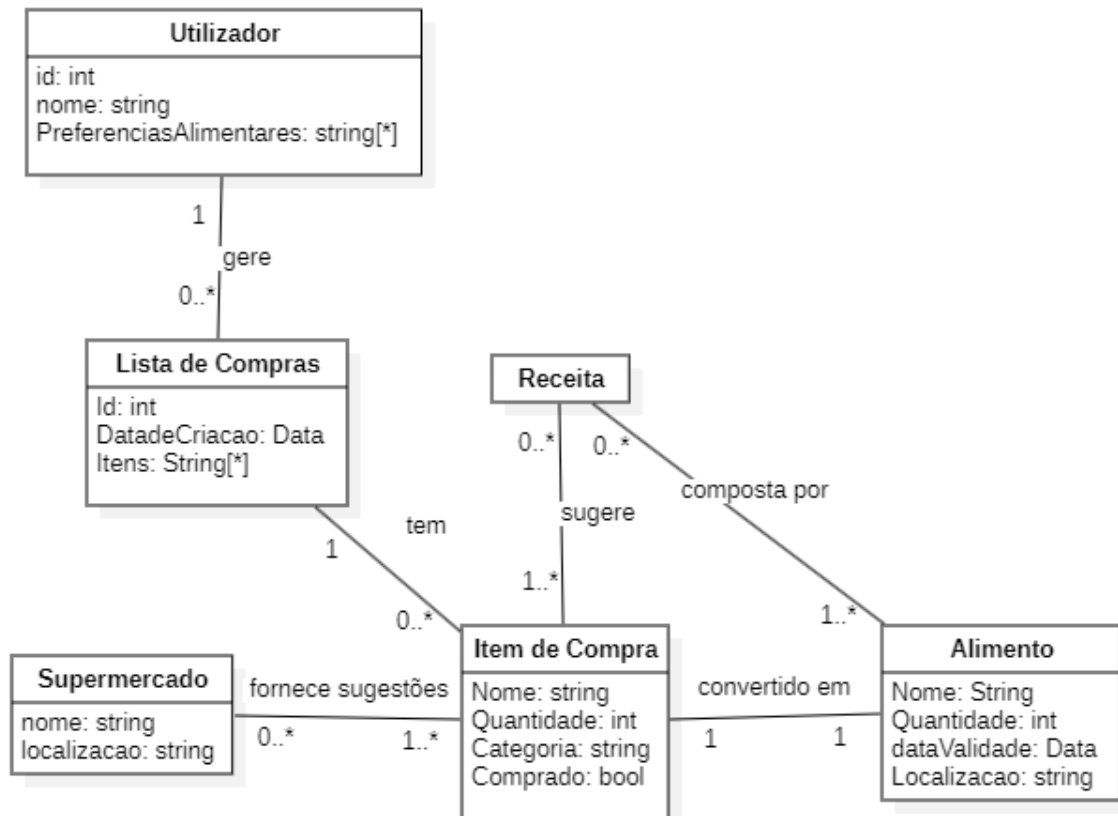




### 3. Modelo do domínio



#### **Classe: Utilizador (Família)**

Atributos: ID, Nome, Preferências Alimentares.

Relações: Gere uma Lista de Compras.

#### **Classe: Lista de Compras**

Atributos: ID, Data de Criação, Itens (lista de Itens de Compra), Estado (ativa/concluída).

Relações: Gerida por um Utilizador, Usada em Supermercados.

#### **Classe: Item de Compra**

Atributos: Nome, Quantidade, Categoria, Comprado (booleano).

Relações: Pertence a uma Lista de Compras, Pode ser convertido em Alimento.

***Classe: Alimento***

Atributos: Nome, Quantidade, Data de Validade, Localização (despensa/frigorífico/congelador).

Relações: Item de compra.

***Classe: Supermercado***

Atributos: Nome, Localização, Preços (lista de pares item/preço).

Relações: Fornece sugestões para Lista de Compras.

***Classe: Receita***

Atributos: Nome, Ingredientes, Tempo de Preparação.

Relações: Pode sugerir Itens de Compra.

## **4. Descrição Pormenorizada do Caso de Uso**

### **4.1**

#### **4.1.1 Nome**

Gerir Lista de Compras

#### **4.1.2 Atores**

Família (ator principal)

Supermercados (ator secundário, fornece dados via integração)

#### **4.1.3 Objetivo**

Permitir ao utilizador criar, editar e otimizar uma lista de compras, integrando sugestões de supermercados e convertendo itens comprados em alimentos no inventário.

