Gestão Inteligente de Stocks

Ana Santos
Inês Soares
Nuno Veloso

Orientadores Matilde Pato
Nuno Datia

Relatório de progresso realizado no âmbito de Projeto e Seminário,
do curso de licenciatura em Engenharia Informática e de Computadores
Semestre de Verão 2017/2018

Março de 2018

Instituto Superior de Engenharia de Lisboa
Licenciatura em Engenharia Informática e de Computadores

Gestão Inteligente de Stocks

42142 Ana Rita Ferreira dos Santos
42162 Inês Lima Amil Soares
42181 Nuno Manuel Olival Veloso

Orientadores: Nuno Miguel Soares Datia
Matilde Pós-de-Mina Pato

Relatório de progresso realizado no âmbito de Projeto e Seminário,
do curso de licenciatura em Engenharia Informática e de Computadores
Semestre de Verão 2017/2018

Março de 2018

Resumo

Com a evolução da *Internet of Things (IoT)* nos dias de hoje, seria expectável que tarefas básicas do dia-a-dia se encontrassem ao abrigo destas tecnologias. Tal avanço minimizaria o tempo desperdiçado pelas pessoas em atividades realizáveis por máquinas. Ao automatizarmos a recolha de dados relacionada com os stocks de produtos em casa, simplificamos a gestão dos mesmos. Desta forma, auxiliamos os utilizadores a manter o stock adequado às suas necessidades, bem como alertá-lo para a proximidade do término dos produtos. Assim, o nosso trabalho vai no sentido de responder a questões como: "De que forma podemos evitar transtornos causados na altura de reabastecer a nossa despensa? Ou como proceder ao controlo de stocks de alimentos e outros produtos? Como impedir artigos fora de prazo?". Se entendermos que a nossa casa funciona como uma empresa, onde existem pessoas que podem realizar as mesmas tarefas, e.g. ir às compras seguindo uma lista previamente elaborada, capacitamos qualquer elemento da família para exercer a compra.

Palavras-chave: automatização; gestão; Iot; stocks; tarefas;.

Abstract

Abstract text (1 page).

Keywords: sorted keyword list, delimited by ;.

Agradecimentos

Texto dos agradecimentos. É opcional.

Índice

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

A gestão de stocks é um processo estruturado, assim, no âmbito da Internet of Things (IoT) e de forma a simplificar o quotidiano de cada um, surge este projeto. Atualmente, são várias as aplicações responsáveis por fornecer listas de compras, porém carecem de controlo de stocks e conhecimento dos hábitos dos seus utilizadores. A aplicação desenvolvida neste projeto distingue-se pelo controlo de stocks existentes nos armários, frigoríficos e despensa. Difere-se também, pela elaboração/presença de um algoritmo de previsão de stocks, com base no histórico de consumo e reposição do utilizador.

1.1 Enquadramento

No contexto do projeto assume-se a existência de duas formas de apresentação para os produtos: avulsos e embalados. Os primeiros são conservados em sistemas de arrumação marcados com *tags* programáveis por *smartphones*. Os detalhes dos itens são especificados pelo utilizador e carregados para a *tag*. Enquanto que para os produtos embalados, admite-se que os produtores utilizam *tags*, Near-Field Communication (NFC) ou Radio-frequency Identification (RFID), para guardar os rótulos de forma digital e em formato standard.

Após a aquisição, os artigos são armazenados em locais que devem dispor de dispositivos de hardware equipados com leitores de *tags*. É recolhida a informação presente na *tag*, identificado o tipo de movimento (entrada ou saída) e enviado para a API Web.

Através da aplicação disponibilizada, o utilizador poderá consultar:

- os produtos em stock,
- as listas do sistema e/ou as por si criadas,
- as suas casas e caraterísticas,
- as alergias dos membros de cada casa,
- os locais de armazenamento e dispositivos de hardware.

