



Pontifícia Universidade Católica do Rio de Janeiro

**API de ressurgimento de produtos de hardware para grandes varejistas
que atuam na área de e-commerce**

Leonardo Abreu Santos

**Centro Técnico e Científico - CTC
Departamento de Informática
Curso de Graduação em Ciência da Computação**

Orientador: Edmundo Bastos Torreão

Rio de Janeiro, Dezembro de 2023

Agradecimentos

Aos meus pais, Armando e Lourdes por todo o apoio dado em todos esses anos de graduação.

Aos meus amigos que estão comigo todos os dias acompanhando minha luta com a faculdade e o projeto final, garantindo a dose diária de serotonina.

Aos meus colegas de trabalho que foram super compreensivos em relação às atividades acadêmicas.

Por fim, ao meu orientador Edmundo Torreão que me acompanhou e auxiliou em todas as etapas do desenvolvimento deste projeto. Suas orientações permitiram o pensamento otimista em todo momento neste projeto.

Em memória do meu querido gato Vítinho que esteve do meu lado em todo momento do desenvolvimento deste projeto e lutou bravamente contra a felv. Acompanhando-me inclusive até a apresentação.

Resumo

Santos, Leonardo Abreu. Torreão, Edmundo. API de Ressuprimento de Produtos de Hardware para Grandes Varejistas Atuantes na Área de E-commerce. Rio de Janeiro, 2023. (Número de páginas: 74) Relatório de Projeto Final – Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

Este projeto tem como objetivo a especificação e implementação de uma API (Application Programming Interface) para o gerenciamento do processo de ressuprimento de uma varejista de produtos de hardware. Neste projeto, será possível realizar operações como o cadastro de usuários (funcionários e fornecedores), cadastro de fornecedores, cadastro de produtos, registro de todas as etapas da cadeia de ressuprimento internacional e nacional, visualização do estado do estoque, registro da venda dos produtos para clientes, registro do estoque diário dos produtos, cálculo do giro de estoque, cálculo do estoque de segurança e emissão de alertas sobre produtos em ponto de reposição e produtos inativos.

Palavras-chave: Gerenciamento de Estoque; Processo de Compras; Cadeia de ressuprimentos

Abstract

Santos, Leonardo Abreu. Torreão, Edmundo. Hardware Product Replenishment System for Large Retailers Operating in the E-commerce Sector. Rio de Janeiro, 2023. (Number of pages: 74) Final Project Report – IT Department. Pontifícia Universidade Católica do Rio de Janeiro.

This project aims at the specification and implementation of an API (Application Programming Interface) for managing the replenishment process of a hardware product retailer. In this project, it will be possible to perform operations such as user registration (employees and suppliers), supplier registration, product registration, registering all stages of the international and national supply chain, viewing the state of the inventory, registering product sales to customers, registering daily product inventory, calculating inventory turnover, calculating safety stock, and issuing alerts for products at the replenishment point and inactive products.

Keywords: Inventory Management; Procurement Process; Supply Chain

SUMÁRIO

1. Introdução.....	6
2. Proposta e Objetivo do Trabalho.....	8
3. Plano de ação.....	10
4. Cronograma.....	11
5. Pesquisa de conceitos aplicáveis a estoque.....	12
5.1. Conceitos abordados neste projeto.....	12
5.2. Conceitos não abordados neste projeto.....	17
6. Definição de Escopo.....	20
7. Diagrama de Classe.....	22
8. Diagrama dos Casos de Uso.....	24
9. Descrição dos Casos de Uso.....	26
10. Diagrama de Transição de Estados.....	54
11. Requisitos não funcionais.....	56
11.1. Requisitos de Performance.....	56
11.2. Requisitos de Segurança.....	56
11.3. Requisitos de Qualidade de Software.....	57
12. Testes realizados.....	58
13. Conceitos das Tecnologias Usadas.....	65
13.1. Utilização do ChatGPT no Projeto.....	66
14. Arquitetura do Software.....	67
15. Estrutura do Projeto.....	68
16. Possíveis Evoluções do Projeto.....	69
17. Considerações Finais.....	70
18. Referências.....	71

1. Introdução

A área de varejo de peças de **hardware e periféricos** é um negócio bem movimentado há anos, principalmente no que se refere ao atendimento das necessidades dos 'PC gamers' (configurações de PCs específicas para jogos). A demanda, nesse segmento, por peças que agreguem suas configurações conhecidas como "de entrada", "mid-end" e "high-end" são altas e constantes, principalmente por conta do avançar das tecnologias e dos softwares que cada vez mais exigem poder de processamento e gráfico, sejam por peças de hardware avulsas, PCs completos ou periféricos.

A gestão eficaz do estoque desempenha um papel central nesse varejo de peças de hardware, oferecendo uma visão abrangente e eficiente das demandas e padrões de consumo. Ao prever com precisão as necessidades, as empresas conseguem manter níveis ideais de estoque, evitando excessos e minimizando custos associados. Além disso, a rápida tomada de decisões baseada em dados contribui para a eficiência operacional, reduzindo a possibilidade de interrupções nas vendas.

Essa abordagem não apenas fortalece a resiliência da cadeia de suprimentos, mas também permite que as empresas estejam na vanguarda da inovação, respondendo de maneira ágil às mudanças no mercado de hardware e garantindo uma posição competitiva sólida. Em última análise, a inteligência de estoque não é apenas uma ferramenta operacional, mas um elemento estratégico fundamental para o sucesso sustentável das organizações neste cenário dinâmico.

Além da gestão interna, a gestão eficiente do estoque desempenha um papel crucial na otimização das operações de importação de peças de hardware. Ao analisar dados de fornecedores, condições de mercado e variáveis logísticas, ela possibilita uma programação otimizada de importações. Isso não apenas reduz os tempos de espera associados à reposição de estoque, mas também ajuda a minimizar custos relacionados à armazenagem e taxas alfandegárias. Uma 'inteligência' de estoque bem desenvolvida facilita a identificação de oportunidades para consolidar importações, aproveitar descontos por volume e garantir que os níveis de estoque estejam alinhados com as demandas sazonais ou flutuações cambiais.

Portanto, ao integrar a 'inteligência' de estoque à gestão de importação, as empresas podem não apenas garantir a disponibilidade de peças, mas também otimizar os processos globais de aquisição, contribuindo para uma eficiência operacional abrangente.

A demanda por peças de computadores se deve principalmente por três grupos separados:

- clientes que buscam um PC "gamer", sendo sua primeira configuração, ou fazer um upgrade numa configuração existente.
- clientes que buscam peças avulsas para montar ou agregar no seu PC.
- clientes que adotaram o modelo de trabalho home-office
- clientes que buscam as peças com maior poder de processamento para estações de mineração de criptomoedas.

2. Proposta e Objetivo do Trabalho

Este projeto tem como foco central a implementação de uma API abrangente para a gestão eficaz do ressuprimento de peças de hardware. O ciclo inicia-se com o registro do pedido, seguido pela aceitação por parte dos fornecedores.

A etapa subsequente envolve o registro de pagamento, seguido pela emissão de notas fiscais para garantir a conformidade legal.

Nas operações de transporte, a API gerencia as etapas tanto para fornecedores nacionais quanto internacionais.

No caso de fornecedores nacionais, a API conduz o registro das etapas de transporte, desde a saída do produto do fornecedor até a chegada no depósito da empresa. Essa jornada inclui a logística nacional que não possui a necessidade de procedimentos alfandegários.

Já para o cenário de importação, a API gerencia de forma integrada as etapas de transporte internacional, passando pela alfândega internacional, até a fase de transporte nacional, incluindo a alfândega nacional. Durante esse processo, são registrados as datas de cada etapa do transporte, a fim de realizar a documentação histórica do processo de ressuprimento.

A API será desenvolvida para disponibilizar operações CRUD (Create, Read, Update, Delete) para peças e fornecedores. Essa funcionalidade permitirá uma gestão ágil e eficaz do catálogo de peças e informações dos fornecedores, tornando a API mais flexível e adaptável às necessidades dinâmicas da cadeia de ressuprimento.

No que tange à saída de peças, no escopo deste projeto, as operações focando serão 'simplificadas nos estágios vitais de pedido de venda realizado, pedido disponibilizado e pedido entregue. O objetivo principal é garantir a contabilização precisa da saída de produtos, ao mesmo tempo que mantém a eficiência na gestão do ressuprimento como prioridade estratégica.

A API será desenvolvida em Javascript, utilizando o framework NodeJs para a implementação dos métodos. A API será desenvolvida baseada na arquitetura REST [2]. Será utilizado o pacote 'express'[4] do NodeJs [5] que fornece ferramentas para a implementação de servidores HTTP. Também será obrigatório um método de autenticação para a segurança da API.

O método de autenticação utilizado será o JWT [6], Json Web Token, que fornece tokens de acesso assinados por uma chave criptografada para a autenticação de usuário e segurança da API.

Em relação à persistência de dados, a aplicação irá utilizar o pacote do NodeJs 'sequelize' [3] que oferece ferramentas para elaborar o ORM (mapeamento objeto-relacional) para bancos de dados. Para esse trabalho, o banco de dados relacional a ser utilizado será o MySQL.

A aplicação disponibilizará as funções de CRUD com rotas de recursos da API bem definidas. A API contará com auxílio do programa Postman[7] para a realização de teste dos recursos disponibilizados pela API.

A API fará uso do verbo HTTP Patch. Em contextos web, particularmente em APIs, este método é usado para realizar atualizações parciais em registros existentes. Ao contrário de métodos que substituem o recurso completo, como o "PUT", o "PATCH" permite modificar apenas partes específicas do registro, economizando largura de banda e otimizando o processo de atualização. Isso é especialmente útil quando as mudanças são específicas e não é necessário reenviar todo o conjunto de dados relativo a determinada funcionalidade.

3. Plano de ação

Etapa 1: Documentação

- Documentação da pesquisa realizada;
- Atualização do escopo conforme a evolução e mudanças; necessárias durante o desenvolvimento;
- Atualização da modelagem conforme a evolução e mudanças; necessárias durante o desenvolvimento;

Etapa 2: Pesquisa

- Pesquisa de conceitos aplicáveis a estoque;

Etapa 3: Definição de escopo

- Definição clara e objetiva de todos os aspectos da aplicação;
- Detalhamento sobre todas as funcionalidades a serem implementadas;

Etapa 4: Modelagem

- Criação do modelo de representação da API.

Etapa 5: Especificação de requisitos

- Definição das características, objetivos e restrições que a API deve atender.

Etapa 6: Implementação

- Implementação da API.

Etapa 7: Testes

- Realização de testes.
- Elaboração de documento apresentando os resultados finais.

4. Cronograma(s) Previsto/Realizado

Previsto

Projeto Final	out 2021	nov 2021	dez 2021	jan 2022	fev 2022	mar 2022	abr 2022	mai 2022	jun 2022
Documentação									
Pesquisa									
Definição de Escopo									
Modelagem									
Especificação de Requisitos									
Implementação									
Testes									

Realizado

Projeto Final	out 2021	nov 2021	jun 2023	jul 2023	ago 2023	set 2023	out 2023	nov 2023	dez 2023
Documentação									
Pesquisa									
Definição de Escopo									
Modelagem									
Especificação de Requisitos									
Implementação									
Testes									

5. Pesquisa de conceitos aplicáveis a estoque

Os conceitos estão separados naqueles que foram implementados (considerados apropriados para a aplicação) e os não implementados.

5.1. Conceitos abordados neste projeto

O que é estoque?

Estoque é um acúmulo de materiais sempre estando presente em toda cadeia logística e cadeia produtiva. Sua função primária é garantir a disponibilidade de produtos, fazendo um regulador de entrada e saída dos materiais. [12]

Por que estocar?

Garantir a disponibilidade de materiais para atender às demandas dos clientes. Estoque é um capital investido que está “parado”. Muitos acreditam que isso é um ponto negativo para se estocar, porém, existem vantagens na manutenção de estoques, sobretudo porque é possível garantir a continuidade das operações da empresa, evitando, por exemplo, atraso de entrega dos fornecedores, imprevisibilidade de vendas, variação de demanda do mercado e falta de produto no mercado.

O que é giro de estoque?

O giro de estoque é um indicador estatístico que descreve quantas vezes o estoque de determinado produto ou do total do estoque é repostado durante determinado período. Essa informação mostra de quanto tempo um estoque é renovado. Dessa forma é possível criar um planejamento financeiro específico para produtos que apresentem um maior giro de estoque.

Fórmula para o cálculo do giro de estoque:

Giro de estoque do produto X = Número total de vendas do produto X em determinado período / Quantidade média de estoque no mesmo período

Por exemplo:

Número total de vendas: 12000 unidades ao ano

Volume médio de estoque: 800 unidades

Cálculo: 12000/800: 15 giros ao ano

Caso a empresa venda vários tipos de produto, o apropriado também fazer o cálculo considerando os valores financeiros dos custos de compra dos produtos.

Por exemplo:

Volume de vendas ao ano, em R\$, do produto X: R\$ 60000,00

Volume médio do estoque, em R\$, do produto X: R\$ 12000,00

Cálculo: 60000/12000: 5 giros ao ano

Com o giro de estoque é possível também calcular a frequência mínima de processos de compras no tempo necessária de aquisição junto ao fornecedor:

Por exemplo:

Número de dias no ano: 365 dias

Número de giros de estoque: 15 giros

Cálculo: 365/15 = 24,3 dias

Dessa forma, em média será necessário um novo giro de estoque a cada 24 dias com a compra de um novo lote de produtos ou matérias-primas. [11] [12]

O que é SKU?

SKU é sigla para Stock Keeping Unit (Unidade de Armazenamento de Estoque). Serve para identificar um produto com um código único de identificação atribuído a um produto. O código SKU é formado por uma sequência de números e letras para representar sua especificidade. [13]

O que é EAN?

Significa European Article Number (Numeração Europeia de Itens). É um código de barras composto por uma sequência de 13 dígitos. Esses 13 dígitos descrevem informações sobre fabricante, país de origem, modelo do produto e dígito verificador. [14]

Quais são as principais estratégias de estoque no varejo?

Estoque de antecipação ou estoque sazonal

A empresa utiliza essa estratégia a fim de se manter abastecida em períodos de alta demanda em decorrência de épocas específicas do ano, como em datas comemorativas tais como Natal e Páscoa. [8] [10]

Para esse projeto vamos considerar que o estoque deve estar abastecido com 10 dias de antecipação da data comemorativa. Desse modo, vamos considerar que o tempo de ressuprimento é 30 dias.

Estoque de segurança

Tem o objetivo de garantir a entrega dos itens em casos inesperados nos quais a falta de estoque poderia ocorrer, ou seja, é um estoque de precaução. Servem principalmente para absorver as flutuações de demanda que ocorrem durante o tempo de espera pela reposição.

A implementação do estoque de segurança será feita a partir da abordagem baseada no nível de disponibilidade de estoque. Pois, é uma abordagem estatística, que oferece maior certeza quanto ao nível de serviço.

O estoque de segurança utiliza a seguinte fórmula:

$$ES = Zns \times \sqrt{L} \times \sigma$$

Em que:

ES = Estoque de Segurança

L = Lead Time (tempo em dias para o ressuprimento no estoque)

σ = desvio padrão da demanda diária

Zns = Constante relacionada ao nível de serviço desejado pela empresa

Figura 1 - Constante Zns

Nível de Serviço	Valor z_{NS}
85%	1,036
90%	1,282
95%	1,645
99%	2,326
99,5%	2,576

Fonte: <http://www.conhecimento.market.com.br/artigo/453> (2023)

O que é nível de serviço?

É a probabilidade de não haver ruptura de estoque ao longo do tempo de espera pela reposição. Quanto maior o nível de serviço desejado, maior o estoque de segurança.

Exemplo:

Suponhamos que a empresa deseja um nível de serviço de 95%.

$L = 5$ e $\sigma = 4$.

Considerando um nível de serviço de 95%, o valor de Z_{ns} é 1,645, conforme a tabela acima.

Então, o estoque de segurança é: $1,645 * \sqrt{5} * 4 = 14,71$.

Arredondando, para cima, temos 15.

Isso quer dizer que ao manter um estoque de segurança de 15 unidades, espera-se que em mais de 95% dos ciclos de reposição não haja ruptura de estoque de produtos. Qualquer valor acima de 15 seria um excesso para o nível de serviço de 95%.

