

# Fatec

Ipiranga

**FACULDADE DE TECNOLOGIA DO IPIRANGA  
CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

Felipe Navas Nascimento  
Gabriel Dias Santana  
Henrique Castilho Pires De Carvalho  
Leonardo Vivo Guerreiro  
Rafael Brito de Campos

**Projeto IHC/UX**

SÃO PAULO  
ANO 2025

## SUMÁRIO

1 REQUISITOS DO SISTEMA DE SOFTWARE.....	12
1.1.1 <i>Requisitos Funcionais</i> .....	12
1.1.2 <i>Requisitos Não-Funcionais</i> .....	14
1.1.3 <i>Regras de Negócio</i> .....	16
1.2.1 <i>Atores</i> .....	17
1.2.2 <i>Diagrama de Caso de Uso</i> .....	18
1.2.3 <i>Especificação do Caso de Uso</i> .....	19
1.3.1 <i>Diagrama “É, Não É, Faz, Não Faz”</i> .....	26
1.3.2 <i>Diagrama 360</i> .....	27
1.4.1 <i>Heurísticas:</i> .....	28
1.5.1 <i>Leis da UX</i> .....	29
1.6.1 <i>Gestalt:</i> .....	30
1.7.1 <i>Fluxo de Telas - Casos de Usos</i> .....	31
1.8.1 <i>Design System</i> .....	32
1.9.1 <i>UX Canvas</i> .....	34
1.10.1 <i>Personas</i> .....	36

## 1 REQUISITOS DO SISTEMA DE SOFTWARE

### 1.1.1 Requisitos Funcionais

Neste item devem ser descritos os requisitos funcionais que especificam ações que um sistema deve ser capaz de executar, ou seja, os objetivos do sistema, incluindo prioridade e regras de negócio. A seguir são apresentados exemplos.

#### [RF001] – Manter Usuário

**Prioridade:**     ☒ Essencial            ☐ Importante            ☐ Desejável

**Descrição:** Este requisito permite que o usuário se registre, acesse o sistema e edite seus dados pessoais.

#### [RF002] – Definir Tempo Disponível

**Prioridade:**     ☒ Essencial            ☐ Importante            ☐ Desejável

**Descrição:** Este requisito permite que o usuário informe quanto tempo tem para a refeição e esse tempo impactar nas sugestões.

#### [RF003] – Definir Preferências Alimentares

**Prioridade:**     ☒ Essencial            ☐ Importante            ☐ Desejável

**Descrição:** Este requisito permite que o usuário configure gostos e restrições alimentares.

#### [RF004] – Visualizar Recomendações com Base no Histórico

**Prioridade:**     ☒ Essencial            ☐ Importante            ☐ Desejável

**Descrição:** Este requisito permite que o sistema mostre sugestões baseadas em interações anteriores, utilizando IA.

#### **[RF005] – Filtrar Sugestões**

**Prioridade:**     ☒ Essencial            ☐ Importante            ☐ Desejável

**Descrição:** Este requisito permite que o usuário aplique filtros manuais como tipo de comida, preço e distância do local.

#### **[RF006] – Visualizar Rota até o Restaurante**

**Prioridade:**     ☒ Essencial            ☐ Importante            ☐ Desejável

**Descrição:** Este requisito permite que o sistema mostre o melhor caminho até o local sugerido, utilizando a API do Google Maps.

#### **[RF007] – Pesquisar Restaurantes para Mim**

**Prioridade:**     ☒ Essencial            ☐ Importante            ☐ Desejável

**Descrição:** Este requisito permite que o sistema localize restaurantes com base em filtros, tempo e localização.

**[RF008] – Salvar Refeições Favoritas**

**Prioridade:**      ☒ Essencial              ☐ Importante              ☐ Desejável

**Descrição:** Este requisito permite que o usuário marque refeições como favoritas para recomendações futuras.

**[RF009] – Avaliar Refeições**

**Prioridade:**      ☒ Essencial              ☐ Importante              ☐ Desejável

**Descrição:** Este requisito permite que o usuário avalie refeições, melhorando o sistema de sugestões.

**[RF010] – Receber Sugestões de IA**

**Prioridade:**      ☒ Essencial              ☐ Importante              ☐ Desejável

**Descrição:** Este requisito permite que o sistema recomende automaticamente refeições ideais baseadas no perfil do usuário.

**[RF011] – Salvar Locais Favoritos**

**Prioridade:**      ☒ Essencial              ☐ Importante              ☐ Desejável

**Descrição:** Este requisito permite que o usuário possa salvar restaurantes preferidos.

**1.1.2 Requisitos Não-Funcionais**

Neste item devem ser apresentados os requisitos não funcionais, que especificam restrições sobre os serviços ou funções providas pelo sistema. A seguir são apresentados alguns exemplos de requisitos não-funcionais.

#### [RNF001] – Usabilidade

**Prioridade:**     ☒ Essencial            ☐ Importante            ☐ Desejável

**Descrição:** O aplicativo deve oferecer uma interface amigável e de fácil navegação, permitindo que usuários de diferentes perfis (inclusive com pouca familiaridade com tecnologia) consigam realizar suas ações com rapidez.

#### [RNF002] – Apresentação da Interface Gráfica

**Prioridade:**     ☐ Essencial            ☒ Importante            ☐ Desejável

**Descrição:** O sistema deve apresentar uma interface gráfica moderna, responsiva e adaptada para dispositivos móveis Android, utilizando componentes visuais claros e acessíveis. A interface deve seguir padrões de design, como ícones autoexplicativos, contraste adequado e facilidade de leitura em diferentes condições de iluminação.

