

CENTRO UNIVERSITÁRIO SENAC

Disciplina: Projeto Integrador III

IMPLANTAÇÃO DE BANCO DE DADOS

São Paulo

2023

CENTRO UNIVERSITÁRIO SENAC

Disciplina: Projeto Integrador III

BRUNA CORRÊA SABEL

CLOVES MENEGHIN DE OLIVEIRA

GABRIELA MARINA ABDO

LEONARDO CAVALCANTE DA SILVA

MAIBI CRISTINA MARTINS LEITE

THIAGO FELIPE SILVA DE OLIVEIRA

IMPLANTAÇÃO DE BANCO DE DADOS

Projeto Integrador III – Fase 2 apresentado ao Centro Universitário Senac, como exigência parcial para obtenção de aprovação na disciplina Projeto Integrador III, do curso de Tecnologia em Banco de Dados.

Orientador (a): Prof. (ª) Debora Batista Da Silva Paulo

São Paulo

2023

RESUMO

O projeto apresentado tem por objetivo desenvolver um aplicativo que auxilie na seleção de receitas com base nos ingredientes disponíveis em casa. O aplicativo contará com um scanner para reconhecer alimentos através do leitor de QR code, tornando a listagem de ingredientes disponíveis mais rápida e ágil. O projeto foi motivado pela necessidade de otimizar e facilitar a alimentação diária. A app fornecerá valores nutricionais, análises de ingredientes e sugestões de receitas com o produto selecionado, tornando-a útil para utilizadores habituais e profissionais de nutrição e gastronomia. A metodologia utilizada é baseada no framework Scrum, promovendo a comunicação e colaboração entre os membros do time por meio de diversas cerimônias. A abordagem ágil permite a melhoria contínua e adaptação às mudanças durante as etapas de produção, resultando em uma entrega mais rápida do produto. O projeto atende a requisitos funcionais e não funcionais, incluindo capacidade de visualização de imagens, processamento de imagem, reconhecimento de texto, interface do usuário, armazenamento de dados, desempenho, confiabilidade, escalabilidade e segurança.

Palavras-Chave: Ingredientes. Receitas. Alimentos. QR Code.

SUMÁRIO

1	Introdução - Visão geral do produto	5
1.1	Contextualização e motivação	5
1.2	Inserção no Mercado	6
1.3	Vantagens para o Usuário	6
1.4	Objetivos	7
1.5	Metodologia	7
2.	Estudo de viabilidade	9
2.1	Entendimento sobre a complexidade de desenvolvimento do produto na esfera técnica e econômica	9
2.2	Estimativa de esforço para o desenvolvimento da solução	10
2.3	Apresentação de soluções alternativas que atendam à expectativa do demandante	11
3	Especificação de requisitos do produto	12
3.1	- Descrição sobre o processo de licitação / extração requisitos.	12
3.2	- Análise do resultado do processo de extração e elaboração das regras de negócio e requisitos	12
3.3	Classificação dos Requisitos	14
4	Modelo de Dados	16
4.1	Levantamento do Grupo de Dados e afinidades entre eles	16
4.2	Elaboração do Diagrama Entidade Relacionamento(DER)	17
4.3	Modelo Lógico e Normalização	18
4. 4	Modelo Físico	22
5	Linguagem e SGBD utilizados	23
5.1	Código Fonte	23
6	Conclusão	27
	REFERÊNCIAS	28

1 Introdução - Visão geral do produto

Com o objetivo de aumentar a eficiência dos processos diários, tem se aumentado a busca por alternativas para economizar tempo. Visando proporcionar uma alimentação balanceada e diversificada, o projeto apresenta um aplicativo que busca auxiliar na escolha de receitas com base nos ingredientes disponíveis em casa.

O aplicativo contará com um scanner que reconhecerá os alimentos por meio do leitor de QR code, tornando a listagem dos ingredientes disponíveis mais rápida e ágil. Dessa forma, os usuários poderão selecionar suas receitas preferidas sem a necessidade de verificar manualmente o que possuem em sua dispensa. Com isso espera-se contribuir para o bem-estar e a praticidade do dia a dia.

1.1 Contextualização e motivação

A preparação de alimentos é universal e desenvolveu-se ao longo da história da humanidade. Cozinhar além de ser uma atividade integral e necessária que une as famílias também pode trazer um grande benefício econômico para muitos.

Pode-se dizer que muitas pessoas não são boas na cozinha e que por esse motivo necessitam e dependem muito de receitas para aprender a cozinhar.

Existem também aqueles que na correria do dia-dia não encontram um tempo necessário para se dedicar à prática de cozinhar, mas também não gostam de optar por uma alimentação de fast food e não saudável.

Por isso, seja para tentar otimizar e facilitar sua alimentação ou se guiar com uma lista de ingredientes por não saber cozinhar, muitas pessoas procuram aplicativos de culinária para poder auxiliá-las.

Foi pensando nesse contexto que se optou por um aplicativo que irá reconhecer os ingredientes disponíveis escaneando um determinado produto.

A ideia é que o usuário possa através da câmera de seu smartphone apontar para um produto e o aplicativo trará como resultado uma lista de receitas relacionadas ao produto escolhido. Assim que o usuário escolher a receita de sua preferência o aplicativo será capaz de apontar todos os ingredientes necessários para realizar a receita.

Existem diversos aplicativos no mercado que são capazes de mostrar, através de pesquisa realizada pelo usuário, os valores nutricionais e análise dos ingredientes. A proposta deste trabalho inova na medida que além de informar os valores nutricionais, o aplicativo fornece diversas sugestões de receitas com o produto escolhido facilitando o dia-dia do usuário.

1.2 Inserção no Mercado

O mercado de aplicativos de receitas está em constante crescimento. De acordo com um estudo da App Annie, o setor faturou US\$1,2 bilhão em 2022 e deve crescer para US\$1,5 bilhão em 2023.

