

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

**RealTimeStatus – API para entrega expressa e
rastreamento de pedidos**

Thomaz Passarelli Espírito Santo

RELATÓRIO DA DISCIPLINA PROJETO FINAL 2

CENTRO TÉCNICO CIENTÍFICO - CTC
DEPARTAMENTO DE INFORMÁTICA
Curso de Graduação em Ciência da Computação

Rio de Janeiro, dezembro de 2023



Thomaz Passarelli Espírito Santo

RealTimeStatus – API para entrega expressa e rastreamento de pedidos.

Relatório da Disciplina Projeto Final 2, apresentado ao Curso de Ciência da Computação da PUC-Rio como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Edmundo Bastos Torreão

Departamento de Informática

Rio de Janeiro, dezembro de 2023

Resumo

Santo, Thomaz Passarelli Espírito. Torreão, Edmundo Bastos. RealTimeStatus. Rio de Janeiro, 2023. (Número de páginas: 43) Relatório de Projeto Final – Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

O projeto "RealTimeStatus" propõe uma solução inovadora para o rastreamento em tempo real e gestão de entregas. O sistema integra atualizações dinâmicas de estoques e gerenciamento eficiente de pedidos, melhorando a transparência e controle no processo de entrega. Desenvolvido com PHP e Laravel, o projeto demonstra potencial para aprimorar processos de entrega.

Palavras-chave: Rastreamento em Tempo Real, Gestão de Entregas, API.

Abstract

Santo, Thomaz Passarelli. Torreão, Edmundo Bastos. RealTimeStatus. Rio de Janeiro, 2023. (Number of pages: 43) Final Project Report – IT Department. Pontifical Catholic University of Rio de Janeiro.

The "RealTimeStatus" project offers an innovative approach to real-time tracking and delivery management. The system includes dynamic inventory updates and efficient order management, enhancing transparency and control in the delivery process. Developed using PHP and Laravel showing potential to enhance delivery processes.

Keywords: Real-Time Tracking, Delivery Management, API.

Sumário

1. Introdução:	5
2. Proposta e Objetivos do Trabalho	6
3. Plano de Ação	7
5. Diagrama de Classes:	9
6. Diagrama de casos de uso:	10
7 Diagrama Relacional	11
8. Descrição de casos de uso:	12
8.1. Descrição de caso de Uso UC01	12
8.1.1. Teste do Caso de Uso UC01:	13
8.2. Descrição de caso de Uso UC02	14
8.2.1. Teste do caso de Uso UC02	15
- RESULTADO: Código de rastreio RECONHECIDO + Localização obtida com sucesso	15
- RESULTADO: Pedido INVÁLIDO	16
- RESULTADO: PEDIDO RECONHECIDO + Pedido ainda não saiu para entrega.	17
8.3. Descrição de caso de Uso UC03	18
8.3.1. Teste do caso de Uso UC03	20
- RESULTADO: Todos os Itens Disponíveis	21
8.4. Descrição de caso de Uso UC04	23
8.4.1. Teste do caso de Uso UC04	24
- RESULTADO: Pedidos encontrados com SUCESSO	24
- RESULTADO: Id Cliente INVÁLIDO	25
- RESULTADO: NÃO existem pedidos associados ao cliente.	26
8.5. Descrição de caso de Uso UC05	27
8.5.1. Teste do caso de Uso UC05	29
- RESULTADO: Pedido ainda não saiu para entrega.	29
- RESULTADO: Pedido já foi entregue.	30
- RESULTADO: Entrega confirmada com sucesso.	31
8.6. Descrição de caso de Uso UC06	32
8.6.1. Teste do caso de Uso UC06	32
- RESULTADO: status da entrega atualizado com sucesso.	34
8.7. Descrição de caso de Uso UC07	35
8.7.1. Teste do caso de Uso UC07	36
- RESULTADO: pedido de abastecimento realizado com sucesso.	36
8.8. Descrição de caso de Uso UC08 - Informa localização	37
8.8.1. Teste do caso de Uso UC08	39
- RESULTADO: Localização informada com sucesso.	39
8.9. Requisitos não funcionais.	40
9. Estratégia de execução e especificação dos Requisitos Funcionais a serem implementados pela API	41
9.1. Introdução	41
9.2. Ferramentas e Tecnologias	41