Será ainda possível:

- receber alertas de produtos perto do fim da validade,
- especificar stocks mínimos e/ou indesejados,
- partilhar listas entre utilizadores da mesma casa.

1.2 Metas e Objetivos

Têm-se como objetivos, os seguintes pontos:

- Desenho e Implementação das Aplicações Móvel e Web
- Desenvolvimento da API Web
- Desenho e Implementação a DB
- Desenvolvimento do acesso à DB
- Aplicação da Lógica de Negócio no acesso à DB
- Elaboração do Algoritmo de Previsão de Stocks
- Realização dos Algoritmos necessários

1.3 Especificações do Projecto e Resumo da Solução

O projeto é composto por 2 blocos principais, que se relacionam. A Figura **??** representa esses blocos.

*(*Figura Aqui)

Um dos blocos é a interação com o utilizador, através de duas aplicações, uma móvel e uma Web. A aplicação móvel desenvolvida apenas para a plataforma *Android*, e utilizando a linguagem *Kotlin*. Para a aplicação Web, implementada recorrendo à linguagem *JavaScript*, com o auxilio da *framework Express*.

O outro bloco é a API Web desenvolvida com a *framework* da *Spring*, chamada de *Spring Boot*. Subjacente a este último bloco estão as camadas: Database (DB), Data Acess Layer (DAL), Business Logic Layer (BLL). A camada da base de dados (DB), realizada com o Sistema de Gestão de Base de Dados (SGDB) *PostgreSQL*. Para a camada de acesso a dados (DAL), responsável pelas leituras e escritas, a ferramenta utilizada é a linguagem de programação *Java*, com a API Java Database Connectivity (JDBC). A camada da lógica de negócio (BLL), também se serve da ferramenta mencionada anterior para a sua implementação, esta camada é responsável pela gestão dos dados obtidos da DB ou da API Web.

1.4 Estrutura do Relatório

O relatório está estruturado em X capítulos.

O capítulo 2 formula o problema, detalhando os requisitos do projeto. Apresenta dificuldades encontradas ao longo do projeto. São ainda descritos conceitos fundamentais ao trabalho.

Capítulo 2

Formulação do Problema

Neste capítulo o problema é descrito de forma detalhada na secção 2.1, bem como os requisitos funcionais e não funcionais na secção 2.2. A secção 2.3 apresenta as dificuldades que surgiram no decorrer do projeto. São expostas na secção 2.4 noções da gestão de stocks

2.1 Descrição do Problema

No âmbito do projeto existe a necessidade de criar um sistema de informação que permita gerir os itens em stock de uma dada casa.

Uma casa está associada a um ou mais utilizadores, podendo um utilizador ter várias casas. Cada casa é caracterizada por um identificador único, um nome, atribuído por um utilizador no momento de registo da casa, quantos bebés, crianças, adultos e seniores vivem nessa casa. Cada utilizador é identificado univocamente por um email ou por um nome de utilizador, pelo nome da própria pessoa, idade e uma password. Para cada casa podem existir um ou mais administradores. Um utilizador pode criar as suas listas e as suas receitas. As listas que este cria pode decidir se quer partilhar com os restantes utilizadores da casa a que pertence, nunca podendo partilhar com utilizadores fora da sua casa. Existem listas do sistema que são comuns a todos os utilizadores registados, contudo são particulares a cada casa. As listas partilhadas pelos vários utilizadores dependem dos produtos que cada utilizador tem em casa. As receitas que um utilizador cria podem ser partilhadas com todos os utilizadores registados ou só com determinados utilizadores. Existem ainda um conjunto de receitas que são partilhadas por todos os utilizadores registados.