#### [RNF003] – Ajuda Online

**Prioridade:**     ☐ Essencial            ☒ Importante            ☐ Desejável

**Descrição:** O aplicativo deve fornecer uma seção de ajuda integrada, com instruções básicas sobre o uso das funcionalidades principais (ex: como definir tempo, visualizar recomendações, salvar favoritos), além de mensagens informativas para orientar o usuário durante a navegação.

### **1.1.3 Regras de Negócio**

#### **[RN001] – Autenticação Obrigatória**

##### **Descrição:**

O sistema só permitirá o acesso às funcionalidades principais caso o usuário esteja devidamente autenticado através do login.

#### **[RN002] – Validação de Tempo Disponível**

##### **Descrição:**

O tempo disponível informado pelo usuário deve ser um número inteiro positivo (maior que zero). O sistema rejeitará valores inválidos, como negativos, zero ou caracteres não numéricos.

#### **[RN003] – Atualização de Preferências Alimentares**

##### **Descrição:**

O usuário poderá adicionar, alterar ou remover suas preferências alimentares a qualquer momento por meio do perfil. Essas alterações impactarão diretamente nas sugestões futuras geradas pelo sistema.

#### **[RN004] – Sugestões com Filtros Obrigatórios**

##### **Descrição:**

O sistema deverá considerar tempo disponível, localização atual e preferências alimentares como filtros obrigatórios. Sugestões que não respeitem essas condições deverão ser descartadas.

#### **[RN005] – Localização como Fator de Prioridade**

##### **Descrição:**

As sugestões exibidas devem considerar a localização atual do usuário como fator de prioridade, apresentando estabelecimentos próximos e compatíveis com o tempo disponível.

#### **[RN006] – Geração de Rota com Base no GPS**

##### **Descrição:**

O sistema utilizará os dados de GPS do dispositivo para gerar a rota até o restaurante selecionado. Caso a localização não seja obtida, o sistema notificará o usuário sobre a necessidade de conceder permissão.

**[RN007] – Validação de Disponibilidade do Restaurante****Descrição:**

Antes de exibir uma sugestão, o sistema verificará se o restaurante está aberto no momento. Sugestões de restaurantes fechados serão descartadas.

**[RN008] – Personalização Contínua das Sugestões****Descrição:**

As sugestões futuras devem se tornar cada vez mais personalizadas, considerando o histórico de refeições avaliadas, locais frequentados e preferências previamente salvas pelo usuário.

**[RN009] – Retentativa em Caso de Falha de Conexão****Descrição:**

Caso ocorra falha técnica ao salvar informações ou consultar sugestões, o sistema exibirá uma mensagem de erro e permitirá nova tentativa de operação.

**[RN010] – Recomendação com Base no Histórico****Descrição:**

O sistema utilizará o histórico de refeições avaliadas, locais frequentados e preferências salvas para refinar continuamente as sugestões, aplicando técnicas de recomendação inteligente.

**[RN011] – Restrições Alimentares como Critério de Exclusão****Descrição:**

O sistema não recomendará refeições ou locais que contenham ingredientes incompatíveis com as restrições alimentares cadastradas pelo usuário, como lactose, glúten ou carne.

Neste item devem ser descritos os requisitos a serem atendidos funcionalmente pelo sistema de uma forma simples, possibilitando a compreensão do comportamento do sistema pela perspectiva do usuário. Devem ser descritos os atores e o diagrama de caso de uso. A seguir um exemplo de especificação de atores, do diagrama de caso de uso e da especificação de caso de uso.

**1.2.1 Atores**

A seguir são apresentados exemplos da especificação de atores.



**USUÁRIO:**

Representa a pessoa que utiliza o aplicativo para buscar sugestões de refeições personalizadas. Pode informar suas preferências alimentares, tempo disponível, localização atual e interagir com as recomendações recebidas (salvar, avaliar, filtrar e visualizar rotas).

**API GOOGLE MAPS:**

Serviço externo responsável por fornecer dados de geolocalização, rotas e localização de restaurantes próximos, utilizados pelo sistema para apresentar opções viáveis com base na posição atual do usuário.