Esse crescimento é impulsionado por diversos fatores, incluindo:

- O aumento da popularidade de aplicativos móveis;
- A crescente demanda por receitas saudáveis e sustentáveis;
- O interesse das pessoas em aprender a cozinhar.

Para se inserir nesse mercado, é importante que o aplicativo ofereça vantagens competitivas. Algumas possibilidades incluem:

- Uma grande variedade de receitas: O aplicativo deve oferecer uma variedade de receitas para atender a diferentes gostos e necessidades.
- Informações nutricionais: As receitas devem incluir informações nutricionais para que os usuários possam fazer escolhas saudáveis.
- Instruções fáceis de seguir: As instruções das receitas devem ser claras e concisas para que os usuários possam prepará-las com facilidade.
- Funcionalidades adicionais: O aplicativo pode oferecer funcionalidades adicionais, como dicas de culinária, ferramentas para gerenciamento de receitas e até mesmo um serviço de entrega de ingredientes.

1.3 Vantagens para o Usuário

Um aplicativo de receitas pode oferecer diversas vantagens para o usuário, incluindo:

Agilidade: O aplicativo pode ajudar o usuário a encontrar receitas rapidamente e preparar refeições com facilidade.

Inspiração: O aplicativo pode inspirar o usuário a experimentar novos sabores e pratos.

Economia: O aplicativo pode ajudar o usuário a economizar dinheiro, pois ele pode encontrar receitas que usam ingredientes que ele já tem em casa.

Além disso, o aplicativo pode ser uma ferramenta útil para pessoas que desejam aprender a cozinhar ou que buscam receitas saudáveis e sustentáveis.

1.4 Objetivos

O produto apresentado tem por objetivo indicar receitas a partir do scan ou pesquisa de um determinado produto, além de fornecer suas informações nutricionais, pode também fornecer as informações referentes ao produto escaneado ou dos itens presentes na composição da receita selecionada.

Sendo assim, a ferramenta se torna útil não somente para o usuário regular, mas também para os profissionais das áreas da nutrição e gastronomia, mas também para o usuário regular permitindo que eles possam elaborar refeições balanceadas e saudáveis.

1.5 Metodologia

Baseada nos princípios dos métodos ágeis, seu objetivo é reduzir o tempo de entrega de produtos e se adaptar a mudanças com facilidade durante as etapas de produção.

Ken Schwaber e Jeff Sutherland, desenvolvedores renomados, criaram a metodologia Scrum como um framework simples para lidar com projetos complexos. Ela é uma das várias metodologias ágeis, frequentemente utilizada por desenvolvedores de softwares e sistemas.

A metodologia é baseada em ciclos de atividades chamados Sprints, que consistem em um período de tempo predefinido em que a equipe deve completar tarefas previamente planejadas. Ao utilizar a metodologia Scrum, é possível potencializar o trabalho em equipe e monitorar a evolução do produto, mantendo um foco constante na qualidade da produção e no cumprimento dos prazos estabelecidos.

Uma das principais vantagens da metodologia Scrum é a entrega ágil da produção, pois a equipe pode corrigir problemas ao longo do processo com base nos feedbacks dos usuários obtidos durante os Sprints. Isso significa que a equipe não precisa esperar até o final do projeto para que o cliente avalie o produto, pois as correções necessárias são feitas durante os Sprints, durante a produção.

O Scrum é composto por três papéis principais: o Product Owner, o Scrum Master e a equipe de desenvolvimento. O Product Owner é responsável por definir as funcionalidades que devem ser entregues e priorizar o backlog de produtos. O Scrum Master é responsável por garantir que a equipe esteja seguindo o processo corretamente e remover obstáculos que possam prejudicar a produtividade. A equipe de desenvolvimento é responsável por entregar as funcionalidades durante as sprints.

O Scrum promove a comunicação e colaboração entre os membros da equipe por meio de diversas cerimônias, como a reunião de planejamento de sprint, a reunião diária (daily scrum), a revisão de sprint e a retrospectiva de sprint. Essas cerimônias ajudam a manter todos os membros da equipe informados e envolvidos no processo.

Existem diversas justificativas para usar o Scrum em um projeto. O foco na entrega de valor ajuda a equipe a se concentrar nas entregas que trarão mais valor ao cliente, trabalhando nas funcionalidades mais importantes primeiro. A flexibilidade do Scrum permite que a equipe se adapte rapidamente às mudanças nas condições do projeto ou mudanças de ideia do cliente.

A melhoria contínua é incentivada por meio da retrospectiva de sprint, onde a equipe pode implementar melhorias para a próxima sprint. A transparência em todo o processo de desenvolvimento promovida pelo Scrum ajuda a evitar mal-entendidos e garante que todos estejam trabalhando na mesma direção.

O Scrum é uma abordagem iterativa e incremental que ajuda a equipe a entregar um produto de alta qualidade em um curto espaço de tempo, aumentando a satisfação do cliente e a produtividade da equipe, por este motivo ele é o ideal para o projeto.

2. Estudo de viabilidade

O aplicativo foca em um campo que pode ser mais bem explorado. Embora existam diversos serviços de nutrição com sugestões de dieta e cardápio, assim como aplicativos de estoque doméstico e listas de compras, a maioria não integra ambas as funcionalidades. Ao contrário de aplicativos cuja funcionalidade é centralizada no atendimento de um profissional de nutrição, este projeto foca na interação com o usuário.

As alternativas web ou mobile identificadas realizam busca de receitas contendo os ingredientes selecionados, porém não fazem uso de um inventário pré-cadastrado, necessitando de input recorrente do usuário. Geralmente são aplicativos gratuitos com anúncios, acompanhados de comentários dos usuários sugerindo a possibilidade de uma versão paga sem eles. Portanto, há mercado para um aplicativo com assinatura ou valor de compra.

Em termos técnicos o projeto é viável, pois existe uma miríade de recursos do tipo API que permitem consultar informações de produtos pelo código de barras ou até buscar receitas por meio de inteligências artificiais, como ChatGPT.