Cada receita é identificada por um identificador único, um nome, uma preparação, uma dificuldade, um tempo, quantas doses, qual o tipo de cozinha e qual o tipo de prato. Cada receita pode ter vários ingredientes. Para cada receita deve ser possível saber a quantidade dos vários ingredientes que a compõem. Cada lista é identificada por um identificador único e um nome. Uma lista pode ter vários produtos. Para os produtos presentes numa lista pode ser possível saber a sua marca e a quantidade de um produto na lista. Um produto é identificado pelo seu identificador único, contém um nome, se é comestível ou não, e a

validade perecível. Um produto pertence a uma categoria, podendo uma categoria ter vários produtos. Uma categoria é identificada por um identificador único ou um nome. Um produto pode ter vários itens em casa. Uma casa pode ter vários itens presentes na mesma.

Um item presente numa casa é identificado por um identificador único ou por uma marca, uma variedade e um segmento, é também caracterizado por uma descrição, o local de conservação, a quantidade e as datas de validade. Para cada item deve ser possível saber os seus movimentos, isto é, se entrou ou saiu de um local de armazenamento. Para cada movimento deve ser possível saber o tipo de movimento (entrada ou saída), a data em que ocorreu o movimento e a quantidade de produtos que ocorrem num movimento. Para cada casa existem vários locais de armazenamento dos itens, por exemplo armários, frigoríficos, etc. Cada local de armazenamento é caraterizado por um identificador único, a temperatura e um nome. Um local de armazenamento pode ter vários itens presentes numa casa e vários movimentos. Para cada local de armazenamento deve ser possível saber a quantidade de cada item.

Para cada casa deve ser possível saber que alergias os seus membros têm e para cada alergia saber o número de membros que têm essa alergia (os membros não precisam necessariamente de estar registados). Deve também ser possível saber os alergénios de um item presente na casa.

2.2 Requisitos Funcionais e Não Funcionais

2.2.1 Requisitos Funcionais

- Informar o utilizador dos produtos existentes, a sua validade e a sua quantidade;
- Alertas sobre os produtos que estão perto da data de validade;
- Geração da lista de compras com os produtos em falta;
- Possibilidade de especificar os produtos a ter sempre em stock bem como as suas quantidades mínimas;
- Lista de Compras *Offline* (permite rasurar para uso no supermercado);
- Listas partilhadas entre utilizadores da mesma casa;
- Criação de Listas (As Minhas Listas);
- Especificação das alergias dos membros da casa.

2.2.2 Requisitos Não Funcionais

- Lista de produtos quase a expirar;
- Lista de produtos indesejados (Lista Negra);

- Lista de contenção em situações de emergência (Lista SOS);
- Sugestão de receitas que utilizem os produtos mais perto do fim da validade;
- Inserção de receitas (As Minhas Receitas);
- Inserir refeições extraordinárias de eventos a realizar num futuro próximo, para acrescentar alimentos não básicos à lista de compras;
- Especificação dietas alimentares (Vegetarianos, *Vegans*, etc.) a cada utilizador da casa.

2.3 Dificuldades Encontradas

Para a realização deste projeto encontrámos dificuldades nos aspetos a seguir referidos.

2.3.1 Rótulos em Formato Não Digital

Nos dias de hoje, os produtos não possuem rótulos digitais. Isto é um problema para a concretização do projeto, na medida em que se torna menos eficiente a recolha dos dados presentes nos produtos. Contudo, assumindo que este dilema é resolvido fora do âmbito do projeto, apenas é preciso definir um formato standard de como os dados devem ser armazenados nas *tags*, que podem ser NFC ou RFID. Num cenário ideal, este formato deve ser respeitado por todos os embaladores. Assim, os produtos têm um rótulo, código de barras e uma *tag* NFC ou RFID, com a informação necessária. Está fora do âmbito do trabalho implementar o suporte hardware para a leitura das *tags* e qual o sentido do movimento (entrada ou saída). Assume-se que essas informações são disponibilizadas num formato conhecido.

2.3.2 Ausência de Identificador Único nos Itens

Os itens não dispõem de um identificador unívoco, alguns deles contêm um lote e um número de série. A ausência deste identificador impede a distinção entre itens iguais, o que impossibilita saber se entrou um novo item no local de armazenamento ou se saiu um dos itens presentes. Tal facto torna a gestão dos stocks dependente do dispositivo de hardware para distinguir o tipo de movimento.