Com essa informação podemos também determinar o ponto de reposição, isto é, quando pedir novos produtos).

O ponto de reposição utiliza a seguinte fórmula:

$$PR = L \times \mu D \times ES$$

Em que:

PR = Ponto de reposição

L = Lead Time (tempo em dias para o ressuprimento no estoque)

μD = Demanda média diária

ES = Estoque de segurança

obs: arredonda-se para cima caso o resultado tenha casas decimais. [15]

Estoque inativo

Estoque inativo é o nome dado à estocagem de produtos que estão parados devido ao mau desempenho em vendas, se tornando obsoletos. Apesar de não ser o cenário ideal, é uma situação comum de se acontecer nas empresas. A solução para esse tipo de estoque normalmente é feita por meio de promoções ou transferir essas mercadorias com baixo preço para lojas especializadas em pontas de estoque ('outlets'). [8] [10]

5.2. Conceitos não abordados neste projeto

Estoque Mínimo

O estoque mínimo representa a quantidade de mercadorias necessárias para satisfazer a demanda, sendo o montante que a empresa deve manter em estoque para abranger o período de entrega dos fornecedores. Em termos simples, é o estoque necessário para cobrir o intervalo entre o consumo atual e a próxima entrega de fornecedores.

Na prática, quando a quantidade do estoque mínimo é atingida, é hora de iniciar um novo pedido junto ao fornecedor. Essa medida indica que a empresa dispõe apenas de estoque suficiente para atender à demanda do mercado até a próxima entrega agendada.

Além disso, o conceito de estoque mínimo possibilita a implementação de alertas em sistemas de gestão, indicando níveis críticos de estoque. Esses alertas são acionados quando a quantidade em estoque atinge um ponto considerado crítico, permitindo à empresa agir proativamente, realizar novos pedidos e evitar possíveis interrupções nas operações devido à escassez de produtos.

Fórmula para o cálculo de estoque mínimo:

$$\text{Estoque Mínimo} = (a / b) \times c$$

Em que:

a = saída das mercadorias em determinado período

b = quantidade de dias deste período

c = tempo de reposição

Por exemplo:

Saída de 120 unidades de determinado produto em 1 mês

Quantidade de dias deste período = 30 dias

Tempo de reposição = 15 dias

Estoque mínimo = $(120 / 30) \times 15 = 60$ unidades

Isso significa que ao estoque de determinado produto, com essas características, chegar a 60 unidades o mesmo estará em seu estado crítico.

Estoque consignado

Esse tipo de estoque acontece no cenário em que um varejista utiliza outros canais de distribuição. Uma empresa pode ter seus produtos oferecidos em outra loja. Em uma loja multimarca, por exemplo. Nessa modalidade, é comum que ocorra a venda consignada, ou seja, a outra loja mantém os produtos em seu estoque por um tempo estabelecido entre as empresas por meio de contratos. É uma boa opção para empresas que não dispõem de lojas físicas e não teriam espaço físico para manter o estoque. [8] [10]

Estoque de ciclo de produção

Este tipo de estoque de peças (sub-partes) / insumos intermediários é comumente implementado, por exemplo, em indústrias na fabricação de suas próprias peças. O estoque armazena as peças (sub-partes) / insumos intermediários necessários para a produção do produto final. Neste cenário é necessário planejar um ciclo produtivo para atender à demanda. [8] [10]

Dropshipping

Esta modalidade está presente principalmente no e-commerce onde, na verdade, não existe estoque. A mercadoria fica com o fabricante ou distribuidor. Quando o cliente realiza a compra, a fornecedora é a responsável por enviar o produto. Dessa forma a empresa que vendeu o produto não se envolve na estocagem dessas mercadorias. [8] [10]

Estoque regulador

Entra em cena em empresas que possuem várias filiais e contam com um centro de distribuição, o estoque regulador. Esse é responsável por ressuprir outras filiais em casos de necessidade de produtos. Como o nome diz, uma filial central fica responsável por regular os estoques de outras filiais. Essa filial é um depósito que comporta um volume maior de produtos em relação às outras filiais. Dessa forma, não é preciso alocar um espaço maior nas demais filiais, economizando gastos, entretanto é necessária a criação de uma logística de depósito-filiais. [8] [9]

Estoque em trânsito

Abrange aqueles produtos que estão na etapa de transporte, no caso um tempo de transporte relativamente longo, entre a empresa fabricante e a empresa que irá revendê-los. Por este motivo, esse estoque também é chamado de estoque em trânsito. [8] [9] [10]

6. Definição de Escopo

Escopo

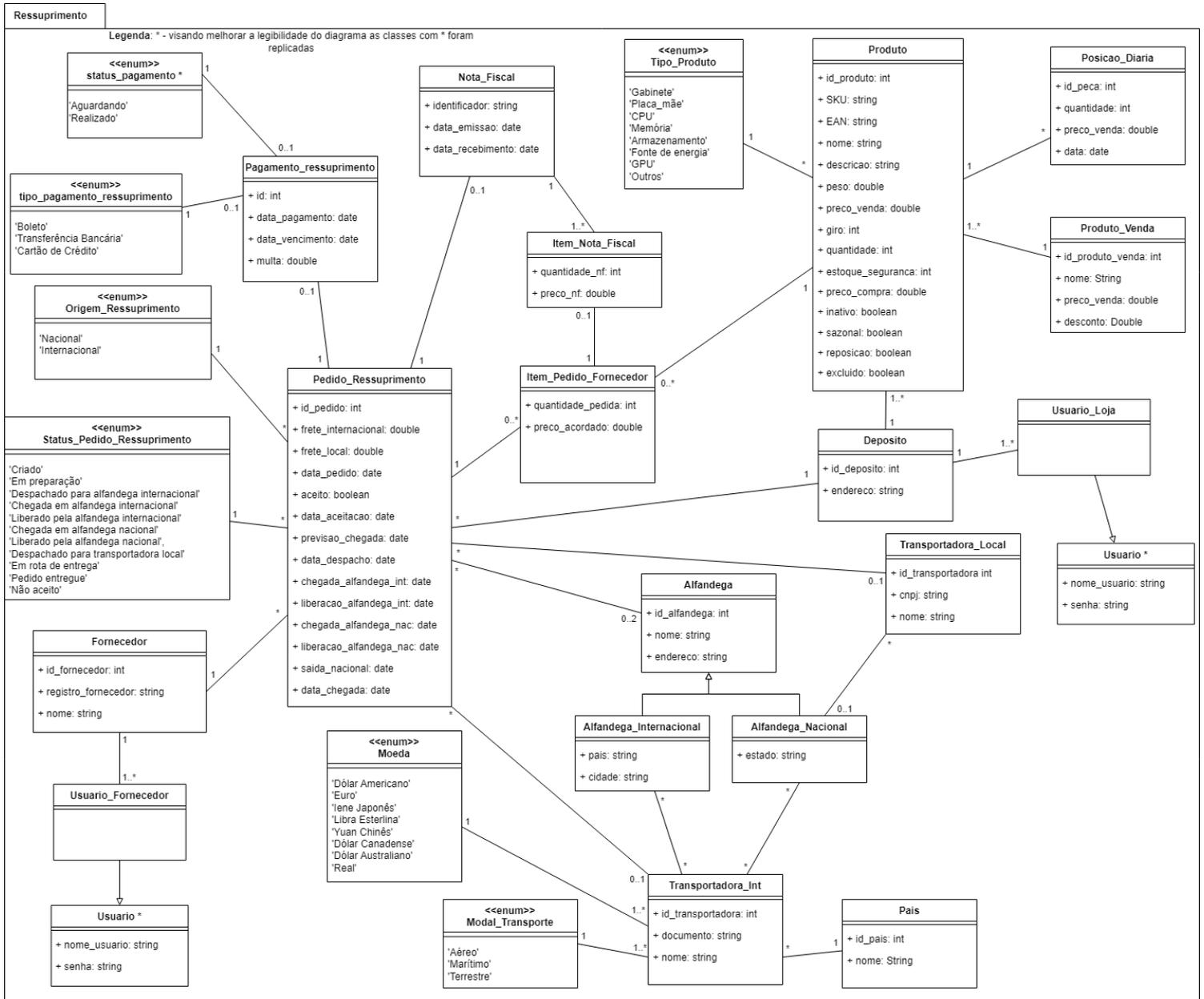
- O projeto irá contar com auxílio do software postman para realizar os testes das requisições e receber respostas http.
- O projeto contará apenas com o backend do projeto, não contendo uma interface visual.
- A linguagem utilizada será Javascript e a API será gerada em NodeJs;
- O projeto contemplará as seguintes funcionalidades:
 - Cadastrar usuário;
 - Autenticar usuário;
 - Cadastrar fornecedor;
 - CRUD produto;
 - Ressuprimento nacional de produtos;
 - Ressuprimento internacional de produtos;
 - Registrar pagamento de ressuprimento;
 - Visualizar estado do estoque;
 - Realizar a venda de produtos;
 - Calcular giro de estoque;
 - Calcular estoque de segurança;
 - Registrar e listar produtos inativos;
 - Registrar e listar produtos em ponto de reposição;
 - Registrar a posição diária dos produtos;

Restrições

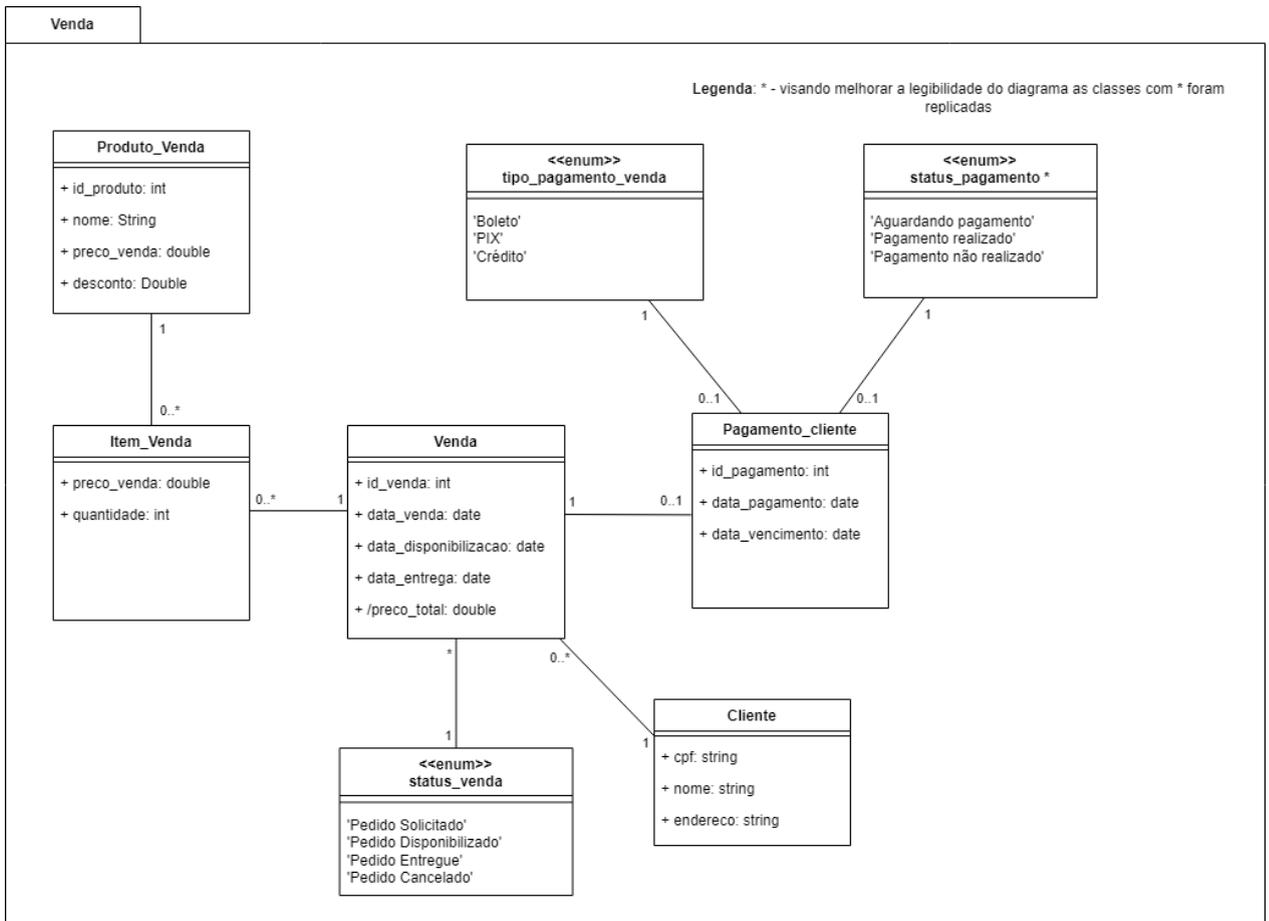
- A aplicação terá suporte apenas ao content-type application/json
- O projeto utilizará o banco de dados MySQL
- O estoque sazonal será sugerido apenas no final de novembro e em dezembro, por conta das datas de 'black friday' e Natal que são épocas de alta demanda de computadores e peças
- A venda de produtos será realizada de forma mais simplificada se comparada à entrada de produtos de estoque. De forma que sirva apenas para contabilizar a saída de produtos e o ressuprimento seja implementado com uma maior riqueza de detalhes.
- A venda de produtos conta com as seguintes etapas:
 - Solicitação de pedido
 - Pedido disponibilizado
 - Entregue