2.1 Entendimento sobre a complexidade de desenvolvimento do produto na esfera técnica e econômica

Para entender a complexidade de desenvolvimento do produto na esfera técnica e econômica, foi realizado um levantamento de requisitos visando compreender e identificar as necessidades do usuário final da aplicação

Abaixo seguem as necessidades levantadas:

- Ajudar o usuário a ter uma alimentação adequada à sua necessidade de forma simples e intuitiva, proporcionando facilidade em obter motivação e prazer na utilização da aplicação;
- Sugerir receitas com base nos ingredientes que estão disponíveis em casa e/ou que são desejados, além de entregar lista de compras necessárias;
- Atender e ajudar no cumprimento de todo tipo de demanda nutricional, como por exemplo: emagrecimento, ganho de massa muscular, atletas, dieta saudável, etc);

- Possibilitar interface com aplicações de exercícios de forma a compor um quadro completo do acompanhamento e possibilitar um acompanhamento diário e do histórico;
- Possibilitar interface com um plano nutricional realizado por profissionais e ter versões para acompanhamento e atuação deles;
- Entregar informações nutricionais das receitas e comparar com o plano com cálculo diário;

Para atender as necessidades descritas acima, também foram levantados os requisitos funcionais mostrados logo abaixo:

- Realizar cadastro de usuários;
- Realizar cadastro de profissionais;
- Aplicação em versão Mobile e Web;
- Acessar receitas sugeridas a partir de site de busca;
- Realizar cruzamento de ingredientes selecionados com a pesquisa;
- Obter informações nutricionais de receitas e ingredientes de bancos de dados confiáveis;
- Realizar interface com aplicativos de exercícios para obter dados de exercícios realizados;
- Possibilitar interface em modelo de rede social para conexão com pessoas e grupos.

2.2 Estimativa de esforço para o desenvolvimento da solução

Ao estimar o esforço necessário para desenvolver um aplicativo utilizando a metodologia Scrum, é importante considerar vários fatores. Esses fatores incluem o escopo do projeto, a equipe disponível, as habilidades técnicas dos membros da equipe, a complexidade da tecnologia a ser usada e o cronograma previsto para o projeto.

O aplicativo levará cerca de seis meses para ser desenvolvido, abaixo detalho a estimativa de cada fase:

- 1- Análise e definição dos requisitos: 44 horas;
- 2- Planejamento: 44 horas;
- 3- Implementação das funcionalidades no código fonte: 308 horas;

4- Execução de testes de segurança, funcionalidade e aceitação, além de rastreamento e correção de bugs: 88 horas;

5- Implantação: 44 horas.

2.3 Apresentação de soluções alternativas que atendam à expectativa do demandante

Será utilizada a tabela Brasileira de Composição de Alimentos disponível (TBCA – disponível em: www.tbca.net.br) como base de dados para a obtenção dos dados nutricionais relevantes dos ingredientes disponíveis ou inseridos.

Para obter as receitas será utilizado um mecanismo de busca do Google.

Será utilizada a API de um aplicativo de exercícios (exemplo: Strava) com o objetivo de obter dados atualizados dos exercícios realizados.

3 Especificação de requisitos do produto

Neste tópico, serão apresentados os requisitos do software que estamos desenvolvendo e as técnicas que utilizamos para obtê-los. Os requisitos foram classificados em funcionais e não-funcionais, e descreveremos todo o processo de obtenção deles.

3.1 - Descrição sobre o processo de licitação / extração requisitos.

Foi realizado um estudo para identificar o público-alvo do nosso projeto e coletar os requisitos necessários. Para isso, foram conduzidas entrevistas com diferentes grupos, incluindo donas de casa que buscam praticidade no dia a dia, nutricionistas que desejam auxiliar seus pacientes na organização alimentar e pessoas que têm dificuldades no planejamento de uma dieta equilibrada. Os dados obtidos foram analisados e os resultados organizados em uma lista completa de requisitos funcionais e não funcionais, descrita no tópico abaixo.

3.2 - Análise do resultado do processo de extração e elaboração das regras de negócio e requisitos

Os requisitos que foram obtidos com as entrevistas realizadas são:

Capacidade de visualizar imagens - O aplicativo deve ser capaz de carregar imagens para serem processadas;

- Processamento de imagem - O aplicativo deve ser capaz de analisar as imagens e identificar os itens presentes;
- Reconhecimento de texto - O aplicativo deve ser capaz de reconhecer qualquer texto presente nas imagens;
- Conversão em lista - O aplicativo deve ser capaz de transformar as informações identificadas em uma lista de itens;
- Interface do usuário - O aplicativo deve ter uma interface fácil de usar para que os usuários possam carregar imagens e visualizar as listas de itens resultantes;
- Armazenamento de dados - O aplicativo deve ser capaz de armazenar as listas de itens para uso futuro ou compartilhamento com outros usuários;

- Compatibilidade com diferentes formatos de imagem - O aplicativo deve ser capaz de trabalhar com diferentes formatos de imagem, como JPEG, PNG, GIF, entre outros;
- Suporte a múltiplos idiomas - O aplicativo deve ser capaz de reconhecer texto em diferentes idiomas e converter em uma lista de itens;
- Capacidade de atualização - O aplicativo deve ser capaz de ser atualizado com novas funcionalidades e correções de bugs;
- Cadastro de usuário: o aplicativo deve permitir que o usuário crie uma conta, faça login e mantenha um perfil pessoal;
- Lista de ingredientes: o aplicativo deve permitir que o usuário adicione ingredientes que já possui em sua cozinha à sua lista de ingredientes pessoais;
- Busca de receitas: o aplicativo deve permitir que o usuário pesquise receitas com base em palavras-chave ou categoria de culinária;
- Sugerir receitas: com base nos ingredientes disponíveis do usuário, o aplicativo deve ser capaz de sugerir receitas que possam ser feitas com os ingredientes disponíveis;
- Lista de compras: o aplicativo deve gerar uma lista de compras de ingredientes necessários para a receita selecionada, com base nos ingredientes que o usuário já possui;
- Personalização da lista de compras: o usuário deve ter a opção de personalizar a lista de compras, adicionando ou removendo itens, ou alterando a quantidade de ingredientes.