<u>9.3. Desenvolvimento</u>	<u>41</u>
<u>9.3.1. Banco de Dados</u>	<u>41</u>
<u>9.3.2. Criação do Projeto Laravel</u>	<u>41</u>
<u>9.3.3. Models</u>	<u>42</u>
<u>9.3.4. Rotas (Endpoints) da API</u>	<u>42</u>
<u>8.3.5. Desenvolvimento do Controller</u>	<u>42</u>
<u>8.3.6 Testes Realizados</u>	<u>42</u>
<u>9. Considerações Finais</u>	<u>43</u>
<u>12. Trabalhos Futuros</u>	<u>46</u>
<u>13. Referências Bibliográficas:</u>	<u>47</u>

1. Introdução:

A motivação para criação do projeto veio de uma demanda do mercado de entregas e da logística atuais que, em diversas situações, impossibilita o rastreo e entrega expressa dos produtos de forma eficiente.

Em um mundo globalizado, a concorrência tornou-se muito mais feroz do que antes, então o planejamento logístico de uma empresa se tornou uma das suas áreas mais importantes para determinar a vantagem competitiva em relação ao concorrente e, também, quanto à satisfação dos seus clientes.

Dessa forma, atualmente, a logística gira em torno da redução de custos, aumento da agilidade no processo e do atendimento total às necessidades do cliente, existindo diversos pontos de gargalo que buscam o aprimoramento. O reflexo disso na prática é um sistema muitas vezes falho, nas entregas ditas 'rápidas', demorando mais do que o prazo dito como 'rápido' (1 a 2 dias) para chegarem ao domicílio devido à longa distância dos centros logísticos e com atrasos na atualização de status do pedido.

Essa grande distância dos centros de logística, muito fora do alcance dos centros urbanos, torna a dinâmica de entrega mais sujeita a atrasos por fatores externos, como: trânsito, acidentes, infraestrutura precária de grandes rodovias, desvios de rota, danos ou perda no produto por manipulação de diversos setores. Em um mundo global, torna-se grande a vantagem competitiva nas empresas que conseguem aprimorar processos logísticos para que estes se tornem cada vez mais eficientes, com o menor custo possível. Além das grandes distâncias temos uma falha de gestão de estoque e de planejamento das entregas expressas, gerando problemas no controle de estoque e reduzindo a velocidade das logísticas 'inbound' e 'outbound' das empresas.

Atualmente o status das encomendas demoram muito para serem atualizados, não dando uma perspectiva exata de onde se encontra o produto e da estimativa de entrega dele na residência do cliente, nem as estatísticas necessárias para que a empresa organize pedidos de fornecedores.

A proposta do sistema do escopo deste projeto permitiria a redução dos meios de transporte com a proximidade de grandes centros urbanos, possibilitando a realização de entregas 'a pé' ou por meio de bicicletas/motocicletas até o destino final, otimizando as entregas expressas e respeitando seus prazos restritos de alta demanda. Nesse cenário, é possível rastrear todas as compras de diversos ramos empresariais por uma única plataforma, de forma rápida e em tempo real, desde o momento da compra até a entrega.

2. Proposta e Objetivos do Trabalho

Uma vez evidenciadas as falhas no sistema atual de entrega e logística a proposta desse projeto seria criar um sistema mais uniformizado e eficiente, com tecnologia real time de rastreamento de pedidos bem como a ampliação dos armazéns de mercadorias em diferentes pontos dos centros urbanos, possibilitando uma redução da distância de frete e aumentando a dinâmica e eficácia das entregas. A ideia é tornar o sistema para entrega de mercadorias adquiridas em farmácias, de forma ágil e dinâmica, seja diretamente em domicílio ou pela retirada em 'lojas físicas'.