7. Diagrama de classe do domínio do problema

- Pacote de ressuprimento



- Pacote de venda

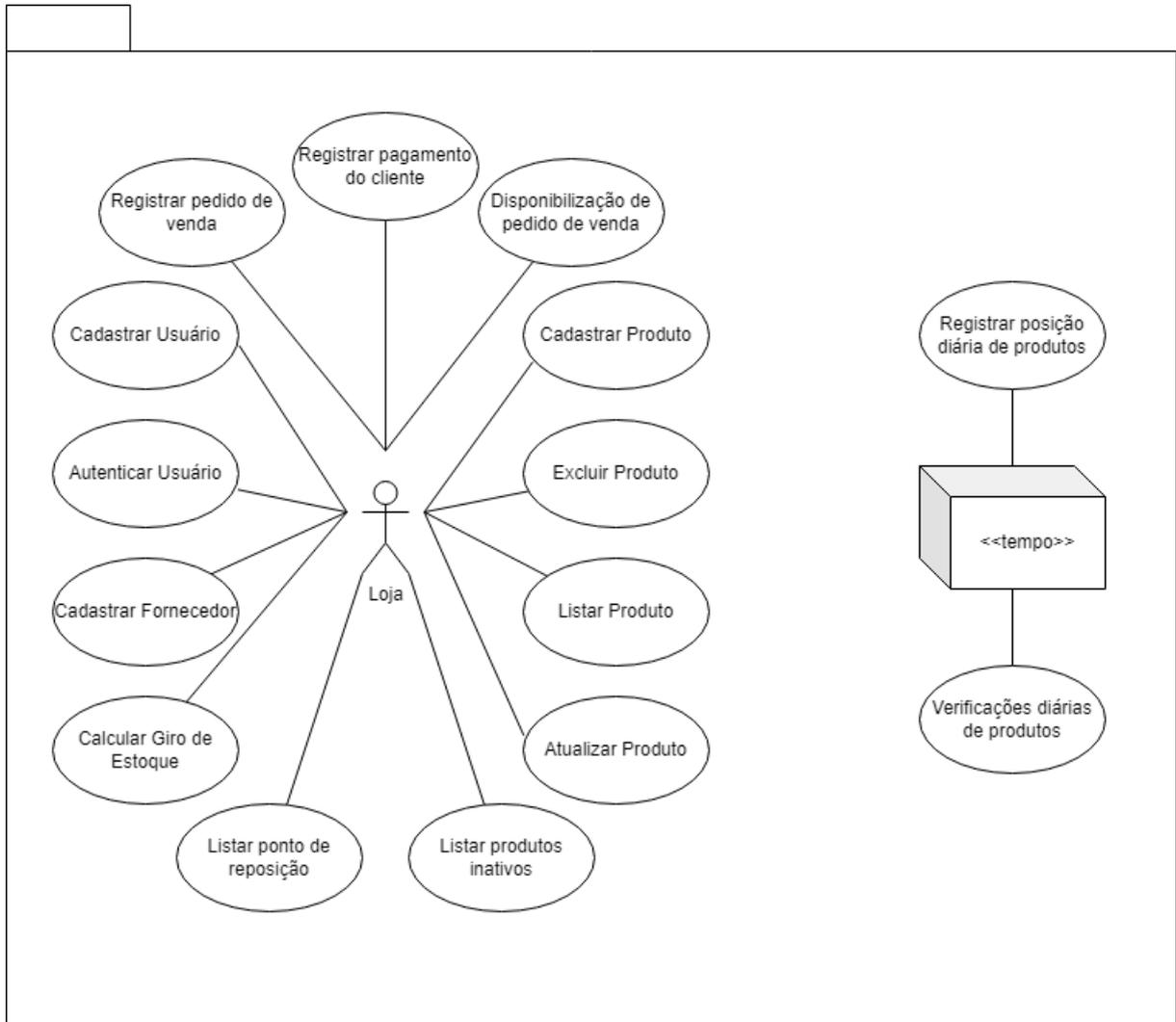


8. Diagrama dos Casos de Uso

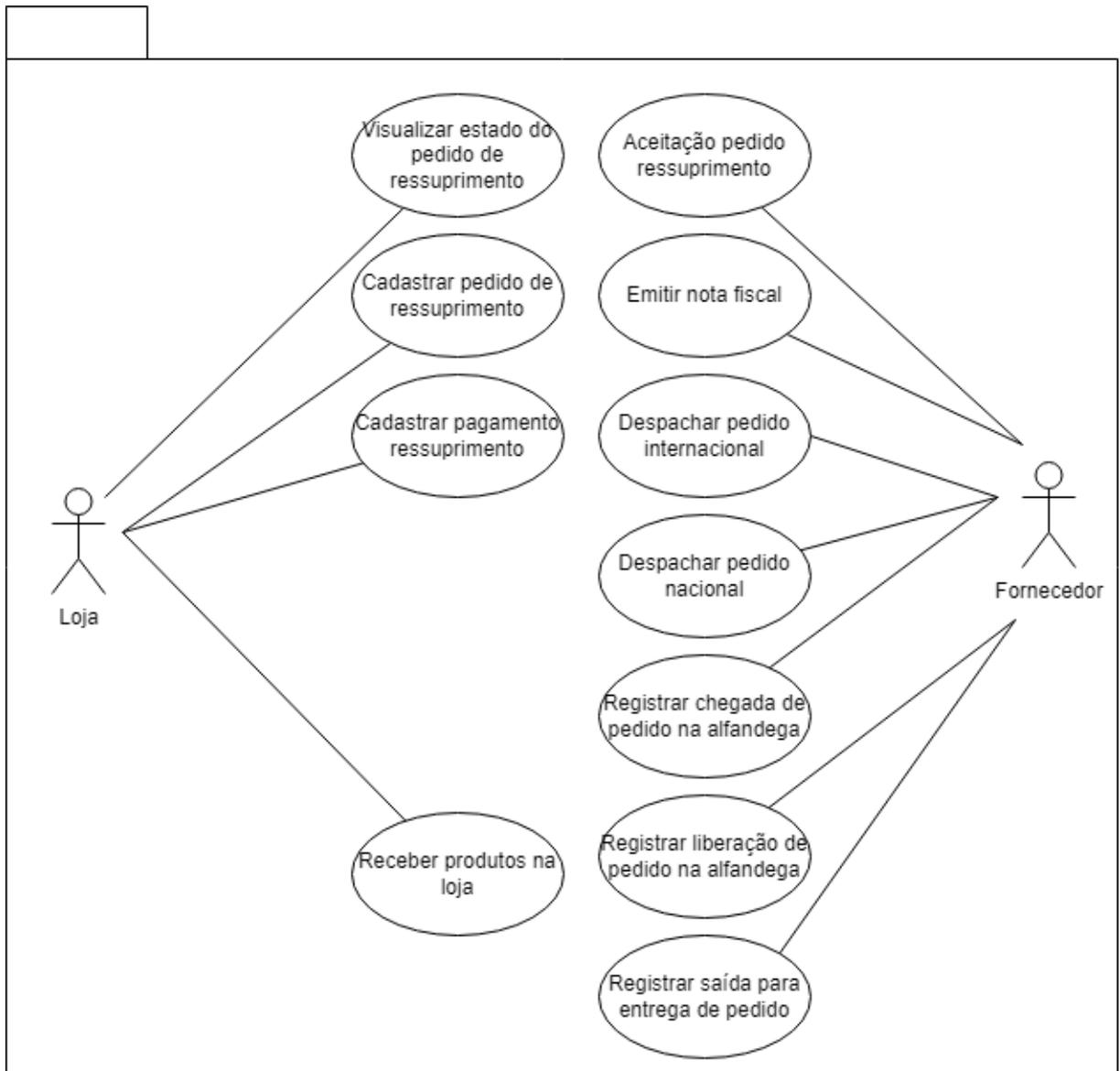
Obs¹.: o diagrama de casos de uso foi dividido em dois packages visando melhor legibilidade.

Obs².: o ator Loja é apresentado nos 2 packages visando maior clareza.

Package 1:



Package 2:



9. Descrição dos Casos de Uso

Obs.: Os casos de uso são considerados os requisitos funcionais conforme [1].

Descrição de Casos de Uso - UC01 - Cadastrar Usuário

Nome:	Cadastrar Usuário
Objetivo:	Permitir o cadastro de um Usuário no sistema.
Atores:	Loja.
Pré-condição:	Loja possui token válido
Pós-condição (cenário de sucesso):	Usuário cadastrado com sucesso na base de dados.
Pós-condição (cenário de insucesso):	Usuário não cadastrado na base de dados.
Trigger:	O end-point de cadastro de usuário da API é acionado
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário deseja cadastrar um usuário da loja 2. O usuário escolhe o verbo HTTP 'POST'[RN2][RN3][RN4][RN5][RN6][RN7] 3. O usuário preenche os dados: <ul style="list-style-type: none"> - dados = nome_usuario + senha 4. O usuário submete a requisição [RN1][E1][E2][E3] 5. O sistema realiza a requisição 6. O sistema apresenta status code de sucesso: 201 - Created 7. O sistema apresenta a mensagem: "Usuário da loja criado"
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 415 - Unsupported Media Type e apresenta a mensagem: "O content-type da requisição deve ser application/json" <p>[E2] nome_usuario com quantidade inválida de caracteres.</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Quantidade inválida de caracteres no campo nome_usuario". <p>[E3] senha não atende os requisitos de criação.</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Senha inválida".
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] nome_usuario deve ter entre 3 e 20 caracteres.</p> <p>[RN3] senha deve ter ao menos 12 caracteres.</p> <p>[RN4] senha deve conter ao menos 1 letra maiúscula</p> <p>[RN5] senha deve conter ao menos 1 letra minúscula</p> <p>[RN6] senha deve conter ao menos 1 símbolo</p> <p>[RN7] senha deve conter ao menos 1 número</p>

Descrição de Casos de Uso - UC02 - Autenticar Usuário

Nome:	Autenticar usuário.
Objetivo:	Permitir o cadastro de um produto no sistema.
Atores:	Loja.
Pré-condição:	
Pós-condição (cenário de sucesso):	Usuário autenticado com sucesso.
Pós-condição (cenário de insucesso):	Usuário não autenticado.
Trigger:	O end-point de autenticação de usuário da API é acionado
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP 'POST' 2. O usuário preenche os dados: [RN2][RN3] <ul style="list-style-type: none"> - dados = nome_usuario + senha 3. O usuário submete a requisição [RN1][E1][E2][E3] 4. O sistema realiza a requisição 5. O sistema apresenta status code de sucesso: 201 - Created 6. O sistema apresenta a mensagem: "Usuário autenticado. Token: " + token
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 415 - Unsupported Media Type e apresenta a mensagem: "O content-type da requisição deve ser application/json" <p>[E2] Usuário não está cadastrado na base de dados</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 404 - Not found e apresenta a mensagem: "Nome de usuário não encontrado na base de dados." <p>[E3] Senha incorreta</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Senha incorreta."
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] nome_usuario deve estar cadastrado na base de dados.</p> <p>[RN3] senha do usuário deve estar correta.</p>

Descrição de Casos de Uso - UC03 - Cadastrar Fornecedor

Nome:	Cadastrar Fornecedor
Objetivo:	Permitir o cadastro de um Fornecedor no sistema.
Atores:	Loja.
Pré-condição:	Loja possui token válido
Pós-condição (cenário de sucesso):	O end-point de cadastro de fornecedor da API é acionado
Pós-condição (cenário de insucesso):	Fornecedor não cadastrado na base de dados.
Trigger:	O end-point de cadastro de fornecedor da API é acionado
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário deseja cadastrar um fornecedor 2. O usuário escolhe o verbo HTTP 'POST' [RN2][RN3][RN4][RN5] 3. O usuário preenche os dados: <ul style="list-style-type: none"> - dados = registro_fornecedor + nome 4. O usuário submete a requisição [RN1][E1][E2] 5. O sistema realiza a requisição 6. O sistema apresenta status code de sucesso: 201 - Created 7. O sistema apresenta a mensagem: "fornecedor criado" + idFornecedor
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 415 - Unsupported Media Type e apresenta a mensagem: "O content-type da requisição deve ser application/json" <p>[E2] nome com quantidade inválida de caracteres.</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Quantidade inválida de caracteres no campo nome".
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] nome deve ter entre 3 e 50 caracteres.</p>

Descrição de Casos de Uso - UC04 - Cadastrar Produto

Nome:	Cadastrar Produto
Objetivo:	Permitir o cadastro de um produto no sistema.
Atores:	Loja.
Pré-condição:	Loja possui token válido
Pós-condição (cenário de sucesso):	Produto cadastrado com sucesso na base de dados.
Pós-condição (cenário de insucesso):	Produto não cadastrado na base de dados.
Trigger:	O end-point de cadastro de produto da API é acionado
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP 'POST' 2. O usuário preenche os dados: [RN2][RN3][RN4][RN5][RN6] <ul style="list-style-type: none"> - dados = SKU + EAN + peso + nome + descricao + preco_venda + sazonal + tipo_produto - tipo_produto = ['gabinete' 'placa mãe' 'CPU' 'memória' 'armazenamento' 'fonte de energia' 'GPU'] 3. O usuário submete a requisição [E1][E2][E3][E4][E5][E6][RN1][RN7] 4. O sistema realiza a requisição 5. O sistema apresenta status code de sucesso: 201 - Created 6. O sistema apresenta a mensagem: "produto criado" + idProduto
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 415 - Unsupported Media Type e apresenta a mensagem: "O content-type da requisição deve ser application/json" <p>[E2] Peso ou Preço de Venda ou Preço de Compra negativos</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "[peso preço de venda] inválido" <p>[E3] Descrição excede o número de caracteres</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Descrição excedeu o limite de caracteres" <p>[E4] EAN com quantidade de caracteres diferente de 13 ou com caractere não numérico</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Código EAN inválido. Favor checar o padrão EAN-13" <p>[E5] SKU com caractere não numérico</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Código SKU inválido" <p>[E6] SKU ou EAN já cadastrados</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Código SKU ou EAN já constam cadastrados na base "
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] O <i>peso</i>, <i>preco_venda</i> e <i>preco_compra</i> devem ser positivos</p> <p>[RN3] A <i>descricao</i> deve ter no máximo 200 caracteres</p> <p>[RN4] EAN deve conter apenas caracteres numéricos</p> <p>[RN5] EAN deve conter 13 caracteres</p> <p>[RN6] SKU pode conter apenas caracteres numéricos</p> <p>[RN7] SKU e EAN não devem constar na base de dados</p>

Descrição de Casos de Uso - UC05 - Listar Produto(s)

Nome:	Listar Produto(s)
Objetivo:	Permitir a listagem de produto(s) no sistema.
Atores:	Loja.
Pré-condição:	Loja possui token válido
Pós-condição (cenário de sucesso):	Produto(s) listado(s) com sucesso.
Pós-condição (cenário de insucesso):	Produto(s) não listado(s).
Trigger:	O end-point de produtos da API é acionado
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP 'GET' + (idProduto) [RN1][RN2][RN3] 2. O sistema submete a requisição [E1][E2] 3. O sistema apresenta o status code de 200 - Ok e apresenta a lista de produto(s): [RN4] <ul style="list-style-type: none"> - produtos = {produto} - produto = idProduto + SKU + EAN + nome + descricao + peso + preco_venda + preco_compra + quantidade + estoque_seguranca + giro + tipo_produto + inativo + sazonal + reposica
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] idProduto não consta na base de dados</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 404 - Not Found e apresenta a mensagem: "Produto não encontrado" <p>[E2] Corpo de requisição preenchido</p> <ol style="list-style-type: none"> 1. O sistema retorna o status 400 - Bad Request e apresenta a mensagem: "O método GET não espera um corpo de requisição"
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Para receber informações de um produto em específico o idProduto deve ser preenchido na URL</p> <p>[RN2] O idProduto deve constar na base de dados caso seja preenchido</p> <p>[RN3] No protocolo HTTP o método GET não é destinado a incluir um corpo de requisição</p> <p>[RN4] Os produtos com a flag excluído = true não são listados</p>

Descrição de Casos de Uso - UC06 - Atualizar Produto

Nome:	Atualizar Produto
Objetivo:	Permitir a atualização de um produto no sistema.
Atores:	Loja.
Pré-condição:	Loja possui token válido
Pós-condição (cenário de sucesso):	Dados do Produto atualizado com sucesso na base de dados.
Pós-condição (cenário de insucesso):	Dados do Produto não atualizados.
Trigger:	End-point de atualização de produto da API é acionado
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP 'PATCH' + idProduto [RN8] 2. O usuário preenche os dados que deseja atualizar: [RN2][RN3][RN4][RN5][RN6][RN7] <ul style="list-style-type: none"> - dados = SKU + EAN + nome + descricao + peso + preco_venda + sazonal + tipo_produto - tipo_produto = ['gabinete' 'placa mãe' 'CPU' 'memória' 'armazenamento' 'fonte de energia' 'GPU'] 3. O sistema submete a requisição [E1][E2][E3][E4][E5][E6][E7][RN1] 4. O sistema atualiza o cadastro do produto com os dados retorna o status code de 204 - No content
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 415 - Unsupported Media Type e apresenta a mensagem: "O content-type da requisição deve ser application/json" <p>[E2] Peso ou Preço de Venda ou Preço de Compra com valores negativos</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "[peso preço de venda] inválido" <p>[E3] Descrição excede quantidade de caracteres aceitos</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Descrição excedeu o limite de caracteres" <p>[E4] EAN com quantidade de caracteres diferente de 13 ou com caractere não numérico</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Código EAN inválido. Favor checar o padrão EAN-13" <p>[E5] SKU com caractere não numérico</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Código SKU inválido" <p>[E6] SKU ou EAN já cadastrados</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Código SKU ou EAN já constam cadastrados na base " <p>[E7] idProduto não cadastrado na base de dados</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 404 - Not Found e apresenta a mensagem: "Produto não encontrado"
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] O <i>peso</i>, <i>preco_venda</i> e <i>preco_compra</i> devem ser positivos</p> <p>[RN3] A <i>descricao</i> deve ter no máximo 200 caracteres</p> <p>[RN4] EAN deve conter apenas caracteres numéricos</p> <p>[RN5] EAN deve conter 13 caracteres</p> <p>[RN6] SKU pode conter apenas caracteres numéricos</p> <p>[RN7] SKU e EAN não devem constar na base de dados</p> <p>[RN8] O idProduto deve constar na base de dados</p>

Descrição de Casos de Uso - UC07 - Excluir Cadastro do Produto

Nome:	Excluir produto
Objetivo:	Permitir a exclusão de um produto no sistema.
Atores:	Loja.
Pré-condição:	Loja possui token válido
Pós-condição (cenário de sucesso):	Produto excluído com sucesso na base de dados.
Pós-condição (cenário de insucesso):	Produto não excluído na base de dados.
Trigger:	End-point de exclusão de produto da API é acionado
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP 'DELETE' + idProduto 2. O sistema submete a requisição [E1][E2] 3. O sistema preenche a flag de exclusão do produto e retorna o status code de 204 - No content
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] idProduto não encontrado na base de dados</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 404 - Not Found e apresenta a mensagem: "Produto não encontrado" <p>[E2] Corpo de requisição preenchido</p> <ol style="list-style-type: none"> 1. O sistema retorna o status 400 - Bad Request e apresenta a mensagem: "O método DELETE não espera um corpo de requisição"
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] O idProduto deve constar na base de dados caso seja preenchido</p> <p>[RN2] No protocolo HTTP o método DELETE não é destinado a incluir um corpo de requisição</p> <p>[RN3] O sistema trabalha com FLAG para exclusão, a fim de implementar o <i>soft delete</i>.</p>