O software a ser desenvolvido deve ter capacidade de processar imagens e identificar os itens presentes, reconhecer textos, transformar informações identificadas em uma lista de itens, ter uma interface fácil de usar, armazenar dados e ser compatível com diferentes formatos de imagem e suporte a múltiplos idiomas.

Além disso, deve permitir o cadastro de usuários, criação de perfil pessoal, adição de ingredientes pessoais à lista de ingredientes, busca de receitas com base em palavras-chave ou categoria de culinária, sugestão de receitas com base nos ingredientes disponíveis, geração de lista de compras com base nos ingredientes necessários para a receita selecionada e personalização da lista de compras.

O software deve ser capaz de ser atualizado com novas funcionalidades e correções de bugs.

3.3 Classificação dos Requisitos

Requisitos não funcionais:

Desempenho: o aplicativo deve processar as imagens de forma rápida e eficiente, a fim de fornecer uma experiência de usuário satisfatória. Isso inclui o tempo de carregamento das imagens, o tempo de processamento e a capacidade de resposta do aplicativo;

Confiabilidade: o aplicativo deve ser confiável e não deve falhar ou perder dados importantes. Isso inclui a capacidade de lidar com uma grande variedade de formatos de imagem, tamanhos e resoluções;

Escalabilidade: o aplicativo deve ser capaz de lidar com um grande volume de imagens e usuários simultâneos. Isso significa que ele deve ser projetado para suportar um grande número de solicitações de usuários e ter uma boa escalabilidade horizontal;

Segurança: o aplicativo deve garantir a segurança e privacidade dos dados dos usuários e das imagens. Isso inclui a autenticação do usuário, criptografia de dados e acesso restrito aos recursos do sistema;

Usabilidade: o aplicativo deve ser fácil de usar e intuitivo, com uma interface de usuário clara e simples. Ele também deve ser acessível para pessoas com deficiências visuais ou motoras;

Manutenibilidade: o aplicativo deve ser fácil de manter e atualizar, com um código bem estruturado e documentado. Isso facilita a identificação e correção de problemas e a adição de novas funcionalidades;

Performance: o aplicativo deve ser rápido e responsivo, com tempos de resposta mínimos para a pesquisa de receitas e geração de listas de compras;

Disponibilidade: o aplicativo deve estar disponível 24 horas por dia, 7 dias por semana, para garantir a conveniência do usuário.

Requisitos funcionais:

Capacidade de visualizar imagens - O aplicativo deve ser capaz de carregar imagens para serem processadas;

Processamento de imagem - O aplicativo deve ser capaz de analisar as imagens e identificar os itens presentes;

Reconhecimento de texto - O aplicativo deve ser capaz de reconhecer qualquer texto presente nas imagens;

Conversão em lista - O aplicativo deve ser capaz de transformar as informações identificadas em uma lista de itens;

Interface do usuário - O aplicativo deve ter uma interface fácil de usar para que os usuários possam carregar imagens e visualizar as listas de itens resultantes;

Armazenamento de dados - O aplicativo deve ser capaz de armazenar as listas de itens para uso futuro ou compartilhamento com outros usuários;

Compatibilidade com diferentes formatos de imagem - O aplicativo deve ser capaz de trabalhar com diferentes formatos de imagem, como JPEG, PNG, GIF, entre outros;

Suporte a múltiplos idiomas - O aplicativo deve ser capaz de reconhecer texto em diferentes idiomas e converter em uma lista de itens;

Capacidade de atualização - O aplicativo deve ser capaz de ser atualizado com novas funcionalidades e correções de bugs;

Cadastro de usuário: o aplicativo deve permitir que o usuário crie uma conta, faça login e mantenha um perfil pessoal;

Lista de ingredientes: o aplicativo deve permitir que o usuário adicione ingredientes que já possui em sua cozinha à sua lista de ingredientes pessoais;

Busca de receitas: o aplicativo deve permitir que o usuário pesquise receitas com base em palavras-chave ou categoria de culinária;

Sugerir receitas: com base nos ingredientes disponíveis do usuário, o aplicativo deve ser capaz de sugerir receitas que possam ser feitas com os ingredientes disponíveis;

Lista de compras: o aplicativo deve gerar uma lista de compras de ingredientes necessários para a receita selecionada, com base nos ingredientes que o usuário já possui;

Personalização da lista de compras: o usuário deve ter a opção de personalizar a lista de compras, adicionando ou removendo itens, ou alterando a quantidade de ingredientes.

4 Modelo de Dados

Neste tópico serão apresentadas às entidades que puderam ser identificadas ao longo do trabalho bem como as relações entre elas, além de uma demonstração gráfica por meio do Diagrama Entidade-Relacionamento(DER).

O DER inclui as seguintes entidades: Usuário, Ingredientes, Receitas, Local de Compra, Lista de Compras. Cada uma dessas entidades possui seus próprios atributos, tais como: id, nome, e-mail, senha, entre outros. Além disso, as entidades estão interligadas por meio de relacionamentos, como um usuário pode escanear vários ingredientes, um ingrediente pode ser utilizado em várias receitas e um ingrediente pode ser obtido em um lugar a escolha do usuário.

Através desse diagrama, pode-se compreender a estrutura do banco de dados e as relações entre as entidades de forma visual e intuitiva. Assim, é possível identificar quais informações são necessárias para o desenvolvimento da aplicação e como elas se relacionam entre si, possibilitando uma melhor organização e planejamento do projeto.

4.1 Levantamento do Grupo de Dados e afinidades entre eles

Aplicação descrita neste projeto contará com os seguintes grupos de dados:

Usuário - A entidade usuário será utilizada para cadastrar a pessoa que estará acessando o aplicativo. Serão coletadas as seguintes informações: Nome, Email, Localização e Senha.

