos()
23    {
24        ...
25    }
26
27    public async Task<T> Buscar(Guid id)
28    {
29        ...
30    }
31
32    public async Task Editar(T entidadeEditada)
33    {
34        ...
35    }
36
37    public async Task Deletar(Guid id)
38    {
39        ...
40    }
41
42    ...
43
44    ...
45
46    ...
47 }

```

Figura 17. Código da classe BaseRepository

Para a realização de testes durante a produção foi usado o *Swagger*: Figura 18:



**Figura 18. Código da classe ProdutoRepository**

Essa arquitetura ajuda a tornar a aplicação mais modular, escalável e fácil de manter à medida que novos recursos são adicionados ou requisitos de negócios evoluem. Além disso, facilita a testabilidade das diferentes partes da aplicação, permitindo a criação de testes unitários para a camada de lógica de negócios e a camada de acesso a dados.

#### 4. Conclusões

O presente trabalho de conclusão de curso abordou a relevante temática do desenvolvimento de uma aplicação web destinada ao controle da doença celíaca, as tecnologias necessárias para o desenvolvimento da solução foram selecionadas pela afinidade com elas, visto que o intuito do trabalho é a viabilidade da solução. Assim, reduzindo o tempo de desenvolvimento, visto que não há uma curva de aprendizado tão grande para a execução da solução, e que posteriormente pode ser migrada para utilização de outras ferramentas. A doença celíaca é uma condição crônica que exige uma gestão cuidadosa da dieta, e a proposta desta aplicação visou oferecer um suporte eficaz aos indivíduos afetados, facilitando o monitoramento e a administração de uma alimentação isenta de glúten.

Após a realização de alguns testes e a avaliação dos resultados, foi possível afirmar que o problema proposto foi resolvido de maneira satisfatória. A aplicação desenvolvida demonstrou ser capaz de facilitar o controle da dieta sem glúten, permitindo que os usuários façam suas compras com mais segurança, ao acessar informações relevantes

sobre os alimentos. Entretanto, é crucial registrar algumas limitações e características importantes identificadas durante o desenvolvimento e implementação. Algumas limitações incluem a dependência da precisão das informações alimentares fornecidas pelos usuários e a necessidade contínua de atualizações na base de dados de alimentos sem glúten. Além disso, considerações de usabilidade e acessibilidade devem ser continuamente aprimoradas para garantir a efetividade da aplicação em diferentes contextos.

No tocante aos objetivos estabelecidos, destaca-se que foram alcançados de maneira geral. A aplicação proporcionou uma solução tecnológica que atende às necessidades específicas dos celíacos, contribuindo para a melhoria da qualidade de vida desses indivíduos.

No parágrafo final, é importante ressaltar que o desenvolvimento desta aplicação representa um ponto de partida significativo, mas há espaço para avanços futuros. Trabalhos adicionais podem se concentrar em aprimorar a aplicação e futuramente integrá-la uma inteligência artificial para a identificação automática de produtos sem glúten e para retirar possíveis dúvidas sobre a doença, produtos e sobre a aplicação, integrar funcionalidades de geolocalização para facilitar a busca por restaurantes e estabelecimentos adequados, e expandir a aplicação para plataformas móveis, ampliando assim o alcance e a acessibilidade da solução proposta.

Em suma, a aplicação desenvolvida revelou-se efetiva no auxílio ao controle da doença celíaca, cumprindo os objetivos estabelecidos, mas reconhecendo a necessidade de contínuo aprimoramento e inovação para melhor atender às demandas e expectativas dos usuários, bem como para enfrentar os desafios inerentes a uma condição complexa como a doença celíaca.

## Referências

- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., e Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols and applications. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376.
- Amit, R. e Zott, C. (2012). Creating value through business model innovation. 53(3):41–49.
- Atzori, L., Iera, A., e Morabito, G. (2010). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.
- Baptista, C. F. (2019). Consumo de produtos sem glúten deve crescer 40% até 2022. Disponível em: <https://agemt.pucsp.br/noticias/consumo-de-produtos-sem-gluten-deve-crescer-40-ate-2022#:~:text=Segundo%20dados%20da%20OMS%20>, Acesso em 19 mar 2023.
- Borghini, F. e et al. (2021). The efficacy of a mobile application in improving adherence to a gluten-free diet in patients with celiac disease: A randomized controlled trial. *Nutrients*, 13(3):1015.