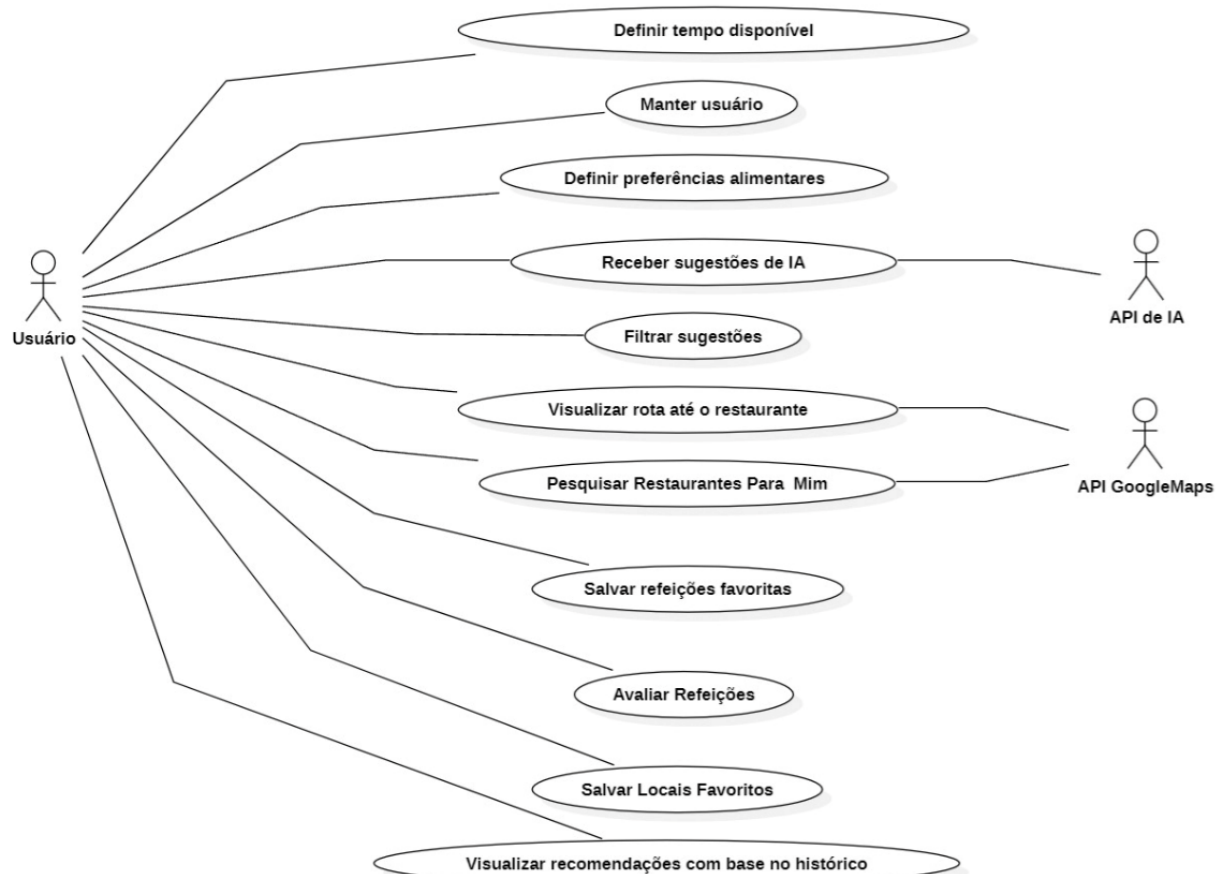
**API DE INTELIGÊNCIA ARTIFICIAL:**

Serviço externo utilizado pelo sistema para gerar sugestões personalizadas com base no histórico do usuário, restrições alimentares, tempo disponível e avaliações anteriores. Também pode processar linguagem natural em comandos por voz ou texto, quando implementado.

## 1.2.2 Diagrama de Caso de Uso

Na Figura 1 é apresentada a notação básica de um diagrama de caso de uso.

Figura 1 - Notação básica do diagrama de caso de uso.



Fonte: Autores (2025)

### 1.2.3 Especificação do Caso de Uso

No Quadro 1 é apresentado um exemplo da especificação de casos de uso

<b>CSU001 – Manter Usuario</b>	
<u>Sumário:</u>	Permitir que o usuário acesse o aplicativo Meal4You informando suas credenciais (e-mail e senha).
Ator Primário:	Usuário
Ator Secundário: Sistema de Autenticação	
Casos de Uso Associados: Nenhum.	
<b>Pré-condição:</b> a. O aplicativo deve estar instalado e funcionando no dispositivo do usuário. b. O usuário deve ter um cadastro prévio com e-mail e senha válidos.	
<b>Fluxo Principal:</b> 1- O Usuário abre o aplicativo Meal4You. 2- O Sistema exibe a tela de login com os campos de e-mail e senha e os botões de login social (Google, Facebook, Microsoft, Apple). 3- O Usuário escolhe entre: a. Preencher os campos com suas credenciais e clicar em "Entrar"; ou b. Clicar em um dos botões de Social Login. 4- O Sistema verifica os dados de login: a. Se for login tradicional, valida o e-mail e a senha. b. Se for login social, redireciona o usuário para o provedor escolhido para autenticação. 5- Se as credenciais forem válidas ou a autenticação social for bem-sucedida, o Sistema autentica o usuário e o redireciona para a tela principal do aplicativo.	
<b>Fluxo Alternativo (3): Recuperar Senha:</b> a. Caso o usuário não lembre a senha, pode clicar no link "Esqueci minha senha". b. O Sistema exibe uma tela para o usuário informar o e-mail cadastrado. c. O Sistema envia um link de recuperação de senha para o e-mail informado. d. O Usuário segue as instruções do e-mail para redefinir a senha. e. Após redefinir, o caso de uso retorna ao passo 2 do fluxo principal.	
<b>Fluxo Alternativo (5): Dados inválidos:</b> a. Se o usuário informar e-mail ou senha incorretos, o sistema exibe uma mensagem de erro: "E-mail ou senha inválidos. Por favor, tente novamente." b. O caso de uso retorna ao passo 2.	
<b>Fluxo de Exceção (4): Problemas de conexão:</b> a. Se houver falha na conexão com a internet, o sistema exibe uma mensagem: "Erro de conexão. Verifique sua internet e tente novamente." b. O caso de uso retorna ao passo 2.	
<b>Pós-condições:</b>	

a. O Usuário estará autenticado e poderá acessar os recursos do aplicativo.
b. O Sistema registrará o horário de login para fins de segurança.
<b>Requisitos: RF001</b>
<b>Regras de Negócio: RN001</b>
<b>Interface: I001</b>

**Figura 2 e 3 - Protótipo de tela I001 – Manter Usuário**

**MEAL4YOU**  
comida consciente

**LOGIN**

E-mail

Senha

**ENTRAR**

Esqueceu a senha? [Redefinir senha](#)

Ainda não possui uma conta? [Cadastrar](#)

OU

**MEAL4YOU**  
comida consciente

**CADASTRO**

E-mail

Senha

Confirmar a senha

**CADASTRAR**

Já possui uma conta? [Login](#)

OU

Fonte: Autores (2025)