Ingredientes - A entidade Ingredientes é utilizada para armazenar informações sobre os ingredientes disponíveis na aplicação. Cada Ingrediente possui um ID único (chave primária), um Nome, a Quantidade disponível, a Unidade de medida, as Calorias, Proteínas, Carboidratos e Gorduras presentes no alimento.

Esses atributos são importantes para o cálculo de valores nutricionais de receitas e controle de estoque de alimentos.

Receitas - A entidade Receita representa uma receita que pode ser criada pelos usuários. Ela contém um ID que a identifica, um Nome para identificação da receita, uma Descrição sobre a receita, um Modo de Preparo que contém os passos para a criação da receita, o Número de Porções que essa receita pode gerar e as Informações Nutricionais que a receita contém.

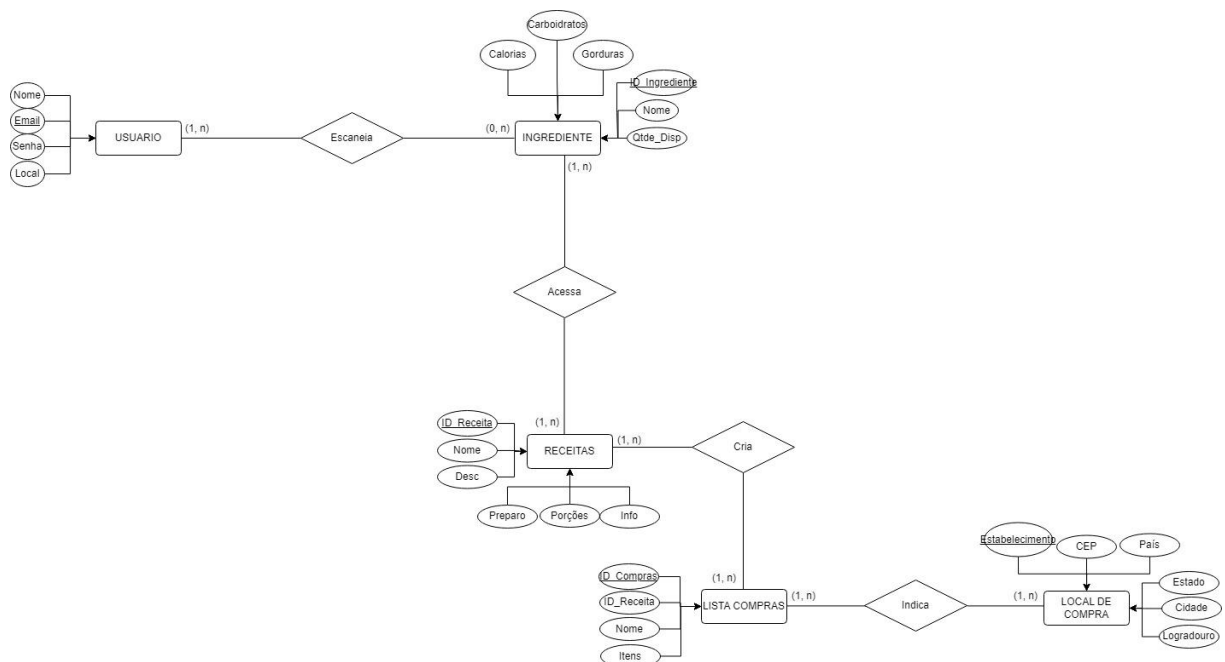
Lista de Compras - A tabela Lista de Compras armazena a lista de compras de cada usuário. Ela possui uma chave primária (ID) e uma chave estrangeira (ID_Receita) que faz referência a receita selecionada pelo usuário. Cada lista pode ser identificada por um nome.

Local de Compra - O local de compra armazenará os locais de compra(supermercados/comércios e etc.) que se encontram nas proximidades da localização do usuário. Contendo o país, estado, cidade e logradouro, bem como o CEP do estabelecimento. Tais dados deverão ser obtidos através de uma API de geolocalização.

4.2 Elaboração do Diagrama Entidade Relacionamento(DER)

Segue abaixo o DER da aplicação apresentada:

Figura 1 - Diagrama Entidade Relacionamento

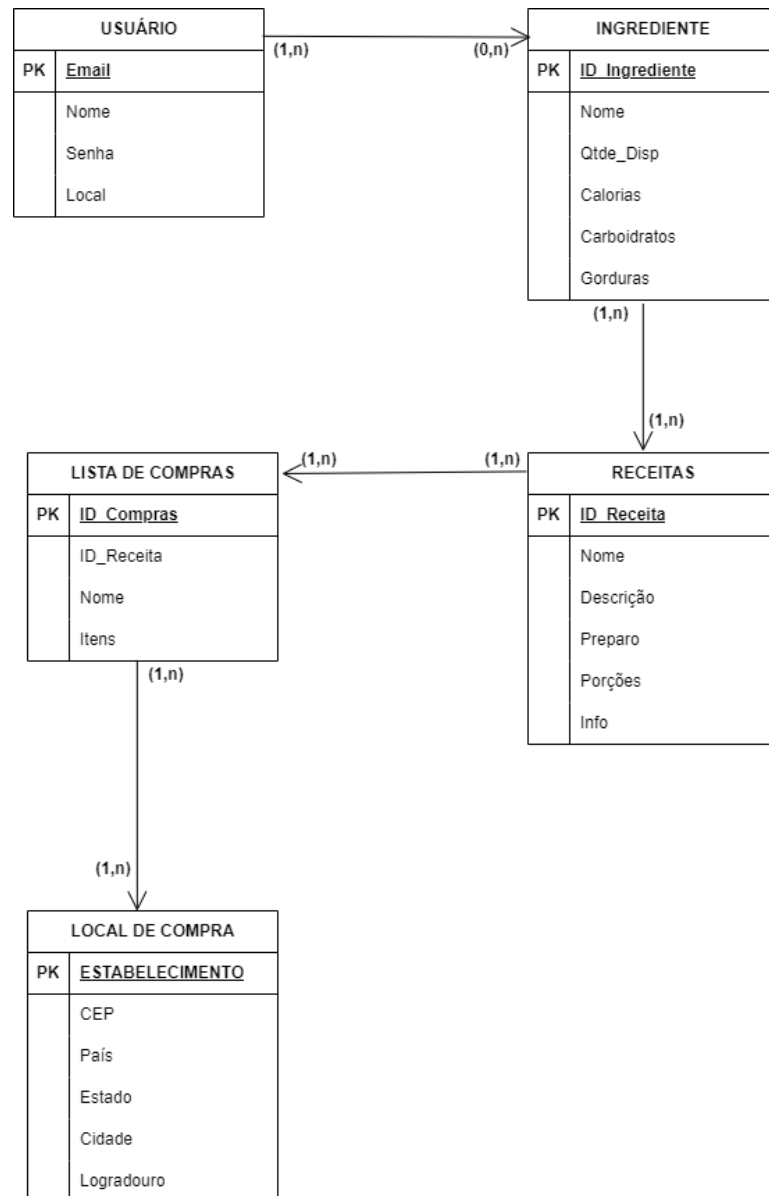


Fonte: Autoral, 2023.

4.3 Modelo Lógico e Normalização

O modelo lógico considera as criações do modelo conceitual, traduzindo-as através do DER e levando em conta restrições, normalização, e a definição de chaves primárias e estrangeiras, juntamente com os atributos associados a cada entidade, conforme o diagrama abaixo:

Figura 2 - Modelo Lógico Pré Normalização



Fonte: Autoral, 2023.

As três formas normais (1NF, 2NF e 3NF) são princípios de design que garantem a integridade e eficiência de um banco de dados relacional. Elas são utilizadas para estruturar as tabelas e as relações entre elas, evitando duplicidade de dados e problemas de atualização.

A 1ª Forma Normal (1NF) estabelece que cada coluna em uma tabela deve conter apenas um valor, sem múltiplos valores ou listas de valores. Além disso, cada tabela deve ter uma chave primária única para identificar de forma exclusiva cada registro.

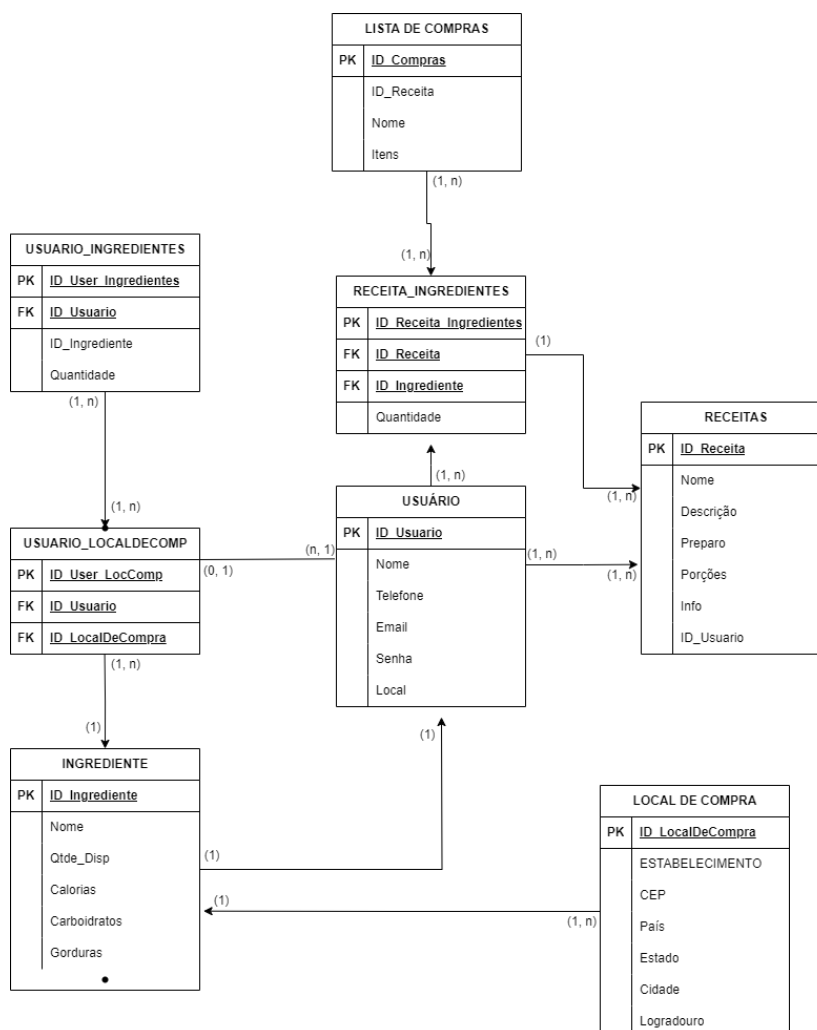
A 2ª Forma Normal (2NF) lida com a dependência funcional parcial em uma tabela. Ela requer que uma tabela esteja na 1NF e, adicionalmente, todos os atributos não-chave dependem completamente da chave primária. Caso haja dependência parcial, é recomendado separar esses atributos em uma nova tabela relacionada.

A 3ª Forma Normal (3NF) tem como objetivo eliminar a dependência transitiva entre os atributos de uma tabela. Ela exige que uma tabela esteja na 2NF e, além disso, que os atributos não-chave não dependam uns dos outros. Em outras palavras, todos os atributos não-chave devem depender diretamente da chave primária, evitando dependências indiretas.

A aplicação das três formas normais é essencial para evitar redundâncias e inconsistências nos dados, garantindo um design robusto para o banco de dados. Ao seguir esses princípios, é possível melhorar a eficiência das operações de consulta e atualização, além de facilitar a manutenção e a escalabilidade do banco de dados ao longo do tempo.