#### **4.1.4 Pré-Condições**

O utilizador está autenticado na aplicação.

Existe uma conexão ativa com APIs de supermercados para obter preços e localizações.

#### **4.1.5 Pós-Condições**

A lista de compras está atualizada e otimizada com sugestões de supermercados.

Os itens marcados como comprados foram transferidos para o inventário de alimentos em casa.

#### **4.1.6 Fluxo de eventos**

##### **4.1.6.1 Cenário principal**

1. O sistema exibe a lista de compras existente ou cria uma nova lista vazia.
2. O utilizador adiciona itens à lista escolhendo um método:
  - O utilizador insere manualmente nome e quantidade (ex.: "500g de arroz");
  - O utilizador seleciona uma receita sugerida, e o sistema adiciona os ingredientes à lista;

- O utilizador lê um código de barras, e o sistema preenche os detalhes automaticamente.
3. O sistema consulta as APIs dos supermercados para obter preços e localizações dos itens.
  4. A API retorna os dados de preços e localizações ao sistema.
  5. O sistema otimiza a lista, sugerindo os melhores supermercados com base em preço e proximidade.
  6. O sistema verifica a disponibilidade dos itens nos supermercados sugeridos.
  7. O sistema mantém os itens disponíveis na lista.
  8. O utilizador edita ou remove itens da lista conforme necessário.
  9. O utilizador marca os itens adquiridos como "comprados".
  10. O sistema solicita ao utilizador detalhes dos itens comprados (quantidade, data de validade, localização).
  11. O utilizador fornece os detalhes solicitados.
  12. O sistema transfere os itens comprados para o inventário de alimentos.
  13. O sistema guarda a lista atualizada e sincroniza as alterações.

#### **4.1.6.2 Cenários alternativos**

- A1: Lista vazia ao iniciar
  - 1.1. O sistema verifica e não encontra uma lista existente.
  - 1.2. O sistema cria automaticamente uma nova lista vazia.
  - 1.3. O fluxo prossegue no passo 2.
- A2: Erro na leitura do código de barras
  - 2.1. O utilizador lê um código de barras, mas o sistema não reconhece o código.
  - 2.2. O sistema exibe uma mensagem ("Código inválido") e solicita inserção manual.
  - 2.3. O utilizador insere os detalhes manualmente.

2.4. O fluxo retorna ao início do passo 2 para nova tentativa ou prossegue para o passo 3.

- A3: Falha na consulta às APIs dos supermercados

3.1. O sistema consulta as APIs, mas a API não responde.

3.2. O sistema notifica o utilizador sobre a falha de integração ("Falha ao sincronizar com supermercados").

3.3. O sistema continua o processo sem sugestões otimizadas.

3.4. O fluxo prossegue a partir do passo 8.

- A4: Item indisponível nos supermercados sugeridos

6.1. O sistema verifica a disponibilidade e deteta que um item não está disponível.

6.2. O sistema sugere uma alternativa com base nas preferências do utilizador (ex.: "arroz integral" por "arroz branco").

6.3. O utilizador decide:

- Se o utilizador aceita a alternativa, o sistema substitui o item original.
- Se o utilizador rejeita a alternativa, o sistema mantém o item original.

6.4. O fluxo continua no passo 8.

#### **4.1.7 Casos de uso relacionados**

Inclui: Consultar Sugestões de Receitas (para adicionar ingredientes à lista).

Relacionado: Gerir Alimentos em Casa (para transferir itens comprados).

Relacionado: Sincronizar com Supermercados (para obter dados externos).

## **4.2**

### **4.2.1 Nome**

Sincronizar com Supermercados

#### **4.2.2 Atores**

Família (ator principal)

Supermercados (ator secundário, fornece dados via integração)

#### **4.2.3 Objetivo**

Aceder ao dados de preços e localizações de supermercados para sugerir as melhores opções de compra.

#### **4.2.4 Pré-Condições**

O utilizador está autenticado na aplicação.

Lista de compras criada.

#### **4.2.5 Pós-Condições**

Sugestões de supermercados apresentadas.