<b>CSU002 – Definir Tempo Disponível</b>	
<u>Sumário:</u>	Permitir que o usuário informe quanto tempo disponível possui para realizar sua refeição.
Ator Primário:	Usuário
Ator Secundário: Nenhum.	
Casos de Uso Associados: CSU001 - Login.	
<b>Pré-condição:</b>	
a. O Usuário deve estar autenticado no sistema.	
<b>Fluxo Principal</b>	
1- O Usuário acessa a opção "Definir Tempo Disponível" no menu do aplicativo.	
2- O Sistema exibe uma tela com opções de tempo (exemplo: 15 min, 30 min, 1 hora).	
3- O Usuário seleciona o tempo desejado.	
4- O Usuário confirma clicando em "Salvar" ou "Próximo".	
5- O Sistema registra a informação e exibe uma mensagem de confirmação.	
<b>Fluxo Alternativo (2): Entrada manual de tempo personalizado:</b>	
a. O Sistema oferece um campo opcional para o usuário digitar manualmente outro valor de tempo (em minutos).	
b. O Sistema valida se o valor é numérico e maior que zero.	
c. Se válido, salva o tempo informado.	
d. Caso contrário, exibe mensagem de erro.	
<b>Fluxo de Exceção (4): Falha ao salvar tempo:</b>	
a. Se ocorrer algum problema técnico na gravação, o Sistema exibe a mensagem: "Erro ao salvar o tempo disponível. Tente novamente."	
b. O caso de uso retorna ao passo 2.	
<b>Pós-condições:</b>	
a. O tempo disponível ficará salvo no perfil do usuário para ser usado nos próximos filtros.	
<b>Requisitos: RF002</b>	
<b>Regras de Negócio: RN003.</b>	
<b>Interface: I002</b>	

Figura 4 - Protótipo de tela I002 – Definir Tempo Disponível

×

Definir Tempo

Selecionar Tempo

- 15 min

15 min

20 min

25 min

30 min

35 min

40 min

45 min

50 min

55 min

1 hora

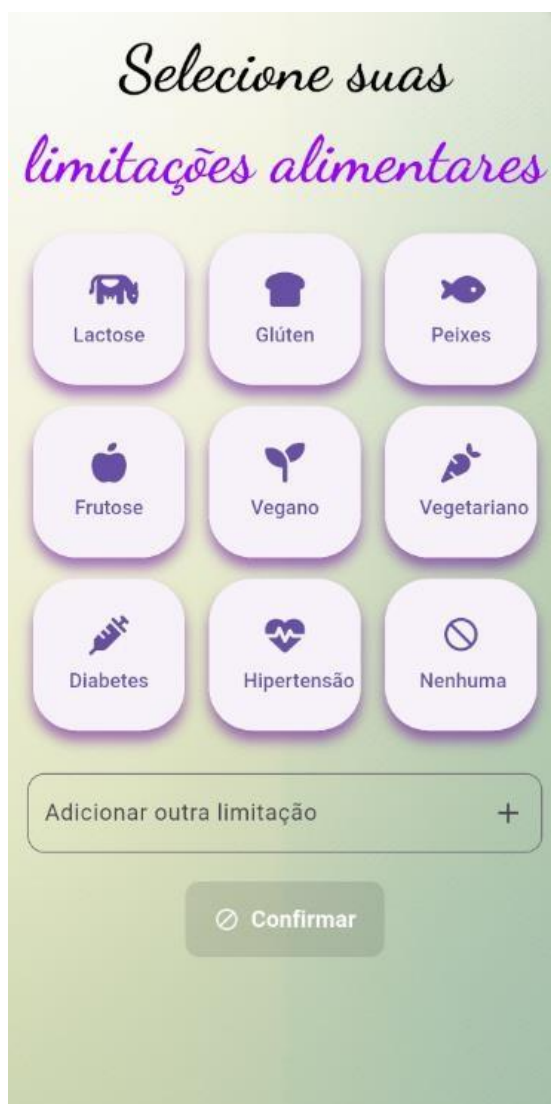
+ 1 hora

Salvar

CSU003 – Definir Preferências Alimentares	
Sumário:	Permitir que o usuário configure suas preferências e restrições alimentares.
Ator Primário:	Usuário
Ator Secundário: Nenhum.	
Casos de Uso Associados: CSU001 - Login.	
<b>Pré-condição:</b> a. O Usuário deve estar autenticado no sistema.	
<b>Fluxo Principal</b> 1- O Usuário acessa a opção "Preferências Alimentares" no menu. 2- O Sistema exibe uma lista com categorias de preferências (exemplo: Vegetariano, Vegano, Sem Glúten, Low Carb, etc).	

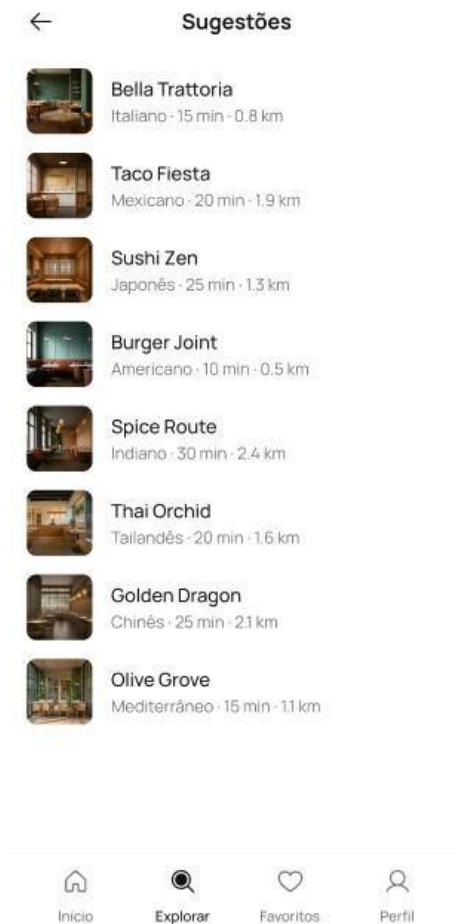
3- O Usuário marca as opções desejadas (usando checkboxes).
4- O Usuário clica em "Salvar Preferências".
5- O Sistema grava as preferências e exibe uma mensagem de confirmação.
<b>Fluxo Alternativo (2): Usuário deseja redefinir as preferências:</b> a. O Sistema permite que o usuário desmarque todas as opções e salve novamente. b. O Sistema apaga as preferências anteriores.
<b>Fluxo de Exceção (4): Falha ao salvar as preferências:</b> a. Se houver problema de comunicação com o servidor, o sistema exibe: "Não foi possível salvar as preferências. Verifique sua conexão e tente novamente." b. O caso de uso retorna ao passo 2.
<b>Pós-condições:</b> a. As preferências ficam salvas no perfil do usuário para uso nas próximas sugestões.
<b>Requisitos: RF003</b>
<b>Regras de Negócio: RN004.</b>
<b>Interface: I003</b>