Segue abaixo o Modelo Lógico do respectivo projeto dentro das 3 formas normais:

Figura 3 - Modelo Lógico Pós Normalização



Fonte: Autoral, 2023.

Inicialmente a terceira forma normal não encontrava-se satisfeita, tendo em vista que a tabela Receitas_Ingredientes tinha uma dependência transitiva entre os atributos ID_Receita e ID_Ingrediente. Para que houvesse a normalização foi necessário que se dividisse a tabela Receitas_Ingredientes em duas outras tabelas: Receitas e Ingredientes_Receitas.

A tabela Receitas terá como chave primária o atributo ID. A tabela Ingredientes_Receitas terá como chave primária o par de atributos ID_Receita e ID_Ingrediente.

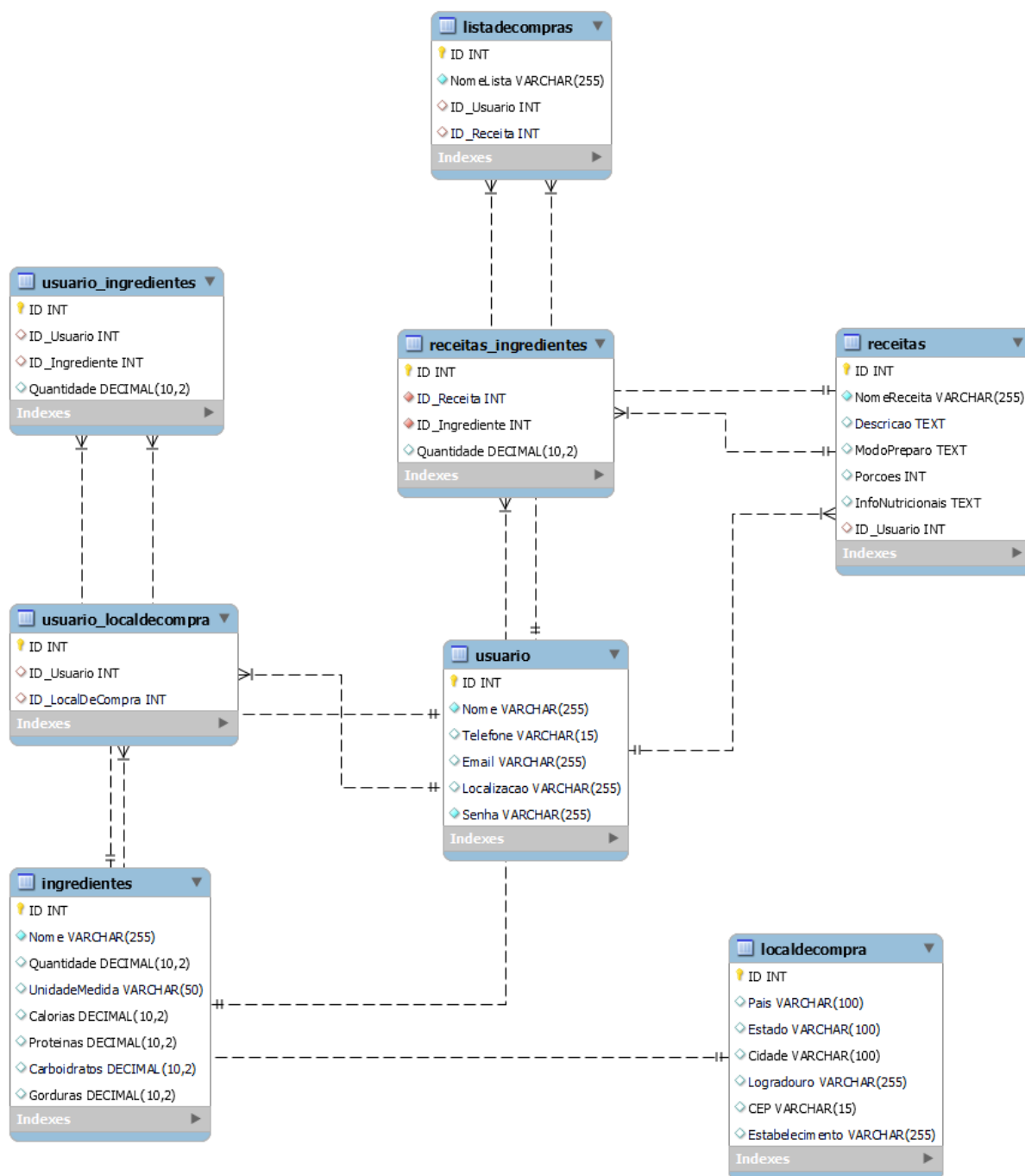
Com essa alteração, a tabela Receitas terá dependências funcionais apenas entre os atributos ID e NomeReceita, e a tabela Ingredientes_Receitas terá

dependências funcionais apenas entre os atributos ID_Receita, ID_Ingrediente e Quantidade.

Vale ressaltar que não foi detectada necessidade de refinar a aplicação no momento, tendo em vista que por enquanto ela cumpre com tudo que foi proposto em seu desenvolvimento inicial.

4. 4 Modelo Físico

Figura 4 - Modelo Físico



Fonte: Autoral, 2023.

5 Linguagem e SGBD utilizados

Optou-se por empregar a linguagem SQL como a plataforma para a elaboração do banco de dados em questão, sendo o Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido o MySQL.