#### **4.2.6 Fluxo de eventos**

##### **4.2.6.1 Cenário principal**

1. O sistema inicia a sincronização ao receber uma lista de itens de compra de "Gerir Lista de Compras".
2. O sistema envia uma solicitação às APIs dos supermercados afiliados, incluindo os nomes e quantidades dos itens da lista.
3. As APIs retornam dados de preços e localizações para cada item solicitado.
4. O sistema processa os dados recebidos, associando preços e supermercados aos itens da lista.
5. O sistema retorna os dados processados ao caso de uso chamador ("Gerir Lista de Compras") para otimização da lista.

##### **4.2.6.2 Cenários alternativos**

- A1: Falha na resposta das APIs
  - 3.1. Se as APIs não responderem dentro de 2 segundos, o sistema regista a falha e retorna uma mensagem de erro ao caso de uso chamador.
  - 3.2. O fluxo termina sem dados de sincronização.

- A2: Dados parciais retornados

3.3. Se apenas alguns supermercados responderem, o sistema processa os dados disponíveis e retorna uma sincronização parcial, informando o utilizador sobre os dados ausentes.

3.4. O fluxo prossegue no passo 4 com os dados parciais.

## **4.3**

### **4.3.1 Nome**

Gerir Alimentos em Casa

### **4.3.2 Atores**

Família (ator principal)

Supermercados (ator secundário)

### **4.3.3 Objetivo**

Adicionar, remover ou editar alimentos na despensa, frigorífico ou congelador.

### **4.3.4 Pré-Condições**

O utilizador está autenticado na aplicação.

### **4.3.5 Pós-Condições**

Lista de alimentos atualizada com informações consistentes.

### **4.3.6 Fluxo de eventos**

#### **4.3.6.1 Cenário principal**

1. O sistema inicia a gestão de alimentos ao receber um item marcado como comprado de "Gerir Lista de Compras".
2. O sistema exibe um formulário para o utilizador inserir detalhes do item (nome, quantidade, data de validade, localização: despensa/frigorífico/congelador).
3. O utilizador preenche os detalhes e confirma.
4. O sistema adiciona o item ao inventário de alimentos do utilizador.
5. O sistema remove o item da lista de compras.



6. O sistema pergunta ao utilizador se deseja configurar um lembrete para a data de validade.
7. O utilizador define o lembrete (ex.: notificação 2 dias antes do vencimento) e confirma.
8. O sistema regista o lembrete e atualiza o inventário.

#### **4.3.6.2 Cenários alternativos**

- A1: Edição de item existente
  - 1.1. Se o utilizador aceder diretamente à funcionalidade para editar um item já no inventário (não vindo de "Gerir Lista de Compras"), o sistema exibe o item atual e permite ajustes.
  - 1.2. Prossegue a partir do passo 4, sem remover da lista de compras.
- A2: Item adicionado via código de barras
  - 2.1. Se o item foi digitalizado via código de barras, o sistema preenche automaticamente os campos de nome e categoria, solicitando apenas quantidade, data de validade e localização.
  - 2.2. O fluxo prossegue no passo 3.
- A3: Utilizador recusa lembrete
  - 6.1. Se o utilizador optar por não configurar um lembrete, o sistema finaliza a adição do item sem registar notificações.
  - 6.2. O fluxo termina.

### **4.4**

#### **4.4.1 Nome**

Consultar Sugestões de Receitas

#### **4.4.2 Atores**

Família (ator principal)

Supermercados (ator secundário)

#### **4.4.3 Objetivo**

Obter receitas sugeridas com base em preferências e alimentos disponíveis.

#### **4.4.4 Pré-Condições**

Preferências e lista de alimentos registados.

#### **4.4.5 Pós-Condições**

Lista de receitas sugeridas apresentada.