Figura 5 - Protótipo de tela I003 – Definir Preferências Alimentares



<b>CSU004 – Receber Sugestões de IA</b>	
<b>Sumário:</b>	Exibir ao usuário uma lista de restaurantes e refeições sugeridas com base em preferências, localização e tempo disponível.
<b>Ator Primário:</b>	<b>Usuário</b>
<b>Ator Secundário:</b> Sistema de Recomendação por IA	
<b>Casos de Uso Associados:</b> CSU002 - Definir Tempo Disponível CSU003 - Definir Preferências Alimentares	
<b>Pré-condição:</b> <b>a. O Usuário deve estar autenticado.</b> <b>b. O Usuário já deve ter definido tempo e preferências.</b>	
<b>Fluxo Principal</b> 1- O Usuário acessa a opção "Sugestões de IA" no app. 2- O Sistema consulta os dados de perfil do usuário (tempo, preferências e localização). 3- O Sistema executa o motor de recomendação com IA. 4- O Sistema exibe uma lista de sugestões, com nome do restaurante, tipo de comida, tempo de espera e distância. 5- O Usuário pode clicar em uma das sugestões para ver mais detalhes.	
<b>Fluxo Alternativo (2): Usuário deseja atualizar filtros:</b> a. O Usuário pode acessar novamente as telas de tempo ou preferências e ajustar os filtros. b. O Sistema atualiza as sugestões com base nos novos dados.	
<b>Fluxo de Exceção (3): Falha ao buscar sugestões:</b> a. Se houver problema na comunicação com a API de IA, o sistema exibe: "Não foi possível carregar sugestões no momento. Tente novamente mais tarde." b. O caso de uso retorna ao passo 1.	
<b>Pós-condições:</b> <b>a. O Usuário terá visualizado uma lista personalizada de sugestões.</b>	
<b>Requisitos:</b> RF004	
<b>Regras de Negócio:</b> RN005.	
<b>Interface:</b> I004	

**Figura 6 - Protótipo de tela I004 – Receber Sugestões de IA**

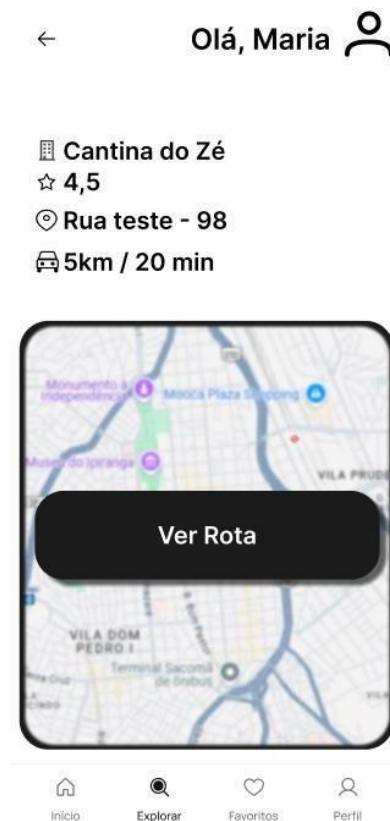


CSU005 – Visualizar Rota até o Restaurante	
<b>Sumário:</b>	Permitir que o usuário visualize o trajeto até o restaurante escolhido, com base na sua localização atual.
<b>Ator Primário:</b>	Usuário
<b>Ator Secundário: API de Mapas (ex: Google Maps API)</b>	
<b>Casos de Uso Associados: CSU004 – Receber Sugestões de IA</b>	
<b>Pré-condição:</b> a. O Usuário deve estar autenticado. b. O Usuário deve ter selecionado um restaurante da lista de sugestões.	
<b>Fluxo Principal</b> 1- O Usuário escolhe um restaurante da lista de sugestões. 2- O Sistema exibe a tela de detalhes do restaurante. 3- O Usuário clica em "Ver Rota". 4- O Sistema consulta a localização atual do usuário.	



5- O Sistema abre o mapa com o trajeto até o restaurante (com distância e tempo estimado).
6- O Usuário pode iniciar a navegação.
<b>Fluxo Alternativo (4): Escolha de outro aplicativo de mapa:</b> a. O Sistema oferece a opção de abrir a rota em outro app de navegação instalado no dispositivo (ex.: Google Maps, Waze). b. O Usuário seleciona o app desejado.
<b>Fluxo de Exceção (4): Falha ao obter localização:</b> a. Se o sistema não conseguir acessar o GPS, exibe a mensagem: "Não foi possível obter sua localização. Verifique as permissões de localização do app." b. O caso de uso retorna ao passo 3.
<b>Pós-condições:</b> a. O Usuário visualiza o trajeto até o restaurante selecionado.
<b>Requisitos: RF005</b>
<b>Regras de Negócio: RN006.</b>
<b>Interface: I005</b>