- Chinnici, F., Guerrieri, M., Grasso, A., e Russo, P. (2020). Usefulness of three gluten-free diet applications for celiac patients. *Digestive and Liver Disease*, 52(9):1045–1051.
- Chomutare, T. e et al. (2015). Designing personalized mobile applications for individuals with food intolerances. *Personal and Ubiquitous Computing*, 19(2):313–323.
- CIÊNCIA, S. C. (2019). Rótulos de alimentos não apresentam informações claras. Disponível em: <https://www.medicina.ufmg.br/>

- rotulos-de-alimentos-nao-apresentam-informacoes-claras/, Acesso em 19 mar 2023.
- Comino, I., Sousa, C., Vivas, S., Lozano, M., de Lourdes Moreno, M., Fernández, J., .. e Cebolla, (2016). Detection of gluten in gluten-free labeled restaurant food: analysis of crowd-sourced data. *American Journal of Gastroenterology*, 111(2):265–273.
- Dargie, W. e Poellabauer, C. (2010). *Fundamentals of wireless sensor networks: theory and practice*, volume 1. John Wiley Sons.
- Díaz, M., Pérez, M., e Rubio, B. (2020). Middleware for the internet of things: a survey. *Journal of Systems Architectures*, 108:101688.
- FENACELBRA (2021). Dados estatísticos de doença celíaca. Disponível em: <https://www.fenacelbra.com.br/dados-estatisticos>, Acesso em 19 mar 2023.
- FGV (2021). Retrospectiva 2021: Brasil tem dois dispositivos digitais por habitante, revela pesquisa da fgv. Disponível em: <https://portal.fgv.br/noticias>, Acesso em 19 mar 2023.
- Gubbi, J., Buyya, R., Marusic, S., e Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660.
- Khan, R., Khan, S. U., Zaheer, R., e Khan, S. (2012). Future internet: The internet of things architecture, possible applications and key challenges. In *10th International Conference on Frontiers of Information Technology*, page 257–260.
- Larman, C. (2004). *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design.*, volume 1. Prentice Hall.
- Lee, J., Bagheri, B., e Kao, H. A. (2015). A cyber-physical systems architecture for industry 4.0. Disponível em: <https://www.sciencedirect.com/science/article/pii/S221384631400025X>, Acesso em 19 abr 2023.
- Li, D., Li, T., Li, C., e Li, W. (2020). Iot-based real-time monitoring and control system for large-scale aquaponics. *IEEE Access*, 8:218194–218204.
- Li, S., Xu, L. D., e Zhao, S. (2017). The internet of things: A survey. information systems frontiers. *Journal of Network and Computer Applications*, 17(2):243–259.
- Licks, F. R. e Busata, R. (2020). Loomos. Disponível em: <https://www.loomos.com.br/>, Acesso em 23 mar 2023.
- Lloret, J., Garcia, M., Bri, D., Sendra, S., e Serrano, J. J. (2019). An architecture based on iot, big data and blockchain to guarantee secure transactions in smart cities. *Sensors*, 19(19):4116.
- Magrini, A. e et al. (2020). Mobile applications for individuals with food intolerances: State of the art and future directions. *Journal of Medical Internet Research*, 22(6):e18243.
- Maschietto, L. G. e et al. (2021). *Arquitetura e infraestrutura de IoT*, volume 1. Grupo A.
- Parker, G., Van Alstyne, M., e Choudary, S. P. (2016). Platform revolution: How networked markets are transforming the economy and how to make them work for you. 53(3):41–49.
- Patterson, D. A. e Hennessy, J. L. (2017). *Computer Organization and Design: The Hardware/Software Interface*, volume 1. Morgan Kaufmann.
- SAÚDE, M. D. (2020). Doença celíaca. Disponível em: [https://bvsm.saude.gov.br/doenca-celiaca/#:~:text=A%20maior%20dificuldade%](https://bvsm.saude.gov.br/doenca-celiaca/#:~:text=A%20maior%20dificuldade%20)

20para%20os, rigorosamente%20pele%20resto%20da%20vida, Acesso em 19 mar 2023.

Silvester, J. A. e Duerksen, D. R. (2016). A review of dietary apps for the ketogenic diet. *Current gastroenterology reports*, 18(7):33.

Sun, Y., Zhang, Y., Zhang, J., Gui, C., e Chen, J. (2019). A survey of the applications of iot in healthcare. *Journal of Medical Systems*, 43(3):58.

Zhang, Y., Wen, Y., e Xu, L. D. (2018). Security models and requirements for health-care application clouds in internet of things. *Future Generation Computer Systems*, 86:1184–1194.