#### **4.4.6 Fluxo de eventos**

##### **4.4.6.1 Cenário principal**

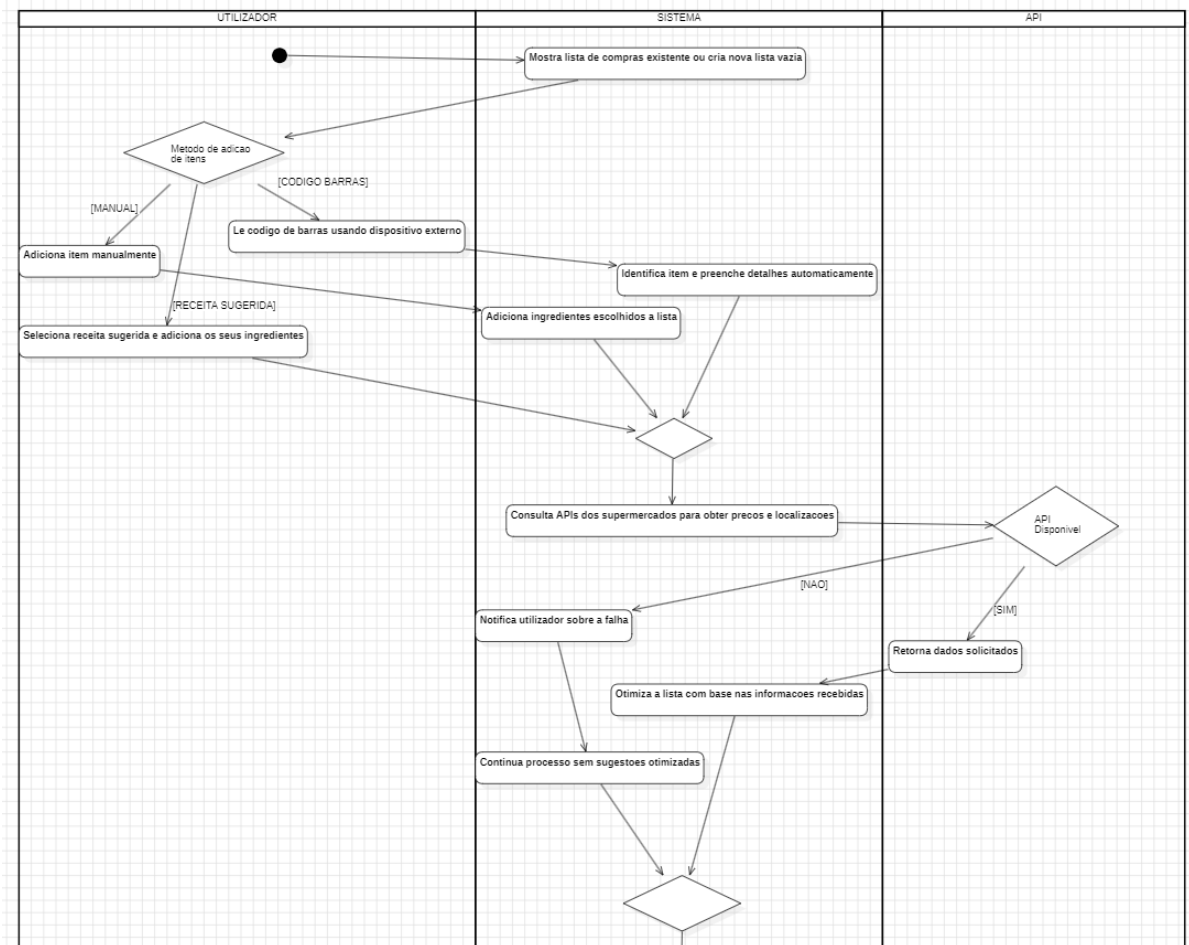
1. O sistema verifica o inventário de alimentos em casa, invocando "Gerir Alimentos em Casa" para obter a lista atualizada.
2. O sistema consulta as preferências alimentares do utilizador (ex.: alergias, tipos de cozinha preferidos).
3. O sistema analisa os alimentos disponíveis e as preferências, gerando uma lista de receitas compatíveis.
4. O sistema apresenta as receitas sugeridas ao utilizador, incluindo detalhes como ingredientes necessários e tempo de preparação.
5. O utilizador seleciona uma receita para visualizar instruções completas.
6. O sistema pergunta se o utilizador deseja adicionar os ingredientes em falta à lista de compras.
7. O utilizador confirma, e o sistema invoca "Gerir Lista de Compras" para adicionar os itens necessários.