O software foi desenvolvido como uma API, em especial, para que possa ser utilizado pelos sistemas das organizações vendedoras (aqui identificadas pelo ator 'Vendedor') que desejem utilizar os serviços dos Centros de Distribuição, foco deste projeto, para realização de entregas expressas. As organizações vendedoras são responsáveis pelo ressuprimento de seus 'pequenos depósitos' nos Centros de Distribuição que são compartilhados por vários 'vendedores'.

3. Requisitos do sistema

- Entregas em domicílio realizadas por diferentes meios de transporte (bicicletas/motocicletas ou 'a pé')
- Redução de distância dos centros logísticos, tornando o frete de mercadorias cada vez mais curtos.
- Rastreamento de mercadorias em real time desde a confirmação do pedido até o ato de entrega.
- Os pagamentos das mercadorias poderão ser realizados no ato da confirmação dos pedidos ou no ato da entrega (tendo eventualmente a cobrança de respectiva taxa).
- Diferente dos recursos atuais, nesse sistema as entregas de produtos farmacêuticos controlados poderão ser realizadas à domicílio com a exigência de 'receitas médicas'.
- Atualização de estoques em tempo real por meio de sistemas API (Application Programming Interface), reduzindo os erros de controle de mercadorias após a entrada em estoque e após as vendas da quantidade disponível.
- O sistema notificará as empresas varejistas quando novas compras de fornecedores forem recebidas e quando vendas forem fechadas, fornecendo informações estatísticas importantes para otimizar assim as logísticas 'inbound' e 'outbound' das empresas.
- O API funcionará através da internet de forma a fornecer informações dinâmicas de estoque e status das mercadorias em tempo real.
- O ator 'Vendedor' integra um papel vital no processo de gestão de pedidos. No escopo deste sistema, o 'Vendedor' da loja 'física' é encarregado de registrar os pedidos dos clientes e também solicitar abastecimento do depósito da Matriz e registrar os recebimentos dos mesmos no CD Vendedor. Optou-se por esta solução, visando tornar os 'Vendedores' das lojas 'físicas', no escopo deste projeto, responsáveis, de forma integral, pelos níveis de estoque de cada produto.

4. Plano de Ação

- ☐ Etapa 1: Elaboração da proposta do projeto
- ☐ Etapa 2: Estudo de 'aplicações semelhantes'
- ☐ Etapa 3: Levantamento de Requisitos e Modelagem de Dados.
- ☐ Etapa 4: Especificação de Requisitos
- ☐ Etapa 5: Elaboração relatório de Projeto Final 1
- ☐ Etapa 6: Estudos sobre as tecnologias atuais no que se diz respeito a API's
- ☐ Etapa 7: Projeto da API
- ☐ Etapa 8: Implementação do software
- ☐ Etapa 9: Teste do software
- ☐ Etapa 10: Relatório do projeto
- ☐ Etapa 11: Preparação da apresentação para a banca