2.4 Conceitos Básicos de Gestão de Stocks

Inventário - Um catálogo detalhado ou uma lista de bens ou propriedades tangíveis, ou os atributos ou qualidades intangíveis.

Stock Keeping Unit (SKU) (Unidade de Manutenção de Stock, em Português) - Um código de identificação de um produto e serviço para uma loja ou produto, muitas vezes retratado como um código de barras legível por máquinas que ajuda a rastrear o item para inventários. Ver exemplo 1.

Exemplo 1
Por exemplo, um armário pode ter pacotes de leite magro da marca X, 2 pacotes de leite magro da marca Y e 1 pacote de leite meio gordo da marca X. Logo, o armário contém 3 SKU, uma vez que um SKU se distingue pelo tamanho, cor, sabor, marca, etc.

Stock Item (Item de Stock, em Português) - Refere-se aos itens que se mantêm em stock físico na loja. O item de stock tem uma quantidade associada. Cada vez que uma venda é feita para aquele item, a sua quantidade será deduzida. Artigo aprovado para aquisição, armazenamento e emissão, e geralmente mantido à mão.

Product Category (Categoria de Produtos, em Português) - Taxonomias de classificação que subdividem um Setor ("yet another market construct") nos diferentes tipos de produtos para os quais existe demanda. Quanto mais especializada for uma categoria, mais especializado é o produto.

Nota: Neste projeto apenas se consideram as categorias de maior dimensão, são elas, por exemplo, Laticínios, Bebidas, Frescos, Congelados, entre outras.

Brand (Marca, em Português) - Um símbolo de identificação, marca, logótipo, nome, palavra e/ou frase que as empresas usam para distinguir os seus produtos dos outros.

Segmentation (Segmento, em Português) - Quando os estrategistas de marca falam sobre segmento,referem-se à segmentação do consumidor/audiência. A maneira antiga de abordar isso era através da demografia (idade, sexo, etnia, faixa de renda, urbano-rural, etc.). Agora a segmentação é VALS (valores, atitudes e estilo de vida).

Nota: Neste projeto o segmento é a quantidade presente numa embalagem, i.e., para um pacote de leite de 1L, o segmento é 1L.

Variety (Variedade, em Português) - A variedade é confusa porque pode ser difícil de entender onde a especialização da segmentação termina e a especialização em prol da Variedade começa. A variação é sobre a personalização de um produto para se adequar ao caráter do consumidor individual. Ver exemplo 2.

Exemplo 2
Note-se um pacote de leite com as caraterísticas, quantidade líquida igual a 1L, da marca X e do tipo UHT magro. Então, identificar-se-ia da seguinte forma:

- Categoria: Laticínios
- Produto: Leite
- Marca: X
- Segmento: 1L
- Variedade: UHT Magro

Capítulo 3

Solução do Problema

3.1 Modelo de Dados

3.1.1 Modelo Entidade-Associação

3.1.2 Modelo Relacional

House(house_id, house_name, house_babiesNumber, house_childrenNumber, house_adultsNumber, house_seniorsNumber)
CP: (house_id)

User(user_username, user_email, user_age, user_name, user_password)
CP: (user_username)
OCC: (user_email)

Allergy(allergy_allergen)
CP: (allergy_allergen)

Recipe(recipe_id, recipe_name, recipe_instructions, recipe_difficulty, recipe_time, recipe_servings, recipe_cuisine, recipe_dishType, recipe_type)
CP: (recipe_id)

SystemRecipe(recipe_id)
CP: (recipe_id)
CE: {(recipe_id) ref Recipe}

UserRecipe(recipe_id, user_username)
CP: (recipe_id)
CE: {(recipe_id) ref Recipe, (user_username) ref User}