Descrição de Casos de Uso - UC08 - Ressuprimento de Produto

Nome:	Ressuprimento de Produto
Objetivo:	Permitir realizar um pedido de ressuprimento
Atores:	Loja
Pré-condição:	Loja possui token válido
Pós-condição (cenário de sucesso):	Pedido de ressuprimento realizado.
Pós-condição (cenário de insucesso):	Pedido de ressuprimento não realizado
Trigger:	End-point de ressuprimento de produto da API é acionado
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário preenche os dados: [RN1][RN2][RN3][RN4][RN5][RN6] <ul style="list-style-type: none"> - os dados: produtos + fornecedor_id + deposito_id produtos = {produto + quantidade_pedida + preco_acordado} produto = [EAN] 2. O usuário realiza o POST. 3. O sistema realiza a requisição. [E1][E2][E3][E4][E5][E6] 4. O sistema preenche o status_pedido_ressuprimento com o valor: "Criado" 5. O sistema apresenta status code de sucesso: 201 - Created 6. O sistema apresenta a mensagem: "pedido criado" e as informações do pedido: <ul style="list-style-type: none"> - os dados: 'id_pedido', 'data_pedido'
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 415 - Unsupported Media Type e apresenta a mensagem: "O content-type da requisição deve ser application/json" <p>[E2] Quantidade pedida não é positiva</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "quantidade pedida inválida" <p>[E3] Preço acordado não é positivo</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "preço acordado inválido" <p>[E4] EAN não encontrado na base de dados</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 404 - Not found e apresenta a mensagem: "EAN não encontrado na base" <p>[E5] Fornecedor não encontrado na base de dados</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 404 - Not found e apresenta a mensagem: "Fornecedor não encontrado na base" <p>[E6] Depósito não encontrado na base de dados</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 404 - Not found e apresenta a mensagem: "Depósito não encontrado na base"
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] A <i>quantidade pedida</i> deve ser positiva</p> <p>[RN3] O <i>preço acordado</i> deve ser positivo</p> <p>[RN4] EAN deve existir na base de dados</p> <p>[RN5] Fornecedor deve estar cadastrado no sistema.</p> <p>[RN6] Depósito deve estar cadastrado no sistema.</p>

Descrição de Caso de Uso UC09 - Aceitação de Pedido de Ressuprimento

Nome:	Aceitação de Pedido de Ressuprimento
Objetivo:	Registrar a aceitação do pedido de Ressuprimento.
Atores:	Fornecedor
Pré-condição:	Fornecedor possui token válido
Pós-condição (cenário de sucesso):	Pedido de Ressuprimento Aceito
Pós-condição (cenário de insucesso):	Aceitação ou Recusa de Pedido de Ressuprimento não pode ser processado.
Trigger:	End-point de aceitação de pedido de ressuplicamento da API é acionado
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP PATCH e preenche o pedido_ressuplicamento_id [RN3] 2. O usuário preenche o dado: [RN1][RN2][RN3][RN4] - aceite = [true false] 3. O usuário realiza o PATCH. 4. O sistema realiza a requisição [E1][E2][E3][E4] 6. O sistema cadastra um registro de pagamento_ressuplicamento com os dados: - dados: data_vencimento + multa 7. O sistema cadastra status_pagamento com valor: 'Aguardando' 8. O sistema apresenta status code de sucesso: 200 - Ok 9. O sistema apresenta a mensagem: "pedido " + pedido_ressuplicamento_id + " aceite" [A1]
Fluxos Alternativos:	<p>[A1] Usuário não aceita o pedido</p> <ol style="list-style-type: none"> 1. O sistema atualiza o status_pedido_ressuplicamento para 'Não aceite' 2. O sistema apresenta status code de sucesso: 200 - Ok 3. O sistema apresenta a mensagem: "pedido " + pedido_ressuplicamento_id + " recusado"
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 415 - Unsupported Media Type e apresenta a mensagem: "O content-type da requisição deve ser application/json" <p>[E2] Valor inválido.</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Valor inválido. Valores esperados: true false". <p>[E3] Pedido Ressuplicamento não encontrado na base.</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 404 - Not found e apresenta a mensagem: "Pedido de Ressuplicamento não encontrado na base" <p>[E4] Pedido de Ressuplicamento encontra-se em um estado diferente de 'Criado'</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Status do pedido é diferente de: Criado"
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] O dado <i>aceito</i> apenas recebe valores booleanos</p> <p>[RN3] Pedido de Ressuplicamento deve constar na base</p> <p>[RN4] Pedido de Ressuplicamento deve estar no estado de: 'Criado'</p>

Descrição de Casos de Uso - UC10 - Pagamento do Ressuprimento

Nome:	Pagamento do Ressuprimento
Objetivo:	Registrar o pagamento do ressuprimento
Atores:	Loja
Pré-condição:	Loja possui token válido
Pós-condição (cenário de sucesso):	Pagamento registrado.
Pós-condição (cenário de insucesso):	Pagamento não registrado.
Trigger:	End-point de pagamento do ressuprimento da API é acionado
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP PATCH e preenche o pedido_ressuprimento_id [RN4] 2. O usuário preenche os dados: [RN1][RN2][RN5][RN6][RN7] <ul style="list-style-type: none"> - os dados: data_pagamento + tipo_pagamento_ressuprimento tipo_pagamento_ressuprimento: ['Boleto' 'Transferência Bancária' 'Cartão de Crédito'] 3. O usuário realiza o PATCH. 4. O sistema realiza a requisição. [E1][E2][E3][E4][E5][E6] 5. O sistema atualiza o status do pagamento para: 'Realizado' 6. O sistema atualiza o status_pedido_ressuprimento para: 'Em preparação' 7. O sistema apresenta status code de sucesso: 200 - Ok 8. O sistema apresenta a mensagem: "Pagamento do ressuprimento registrado!" [A1]
Fluxos Alternativos:	<p>[A1] Pagamento realizado com atraso</p> <ol style="list-style-type: none"> 1. O sistema calcula a multa total 2. O sistema apresenta a mensagem: <ul style="list-style-type: none"> "Pagamento do ressuprimento registrado! Multa total: " + multa_total [RN3] 3. O sistema apresenta atualiza o status do pagamento para: 'Realizado'
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de erro 415 - Unsupported Media Type e apresenta a mensagem: <ul style="list-style-type: none"> "O content-type da requisição deve ser application/json" <p>[E2] Data pagamento não existe ou fora do padrão DD/MM/AAAA</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Data inválida de pagamento". <p>[E3] Pedido Ressuprimento não encontrado na base.</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 404 - Not found e apresenta a mensagem: <ul style="list-style-type: none"> "Pedido de Ressuprimento não encontrado na base" <p>[E4] Pedido de Ressuprimento encontra-se em um estado diferente de 'Criado'</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Status do pedido é diferente de: Criado" <p>[E5] Pedido de Ressuprimento não está aceito</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Pedido de Ressuprimento não foi aceito pelo fornecedor" <p>[E6] Data de pagamento é anterior à data de aceitação do pedido:</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Data de pagamento não pode ser anterior a data de aceitação do pedido"
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] data_pagamento deve ser válida e estar no formato DD/MM/AAAA</p> <p>[RN3] multa_total é o valor referente ao total de multa pago, já que a multa é um valor contabilizado por dia de atraso</p> <p>[RN4] Pedido Ressuprimento deve constar na base</p> <p>[RN5] Pedido de Ressuprimento deve estar no estado de: 'Criado'</p> <p>[RN6] Pedido de Ressuprimento deve ter sido aceito</p> <p>[RN7] Data de Pagamento não deve ser anterior a data de Aceitação do Pedido</p>

Descrição de Caso de Uso UC11 - Emissão Nota Fiscal

Nome:	Emissão Nota Fiscal
Objetivo:	Emitir Nota Fiscal do Ressuprimento
Atores:	Fornecedor
Pré-condição:	Fornecedor tem token válido
Pós-condição (cenário de sucesso):	Nota fiscal emitida
Pós-condição (cenário de insucesso):	Nota fiscal não emitida
Trigger:	A solicitação de acionamento do pedido de ressuprimento na API do endpoint é iniciada.
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP POST 2. O usuário preenche os dados: [RN1][RN2][RN3][RN4][RN5][RN6][RN7][RN8] - Dados = pedido_ressuprimento_id + data_emissao + data_recebimento 3. O usuário realiza o POST. 4. O sistema realiza a requisição [E1][E2][E3][E4][E5][E6][E7][E8] 5. O sistema busca os produtos do pedido_ressuprimento informado 6. O sistema cadastra os produtos em item_nota_fiscal 7. O sistema apresenta o status code de sucesso 200 - Ok e apresenta a mensagem: "Nota emitida com sucesso"
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON 1. O sistema retorna o status code 415 - Unsupported Media Type e apresenta a mensagem: "O content-type da requisição deve ser application/json"</p> <p>[E2] Data de emissão não existe ou fora do padrão DD/MM/AAAA 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Data de emissão inválida".</p> <p>[E3] Data de recebimento não existe ou fora do padrão DD/MM/AAAA 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Data de recebimento inválida".</p> <p>[E4] Pedido Ressuprimento não encontrado na base. 1. O sistema retorna o status code de 404 - Not found e apresenta a mensagem: "Pedido de Ressuprimento não encontrado na base"</p> <p>[E5] Pedido de Ressuprimento encontra-se em um estado diferente de 'Em preparação' 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Status do pedido é diferente de: Em preparação"</p> <p>[E6] Nota fiscal já emitida 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Nota fiscal já emitida para esse pedido"</p> <p>[E7] Data de emissão é anterior a data de pagamento do pedido 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Data de emissão não pode ser anterior a data de pagamento do ressuprimento"</p> <p>[E8] Data de recebimento não é posterior a data de emissão 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Data de recebimento deve ser superior a data de emissão"</p>
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] Data de emissão deve ser válida e estar no formato DD/MM/AAAA</p> <p>[RN3] Data de recebimento deve ser válida e estar no formato DD/MM/AAAA</p> <p>[RN4] Pedido Ressuprimento deve constar na base</p> <p>[RN5] Pedido de Ressuprimento deve estar no estado de: 'Em preparação'</p> <p>[RN6] Nota deve ser emitida uma única vez por pedido</p> <p>[RN7] Data de emissão não pode ser anterior a data de pagamento do pedido</p> <p>[RN8] Data de recebimento deve ser posterior a data de emissão</p>

Descrição de Caso de Uso UC12 - Despacho de pedido de ressurgimento internacional

Nome:	Despacho de pedido de ressurgimento internacional
Objetivo:	Registrar o despacho do pedido de Ressurgimento internacional.
Atores:	Fornecedor
Pré-condição:	Fornecedor possui token válido
Pós-condição (cenário de sucesso):	Pedido despachado
Pós-condição (cenário de insucesso):	Pedido não despachado
Trigger:	Endpoint de despacho de pedidos da API é acionado.
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP PATCH e preenche o pedido_ressurgimento_id [RN2] 2. O usuário preenche os dados: [RN1][RN3][RN4][RN5][RN6][RN7][RN8][RN9][RN10] <ul style="list-style-type: none"> - Dados = data_despacho + alfandega_internacional_id + transportadora_internacional_id + frete_internacional + (previsao_chegada) 3. O usuário realiza o PATCH. 4. O sistema realiza a requisição [E1][E2][E3][E4][E5][E6][E7][E8][E9][E10] 5. O sistema atualiza o dado status_pedido_ressurgimento para: <ul style="list-style-type: none"> "Despachado para alfândega internacional" 6. O sistema atualiza o dado tipo_ressurgimento para: <ul style="list-style-type: none"> "Internacional" 7. O sistema apresenta status code de sucesso: 200 - Ok e apresenta a mensagem: <ul style="list-style-type: none"> "pedido " + pedido_ressurgimento_id + " despachado para alfândega"
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 415 - Unsupported Media Type e apresenta a mensagem: <ul style="list-style-type: none"> "O content-type da requisição deve ser application/json" <p>[E2] Pedido Ressurgimento não encontrado na base.</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 404 - Not found e apresenta a mensagem: <ul style="list-style-type: none"> "Pedido de Ressurgimento não encontrado na base" <p>[E3] Data de despacho não existe ou fora do padrão DD/MM/AAAA</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Data inválida de despacho". <p>[E4] Data de previsão não existe ou fora do padrão DD/MM/AAAA</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Data inválida de previsão de chegada". <p>[E5] Alfândega Internacional não existe na base dados</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 404 - Not found e apresenta a mensagem: <ul style="list-style-type: none"> "Alfândega não cadastrada na base de dados" <p>[E6] Transportadora Internacional não existe na base de dados</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 404 - Not found e apresenta a mensagem: <ul style="list-style-type: none"> "Transportadora internacional não cadastrada na base de dados" <p>[E7] frete internacional inválido:</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Frete internacional inválido. Valor do frete deve ser numérico e maior ou igual a 0" <p>[E8] Transportadora não entrega na alfândega internacional</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "A transportadora não realiza transporte para a alfândega" <p>[E9] Pedido de Ressurgimento encontra-se em um estado diferente de 'Em preparação'</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Status do pedido é diferente de: Em preparação" <p>[E10] Data de despacho é anterior a data de emissão da nota fiscal</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Data de despacho não deve ser anterior a data de emissão da nota fiscal"
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] Pedido Ressurgimento deve constar na base</p> <p>[RN3] Data de despacho deve ser válido e estar no formato DD/MM/AAAA</p>

	<p>[RN4] Previsão de chegada deve ser válido e estar no formato DD/MM/AAAA</p> <p>[RN5] Alfândega Internacional deve existir da base de dados</p> <p>[RN6] Transportadora Internacional deve existir na base de dados</p> <p>[RN7] Frete internacional deve ser numérico e maior ou igual a 0</p> <p>[RN8] A transportadora internacional deve entregar na alfândega internacional</p> <p>[RN9] Pedido de Ressuprimento deve estar no estado de: 'Em preparação'</p> <p>[RN10] Data de despacho não pode ser anterior a data de emissão da nota fiscal</p>
--	--

Descrição de Caso de Uso UC13 - Despacho de pedido de ressurgimento nacional

Nome:	Despacho de pedido de ressurgimento nacional
Objetivo:	Registrar o despacho do pedido de Ressurgimento nacional.
Atores:	Fornecedor
Pré-condição:	Fornecedor possui token válido
Pós-condição (cenário de sucesso):	Pedido despachado
Pós-condição (cenário de insucesso):	Pedido não despachado
Trigger:	Endpoint de despacho de pedido nacional da API é acionado.
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP PATCH e preenche o pedido_ressurgimento_id [RN2] 1. O usuário preenche os dados: [RN1][RN3][RN4][RN5][RN6][RN7][RN8] <ul style="list-style-type: none"> - Dados = data_despacho + transportadora_local_id + frete_local + (previsao_chegada) 2. O usuário realiza o PATCH. 3. O sistema realiza a requisição [E1][E2][E3][E4][E5][E6][E7][E8] 4. O sistema atualiza o dado status_pedido_ressurgimento para: <ul style="list-style-type: none"> “Despachado para a transportadora local” 5. O sistema atualiza o dado tipo_ressurgimento para: <ul style="list-style-type: none"> “Nacional” 6. O sistema apresenta status code de sucesso: 200 - Ok e apresenta a mensagem: <ul style="list-style-type: none"> “pedido ” + pedido_ressurgimento_id + “ Despachado para transportadora local”
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 415 - Unsupported Media Type e apresenta a mensagem: <ul style="list-style-type: none"> "O content-type da requisição deve ser application/json" <p>[E2] Pedido Ressurgimento não encontrado na base.</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 404 - Not found e apresenta a mensagem: <ul style="list-style-type: none"> “Pedido de Ressurgimento não encontrado na base” <p>[E3] Data de despacho não existe ou fora do padrão DD/MM/AAAA</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> “Data inválida de despacho”. <p>[E4] Data de previsão não existe ou fora do padrão DD/MM/AAAA</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> “Data inválida de previsão de chegada”. <p>[E5] Transportadora local não existe na base de dados</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 404 - Not found e apresenta a mensagem: <ul style="list-style-type: none"> "Transportadora local não cadastrada na base de dados" <p>[E6] frete local inválido:</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Frete local inválido. Valor do frete deve ser numérico e maior ou igual a 0" <p>[E7] Pedido de Ressurgimento encontra-se em um estado diferente de 'Em preparação'</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Status do pedido é diferente de: Em preparação" <p>[E8] Data de despacho é anterior a data de emissão da nota fiscal</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> “Data de despacho não deve ser anterior a data de emissão da nota fiscal”
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] Pedido Ressurgimento deve constar na base</p> <p>[RN3] Data de despacho deve ser válido e estar no formato DD/MM/AAAA</p> <p>[RN4] Previsão de chegada deve ser válido e estar no formato DD/MM/AAAA</p> <p>[RN5] Transportadora Local deve existir na base de dados</p> <p>[RN6] Frete local deve ser numérico e maior ou igual a 0</p> <p>[RN7] Pedido de Ressurgimento deve estar no estado de: 'Em preparação'</p> <p>[RN8] Data de despacho não pode ser anterior a data de emissão da nota fiscal</p>