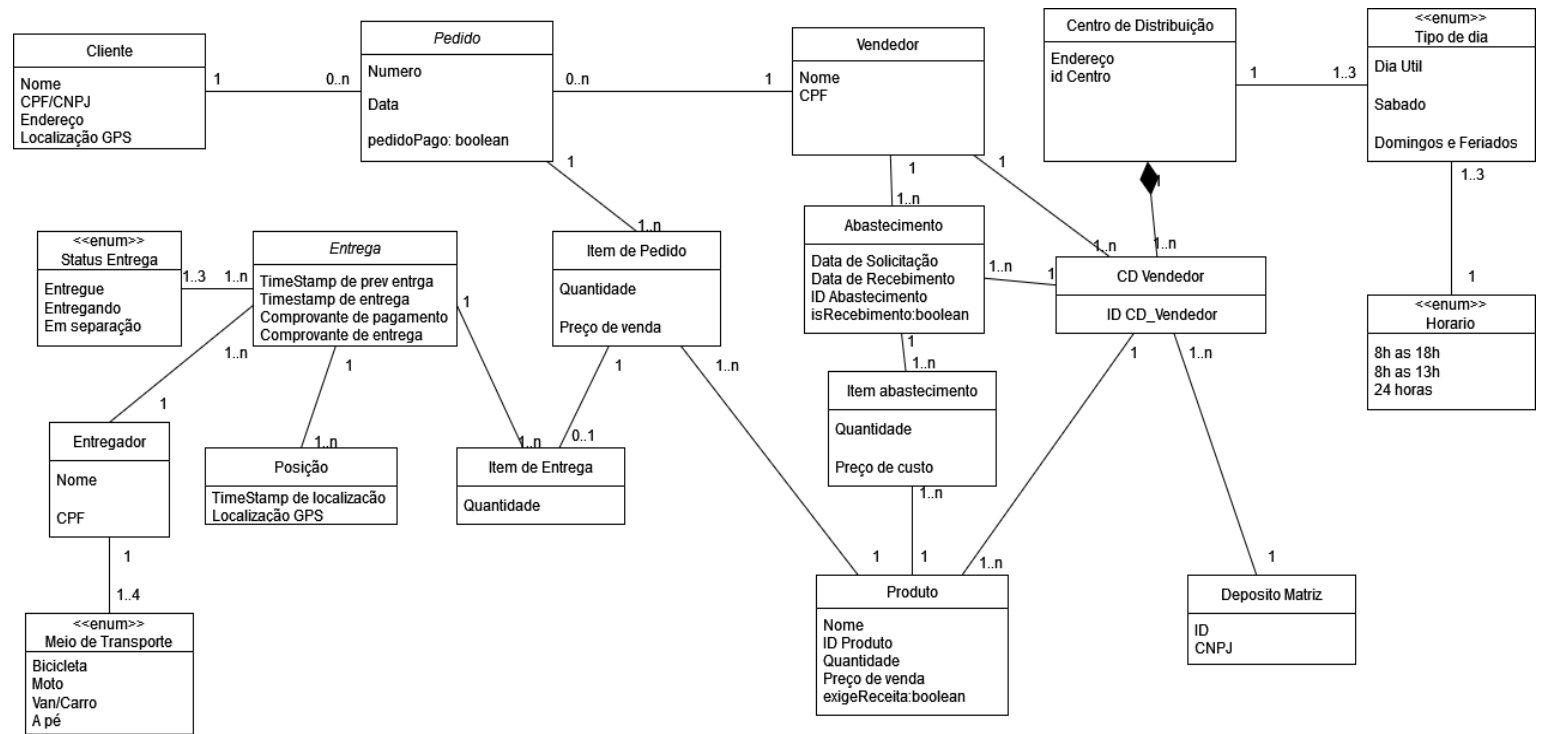
2022

Atividades Projeto Final 1	setembro				outubro				novembro				dezembro			
Proposta																
Estudo de aplicações semelhantes																
Especificação de Requisitos																
Montagem de Dados																
Relatório do Projeto Final 1																