Figura 7 - Protótipo de tela I005 – Visualizar Rota até o Restaurante



### 1.3.1 Diagrama “É, Não É, Faz, Não Faz”

<p><b>É</b></p> <p>Aplicativo inteligente  Personalizado para alimentação  Prático e rápido  Baseado em geolocalização e tempo disponível  Apoiado por inteligência artificial  Adaptável às preferências e restrições alimentares do usuário  Compatível com dispositivos móveis  Capaz de gerar recomendações automáticas com base no histórico do usuário  Uma solução útil para pessoas com rotina acelerada</p>	<p><b>FAZ</b></p> <p>Cadastro de usuário  Registro de preferências alimentares  Sugestões de refeições baseadas em tempo e localização  Consulta de estabelecimentos próximos  Recomendação personalizada com base no histórico  Avaliação de refeições e salvamento de favoritas  Visualização de rotas até os estabelecimentos</p>
<p><b>NÃO É</b></p> <p>Rede social  Aplicativo de entrega (delivery tradicional)  Aplicativo gourmet  App de receitas caseiras  App com foco em nutrição clínica</p>	<p><b>NÃO FAZ</b></p> <p>Envio de mensagens entre usuários  Realizar pedidos diretamente  Reservar mesas ou contratar serviços diretamente  Exibir receitas ou tutoriais culinários  Substituir acompanhamento nutricional profissional</p>

### 1.3.2 Diagrama 360



### 1.4.1 Heurísticas:

- **Telas 1 e 2: Login e Cadastro**

#### Heurísticas Aplicadas:

2. Correspondência entre o Sistema e o Mundo Real: Uso de termos familiares (E-mail, Senha, Entrar, Cadastrar) e ícones reconhecíveis para login social (Google, Facebook, Apple).
4. Consistência e Padrões: Layout consistente dos campos de entrada, botões de ação e botões sociais entre as telas de login e cadastro. A barra inferior (que não é visível nestas telas, mas presumimos que seguiria um padrão) não se aplica aqui diretamente.
6. Reconhecimento em vez de Recordação: Os rótulos dos campos são claros, indicando o que deve ser inserido. Os ícones dos logins sociais são reconhecíveis.
8. Estética e Design Minimalista: Design limpo, com bom uso de espaço em branco e foco nos elementos essenciais do formulário.

- **Tela 3: Visualizar Rota**

#### Heurísticas Aplicadas:

2. Correspondência entre o Sistema e o Mundo Real: Uso de informações familiares como avaliação por estrelas, endereço, distância em "km". O ícone de mapa é universalmente reconhecido.
3. Controle e Liberdade do Usuário: O botão de "Voltar" (seta para a esquerda) no topo permite ao usuário retornar à tela anterior. O botão "Ver Rota" é uma ação clara.
4. Consistência e Padrões: A barra de navegação inferior (Início, Explorar, Favoritos, Perfil) mantém o padrão visual e de posicionamento.
6. Reconhecimento em vez de Recordação: Todas as informações são exibidas claramente, sem exigir que o usuário se lembre de detalhes. O mapa ajuda no reconhecimento visual da localização.
8. Estética e Design Minimalista: Layout limpo, com as informações essenciais do restaurante destacadas.

- **Tela 4: Definir Tempo**

#### Heurísticas Aplicadas:

1. Correspondência entre o Sistema e o Mundo Real: As opções de tempo (15 min, 30 min, 1 hora) são unidades de tempo comuns e de fácil compreensão. Utilizar linguagem simples, próxima da forma como o usuário pensa.,
2. Exemplo no App: Nomenclaturas claras, simples e diretas para definir a quantidade de tempo desejada.
3. Controle e Liberdade do Usuário: O botão "Voltar" (seta para a esquerda) e os botões "Salvar" e "Cancelar" oferecem controle total sobre a ação. O usuário pode mudar de ideia.
4. Consistência e Padrões: Os botões de tempo têm um estilo consistente. O botão "Salvar" segue o padrão dos outros botões de ação do app.
6. Reconhecimento em vez de Recordação: As opções de tempo são apresentadas como botões clicáveis, eliminando a necessidade de o usuário digitar ou memorizar.
7. Flexibilidade e Eficiência de Uso: Oferece opções predefinidas para eficiência (tempo comum) e uma opção "Outro" para flexibilidade (tempo personalizado).

8. Estética e Design Minimalista: Design limpo e direto ao ponto, com foco nas opções de tempo.

- **Tela 5: Preferências Alimentares**

Heurísticas Aplicadas:

2. Correspondência entre o Sistema e o Mundo Real: As categorias de preferências alimentares (Vegetariano, Vegano, Sem Glúten) são termos amplamente reconhecidos.

3. Controle e Liberdade do Usuário: O botão "Voltar" (seta para a esquerda) e o botão "Salvar Preferências" fornecem controle.

4. Consistência e Padrões: A lista de opções e o botão de ação seguem o padrão do aplicativo.

6. Reconhecimento em vez de Recordação: As opções são listadas claramente, permitindo ao usuário reconhecer e selecionar. Checkboxes ou toggles seriam o mecanismo usual, e a interface parece seguir isso.

8. Estética e Design Minimalista: A tela é limpa, com as opções bem espaçadas e focadas no propósito.

- **Tela 6: Sugestões**

Heurísticas Aplicadas:

2. Correspondência entre o Sistema e o Mundo Real: Apresenta nomes de restaurantes, tipos de culinária (Italiano, Japonês) e distância/tempo, que são informações do mundo real.

4. Consistência e Padrões: Cada sugestão é apresentada em um card consistente, facilitando a varredura visual. A barra de navegação inferior é mantida.

6. Reconhecimento em vez de Recordação: As sugestões são exibidas visualmente, com nomes e informações, permitindo ao usuário reconhecer e escolher. As imagens ajudam muito no reconhecimento.

8. Estética e Design Minimalista: O design dos cartões é limpo, com informações claras e concisas para cada restaurante.

### 1.5.1 Leis da UX:

- Lei de Postel

- Descrição simplificada:

"Seja conservador no que faz, seja liberal no que aceita dos outros."