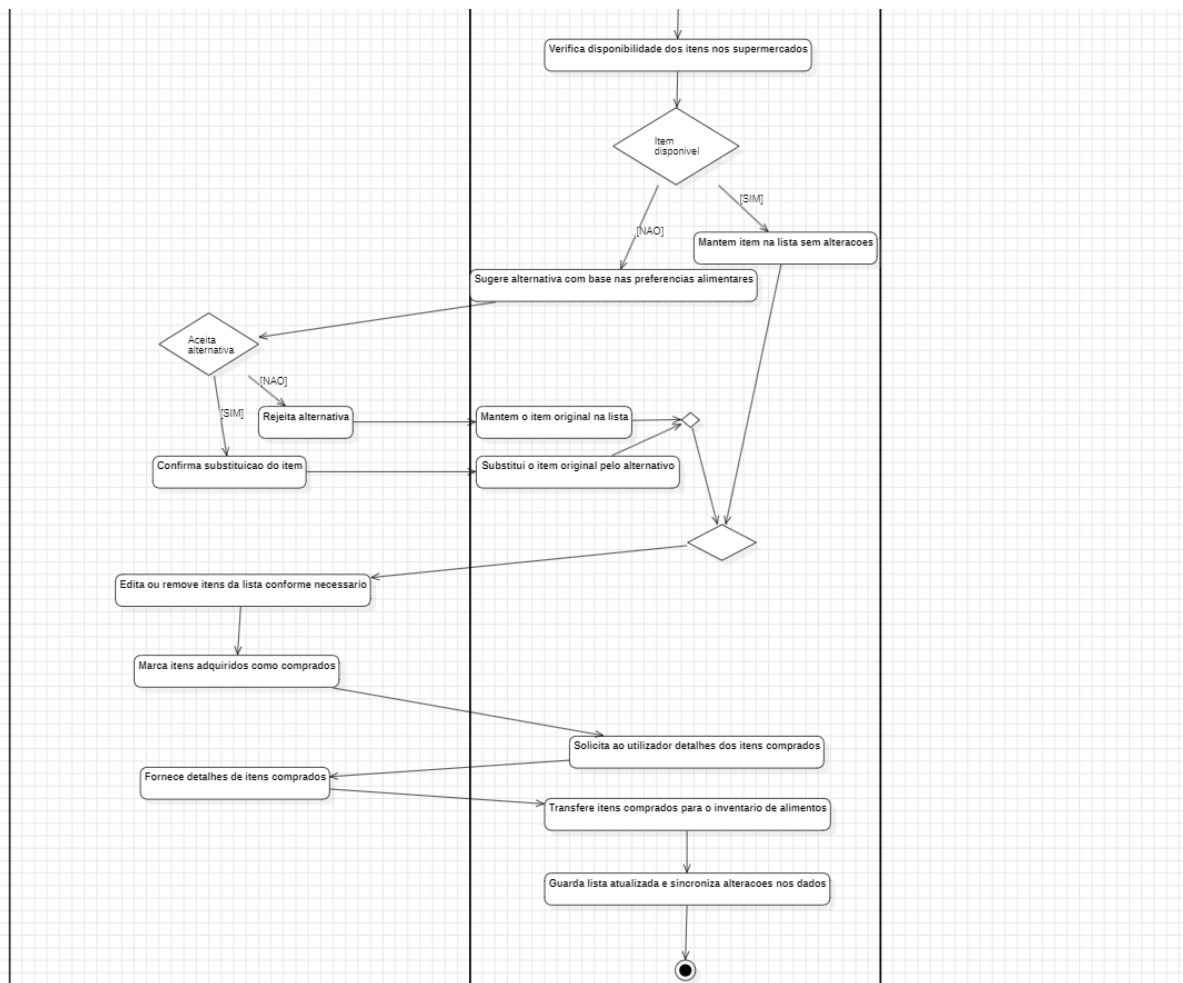
##### **4.4.6.2 Cenários alternativos**

- A1: Nenhum alimento disponível ou compatível
  - 4.1. Se não houver alimentos suficientes ou compatíveis com as preferências, o sistema exibe uma mensagem ("Nenhuma receita disponível. Adicione mais alimentos ao inventário") e sugere voltar ao menu principal.
  - 4.2. O fluxo termina sem sugestões.
- A2: Receita patrocinada selecionada
  - 5.1. Se a receita escolhida for patrocinada por uma empresa afiliada, o sistema exibe um bloco de publicidade com um link para o site da empresa antes de mostrar as instruções.
  - 5.2. O fluxo prossegue normalmente no passo 6.
- A3: Utilizador rejeita adicionar itens à lista
  - 7.1. Se o utilizador optar por não adicionar ingredientes à lista de compras, o sistema retorna ao menu de sugestões ou ao menu principal.
  - 7.2. O fluxo termina.