SharedRecipe(recipe_id, user_username)
CP:(recipe_id, user_username)
CE: {(recipe_id) ref Recipe, (user_username) ref User}

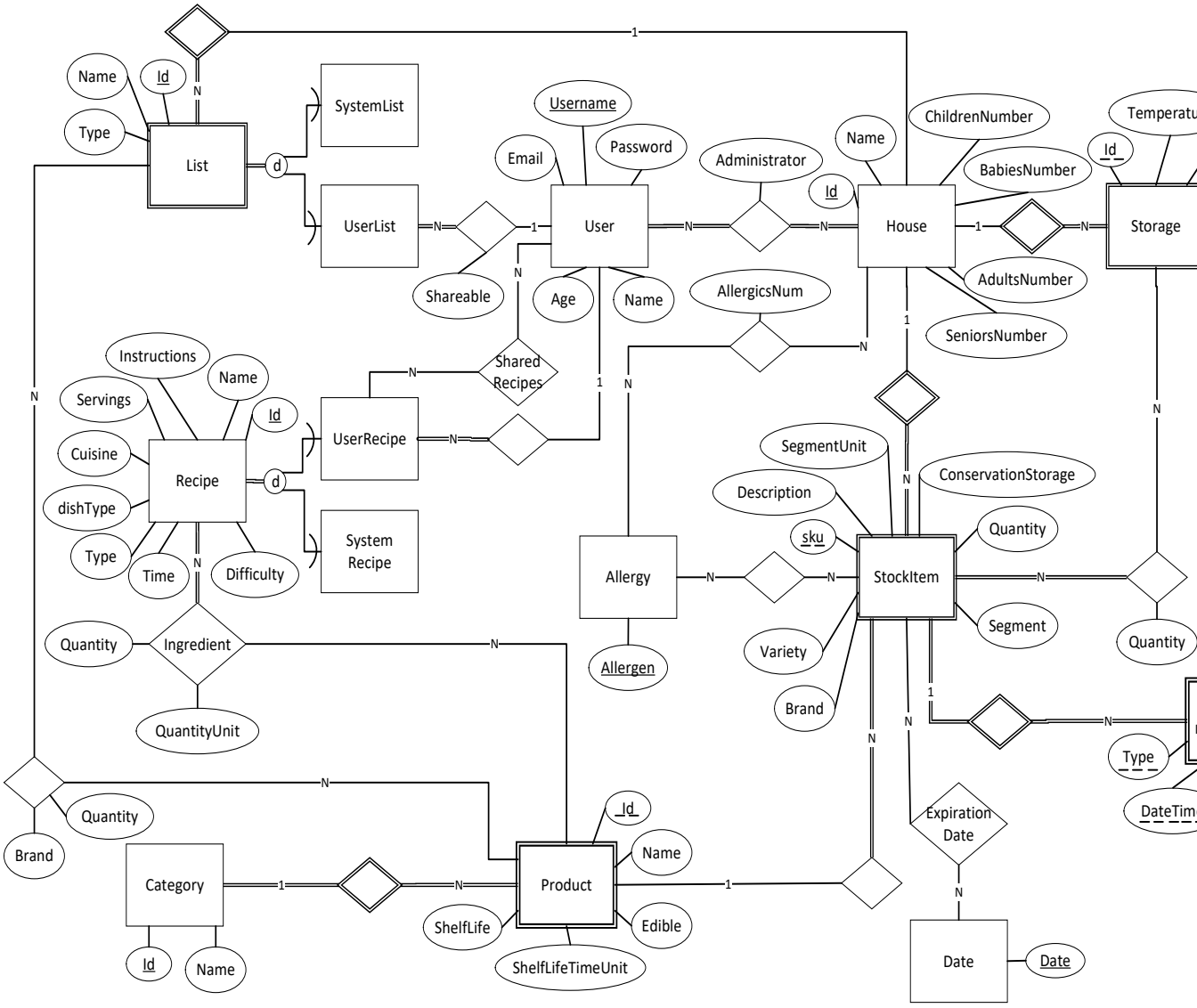

Figura 3.1: Modelo Entidade-Associação

List(house_id, list_id, list_name, list_type)