- Funcionalidade do sistema:

Pesquisa por texto e por voz

- Lei de Fitts

- Descrição simplificada:

"O tempo para acessar um alvo é uma função da distância e do tamanho do alvo."

- Funcionalidade do sistema:

Os alvos de toque estão posicionados de maneira planejada para o alcance dos dedos na tela do celular e são grandes o suficiente para que os usuários cliquem sem dificuldades.

- Lei de Jakob

- Descrição simplificada:

"Os usuários passam a maior parte do tempo em outros sites e preferem que o seu site funcione da mesma maneira que todos os outros sites que já conhecem."

- Funcionalidade do sistema:

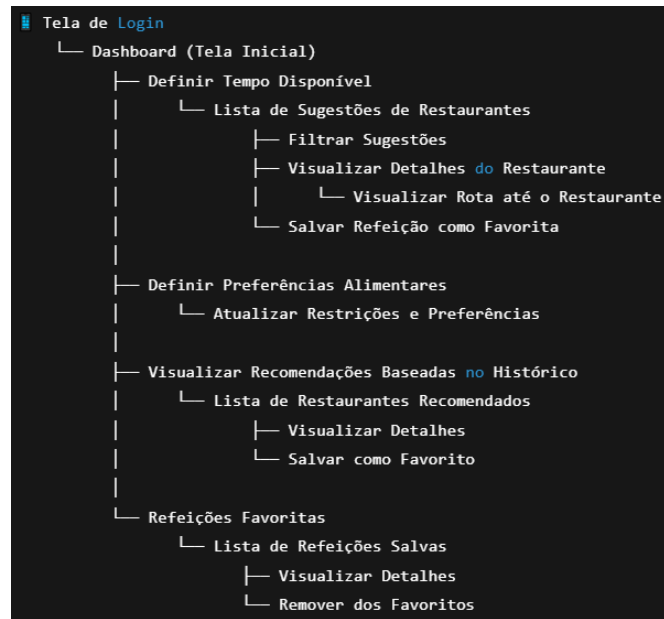
A disposição dos elementos (barra de pesquisa, menu, configurações e etc.) foram dispostos de maneira semelhante à de aplicativos muito usados para causar uma sensação de familiaridade nos usuários.

### 1.6.1 Gestalt:

#### **Princípio da similaridade:**

O princípio de Gestalt da similaridade explora o fato de que **elementos parecidos são percebidos como parte do mesmo grupo e tendo a mesma função.**

### 1.7.1 Fluxo de Telas - Casos de Usos:



Login:

Tela de Login → Dashboard

Definir Tempo Disponível:

Dashboard → Definir Tempo Disponível → Lista de Sugestões → Filtrar / Detalhes / Rota / Favoritar

Definir Preferências Alimentares:

Dashboard → Definir Preferências Alimentares → Formulário de Atualização

Visualizar Recomendações Baseadas no Histórico:


Dashboard → Recomendações do Histórico → Lista de Restaurantes → Detalhes / Favoritar

Salvar Refeições Favoritas:

Dashboard → Refeições Favoritas → Lista de Favoritos → Visualizar / Remover


1.8.1 Design System

Neste item são apresentadas as definições visuais e funcionais que irão padronizar a interface do sistema. O Design System é um conjunto de diretrizes que especifica o uso de cores, tipografias, ícones, componentes gráficos e o estilo geral da aplicação. Seu objetivo é garantir consistência visual, usabilidade e coerência entre todas as telas do sistema, facilitando o desenvolvimento, a manutenção e a experiência do usuário.

 1. Tipografia

- Fonte Padrão: Inter (Fonte nativa do Figma)

Estilo	Tamanho	Peso	Uso
Título	42px	Bold	Títulos principais das telas
Subtítulo	32px	Semi Bold	Seções ou títulos de página
Texto Normal	20px	Regular	Textos de conteúdo e botões

 2. Paleta de Cores

Tipo de Cor	Código HEX	Uso sugerido
Primária (Verde Claro)	#0FE687	Botões principais, destaques de ação positiva
Secundária (Verde Escuro)	#048049	Backgrounds de botões secundários ou hover
Destaque/Vibrante (Roxo)	#AC27FF	Links, bordas em foco, campos ativos
Fundo Escuro	#1B1B1B	Backgrounds gerais, modais, rodapés
Texto Claro (Branco)	#FFFFFF	Textos em fundo escuro
Texto Escuro (Preto)	#000000	Textos em fundo claro
Sucesso (Green - Prime)	#28a745	Mensagens de sucesso
Erro (Red - Prime)	#dc3545	Mensagens de erro, validações
Alerta (Yellow - Prime)	#ffc107	Alertas, notificações
Informação (Blue - Prime)	#17a2b8	Mensagens informativas, banners

### 3. Componentes UI (Botões, Inputs, Cards)

#### ✓ Botões:

Tipo de Botão	Cor de Fundo	Texto	Borda	Radius (Arredondamento)	Exemplo de Uso
Botão Primário	#0FE687	Preto	Nenhum	8px	"Confirmar", "Entrar"
Botão Secundário	#048049	Branco	Nenhum	8px	"Voltar", "Cancelar"
Botão de Destaque	#AC27FF	Branco	Nenhum	8px	"Salvar", "Nova Busca"
Botão Desabilitado	Cinza (#CCCCCC)	Cinza escuro	Nenhum	8px	Quando ação estiver inativa

#### ✓ Inputs e Campos de Formulário:

- **Borda padrão:** Cinza claro (#DDDDDD)
- **Ao focar:** Borda Roxa (#AC27FF)
- **Placeholder:** Cinza médio (#AAAAAA)
- **Texto digitado:** Preto (#000000)
- **Validação de erro:** Borda Vermelha (#dc3545) + Mensagem abaixo em vermelho
- **Validação de sucesso:** Borda Verde (#28a745)