5.1 Código Fonte

```
CREATE DATABASE IF NOT EXISTS App_Alimentacao;
```

```
USE App_Alimentacao;
```

-- Foi utilizado o AUTO_INCREMENT para que não haja necessidade de inserção manual do ID. Também foi utilizado o NOT NULL em informações que não podem ser nulas.

```
CREATE TABLE IF NOT EXISTS Usuario (  
    ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(255) NOT NULL,  
    Telefone VARCHAR(15),  
    Email VARCHAR(255),  
    Localizacao VARCHAR(255),  
    Senha VARCHAR(255) NOT NULL  
);
```

```
CREATE TABLE IF NOT EXISTS Ingredientes (  
    ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    Nome VARCHAR(255) NOT NULL,  
    Quantidade DECIMAL(10, 2),  
    UnidadeMedida VARCHAR(50),  
    Calorias DECIMAL(10, 2),  
    Proteinas DECIMAL(10, 2),  
    Carboidratos DECIMAL(10, 2),  
    Gorduras DECIMAL(10, 2)  
);
```

```
CREATE TABLE IF NOT EXISTS LocalDeCompra (  
    ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    Pais VARCHAR(100),
```



```

Estado VARCHAR(100),
Cidade VARCHAR(100),
Logradouro VARCHAR(255),
CEP VARCHAR(15)
);

```

```

CREATE TABLE IF NOT EXISTS Receitas (
  ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  NomeReceita VARCHAR(255) NOT NULL,
  Descricao TEXT,
  ModoPreparo TEXT,
  Porcoes INT,
  InfoNutricionais TEXT,
  ID_Usuario INT,
  FOREIGN KEY (ID_Usuario) REFERENCES Usuario(ID)
);

```

```

CREATE TABLE IF NOT EXISTS ListaDeCompras (
  ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  NomeLista VARCHAR(255) NOT NULL,
  ID_Usuario INT,
  ID_Receita INT,
  FOREIGN KEY (ID_Usuario) REFERENCES Usuario(ID),
  FOREIGN KEY (ID_Receita) REFERENCES Receitas(ID)
);

```

-- Tabela Usuario_Ingredientes para relacionar os ingredientes que cada usuário possui

```

CREATE TABLE IF NOT EXISTS Usuario_Ingredientes (
  ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
  ID_Usuario INT,
  ID_Ingrediente INT,
  Quantidade DECIMAL(10, 2),
  FOREIGN KEY (ID_Usuario) REFERENCES Usuario(ID),
  FOREIGN KEY (ID_Ingrediente) REFERENCES Ingredientes(ID)
);

```

-- Tabela Usuario_Ingredientes para relacionar os ingredientes que cada usuário possui

```
CREATE TABLE IF NOT EXISTS Receitas_Ingredientes (  
    ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    ID_Receita INT NOT NULL,  
    ID_Ingrediente INT NOT NULL,  
    Quantidade DECIMAL(10, 2),  
    FOREIGN KEY (ID_Receita) REFERENCES Receitas(ID),  
    FOREIGN KEY (ID_Ingrediente) REFERENCES Ingredientes(ID)  
);
```

-- Tabela Usuario_LocalDeCompra para relacionar os locais de compra que cada usuário utiliza

```
CREATE TABLE IF NOT EXISTS Usuario_LocalDeCompra (  
    ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    ID_Usuario INT,  
    ID_LocalDeCompra INT,  
    FOREIGN KEY (ID_Usuario) REFERENCES Usuario(ID),  
    FOREIGN KEY (ID_LocalDeCompra) REFERENCES LocalDeCompra(ID)  
);
```

-- Seleção de dados

```
SELECT * FROM usuario;  
SELECT * FROM receitas;  
SELECT * FROM localdecompra;  
SELECT * FROM listadecompras;  
SELECT * FROM ingredientes;
```

-- Pesquisa com base em palavras-chave,

```
SELECT * FROM receitas WHERE NomeReceita LIKE '%Palavra-Chave%';  
SELECT * FROM ingredientes WHERE Nome LIKE '%Palavra-Chave%';
```

-- Sugerir receitas com base nos ingredientes disponíveis do usuário

```
SELECT r.*  
FROM Receitas r  
INNER JOIN Receitas_Ingredientes ir ON r.id = ir.id_receita
```

```
LEFT JOIN Usuario_Ingredientes iu ON ir.id_ingrediente = iu.id_ingrediente AND
iu.id_usuario = 1
GROUP BY r.id
HAVING COUNT(DISTINCT ir.id_ingrediente) = COUNT(DISTINCT
iu.id_ingrediente);
```

-- Atualizar o nome de um usuário

```
UPDATE Usuario
```

```
SET nome = 'Novo Nome'
```

```
WHERE id = 1;
```

-- Atualizar a quantidade de um ingrediente

```
UPDATE Ingredientes
```

```
SET quantidade = 150
```

```
WHERE id = 1;
```

-- Atualizar o nome de uma receita

```
UPDATE Receitas
```

```
SET nome = 'Nova Receita'
```

```
WHERE id = 1;
```

6 Conclusão

De acordo com o apresentado, o software de organização de compras e planejamento alimentar apresenta uma solução inovadora e prática para auxiliar as pessoas em suas rotinas diárias. Com a utilização do scanner de QR code, o processo de listagem de ingredientes se torna mais rápido e eficiente, além de fornecer valores nutricionais e sugestões de receitas com base nos produtos disponíveis. A metodologia Scrum utilizada no desenvolvimento permite uma entrega mais rápida do produto e aprimoramento contínuo, garantindo a satisfação dos usuários. Com a capacidade de personalização da lista de compras e suporte a diferentes idiomas, o software pode atender a um grande número de usuários em todo o mundo. Combinando funcionalidades práticas e inovadoras com requisitos de desempenho, confiabilidade, escalabilidade e segurança, o software oferece uma solução completa e eficiente para ajudar as pessoas em sua alimentação diária.

REFERÊNCIAS

ELMASRI, R.; NAVATHE, S. B. **Sistemas de banco de dados**. 7 ed. São Paulo: Pearson, 2018.

MACHADO, F. N. R. **Banco de Dados –Projeto e Implementação**. São Paulo, Saraiva, 3ª Ed., 2014.

LEAL, G. C. L.; **Linguagem, programação e banco de dados: guia prático de aprendizagem**. Curitiba, InterSaberes, 2015. <Pearson>

SETZER, V. W.; SILVA, F. S. C. da. **Banco de dados: aprenda o que são, melhore seu conhecimento, construa os seus**. São Paulo: Edgard Blucher, 2005.