Descrição de Caso de Uso UC14 - Chegada do pedido de ressurgimento na alfândega

Nome:	Chegada do pedido de ressurgimento na alfândega.
Objetivo:	Registrar a chegada do pedido de ressurgimento na alfândega
Atores:	Fornecedor
Pré-condição:	Fornecedor tem token válido
Pós-condição (cenário de sucesso):	Chegada do pedido registrada
Pós-condição (cenário de insucesso):	Chegada do pedido não registrada
Trigger:	Endpoint de chegada de pedido de ressurgimento na alfândega da API é acionado.
Fluxo Principal:	<ol style="list-style-type: none"> O usuário escolhe o verbo HTTP PATCH e preenche o pedido_ressurgimento_id [RN2] O usuário preenche os dados: [RN1][RN3][RN4][RN6] [A1] <ul style="list-style-type: none"> Dados = chegada_alfandega_internacional O sistema realiza a requisição [E1][E2][E3][E4][E6] O sistema atualiza o status_pedido_ressurgimento para 'Chegada em alfândega internacional' [A1] O sistema apresenta o status code de sucesso 200 e apresenta a mensagem: <ul style="list-style-type: none"> "Pedido " + pedido_ressurgimento_id + " chegou na alfandega internacional"
Fluxos Alternativos:	<p>[A1] Pedido estava em direção a alfândega nacional</p> <ol style="list-style-type: none"> O usuário preenche os dados: [RN1][RN3][RN5][RN7] <ul style="list-style-type: none"> Dados = chegada_alfandega_nacional. O sistema realiza a requisição [E1][E2][E3][E5][E7] O sistema atualiza o status_pedido para 'Chegada em alfândega nacional' O sistema apresenta o status code de sucesso 200 e apresenta a mensagem: <ul style="list-style-type: none"> "Pedido " + pedido_ressurgimento_id + " chegou na alfandega nacional"
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> O sistema retorna o status code 415 - Unsupported Media Type e apresenta a mensagem: <ul style="list-style-type: none"> "O content-type da requisição deve ser application/json" <p>[E2] Pedido Ressurgimento não encontrado na base.</p> <ol style="list-style-type: none"> O sistema retorna o status code de 404 - Not found e apresenta a mensagem: <ul style="list-style-type: none"> "Pedido de Ressurgimento não encontrado na base" <p>[E3] Data inválida de chegada_alfandega_internacional ou chegada_alfandega_nacional</p> <ol style="list-style-type: none"> O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Data de chegada_alfandega é inválida". <p>[E4] Pedido de Ressurgimento encontra-se em um estado diferente de 'Despachado para alfândega internacional'</p> <ol style="list-style-type: none"> O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Status do pedido é diferente de: Despachado para alfândega internacional" <p>[E5] Pedido de Ressurgimento encontra-se em um estado diferente de 'Despachado para alfândega internacional'</p> <ol style="list-style-type: none"> O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Status do pedido é diferente de: Liberado pela alfândega internacional" <p>[E6] Data de chegada é anterior a data de despacho do pedido</p> <ol style="list-style-type: none"> O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Data de chegada na alfândega internacional não pode ser anterior a data de despacho" <p>[E7] Data de chegada é anterior a data de liberação pela alfândega internacional</p> <ol style="list-style-type: none"> O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: <ul style="list-style-type: none"> "Data de chegada na alfândega nacional não pode ser anterior a data de liberação da alfândega internacional"
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] Pedido de ressurgimento deve existir na base de dados</p> <p>[RN3] chegada_alfandega_internacional e chegada_alfandega_nacional deve ser válida e estar no formato DD/MM/AAAA</p> <p>[RN4] Para pedido em direção a alfândega internacional, o estado do pedido deve ser: <ul style="list-style-type: none"> 'Despachado para alfândega internacional' </p> <p>[RN5] Para pedido em direção a alfândega nacional, estado do pedido deve ser: <ul style="list-style-type: none"> 'Liberado pela alfândega internacional' </p> <p>[RN6] Para pedido em direção a alfândega internacional, a data de chegada não pode ser anterior a data de despacho do pedido</p>

	[RN7] Para pedido em direção a alfândega nacional, a data de chegada não pode ser anterior a data de liberação pela alfândega internacional
--	---

Descrição de Caso de Uso UC15 - Liberação do pedido de ressurgimento da alfândega

Nome:	Liberação do pedido de ressurgimento na alfândega.
Objetivo:	Registrar a chegada do pedido de ressurgimento na alfândega
Atores:	Fornecedor
Pré-condição:	Fornecedor
Pós-condição (cenário de sucesso):	Chegada do pedido registrada
Pós-condição (cenário de insucesso):	Chegada do pedido não registrada
Trigger:	Endpoint de liberação do pedido de ressurgimento da API é acionado.
Fluxo Principal:	<ol style="list-style-type: none"> O usuário escolhe o verbo HTTP PATCH e preenche o pedido_ressurgimento_id [RN2] O usuário preenche os dados: [A1][RN1][RN2][RN3][RN4][RN6][RN7][RN8] <ul style="list-style-type: none"> Dados = liberacao_alfandega_internacional + alfandega_nacional_id O sistema realiza a requisição [E1][E2][E3][E4][E6][E7][E8] O sistema atualiza o status_pedido para 'Liberada pela alfândega internacional' O sistema apresenta o status code de sucesso 200 e apresenta a mensagem: "Pedido " + pedido_ressurgimento_id + " liberado pela alfandega internacional"
Fluxos Alternativos:	<p>[A1] Pedido estava em alfândega nacional</p> <ol style="list-style-type: none"> O usuário preenche os dados: [RN1][RN2][RN3][RN5][RN9] <ul style="list-style-type: none"> Dados = liberacao_alfandega_nacional + O sistema realiza a requisição [E1][E2][E3][E5][E9] O sistema atualiza o status_pedido para 'Liberada pela alfândega nacional' O sistema apresenta o status code de sucesso 200 e apresenta a mensagem: "Pedido " + pedido_ressurgimento_id + " liberado pela alfandega nacional"
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> O sistema retorna o status code 415 - Unsupported Media Type e apresenta a mensagem: "O content-type da requisição deve ser application/json" <p>[E2] Pedido Ressurgimento não encontrado na base.</p> <ol style="list-style-type: none"> O sistema retorna o status code de 404 - Not found e apresenta a mensagem: "Pedido de Ressurgimento não encontrado na base" <p>[E3] Data inválida de liberacao_alfandega_internacional ou liberacao_alfandega_nacional</p> <ol style="list-style-type: none"> O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Data de liberacao_alfandega é inválida". <p>[E4] Pedido de Ressurgimento encontra-se em um estado diferente de 'Chegada em alfândega internacional'</p> <ol style="list-style-type: none"> O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "status_pedido_ressurgimento com problema de escrita" <p>[E5] Pedido de Ressurgimento encontra-se em um estado diferente de 'Chegada em alfândega nacional'</p> <ol style="list-style-type: none"> O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "status_pedido_ressurgimento com problema de escrita" <p>[E6] Alfândega Nacional não existe na base dados</p> <ol style="list-style-type: none"> O sistema retorna o status code 404 - Not found e apresenta a mensagem: "Alfândega nacional não cadastrada na base de dados" <p>[E7] Transportadora não entrega na alfândega nacional</p> <ol style="list-style-type: none"> O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "A transportadora não realiza transporte para a alfândega" <p>[E8] Data de liberação é anterior a data de chegada na alfândega internacional</p> <ol style="list-style-type: none"> O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Data de liberação por alfândega internacional não pode ser anterior a data de chegada na alfândega internacional" <p>[E9] Data de liberação é anterior a data de chegada na alfândega nacional</p> <ol style="list-style-type: none"> O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Data de liberação por alfândega nacional não pode ser anterior a data de chegada na alfândega nacional"
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] Pedido de ressurgimento deve existir na base de dados</p>

	<p>[RN3] Data de liberação da alfândega internacional e nacional devem ser válida e estar no formato DD/MM/AAAA</p> <p>[RN4] Para pedido em alfândega internacional, o estado do pedido deve ser: 'Chegada em alfândega internacional'</p> <p>[RN5] Para pedido em alfândega nacional, o estado do pedido deve ser: 'Chegada em alfândega nacional'</p> <p>[RN6] Alfândega Nacional deve existir da base de dados</p> <p>[RN7] A transportadora internacional deve entregar na alfândega nacional</p> <p>[RN8] Para pedido em alfândega internacional, a data de liberação não pode ser anterior a data de chegada na alfândega internacional</p> <p>[RN9] Para pedido em alfândega nacional, a data de liberação não pode ser anterior a data de chegada na alfândega nacional</p>
--	---

Descrição de Caso de Uso UC16 - Saída para entrega de pedido de ressurgimento

Nome:	Saída para entrega de pedido de ressurgimento
Objetivo:	Registrar a saída para entrega do pedido de ressurgimento
Atores:	Fornecedor
Pré-condição:	Fornecedor tem token válido
Pós-condição (cenário de sucesso):	Saída para entrega do pedido de ressurgimento realizada
Pós-condição (cenário de insucesso):	Saída para entrega do pedido de ressurgimento não realizada
Trigger:	Endpoint de saída para entrega de pedido de ressurgimento da API é acionado.
Fluxo Principal:	<ol style="list-style-type: none"> O usuário escolhe o verbo HTTP PATCH + pedido_ressurgimento_id [RN2] O usuário preenche os dados: [A1][RN1][RN3][RN4][RN5][RN6][RN7][RN9] <ul style="list-style-type: none"> Dados = saída_nacional + frete_local + (previsao_chegada) + transportadora_local_id O sistema realiza a requisição [E1][E2][E3][E4][E5][E6][E7][E8][E9] O sistema atualiza o status_pedido para 'Em rota de entrega' O sistema apresenta o status code de sucesso 200 e apresenta a mensagem: "Pedido " + pedido_ressurgimento_id + " saiu para entrega ao depósito"
Fluxos Alternativos:	<p>[A1] O Pedido de ressurgimento é nacional</p> <ol style="list-style-type: none"> O usuário preenche os dados: [A1][RN1][RN4][RN8][RN10] <ul style="list-style-type: none"> Dados = saída_nacional O sistema realiza a requisição [E1][E2][E8][E10] O sistema atualiza o status_pedido para 'Em rota de entrega' O sistema apresenta o status code de sucesso 200 e apresenta a mensagem: "Pedido " + pedido_ressurgimento_id + " saiu para entrega ao depósito"
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> O sistema retorna o status code 415 - Unsupported Media Type e apresenta a mensagem: "O content-type da requisição deve ser application/json" <p>[E2] Pedido Ressurgimento não encontrado na base.</p> <ol style="list-style-type: none"> O sistema retorna o status code de 404 - Not found e apresenta a mensagem: "Pedido de Ressurgimento não encontrado na base" <p>[E3] Frete menor que zero</p> <ol style="list-style-type: none"> O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "frete_local inválido". <p>[E4] Data inválida de saída nacional</p> <ol style="list-style-type: none"> O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Data de saída inválida". <p>[E5] Data inválida de previsão chegada</p> <ol style="list-style-type: none"> O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Data de previsão de chegada inválida". <p>[E6] Transportadora Local não encontrada</p> <ol style="list-style-type: none"> O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "idTransportadoraNacional não encontrada na base de dados". <p>[E7] Pedido de Ressurgimento Internacional encontra-se em um estado diferente de 'Liberado pela alfândega nacional'</p> <ol style="list-style-type: none"> O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Status do pedido é diferente de: Liberado pela alfândega nacional" <p>[E8] Pedido de Ressurgimento Nacional encontra-se em um estado diferente de 'Despachado para transportadora local'</p> <ol style="list-style-type: none"> O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Status do pedido é diferente de: Despachado para transportadora local" <p>[E9] Data de saída nacional é anterior a data de liberação pela alfândega nacional</p> <ol style="list-style-type: none"> O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Data de saída nacional não pode ser anterior a data de liberação da alfândega nacional" <p>[E10] Data de saída nacional é anterior a data de despacho do pedido</p> <ol style="list-style-type: none"> O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Data de saída nacional não pode ser anterior a data de despacho nacional"
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] Pedido de ressurgimento deve existir na base de dados</p>

	[RN3] Frete local deve ser maior ou igual a zero [RN4] Data de saída nacional deve ser uma data válida e estar no formato DD/MM/AAAA [RN5] Data de previsão chegada deve ser uma data válida e estar no formato DD/MM/AAAA [RN6] Transportadora local deve existir na base de dados [RN7] Para pedido internacional, o estado do pedido deve ser: 'Liberado pela alfândega nacional' [RN8] Para pedido nacional o estado do pedido deve ser: 'Despachado para transportadora local' [RN9] Para pedido internacional, a data de saída nacional não pode ser anterior a data de liberação pela alfândega nacional [RN10] Para pedido nacional, a data de saída nacional não pode ser anterior a data de despacho do pedido
--	---

Descrição de Caso de Uso UC17 - Recebimento de Produtos na loja

Nome:	Recebimento de Produtos na loja
Objetivo:	Registrar o recebimento de produtos na loja
Atores:	Loja
Pré-condição:	Loja tem token válido
Pós-condição (cenário de sucesso):	Produtos entregues na loja
Pós-condição (cenário de insucesso):	Produtos não entregues na loja
Trigger:	Endpoint recebimento de produtos na loja da API é acionado.
Fluxo Principal:	<ol style="list-style-type: none"> O usuário escolhe o verbo HTTP PATCH + pedido_ressuprimento_id [RN2] O usuário preenche os dados: [RN1][RN3][RN4] - Dados = data_chegada O sistema realiza a requisição[E1][E2][E3][E4] O sistema atualiza o status_pedido para 'Pedido entregue' O sistema apresenta o status code de sucesso 204 - No Content
Fluxos Alternativos:	
Fluxos de Exceção	[E1] Requisição não foi realizada por um JSON 1. O sistema retorna o status code 415 - Unsupported Media Type e apresenta a mensagem: "O content-type da requisição deve ser application/json" [E2] Pedido Ressuprimento não encontrado na base. 1. O sistema retorna o status code de 404 - Not found e apresenta a mensagem: "Pedido de Ressuprimento não encontrado na base" [E3] Data inválida de chegada 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Data inválida de chegada". [E4] Pedido de Ressuprimento encontra-se em um estado diferente de 'Em rota de entrega' 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Status do pedido é diferente de: Em rota de entrega" [E5] Data de chegada é anterior a data de saída nacional 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Data de chegada não pode ser anterior a data de saída nacional"
Pontos de Extensão:	
Regras de Negócio	[RN1] Requisição será realizada através de um JSON [RN2] Pedido de ressuplicamento deve existir na base de dados [RN3] Data de chegada deve ser válida e estar no formato DD/MM/AAAA [RN4] Pedido de Ressuplicamento deve estar no estado de: 'Em rota de entrega' [RN5] A data de chegada não pode ser anterior a data de saída nacional