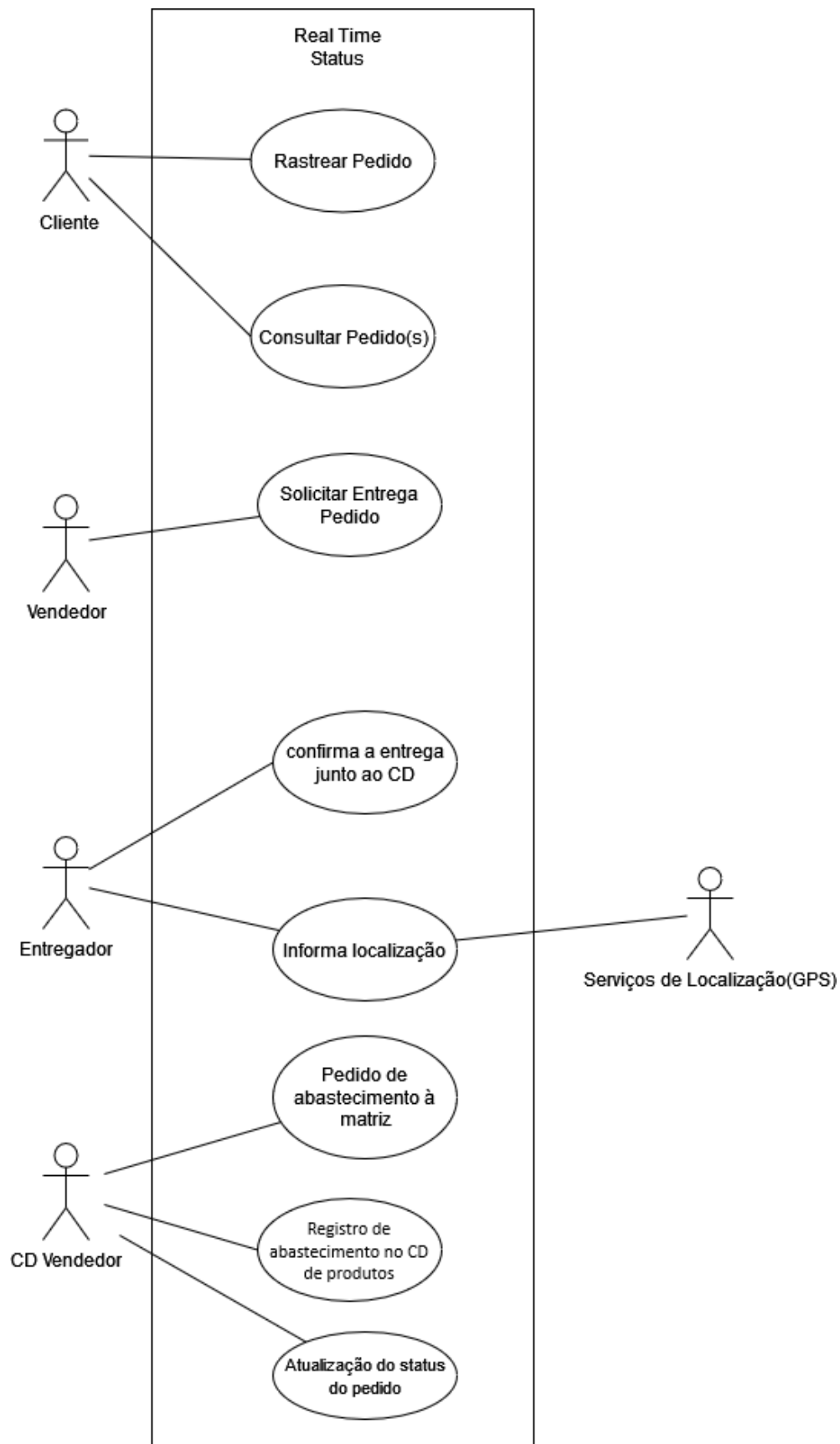
2023

Atividades Projeto Final 2	setembro				outubro				novembro				dezembro			
Estudo sobre tecnologias atuais em API																
Projeto de API																
Implementação do software																
Relatório do Projeto																
Preparação da apresentação para a banca																