#### ✓ Cards (Exemplo para Restaurantes ou Refeições):

- **Fundo:** Branco (#FFFFFF)
- **Borda:** Sombra leve ou borda cinza claro (#E0E0E0)
- **Título:** 20px Bold
- **Descrição:** 16px Regular
- **Botão dentro do card:** Botão Primário ou Secundário
- **Ícones:** PrimeIcons (exemplo: `pi pi-star`, `pi pi-heart`)

### 4. Ícones – PrimeIcons

O projeto utilizará os ícones da biblioteca PrimeIcons, acessados por meio da integração com o PrimeReact. Esses ícones serão aplicados para representar ações, estados e elementos visuais da interface, como botões, alertas e navegação. Exemplos incluem: usuário, confirmação, erro, busca, favoritos, entre outros.



## 📌 5. Estilo Geral (Look & Feel)

- **Estilo predominante:** Moderno e acessível
- **Bordas arredondadas:** 8px (aplicadas em botões, cards e campos de entrada)
- **Espaçamento interno (padding):** Entre 12px e 20px, dependendo do componente
- **Sombreamento:** Usado para dar destaque em cards e botões
- **Foco no contraste:** Garantia de boa legibilidade com uso de cores claras sobre fundo escuro e vice-versa
- **Transições:** Suaves, com duração de 0.2 segundos em interações como hover de botões e campos de formulário

### 1.9.1 UX Canvas

#### 1. Problema

Pessoas com rotinas apertadas ou com restrições alimentares enfrentam dificuldade em encontrar, de forma rápida e eficiente, refeições adequadas ao tempo disponível, localização e preferências alimentares. Isso pode gerar frustração, escolhas inadequadas e perda de tempo.

#### 2. Usuários

- Jovens adultos e adultos com vida urbana agitada
- Profissionais que trabalham fora
- Pessoas com dietas específicas (ex: vegana, sem glúten, low carb, etc.)
- Usuários que buscam praticidade e rapidez na escolha de onde e o que comer

#### 3. Objetivo do Produto (Solução)

Facilitar a escolha de refeições e locais para comer ou pedir comida, considerando:

- Tempo disponível
- Localização do usuário
- Preferências e restrições alimentares
- Histórico de interações no App

#### 4. Jornada do Usuário

- Abre o aplicativo (faz login ou usa social login)
- Informa o tempo disponível para a refeição ou permite o uso da localização
- Visualiza sugestões de refeições e restaurantes compatíveis
- Filtra as opções por preferências alimentares, tipo de comida ou preço
- Seleciona uma das opções sugeridas ou adiciona à lista de favoritos
- Finaliza a decisão (redirecionado para o local ou para um App de entrega parceiro, como iFood)

#### 5. Funcionalidades-Chave

- Social Login (Google, Facebook, Microsoft, Apple)
- Cadastro e edição de preferências alimentares
- Sugestões personalizadas com base no tempo, localização e histórico
- Inclusão de refeições e restaurantes como favoritos
- Exibição de tempo estimado até o local (ou tempo de preparo, no caso de delivery)
- Filtros avançados por tipo de dieta, ingredientes, preço ou distância
- Integração com mapas para cálculo de distância e rotas (GPS)

#### 6. Barreiras e Riscos

- Dificuldade em obter dados atualizados de restaurantes via API
- Personalização limitada caso o usuário não informe suas preferências
- Dependência de acesso à internet para funcionamento das principais funcionalidades
- Resistência de usuários acostumados a outros Apps de delivery

## 7. Requisitos Técnicos e Recursos

- API de localização (Google Maps)
- Backend com suporte à personalização e aplicação das regras de negócio
- Integração com APIs de dados de estabelecimentos e cardápios
- Frontend desenvolvido em React + PrimeReact (voltado para mobile, com layout adaptado para smartphones)
- Banco de dados para armazenar usuários, preferências, histórico e avaliações

### 1.10.1 Personas

Persona 1: Lucas Silva - Profissional Atarefado

Idade: 28 anos

Profissão: Analista de Sistemas

Localização: São Paulo – SP

Rotina: Trabalha em escritório, com horários apertados para o almoço

Comportamento: Busca otimizar o tempo de almoço e evitar perda de tempo escolhendo onde comer

Dores:

Falta de tempo

Repetição dos mesmos lugares

Dificuldade em encontrar restaurantes próximos com serviço rápido

Necessidades:

Sugestões rápidas e próximas

Informações sobre tempo de deslocamento

Filtros por tempo de espera e distância

Persona 2: Mariana Costa - Usuária com Restrições Alimentares

Idade: 34 anos

Profissão: Publicitária freelancer

Localização: Belo Horizonte - MG

Rotina: Trabalha de casa, mas sai com frequência para reuniões externas

Comportamento: Possui intolerância à lactose e evita glúten, sempre verifica o cardápio antes de escolher onde comer

Dores:

Falta de informações claras sobre ingredientes

Risco de consumir alimentos que causem problemas de saúde

Necessidades:

Filtros por restrições alimentares

Garantia de que o app não mostre opções incompatíveis com suas restrições

Visualizar detalhes do cardápio antes de decidir

Persona 3: Rafael Souza - Usuário que busca Economia e Promoções

Idade: 22 anos

Profissão: Estudante de Administração

Localização: Curitiba – PR

Rotina: Estuda em período integral e faz estágio à tarde

Comportamento: Sempre procura promoções e opções com bom custo-benefício

Dores:

Preços altos nos Apps de delivery

Falta de tempo para buscar promoções

Necessidades:

Visualizar promoções e descontos rapidamente

Filtros por faixa de preço

Sugestões baseadas em histórico de consumo com foco em economia