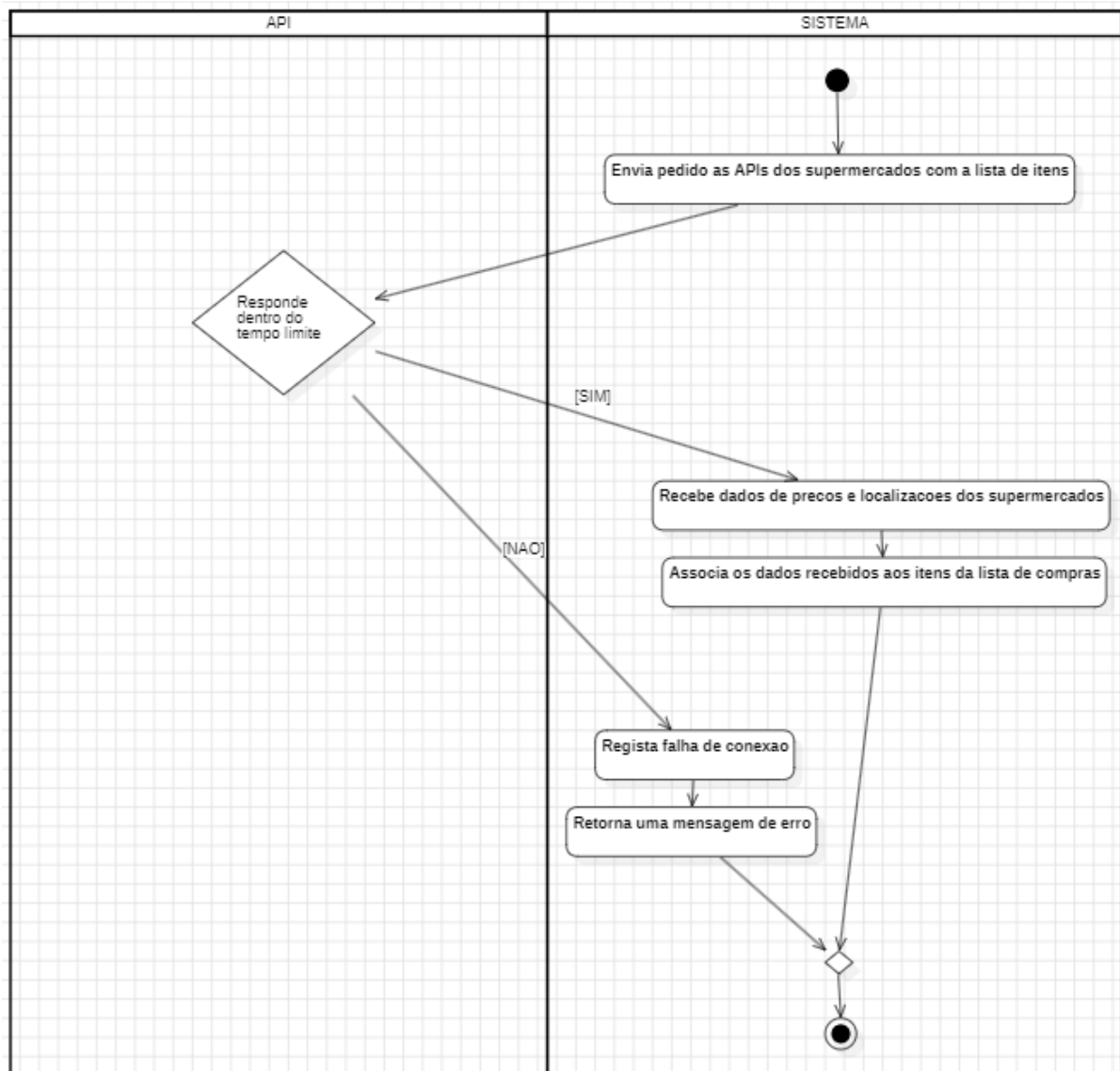
## 5. Diagrama de atividades

### 5.1 Gerir Lista de Compras

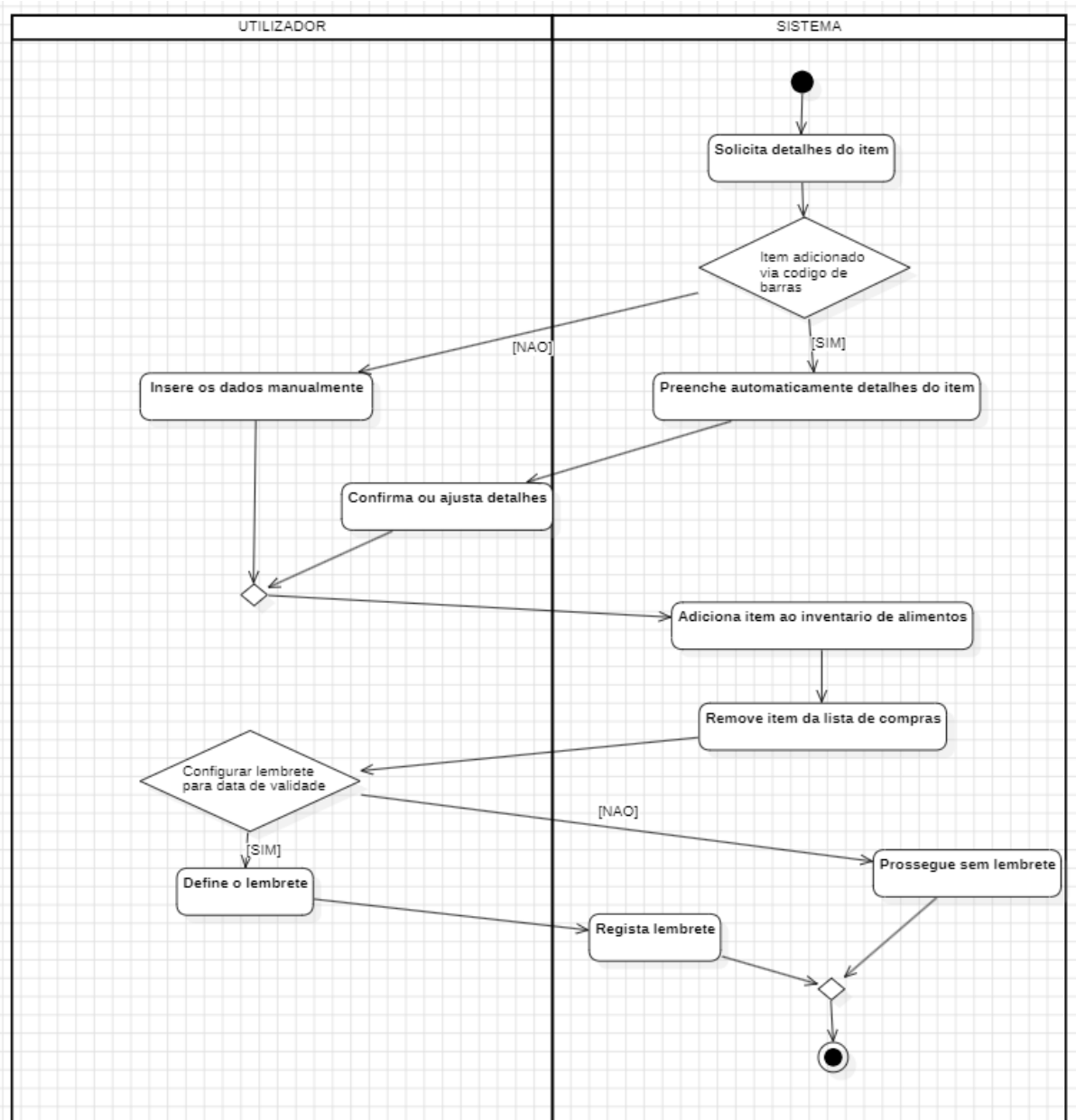




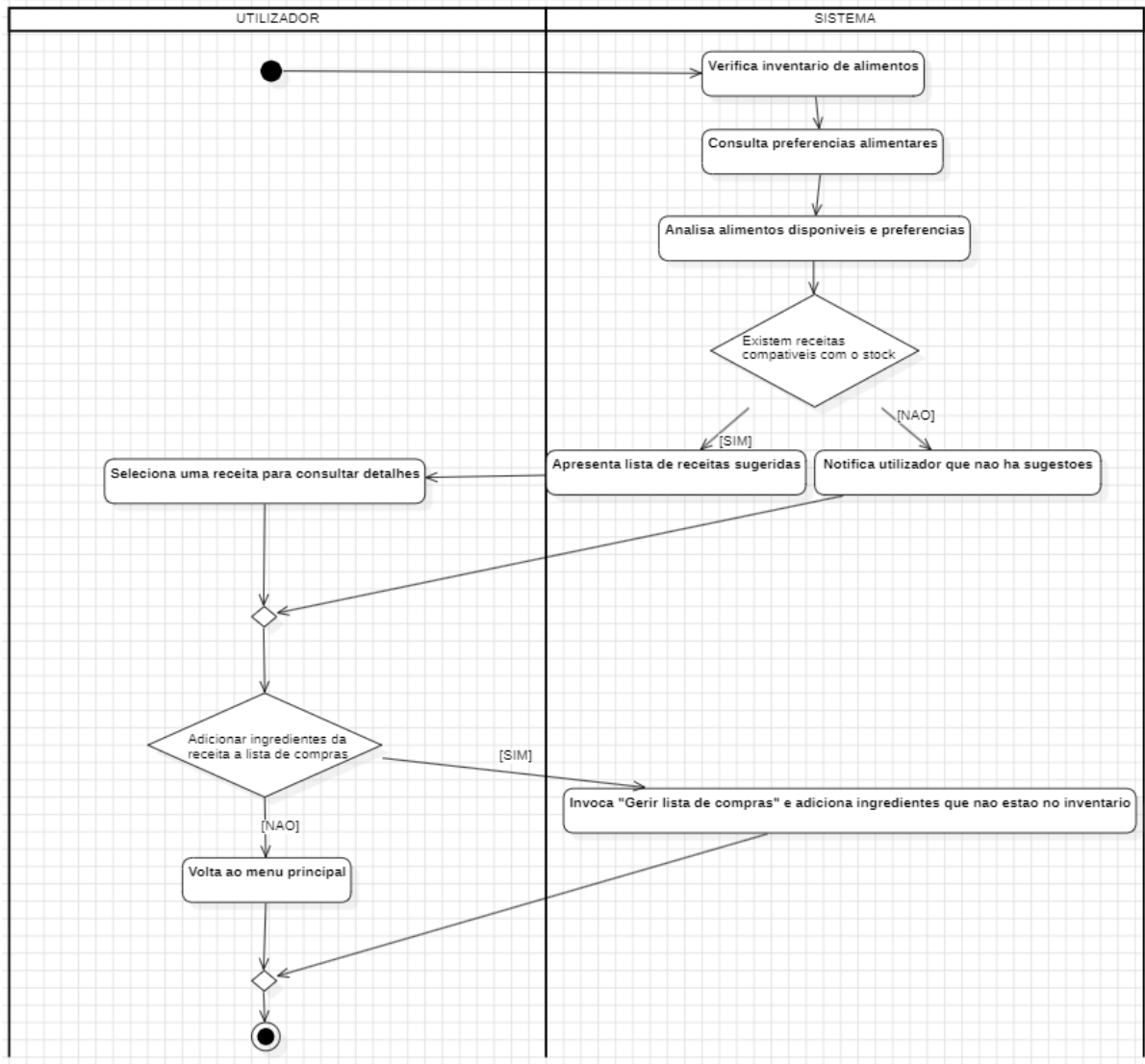
## 5.2 Sincronizar com Supermercados



### 5.3 Gerir Alimentos em Casa



### 5.4 Consultar Sugestões de Receitas



## 6. Glossário

- Lista de Compras
  - Descrição: Conjunto de itens que o utilizador planeia adquirir.
  - Formato: ID (int), Itens (lista)
  - Relação com Outros Termos: Associada a Utilizador, Supermercados
- Item de Compra
  - Descrição: Produto específico a comprar, com nome e quantidade.
  - Formato: Nome (string), Quantidade (float)
  - Relação com Outros Termos: Pertence a Lista de Compras
- Alimento
  - Descrição: Item armazenado em casa, com data de validade e localização.
  - Formato: Nome (string), Data (date)
  - Relação com Outros Termos: Convertido de Item de Compra
- Supermercado
  - Descrição: Estabelecimento comercial que fornece preços e localizações.
  - Formato: Nome (string), Localização (coordenadas)
  - Relação com Outros Termos: Relacionado com Lista de Compras
- Otimização
  - Descrição: Processo de sugerir o melhor supermercado com base em preço e distância.
  - Formato: Nenhum.
  - Relação com Outros Termos: Depende de Supermercado



## Trabalho Prático - Modelação e Design - Fase 2

- Código de Barras
  - Descrição: Identificador único de um produto, lido por scanner.
  - Formato: String
  - Relação com Outros Termos: Usado para identificar Item de Compra