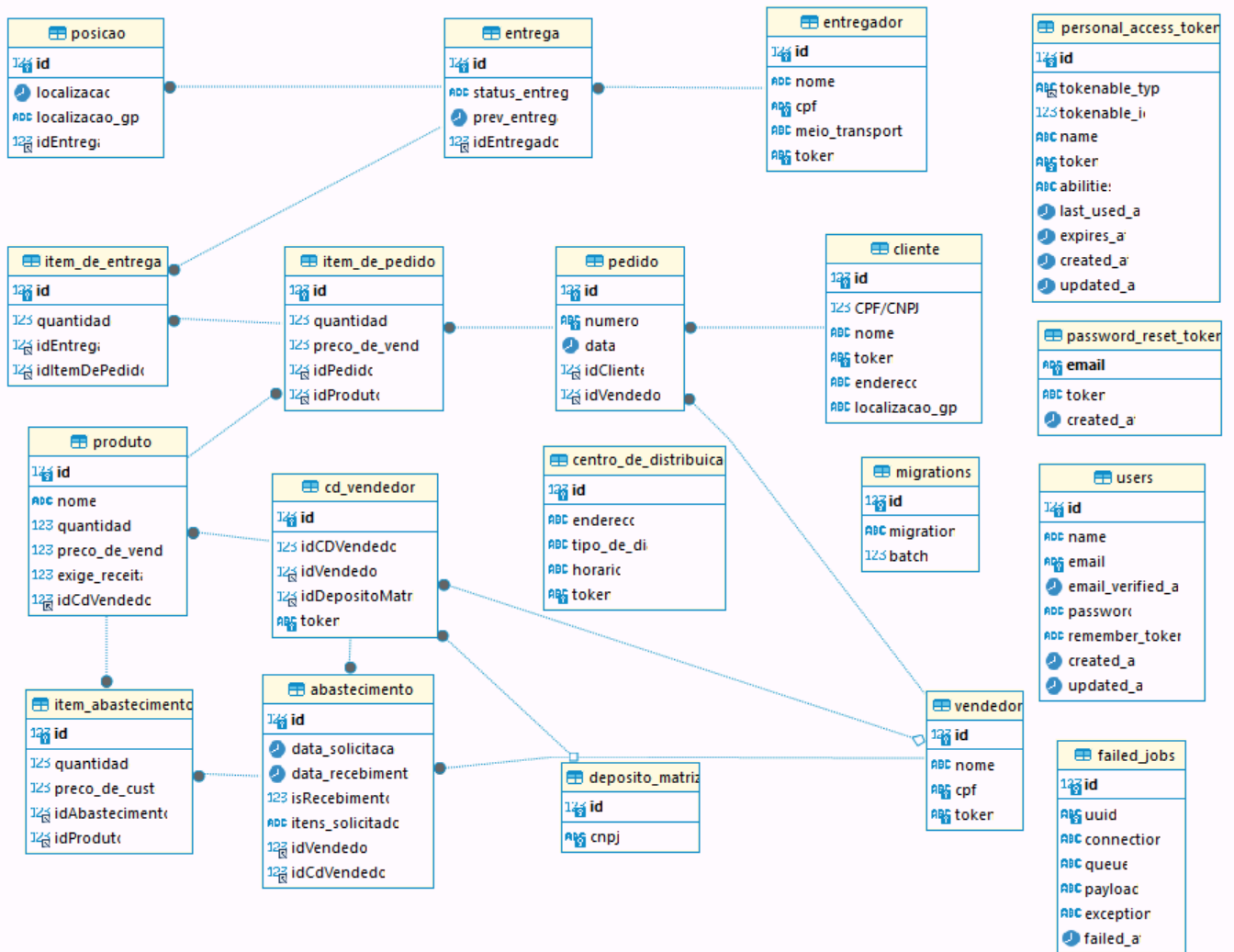
5. Diagrama de Classes:



6. Diagrama de casos de uso:



7 Diagrama Relacional



8. Descrição de casos de uso:

Os casos de uso são considerados os requisitos funcionais conforme [8].