CP: (house_id, list_id)

CE: {(house_id) ref House}

SystemList(house_id, list_id)

CP: (house_id, list_id)

CE: {(house_id, list_id) ref List}

UserList(house_id, list_id, user_username, list_shareable)

CP: (house_id, list_id)

CE: {(house_id, list_id) ref List, (user_username) ref User}

Category(category_id, category_name)

CP: (category_id)

OCC: (category_name)

Product(category_id, product_id, product_name, product_edible, product_shelfLife, product_shelfLifeTimeUnit)

CP: (category_id, product_id)

CE: {(category_id) ref Category}

StockItem(house_id, stockItem_sku, category_id, product_id, stockItem_brand, stockItem_segment, stockItem_variety, stockItem_quantity, stockItem_ segmentUnit, stockItem_decription, stockItem_conservationStorage)

CP: (house_id, stockItem_sku)

OCC: (house_id, category_id, product_id, stockItem_brand, stockItem_segment, stockItem_variety)

CE: {(house_id) ref House, (category_id, product_id) ref Product}

Ingredient(recipe_id, category_id, product_id, ingredient_quantity, ingredient_quantityUnit)

CP: (recipe_id, category_id, product_id)

CE: {(recipe_id) ref Recipe, (category_id, product_id) ref Product}

Storage(house_id, storage_id, storage_name, storage_temperature)

CP:(house_id, storage_id)

CE: {(house_id) ref House}

UserHouse(house_id, user_username, userHouse_administrator)

CP: (house_id, user_username)

CE: {(house_id) ref House, (user_username) ref User}

StockItemStorage(house_id, stockItem_sku, storage_id, stockItemStorage_quantity)

CP: (house_id, stockItem_sku, storage_id)

CE: {(house_id, stockItem_sku) ref StockItem, (house_id, storage_id) ref Storage}

StockItemMovement(house_id, stockItem_sku, storage_id, stockItemMovement_type, stockItemMovement_dateTime, StockItemMovement_quantity)
CP: (house_id, stockItem_sku, storage_id, stockItemMovement_type, stockItemMovement_dateTime, StockItemMovement_quantity)
CE: {(house_id, stockItem_sku) ref StockItem, (storage_id) ref Storage}

HouseAllergy(house_id, allergy_allergen, houseAllergy_alergicsNum)
CP: (house_id, allergy_allergen)
CE: {(house_id) ref House, (allergy_allergen) ref Allergy}

ListProduct(house_id, list_id, category_id, product_id, listProduct_brand, listProduct_quantity)
CP: (house_id, list_id, category_id, product_id)
CE: {(house_id, list_id) ref List, (category_id, product_id) ref Product}

StockItemAllergy(house_id, stockItem_sku, allergy_allergen)
CP: (house_id, stockItem_sku, allergy_allergen)
CE: {(house_id, stockItem_sku) ref StockItem, (allergy_allergen) ref Allergy}

Date(date_date)
CP: (date_date)

ExpirationDate(date_date, house_id, stockItem_sku)
CP: (date_date, house_id, stockItem_sku)
CE: {(date_date) ref Date, (house_id, stockItem_sku) ref StockItem}

Restrições de Integridade

RI1: house_name é uma cadeia de caracteres de comprimento igual ou inferior a 35, podendo incluir letras, números, pontos e underscores;

RI2: house_babiesNumber é um número inteiro pertencente ao intervalo [0, 100];

RI3: house_childrenNumber é um número inteiro pertencente ao intervalo [0, 100];

RI4: house_adultsNumber é um número inteiro pertencente ao intervalo [0, 100];

RI5: house_seniorsNumber é um número inteiro pertencente ao intervalo [0, 100];