Descrição de Casos de Uso - UC18 - Estado do pedido ressuprimento

Nome:	Estado do pedido ressuprimento
Objetivo:	Visualizar estado do pedido ressuprimento
Atores:	Loja.
Pré-condição:	Loja possui token válido
Pós-condição (cenário de sucesso):	Estado do pedido de ressuprimento apresentado.
Pós-condição (cenário de insucesso):	Estado do pedido de ressuprimento não apresentado.
Trigger:	O end-point de estado do pedido de ressuprimento da API é acionada
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP 'GET' + idPedido [RN1][RN2][RN3] 2. O sistema submete a requisição [E1][E2] 3. O sistema apresenta o status code de 200 - Ok e apresenta os dados do pedido: <ul style="list-style-type: none"> - pedido = id_pedido, frete_internacional, frete_local, data_pedido, aceito, data_aceitacao, previsao_chegada, data_despacho, chegada_alfandega_int, liberacao_alfandega_int, chegada_alfandega_nac, liberacao_alfandega_nac, data_chegada, status_pedido_ressuprimento, status_pagamento
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] idPedido não consta na base de dados</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 404 - Not Found e apresenta a mensagem: "Pedido de ressuprimento não encontrado" <p>[E2] Corpo de requisição preenchido</p> <ol style="list-style-type: none"> 1. O sistema retorna o status 400 - Bad Request e apresenta a mensagem: "O método GET não espera um corpo de requisição"
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Para receber informações de um produto em específico o idPedido deve ser preenchido na URL</p> <p>[RN2] O idPedido deve constar na base de dados caso seja preenchido</p> <p>[RN3] No protocolo HTTP o método GET não é destinado a incluir um corpo de requisição</p>

Descrição de Casos de Uso - UC19 - Pedido de venda

Nome:	Pedido de venda
Objetivo:	Permitir realizar um pedido de venda
Atores:	Loja
Pré-condição:	Loja possui token válido
Pós-condição (cenário de sucesso):	Pedido de venda realizado.
Pós-condição (cenário de insucesso):	Pedido de venda não realizado
Trigger:	End-point de pedido de venda da API é acionado
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário preenche os dados: [RN2][RN3][RN4] <ul style="list-style-type: none"> - os dados: produtos + CPF_Cliente + tipo_pagamento_venda produtos = {id_produto + quantidade} tipo_pagamento_venda = ['Boleto' 'PIX' 'Credito'] 2. O usuário realiza o POST. 3. O sistema realiza a requisição. [RN1][E1][E2][E3] 4. O sistema apresenta status code de sucesso: 201 - Created 5. O sistema apresenta a mensagem: "pedido de venda criado" e as informações do pedido: <ul style="list-style-type: none"> - os dados: 'id_venda', 'data_venda' 6. O sistema cria um registro de pagamento para o pedido de venda. 7. O sistema atualiza o status da venda para: 'Pedido solicitado' 8. O sistema gera uma data de vencimento para o pagamento 9. O sistema atualiza o status de pagamento para 'Aguardando pagamento'
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 415 - Unsupported Media Type e apresenta a mensagem: "O content-type da requisição deve ser application/json" <p>[E2] Produto não encontrado na base de dados</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Produto não encontrado na base de dados" <p>[E3] Produto com quantidade menor ou igual a 0</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code 400 - Bad Request e apresenta a mensagem: "Quantidade inválida"
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] Produto deve estar cadastrado na base de dados</p> <p>[RN3] Quantidade deve ser maior que zero</p> <p>[RN4] A data de vencimento do pagamento é de 1 dia útil.</p>

Descrição de Casos de Uso - UC20 - Pagamento do cliente

Nome:	Pagamento do cliente
Objetivo:	Registrar o pagamento do cliente
Atores:	Loja
Pré-condição:	Loja possui token válido
Pós-condição (cenário de sucesso):	Pagamento registrado.
Pós-condição (cenário de insucesso):	Pagamento não registrado.
Trigger:	End-point de pagamento de cliente da API é acionado
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP PATCH + id_venda 2. O usuário preenche os dados: [RN2][RN3] - os dados: data_pagamento 3. O usuário realiza o PATCH. 4. O sistema realiza a requisição. [E1][RN1][RN4] 5. O sistema apresenta status code de sucesso: 200 - Ok 6. O sistema apresenta a mensagem: "Pagamento registrado!" 7. O sistema apresenta atualiza o status do pagamento para: 'Pagamento realizado'
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de erro 415 - Unsupported Media Type e apresenta a mensagem: "O content-type da requisição deve ser application/json" <p>[E2] data_pagamento é superior à data_vencimento</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de erro 404 - Bad Request e apresenta a mensagem: "Pagamento não pode ser realizado. Data de vencimento atingido" <p>[E3] Data inválida de pagamento</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Data inválida de pagamento".
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] data_pagamento deve ser anterior à data vencimento</p> <p>[RN3] data_pagamento deve ser válida e estar no formato DD/MM/AAAA</p>

Descrição de Casos de Uso - UC21 - Disponibilização de pedido

Nome:	Disponibilização de pedido
Objetivo:	Registrar a disponibilização de pedido
Atores:	Loja
Pré-condição:	Loja possui token válido
Pós-condição (cenário de sucesso):	Pedido disponibilizado / entregue.
Pós-condição (cenário de insucesso):	Pedido não disponibilizado / entregue.
Trigger:	End-point de disponibilização de pedido da API é acionado
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP PATCH + id_venda 2. O usuário preenche os dados: [A1][RN2] <ul style="list-style-type: none"> - os dados: data_disponibilizacao 3. O usuário realiza o PATCH. 4. O sistema realiza a requisição. [E1][RN1] 5. O sistema apresenta status code de sucesso: 200 - Ok 6. O sistema apresenta a mensagem: "Pedido disponibilizado " + id + " disponibilizado" 7. O sistema altera o status de venda para 'Pedido Disponibilizado'
Fluxos Alternativos:	<p>[A1] O usuário deseja registrar a entrega do produto</p> <ol style="list-style-type: none"> 1. O usuário preenche os dados: [RN3] <ul style="list-style-type: none"> - os dados: data_entrega 2. O usuário realiza o PATCH. 3. O sistema realiza a requisição. [E1][RN1] 4. O sistema apresenta status code de sucesso: 200 - Ok 5. O sistema apresenta a mensagem: "Pedido " + id + " entregue" 6. O sistema altera o status de venda para 'Pedido Entregue'
Fluxos de Exceção	<p>[E1] Requisição não foi realizada por um JSON</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de erro 415 - Unsupported Media Type e apresenta a mensagem: "O content-type da requisição deve ser application/json" <p>[E2] Data inválida de disponibilização</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Data inválida de disponibilização". <p>[E3] Data inválida de Entrega</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 400 - Bad Request e apresenta a mensagem: "Data inválida de Entrega".
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] Requisição será realizada através de um JSON</p> <p>[RN2] data_disponibilizacao deve ser válida e estar no formato DD/MM/AAAA</p> <p>[RN3] data_entrega deve ser válida e estar no formato DD/MM/AAAA</p>

Descrição de Casos de Uso - UC22 - Calcular giro de estoque

Nome:	Calcular giro de estoque
Objetivo:	Calcular o giro de estoque
Atores:	Loja
Pré-condição:	Loja possui token válido
Pós-condição (cenário de sucesso):	Giro de estoque calculado
Pós-condição (cenário de insucesso):	Giro de estoque não calculado
Trigger:	O end-point de cálculo de giro da API é acionada
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP 'GET' + idProduto [RN1][RN2] 2. O sistema submete a requisição [E1][E2] 3. O sistema apresenta o status code de 200 - Ok e apresenta os dados: <div style="margin-left: 40px;">Dados: total_vendas_ano + quantidade_media_ano + giro_estoque_ano + dias_ressuprimento</div> 4. O sistema atualiza o campo giro do produto em questão
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] idProduto não consta na base de dados</p> <ol style="list-style-type: none"> 1. O sistema retorna o status code de 404 - Not Found e apresenta a mensagem: "Produto não encontrado" <p>[E2] Corpo de requisição preenchido</p> <ol style="list-style-type: none"> 1. O sistema retorna o status 400 - Bad Request e apresenta a mensagem: "O método GET não espera um corpo de requisição"
Pontos de Extensão:	
Regras de Negócio	[RN1] O idProduto deve constar na base de dados [RN2] No protocolo HTTP o método GET não é destinado a incluir um corpo de requisição

Obs.: O tempo é considerado um ator conforme [1]

Descrição de Casos de Uso - UC23 - Verificações diárias

Nome:	Verificações diárias
Objetivo:	Realizar a verificação de inatividade de estoques, sazonalidade, ponto de reposição e pedidos venda não pagas
Atores:	Tempo
Pré-condição:	Existe ao menos um produto na base de dados
Pós-condição (cenário de sucesso):	Verificações realizadas com sucesso
Pós-condição (cenário de insucesso):	Verificação não pode ser realizada
Trigger:	Verificação diária do sistema
Fluxo Principal:	<ol style="list-style-type: none"> 1. O sistema submete a requisição. [RN1] 2. O sistema itera por todos os produtos. 3. O sistema atualiza o campo inativo do produto para true. [RN2][RN3] 4. O sistema calcula a demanda média diária do produto 5. O sistema calcula o estoque de segurança do produto 6. O sistema atualiza o campo estoque_seguranca no produto 7. O sistema compara a data atual com as datas sazonais [RN5] 8. O sistema atualiza o campo reposicao para true. [RN4][RN5] 9. O sistema itera por todas as vendas com o status de pagamento = 'Aguardando pagamento'. 10. O sistema verifica os pedidos de venda sem pagamento e com data de vencimento atingidas. 11. O sistema atualiza o status_pagamento das vendas para 'Pagamento não realizado' e status venda para 'Pedido cancelado'
Fluxos Alternativos:	
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] O sistema roda as verificações diariamente na hora parametrizada no sistema</p> <p>[RN2] O sistema atualiza o campo inativo para true os produtos que estiverem inativos.</p> <p>[RN3] O sistema considera o produto inativo aqueles produtos sem vendas em um intervalo de tempo parametrizado no sistema.</p> <p>[RN4] O sistema atualiza o campo reposicao para true os produtos que estiverem atingido o ponto de reposição.</p> <p>[RN5] O sistema leva em consideração a sazonalidade para definir o ponto de reposição.</p>

Descrição de Casos de Uso - UC24 - Listar ponto de reposição

Nome:	Listar ponto de reposição
Objetivo:	Listar os produtos em ponto de reposição
Atores:	Loja
Pré-condição:	
Pós-condição (cenário de sucesso):	Listagem realizada com sucesso.
Pós-condição (cenário de insucesso):	Listagem não pode ser realizada.
Trigger:	O end-point de Alertas da API é acionada
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP 'GET' [RN1] 2. O sistema submete a requisição [E1] 3. O sistema apresenta o status code de 200 - Ok e apresenta os dados: [RN2] <ul style="list-style-type: none"> - Dados = produtos - produtos = {produto} - produto = produto_id + SKU + EAN + nome + quantidade + reposicao
Fluxos Alternativos:	
Fluxos de Exceção	[E1] Corpo de requisição preenchido <ol style="list-style-type: none"> 1. O sistema retorna o status 400 - Bad Request e apresenta a mensagem: "O método GET não espera um corpo de requisição"
Pontos de Extensão:	
Regras de Negócio	[RN1] No protocolo HTTP o método GET não é destinado a incluir um corpo de requisição [RN2] O sistema apresenta produtos com o campo reposicao = true

Descrição de Casos de Uso - UC25 - Listar produtos inativos

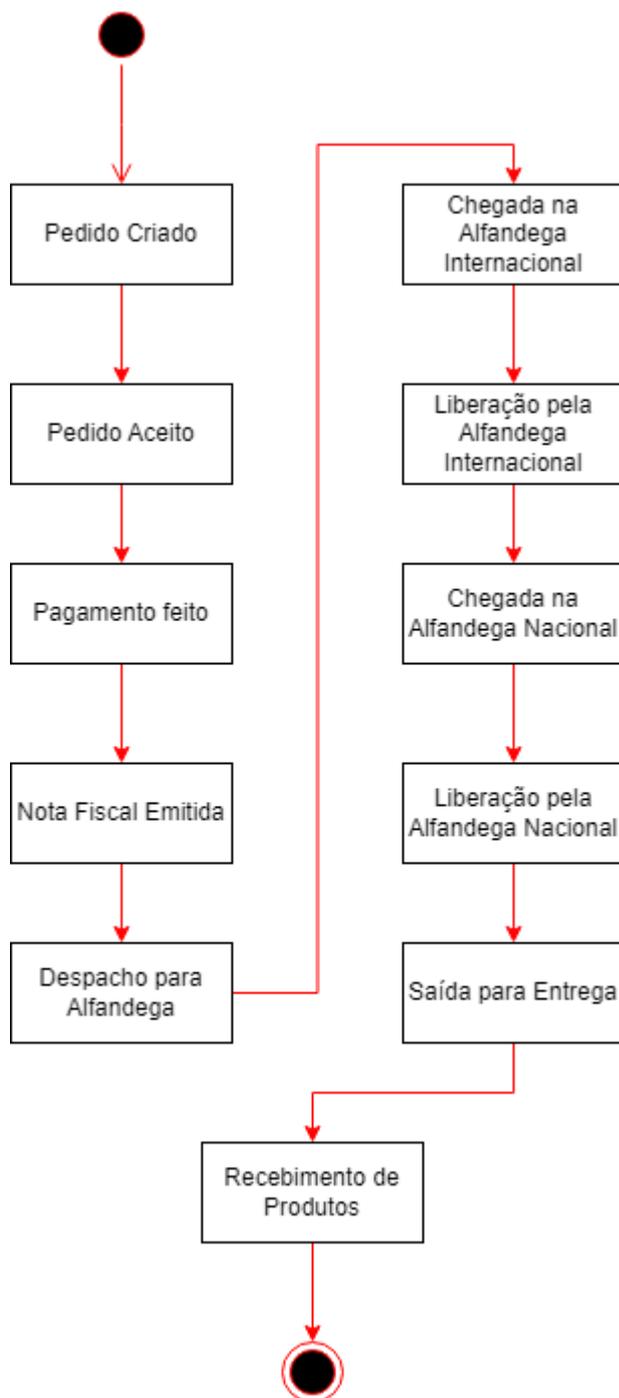
Nome:	Listar produtos inativos
Objetivo:	Listar os produtos com estado de inativo
Atores:	Loja
Pré-condição:	
Pós-condição (cenário de sucesso):	Listagem realizada com sucesso.
Pós-condição (cenário de insucesso):	Listagem não pode ser realizada.
Trigger:	O end-point de Alertas da API é acionada
Fluxo Principal:	<ol style="list-style-type: none"> 1. O usuário escolhe o verbo HTTP 'GET' [RN1] 2. O sistema submete a requisição [E1] 3. O sistema apresenta o status code de 200 - Ok e apresenta os dados: [RN2] <ul style="list-style-type: none"> - Dados = produtos - produtos = {produto} - produto = produto_id + SKU + EAN + nome + quantidade + inativo
Fluxos Alternativos:	
Fluxos de Exceção	[E1] Corpo de requisição preenchido <ol style="list-style-type: none"> 1. O sistema retorna o status 400 - Bad Request e apresenta a mensagem: "O método GET não espera um corpo de requisição"
Pontos de Extensão:	
Regras de Negócio	[RN1] No protocolo HTTP o método GET não é destinado a incluir um corpo de requisição [RN2] O sistema apresenta produtos com o campo inativo = true

Descrição de Casos de Uso - UC26 - Registrar posição diária de produtos

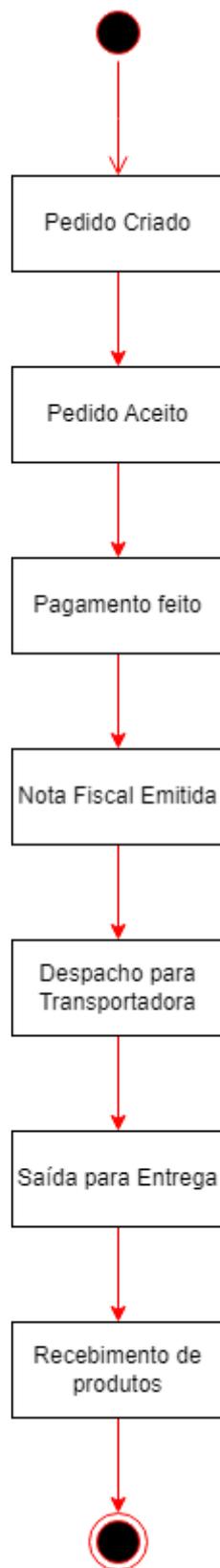
Nome:	Registrar posição diária de produtos
Objetivo:	Registrar as informações diárias dos produtos diariamente
Atores:	Tempo
Pré-condição:	
Pós-condição (cenário de sucesso):	Posição de produtos gravada com sucesso.
Pós-condição (cenário de insucesso):	A posição dos produtos não pode ser gravada.
Trigger:	Procedimento diário do sistema
Fluxo Principal:	<ol style="list-style-type: none"> 1. O sistema submete a requisição. [RN1] 2. O sistema itera por todos os produtos. 3. O sistema grava os dados: <ul style="list-style-type: none"> - Dados = produtos + data - produtos = {produto} - produto = produto_id + quantidade + preco_venda
Fluxos Alternativos:	
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	[RN1] O sistema roda o procedimento diariamente na hora parametrizada no sistema

10. Diagrama de Transição de Estados

Ressuprimento Internacional:



Ressuprimento Nacional:



11. Requisitos não funcionais

Definem as características e restrições que não estão diretamente relacionadas às funcionalidades da API, abrangendo aspectos como desempenho, segurança, usabilidade e escalabilidade, essenciais para garantir a qualidade e o sucesso do projeto.