8.1. Descrição de caso de Uso UC01

7.1 Descrição de Casos de Uso - UC01 - Registro de abastecimento no CD de produtos

Nome:	Registro de abastecimento no CD de produtos
Objetivo:	Registrar abastecimento de estoque no Centro de Distribuição
Atores:	Vendedor(CD)
Pré-condição:	Abastecimento com data agendada
Pós-condição (cenário de sucesso):	Abastecimento registrado com sucesso
Pós-condição (cenário de insucesso):	Não foi possível registrar abastecimento
Trigger:	
Fluxo Principal:	<ol style="list-style-type: none">1. O centro de distribuição recebe o abastecimento2. O vendedor informa os itens abastecidos e suas respectivas quantidades[E1][E2]2. Sistema registra o abastecimento e envia confirmação de que abastecimento foi registrado com sucesso
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] Abastecimento não realizado</p> <ol style="list-style-type: none">1.O centro de distribuição registra o não recebimento do abastecimento2.Sistema finaliza caso de uso <p>[E2] Falha na confirmação</p> <ol style="list-style-type: none">1.Sistema não envia confirmação de registro de abastecimento2.Retorna ao passo 2 do fluxo principal
Pontos de Extensão:	
Regras de Negócio	

8.1.1. Teste do Caso de Uso UC01:

Vendedor registra abastecimento informando produtos, data do recebimento e seu token de acesso pessoal e recebe uma resposta confirmando o registro

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

- URL:** `http://localhost:8000/api/abastecimento`
- Method:** `POST`
- Body (JSON):**

```
1 {
2   "data_recebimento": "2023-11-23",
3   "token": "1YdMPwDttZWVORI0Li1lIs46kgFJK2AKrzPSWgmb2WR7GW90MXhHq6VxUwuksp11",
4   "itens": [
5     {
6       "product_name": "Produto-AZ",
7       "quantity": 1
8     },
9     {
10      "product_name": "Produto-ZA",
11      "quantity": 2,
12    }
13  ]
14 }
```
- Response (JSON):**

```
1 {
2   "message": "Abastecimento registrado com sucesso!"
3 }
```

8.2. Descrição de caso de Uso UC02

7.2 Descrição de Casos de Uso - UC02 - Rastrear Pedido

Nome:	Rastrear pedido
Objetivo:	Consultar localização do pedido
Atores:	Cliente
Pré-condição:	Pedido saiu para entrega
Pós-condição (cenário de sucesso):	localização do pedido com sucesso
Pós-condição (cenário de insucesso):	Pedido ainda não saiu para entrega
Trigger:	
Fluxo Principal:	<ol style="list-style-type: none">1. O cliente envia requisição de localização do pedido2. O sistema informa a última coordenada gps registrada pelo entregador [A1][E1] <p>Repete passo 2 até que Cliente finalize o Caso de Uso</p>
Fluxos Alternativos:	[A1] Pedido não possui coordenadas gps registradas ainda <ol style="list-style-type: none">1. Sistema exibe mensagem2. Sistema finaliza caso de uso
Fluxos de Exceção	[E1] Cliente não recebe nenhuma resposta do servidor <ol style="list-style-type: none">1. Sistema informa situação2. Cliente realiza nova tentativa3. Retorna ao passo 1 do Fluxo Principal
Pontos de Extensão:	
Regras de Negócio	

8.2.1. Teste do caso de Uso UC02

Cliente informa o número do pedido e seu token de acesso pessoal e recebe uma resposta informando a localização(latitude e longitude)

- **RESULTADO: Código de rastreio RECONHECIDO + Localização obtida com sucesso**

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

- URL:** `http://localhost:8000/api/rastrearpedido`
- Method:** `POST`
- Body (Request):**

```
1 {
2   "numero": "1",
3   "token": "1YdMPwDttZWVORIOLi1lIs46kgFJK2AKrzPSWgmb2WR7GW90MXhHq6VxUwukspY3"
4 }
```
- Body (Response):**

```
1 {
2   "message": "Localização do pedido obtida com sucesso.",
3   "localizacao_gps": "-28.55052,-46.633308"
4 }
```