RI6: user_username é uma cadeia de caracteres de comprimento igual ou inferior a 30, podendo incluir letras, números, pontos e underscores;

RI7: user_email é uma cadeia de caracteres de comprimento igual ou inferior a 254, podendo incluir letras, números, pontos, underscores e arroba;

RI8: user_age é um número inteiro pertencente ao intervalo [0, 150];

RI9: user_name é uma cadeia de caracteres de comprimento igual ou inferior a 70, sendo apenas composto por letras;

RI10: user_password é uma cadeia de caracteres de comprimento igual ou inferior a 50, podendo incluir letras, números, caracteres especiais;

RI11: allergy_allergen é uma cadeia de caracteres de comprimento igual ou inferior a 75;

RI12: recipe_name é uma cadeia de caracteres de comprimento igual ou inferior a 35, podendo incluir letras, números, pontos e underscores;

RI13: recipe_difficulty pode tomar um destes valores [‘easy’, ‘average’, ‘difficult’];

RI14: recipe_time número inteiro superior a 0;

RI15: recipe_servings número inteiro superior a 0;

RI16: recipe_cuisine é uma cadeia de caracteres de comprimento igual ou inferior a 35;

RI17: recipe_dishType é uma cadeia de caracteres de comprimento igual ou inferior a 35;

RI18: recipe_type tem de tomar um destes valores [‘system, ‘user];

RI19: list_name é uma cadeia de caracteres de comprimento igual ou inferior a 35, podendo incluir letras, números, pontos e underscores;

RI20: list_type tem de tomar um destes valores [‘system, ‘user];

RI21: category_name é uma cadeia de caracteres de comprimento igual ou inferior a 35, sendo apenas composto por letras;

RI22: product_name é uma cadeia de caracteres de comprimento igual ou inferior a 35, sendo apenas composto por letras;

RI23: product_shelfLife é um número superior a 0;

RI24: product_shelfLifeTimeUnit tem de tomar um destes valores [‘day’, ‘week’, ‘month’, ‘year’];

RI25: stockItem_sku é uma cadeia de caracteres de comprimento igual ou inferior a 128, gerada pela composição de category_id, product_id, stockItem_brand, stockItem_segment e stockItem_variety;

RI26: stockItem_brand é uma cadeia de caracteres de comprimento igual ou inferior a 35;

RI27: stockItem_segment é uma cadeia de caracteres de comprimento igual ou inferior a 35;

RI28: stockItem_variety é uma cadeia de caracteres de comprimento igual ou inferior a 35;

RI29: stockItem_quantity é um número superior a 0;

RI30: stockItem_segmentUnit tem de tomar um destes valores [‘kg’, ‘dag’, ‘hg’, ‘g’, ‘dg’, ‘cg’, ‘mg’, ‘kl’, ‘hl’, ‘dal’, ‘l’, ‘dl’, ‘cl’, ‘ml’, ‘oz’, ‘lb’, ‘pt’, ‘fl oz’, ‘units’];

RI31: stockItem_conservationStorage é uma cadeia de caracteres de comprimento igual ou inferior a 128;

RI32: ingredient_quantity é um número superior a 0;

RI33: ingredient_quantityUnit tem de tomar um destes valores [‘kg’, ‘dag’, ‘hg’, ‘g’, ‘dg’, ‘cg’, ‘mg’, ‘kl’, ‘hl’, ‘dal’, ‘l’, ‘dl’, ‘cl’, ‘ml’, ‘oz’, ‘lb’, ‘pt’, ‘fl oz’, ‘units’];

RI34: storage_name é uma cadeia de caracteres de comprimento igual ou inferior a 35;

3.1.3 Domínio dos Atributos

Entidade	Atributo	Domínio	Tipo Variável (Post
	house_id	Número inteiro	bigserial
House	house_name	Cadeia de caracteres de comprimento variável	até 35 carateres (letras, números
	cell8	cell9	

Referências