11.1. Requisitos de Performance

Cache	A API deve implementar estratégias de cache para armazenar em memória dados que são frequentemente acessados. Ao fazer isso, busca-se minimizar a necessidade de acessar o armazenamento principal (banco de dados) repetidamente, otimizando assim o tempo de resposta e reduzindo a carga no sistema.
Paginação	A API deve implementar mecanismos de paginação em consultas de dados, dividindo grandes conjuntos em páginas menores. Essa abordagem visa melhorar a experiência do usuário ao reduzir o tempo de carregamento e garantir uma resposta rápida

11.2. Requisitos de Segurança

Login com Tokens JWT	Para garantir a segurança da API, é necessário implementar um sistema de autenticação que utiliza tokens JWT em todas as rotas do aplicativo. O uso de tokens JWT proporciona um método eficaz para gerenciar sessões de usuários, validando a identidade e autorização em todas as interações.
----------------------	---

11.3. Requisitos de Qualidade de Software

Transações	Operações que alteram o estado do banco de dados será necessário utilizar o conceito de transação para garantir a consistência dos dados. A transação é iniciada no começo da função. Caso a operação seja efetuada com sucesso ao final será realizado o commit. Em caso de erro em qualquer momento da operação a transação executa o rollback desfazendo qualquer alteração feita durante o tempo de execução da operação.
Estruturação em Camadas	A API deverá ser organizado em camadas distintas, incluindo, mas não se limitando à models, controllers, services e routes. Essa abordagem arquitetônica visa melhorar a qualidade do software, promovendo a clareza, reusabilidade, manutenção e teste do código.
Utilização de Seeders para Carga Inicial de Dados e Massa de Testes	Seeders se referem a scripts ou ferramentas que inicializam ou populam um sistema com dados iniciais, especialmente em ambientes de desenvolvimento e teste. A API deverá implementar a prática de seeders para a carga inicial de dados, possibilitando a rápida configuração de ambientes de desenvolvimento e a criação de massa de testes. Esta abordagem visa aprimorar a qualidade do software, promovendo a eficiência durante o desenvolvimento, facilitando a execução de testes e garantindo a consistência nos ambientes de teste.
Migrações	A API deverá incorporar práticas de migrations para a evolução controlada do esquema do banco de dados. Este requisito busca facilitar a aplicação de alterações, criação e modificação de tabelas de maneira estruturada, garantindo a consistência dos dados e contribuindo para uma gestão organizada do banco de dados.

12. Testes realizados

1. Cadastro de pedido de ressurgimento (Caso de Sucesso)

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:3000/pedido_ressurgimento
- Body (Request):** A JSON object with the following structure:

```
1 {
2   "fornecedorId": 1, // Deve constar da base
3   "depositoId": 1, // Deve constar na base
4   "produtos": [
5     {
6       "ean": "8888999910101", // Deve constar na base
7       "quantidade_pedido": 50, // Deve ser positivo
8       "preco_acordado": 200.00 // Deve ser positivo
9     },
10    {
11      "ean": "1111222233334",
12      "quantidade_pedido": 30,
13      "preco_acordado": 666.00
14    }
15  ]
16 }
```
- Response (Status):** 201 Created, 96 ms, 455 B
- Body (Response):** A JSON object with the following structure:

```
1 {
2   "message": "Pedido de Ressurgimento criado!",
3   "info": {
4     "id": 1,
5     "data_pedido": "2023-12-14T03:00:00.000Z",
6     "fornecedor_id": 1,
7     "deposito_id": 1,
8     "updatedAt": "2023-12-14T18:23:03.106Z",
9     "createdAt": "2023-12-14T18:23:03.106Z"
10  }
11 }
```

2. EAN não encontrado

The screenshot shows a REST client interface with the following details:

- Method:** POST
- URL:** http://localhost:3000/pedido_ressurgimento
- Body (Request):** A JSON object with the following structure:

```
1 {
2   "fornecedorId": 1, // Deve constar da base
3   "depositoId": 1, // Deve constar na base
4   "produtos": [
5     {
6       "ean": "8888999910101", // Deve constar na base
7       "quantidade_pedido": 50, // Deve ser positivo
8       "preco_acordado": 200.00 // Deve ser positivo
9     },
10    {
11      "ean": "1111222233334",
12      "quantidade_pedido": 30,
13      "preco_acordado": 666.00
14    }
15  ]
16 }
```
- Response (Status):** 404 Not Found, 91 ms, 281 B
- Body (Response):** A JSON object with the following structure:

```
1 {"error": "EAN não encontrado na base"}
```

3. Validação de Content-Type

REST client interface showing a PATCH request to `http://localhost:3000/pedido_ressuprimento/aceitar/1`. The request body is XML:

```
1 <root>
2   <aceito>true</aceito>
3 </root>
```

The response is a 415 Unsupported Media Type error:

```
1 {
2   "error": "0 content-type da requisição deve ser application/json"
3 }
```

4. Validação de data_pagamento em pagamento do ressuprimento

REST client interface showing a PATCH request to `http://localhost:3000/pedido_ressuprimento/pagamento/2`. The request body is JSON:

```
1 {
2   "data_pagamento": "32/12/2023",
3   "tipo_pagamento_ressuprimento": "Transferência Bancária"
4 }
```

The response is a 400 Bad Request error:

```
1 {
2   "error": "Data inválida de pagamento"
3 }
```

5. Pagamento do pedido de ressuprimento (Caso de Sucesso)

The screenshot shows a REST client interface for a PATCH request to `http://localhost:3000/pedido_ressuprimento/pagamento/2`. The request body is a JSON object with two fields: `data_pagamento` set to `"14/12/2023"` and `tipo_pagamento_ressuprimento` set to `"Transferência Bancária"`. The response body is a JSON object with a single field: `message` set to `"Pagamento do ressuprimento registrado!"`.

```
1 {
2   "data_pagamento": "14/12/2023", // Não pode ser anterior a data de aceitação do pedido
3   "tipo_pagamento_ressuprimento": "Transferência Bancária" // Deve fazer parte do domínio ENUM
4 }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON ↕

```
1 {
2   "message": "Pagamento do ressuprimento registrado!"
3 }
```

6. Pagamento com multa (Caso de Sucesso)

The screenshot shows a REST client interface for a PATCH request to `http://localhost:3000/pedido_ressuprimento/pagamento/4`. The request body is a JSON object with two fields: `data_pagamento` set to `"30/12/2023"` and `tipo_pagamento_ressuprimento` set to `"Transferência Bancária"`. The response body is a JSON object with a single field: `message` set to `"Pagamento do ressuprimento registrado! Multa total: 11"`.

```
1 {
2   "data_pagamento": "30/12/2023", // Não pode ser anterior a data de aceitação do pedido
3   "tipo_pagamento_ressuprimento": "Transferência Bancária" // Deve fazer parte do domínio ENUM
4 }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON ↕

```
1 {
2   "message": "Pagamento do ressuprimento registrado! Multa total: 11"
3 }
```

7. Emissão de nota fiscal (Caso de Sucesso)

POST http://localhost:3000/nota_fiscal

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▾

```
1 {
2   ... "pedido_ressuprimento_id": 2, // Deve constar na base
3   ... "data_emissao": "14/12/2023", // Não pode ser anterior a data de pagamento
4   ... "data_recebimento": "15/12/2023" // Deve ser posterior a data de emissão
5 }
```

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize **JSON** ▾

```
1 {
2   "message": "Nota emitida com sucesso",
3   "info": {
4     "id": 1,
5     "identificador": "113e17f2-2492-43c9-875f-ee6c8f26b999",
6     "data_emissao": "2023-12-14T03:00:00.000Z",
7     "data_recebimento": "2023-12-15T03:00:00.000Z",
8     "pedido_ressuprimento_id": 2,
9     "updatedAt": "2023-12-14T18:30:09.501Z",
10    "createdAt": "2023-12-14T18:30:09.501Z"
11  }
12 }
```

8. Validação de data_recebimento em emissão de nota fiscal

POST http://localhost:3000/nota_fiscal

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

● none ● form-data ● x-www-form-urlencoded ● raw ● binary ● GraphQL **JSON** ▾

```
1 {
2   ... "pedido_ressuprimento_id": 2, // Deve constar na base
3   ... "data_emissao": "15/01/2024", // Não pode ser anterior a data de pagamento
4   ... "data_recebimento": "15/01/2024" // Deve ser posterior a data de emissão
5 }
```

Body Cookies Headers (7) Test Results 🌐 400 Bad Request 30 ms 312 B

Pretty Raw Preview Visualize **JSON** ▾

```
1 {
2   "error": "Data de recebimento deve ser superior a data de emissão"
3 }
```

9. Validação de data (data_despacho)

PATCH `http://localhost:3000/pedido_ressuprimento/despacho/internacional/2`

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL **JSON** ▾

```
1 {
2   ... "data_despacho": "15/12/2023", // Não pode ser anterior a data de emissão da nota f:
3   ... "alfandega_internacional_id": 1, // Deve constar na base
4   ... "transportadora_internacional_id": 2, // Deve constar na base e entregar na alfandeg
      internacional
5   ... "frete_internacional": 10.55 // Deve ser um número positivo
6 }
```

Body Cookies Headers (7) Test Results 🌐 400 Bad Request 11 ms 329 B 📄 Save

Pretty Raw Preview Visualize **JSON** ▾ 🔍

```
1 {
2   "error": "Data de despacho não deve ser anterior a data de emissão da nota fiscal"
3 }
```

10. Validação cobertura transportadora internacional

PATCH `http://localhost:3000/pedido_ressuprimento/despacho/internacional/2` Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Coc

none form-data x-www-form-urlencoded raw binary GraphQL **JSON** ▾ Beat

```
1 {
2   ... "data_despacho": "16/01/2024", // Não pode ser anterior a data de emissão da nota fiscal
3   ... "alfandega_internacional_id": 1, // Deve constar na base
4   ... "transportadora_internacional_id": 2, // Deve constar na base e entregar na alfandega
      internacional
5   ... "frete_internacional": 10.55 // Deve ser um número positivo
6 }
```

Body Cookies Headers (7) Test Results 🌐 400 Bad Request 27 ms 366 B 📄 Save as exampl

Pretty Raw Preview Visualize **JSON** ▾ 🔍 📄

```
1 {
2   "error": "A transportadora Dragon Logistics Co., Ltd. não realiza transporte para a alfandega
      Customs Clearing Hub USA"
3 }
```

11. Validação de frete

PATCH `http://localhost:3000/pedido_ressuprimento/despacho/internacional/2` Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cook

none form-data x-www-form-urlencoded raw binary GraphQL **JSON** Beauti

```
1 {
2   ... "data_despacho": "16/01/2024", // Não pode ser anterior a data de emissão da nota fiscal
3   ... "alfandega_internacional_id": 2, // Deve constar na base
4   ... "transportadora_internacional_id": 2, // Deve constar na base e entregar na alfandega
      internacional
5   ... "frete_internacional": -2 // Deve ser um número positivo
6 }
```

Body Cookies Headers (7) Test Results 🌐 400 Bad Request 15 ms 341 B 📄 Save as example

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "error": "Frete internacional inválido. Valor do frete deve ser numérico e maior ou igual a 0"
3 }
```

12. Validação de Estado Ressuprimento

PATCH `http://localhost:3000/pedido_ressuprimento/entrega/2`

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL **JSON**

```
1 {
2   ... "saida_nacional": "30/01/2024" // Não pode ser anterior a data de despacho
3 }
```

Body Cookies Headers (7) Test Results 🌐 400 Bad Request 27 ms 323 B 📄 Save

Pretty Raw Preview Visualize **JSON**

```
1 {
2   "error": "Status do pedido é diferente de: Liberado pela alfândega nacional"
3 }
```

13. Estado final da entrega (internacional)

GET http://localhost:3000/pedido_ressuprimento/id

Params • Authorization Headers (6) Body Pre-request Script Tests Settings

Path Variables

Key	Value
id	3

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {"id": 3,
2  "frete_internacional": 10.55,
3  "frete_local": 9.99,
4  "data_pedido": "2023-12-14T03:00:00.000Z",
5  "aceito": true,
6  "data_aceitacao": "2023-12-14T03:00:00.000Z",
7  "previsao_chegada": null,
8  "data_despacho": "2023-12-15T03:00:00.000Z",
9  "chegada_alfandega_int": "2023-12-16T03:00:00.000Z",
10 "liberacao_alfandega_int": "2023-12-17T03:00:00.000Z",
11 "chegada_alfandega_nac": "2023-12-18T03:00:00.000Z",
12 "liberacao_alfandega_nac": "2023-12-18T03:00:00.000Z",
13 "saida_nacional": "2023-12-18T03:00:00.000Z",
14 "data_chegada": "2023-12-30T03:00:00.000Z",
15 "origem_ressuprimento": "Internacional",
16 "status_pedido_ressuprimento": "Pedido entregue",
17 "createdAt": "2023-12-14T18:32:10.000Z",
18 "updatedAt": "2023-12-14T18:35:34.000Z",
19 "alfandega_internacional_id": 2,
20 "alfandega_nacional_id": 1,
21 "deposito_id": 1,
22 "fornecedor_id": 1,
23 "transportadora_internacional_id": 2,
24 "transportadora_local_id": 1
25 }
26
```

14. Estado final da entrega (nacional)

GET http://localhost:3000/pedido_ressuprimento/id

Params • Authorization Headers (6) Body Pre-request Script Tests Settings

Path Variables

Key	Value
id	2

Body Cookies Headers (7) Test Results

Pretty Raw Preview Visualize JSON

```
1 {"id": 2,
2  "frete_internacional": null,
3  "frete_local": 12.56,
4  "data_pedido": "2023-12-14T03:00:00.000Z",
5  "aceito": true,
6  "data_aceitacao": "2023-12-14T03:00:00.000Z",
7  "previsao_chegada": "2023-12-30T03:00:00.000Z",
8  "data_despacho": "2023-12-14T03:00:00.000Z",
9  "chegada_alfandega_int": null,
10 "liberacao_alfandega_int": null,
11 "chegada_alfandega_nac": null,
12 "liberacao_alfandega_nac": null,
13 "saida_nacional": "2023-12-16T03:00:00.000Z",
14 "data_chegada": "2023-12-30T03:00:00.000Z",
15 "origem_ressuprimento": "Nacional",
16 "status_pedido_ressuprimento": "Pedido entregue",
17 "createdAt": "2023-12-14T18:25:18.000Z",
18 "updatedAt": "2023-12-14T18:31:40.000Z",
19 "alfandega_internacional_id": null,
20 "alfandega_nacional_id": null,
21 "deposito_id": 1,
22 "fornecedor_id": 1,
23 "transportadora_internacional_id": null,
24 "transportadora_local_id": 1
25 }
26
```

13. Conceitos das Tecnologias Usadas na API:

JavaScript: É uma linguagem de programação de alto nível, interpretada e orientada a objetos. Criada originalmente para ser executada nos navegadores web, ela desempenha um papel fundamental no desenvolvimento web moderno. [21]

JSON: "JavaScript Object Notation", é um formato leve de troca de dados que se tornou amplamente utilizado na comunicação entre sistemas e na representação de dados estruturados. [22]

Node.js: É um ambiente de execução de JavaScript do lado do servidor, construído sobre o motor V8 da Google. Ele permite que os desenvolvedores usem JavaScript para escrever scripts 'do lado' do servidor, possibilitando a criação de aplicações web escaláveis e de alto desempenho. O Node.js utiliza um modelo de I/O não bloqueante e assíncrono, o que significa que pode lidar com várias operações simultaneamente sem esperar pela conclusão de uma antes de passar para a próxima. [23]

NPM: É o gerenciador de pacotes para o ecossistema Node.js. Ele facilita o download, instalação, e gerenciamento de bibliotecas e módulos de terceiros, permitindo que desenvolvedores incorporem facilmente dependências em seus projetos Node.js. [17]

Express: É um framework para Node.js que simplifica o desenvolvimento de aplicações web. Ele fornece uma estrutura mínima para construir servidores e APIs de forma rápida e eficiente, facilitando o roteamento, manipulação de requisições e respostas, além de oferecer um conjunto de middleware para adicionar funcionalidades extra. [24]

Sequelize: É um ORM (Object-Relational Mapping ou Mapeamento Objeto-Relacional) para Node.js, que simplifica a interação com bancos de dados relacionais. Ele permite que os desenvolvedores usem objetos JavaScript para realizar operações de consulta e manipulação de dados, abstraindo a complexidade do SQL e fornecendo uma interface mais orientada a objetos. Com o Sequelize, é possível modelar, definindo os modelos e suas respectivas tabelas e seus relacionamentos. Com poucos comandos é possível criar e desfazer as tabelas definidas nos modelos além de gerar arquivos de "seed" para realizar inserções na tabela para fins de massa de teste. O sequelize também disponibiliza os métodos necessários e intuitivos para realizar o CRUD e definir transações. [3]

GitHub: É uma plataforma de hospedagem e colaboração para controle de versão de código-fonte usando o sistema de controle de versão Git. O histórico de implementação deste projeto está documentado na plataforma e disponível para download. [25]

Postman: É uma plataforma de colaboração para o desenvolvimento de APIs. Oferecendo uma variedade de ferramentas para facilitar o processo de criação, teste e documentação de APIs. [7]

date-fns: É uma biblioteca JavaScript focada em manipulação e formatação de datas. Ela fornece uma ampla gama de funções para realizar operações com datas [18]

faker: É uma biblioteca JavaScript utilizada para gerar dados fictícios de maneira realista. No contexto de simular um registro de nota fiscal eletrônica, o faker pode ser empregado para criar informações fictícias. É útil para testes e desenvolvimento, permitindo criar conjuntos de dados simulados que se assemelham a dados reais. [19]

ChatGPT: É um modelo de linguagem desenvolvido pela OpenAI, baseado na arquitetura GPT-3.5. Ele é capaz de entender e gerar texto em linguagem natural, sendo utilizado para uma variedade de aplicações, desde conversação até geração de conteúdo. [20]

13.1 Utilização do ChatGPT no Projeto

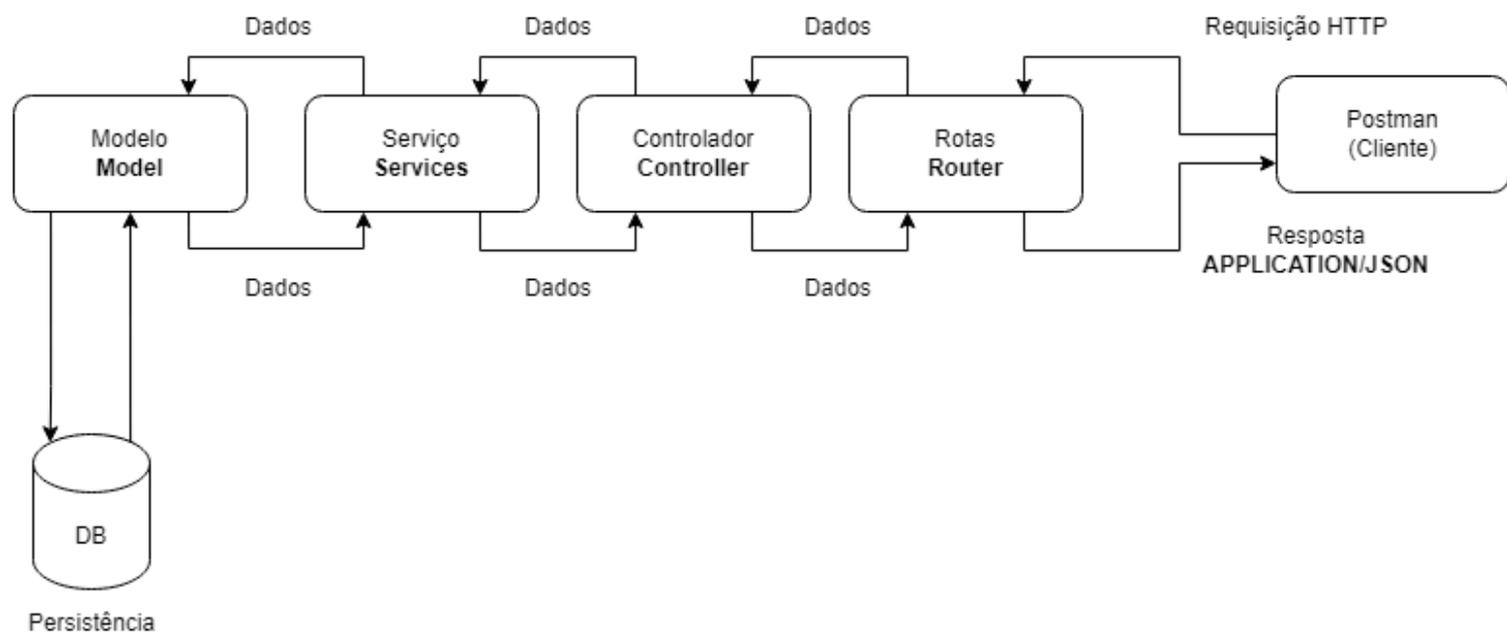
Neste projeto, a ferramenta desempenhou um papel crucial ao facilitar e acelerar a inclusão de dados de massa de teste. Utilizando os arquivos de Seed disponibilizados pelo Sequelize, conseguimos criar objetos JSON que definem os valores a serem inseridos em tabelas específicas, como no modelo de produtos. Cada JSON corresponde a uma linha a ser inserida na tabela.

Ao utilizar o prompt do ChatGPT foram detalhados de forma criteriosa a informações a serem inseridas, isto é: nomes das colunas, seus tipos e os valores a serem inseridos. Além disso, foi possível contextualizar o projeto, garantindo que as informações geradas fossem consistentes com o tema abordado. Ao final, a ferramenta gera um JSON com as regras definidas, que é então copiado para o arquivo de Seed do projeto.

Uma dificuldade encontrada foi o limite de caracteres nas respostas do ChatGPT. Ao solicitar a geração de 50 exemplos diferentes, a ferramenta fornecia no máximo 2 exemplares, acompanhados por um código JavaScript para gerar mais exemplos, o que não era desejado. Essa limitação foi superada ao solicitar a geração de apenas 3 exemplos inicialmente e, em prompts subsequentes, pedindo outros 3 exemplos com características distintas dos fornecidos anteriormente. Repetimos esse processo até alcançar a quantidade desejada de JSONs.

14. Arquitetura do Software

Figura 2: Arquitetura do software



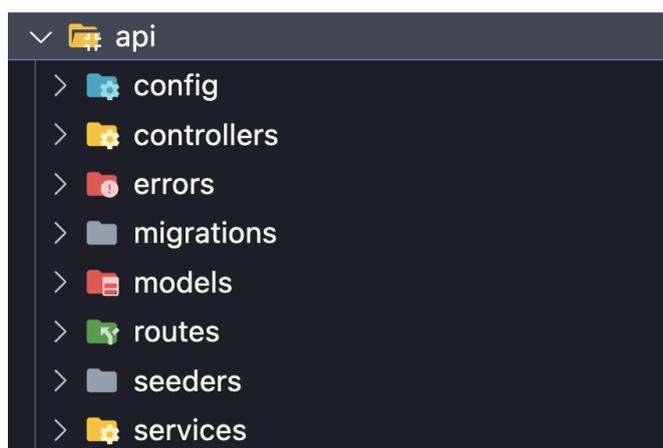
Fonte: Elaborado pelo autor, 2023

15. Estrutura da Implementação do Projeto

Esta API contém um diretório principal onde estão organizados os arquivos de configuração do banco de dados, arquivos de massa de teste e das camadas da arquitetura do software.

Principais arquivos do projeto

Figura 3 - Arquivos do projeto



Fonte: Elaborado pelo autor, 2023

models: Definem a estrutura e o comportamento dos dados no banco de dados. Representam tabelas e seus relacionamentos no banco de dados relacional.

migrations: Responsável por gerenciar e versionar as mudanças no esquema do banco de dados. Tem o papel de criar, modificar ou desfazer alterações no esquema do banco de dados de forma controlada.

services: Agrupa a lógica da regra de negócio da API. É a camada intermediária entre os controladores e os modelos.

controllers: Responsável por receber as requisições, entregar os dados à camada de serviço e enviar as respostas de sucesso ou falha.

routes: Define os endpoints da API e conecta as requisições HTTP aos controladores apropriados.

seeders: Preencher o banco de dados com dados iniciais ou de massa de teste.

errors: Permite criar erros personalizados para as mais variadas situações. Possui um status code para erros e uma mensagem de descrição do erro.

16. Possíveis Evoluções do Projeto:

Considerando o cenário atual, as futuras melhorias no projeto podem incluir:

- Lógica de vendas mais detalhada. Neste projeto, o fluxo de saída de produtos está simples, pois o foco foi o processo de ressuprimento.
- Refatoração dos ENUMS. A API, no estado atual, obriga o cliente a digitar uma string com uma ortografia e acentuação específica tendo uma abertura para erros de digitação. Mostra-se necessário a criação de uma classe de ENUMs no sistema onde é associado um código inteiro com uma string referente aos valores desejados
- Interface visual para a realização das operações disponibilizadas pelo sistema de forma mais amigável.
- Sistema de recomendação de modalidade de estoque.
- Sistema de conversão de câmbio incorporado no sistema.
- Refatorar a persistência da multa de forma a guardar a multa total paga no pedido de ressuprimento.
- Incorporar testes automatizados no sistema para garantir a robustez do mesmo.
- Incorporar lógica de devolução de produtos ao fornecedor. Atualmente, a API não abrange adequadamente situações em que ocorrem problemas com os produtos durante a entrega ao depósito, o que constitui uma limitação na fidelidade da representação da realidade.
- Conceitos pesquisados sobre o tema que não foram tratados no projeto podem ser utilizados em trabalhos futuros

17. Considerações Finais

Estou satisfeito com o resultado alcançado deste projeto. A implementação da cadeia de ressuprimento foi bem sucedida de acordo com o planejamento feito ao longo do semestre. A API atualmente possui 19 endpoints que representam a cadeia de ressuprimento em casos nacionais e internacionais, e endpoints auxiliares para facilitar a visualização do resultado das operações realizadas.

O desenvolvimento deste projeto final proporcionou um grande aprendizado sobre planejamento de projeto e sua importância para a etapa de desenvolvimento da API. Foi possível perceber o quanto engrandecedor é pôr em prática um projeto, seja acadêmico ou pessoal. De forma que ao começar um novo projeto pessoal estarei levando todo o aprendizado aqui adquirido.

Cada etapa do desenvolvimento deste projeto me permitiu pôr em prática os conhecimentos e ensinamentos adquiridos durante o curso e aprender de forma prática. Além dos conhecimentos auto-didata e experiência profissional.

Esse trabalho não seria possível sem a orientação do meu orientador Edmundo Torreão que foi indispensável para os resultados aqui alcançados e todo aprendizado adquirido neste projeto.

18. Referências:

[1] Applying UML and patterns - Craig Larman - 2a edição. Disponível em:

<<https://personal.utdallas.edu/~chung/SP/applying-uml-and-patterns.pdf>>

Acesso em: 13/10/2023

[2] REST. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Glossary/REST>>

Acesso em: 06/10/2023

[3] Sequelize. Disponível em: <<https://sequelize.org/>>

Acesso em: 06/10/2023

[4] Express. Disponível em: <<https://expressjs.com/pt-br/>>

Acesso em: 06/10/2023

[5] About Node.js. Disponível em: <<https://nodejs.org/en/about/>>

Acesso em: 06/10/2023

[6] JSON Web Tokens are an open, industry standard RFC 7519 method for representing claims securely between two parties. Disponível em: <<https://jwt.io/>>

Acesso em: 06/10/2023

[7] What is Postman? Disponível em:

<<https://www.postman.com/product/what-is-postman/>>

Acesso em: 06/10/2023

[8] 9 tipos de estoque: conheça cada um deles e sua importância no varejo.

Disponível em: <<https://nerus.com.br/blog/estoque/quais-tipos-de-estoque/>>

Acesso em: 06/10/2023

[9] 9 tipos de estoque que você precisa conhecer. Disponível em:

<https://eccosys.com.br/guia/tipos-de-estoque-que-voce-precisa-conhecer/#4_Estoque_de_canal>

Acesso em: 06/10/2023

[10] 8 tipos de estoque que você precisa conhecer. Disponível em:

<<https://blog-pt.checklistfacil.com/tipos-de-estoque/>>

Acesso em: 06/10/2023

[11] Giro de estoque: o que é e como calcular. Disponível em:

<<https://www.cashme.com.br/blog/giro-de-estoque/>>

Acesso em: 06/10/2023

[12] Dimensionamento do Estoque de Segurança Disponível em:

<<https://aprendendogestao.com.br/2016/08/23/dimensionamento-do-estoque-de-seguranca/>>

Acesso em: 06/10/2023

[13] O que é SKU do produto e como usá-lo no e-commerce? Disponível em:

<https://pagar.me/blog/o-que-e-sku/?utm_source=google&utm_medium=cpc&utm_campaign=dsa_MegaOferta&origin=search&media=google&type=pago&campaign=11366699401&ad_group=118179479464&ad=534449494969&theme=&gclid=Cj0KCQiAtJeNBhCVARIsANJUJ2G6wbdh0GkPaB0zaKWQU_02ZX9vIaCFaGLEmFMh_s15udds1vS050YaAgjJEALw_wcB>

Acesso em: 06/10/2023

[14] CÓDIGOS EAN: O QUE É E PARA QUE SERVE? Disponível em:

<<https://magis5.com.br/o-que-e-codigo-ean-para-que-serve/>>

Acesso em: 06/10/2023

[15] Estoque de Segurança. Disponível em:

<<https://www.z3bs.com.br/lead-time-estoque-de-seguranca-e-ponto-de-reposicao/>>

Acesso em: 06/10/2023

[16] Nível de serviço e estoque de segurança. Disponível em:

<<http://www.conhecimento.market.com.br/artigo/453>>

Acesso em: 06/10/2023

[17] About npm. Disponível em:

<<https://docs.npmjs.com/about-npm>>

Acesso em: 14/12/2023

[18] date-fns. Disponível em:

<<https://www.npmjs.com/package/date-fns>>

Acesso em: 14/12/2023

[19] faker-js/faker. Disponível em:

<<https://www.npmjs.com/package/@faker-js/faker>>

Acesso em: 14/12/2023

[20] Introducing ChatGPT. Disponível em:

<<https://openai.com/blog/chatgpt>>

Acesso em: 14/12/2023

[21] O que é JavaScript?. Disponível em:

<https://developer.mozilla.org/pt-BR/docs/Learn/JavaScript/First_steps/What_is_JavaScript>

Acesso em 15/12/2023

[22] Introducing JSON. Disponível em:

<<https://www.json.org/json-en.html>>

Acesso em 15/12/2023

[23] About Node.js Disponível em:

<<https://nodejs.org/en/about>>

Acessi em 15/12/2023

[24] Express. Framework web rápido, flexível e minimalista para Node.js. Disponível em: <<https://expressjs.com/pt-br/>>

Acesso em: 15/12/2023

[25] The complete developer platform to build, scale, and deliver secure software.

Disponível em: <<https://github.com/about>>

Acesso em: 15/12/2023