- RESULTADO: Pedido INVÁLIDO

HTTP Casos de uso / UC02: rastrearpedido

POST http://localhost:8000/api/rastrearpedido

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

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

```
1 {
2   "numero": "132",
3   "token": "1YdMPwDttZWVORIOLi11Is46kgFJK2AKIzPSWgmb2WR7GW90MXhHq6VxUwukspY3"
4 }
5
```

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Pedido não encontrado ou número inválido.",
3   "errors": {
4     "numero": [
5       "The selected numero is invalid."
6     ]
7   }
8 }
```

- **RESULTADO: PEDIDO RECONHECIDO + Pedido ainda não saiu para entrega.**

HTTP Casos de uso / UC02: rastrearpedido

POST http://localhost:8000/api/rastrearpedido

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

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

```
1 {
2   "numero": "7",
3   "token": "1YdMPwDttZWVORIOLi1lIs46kgFJK2AKrzPSWgmb2WR7GW90MXhHq6VxUwukspY3"
4 }
5
```

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON ▾ ↻

```
1 {
2   "message": "Pedido ainda não saiu para entrega."
3 }
```

8.3. Descrição de caso de Uso UC03

Descrição de Casos de Uso - UC03 - Solicitar Entrega Pedido

Nome:	Solicitar Entrega Pedido
Objetivo:	Solicitar um novo pedido de entrega ao Centro de Distribuição para um cliente
Atores:	Vendedor
Pré-condição:	Vendedor logado e compra fechada junto ao cliente
Pós-condição (cenário de sucesso):	Novo pedido de entrega gerado com sucesso
Pós-condição (cenário de insucesso):	
Trigger:	
Fluxo Principal:	<ol style="list-style-type: none"> 1. O Vendedor informa os itens do pedido e seus respectivos valores , além do tipo de pagamento escolhido cliente [A1][A2] 2. Sistema registra o pedido e aloca entregador
Fluxos Alternativos:	<p>[A1] Falta de produto em pedido não pago</p> <ol style="list-style-type: none"> 1. O sistema informa os itens do pedido e seus respectivos valores , além do tipo de pagamento escolhido cliente 2. Sistema informa que há falta de produtos no pedido e informa o novo valor e produtos(disponíveis) 3. Retorno ao passo 2 do fluxo principal <p>[A2] Falta de produto em pedido pago</p> <ol style="list-style-type: none"> 1. O sistema informa os itens do pedido e seus respectivos valores , além do tipo de pagamento escolhido cliente 2. Sistema informa que há falta de produtos no pedido e informa o novo valor e produtos(disponíveis) 3. Sistema emite reembolso referente a itens em falta 4. Retorno ao passo 2 do fluxo principal
Fluxos de Exceção	
Pontos de Extensão:	

Regras de Negócio	
------------------------------	--

8.3.1. Teste do caso de Uso UC03

Vendedor informa id do pedido, tipo de pagamento, seu token de acesso pessoal e recebe uma resposta confirmando a solicitação e com informações do pedido

- **RESULTADO: Todos os Itens Disponíveis**

HTTP Casos de uso / UC03: solicitarentregapedido

POST http://localhost:8000/api/solicitarentregapedido

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

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

```
1 {
2   "idPedido": 1,
3   "tipoPagamento": "pago",
4   "token": "1YdMPwDttZWVORI0Li1Is46kgFJK2AKrzPSWgmb2WR7GW90MXhHq6VxUwuksp45",
5   "itens": [
6     { "nome": "Produto K", "quantidade": 2 },
7     { "nome": "Produto L", "quantidade": 3 }
8   ]
9 }
```

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Entrega do pedido solicitada com sucesso.",
3   "pedido": {
4     "id": 1,
5     "numero": "1",
6     "data": "2023-11-24 10:00:00",
7     "idCliente": 1,
8     "idVendedor": 1
9   },
10  "itens": [
11    {
12      "nome": "Produto K",
13      "quantidade": 2,
14      "preco_de_venda": "5.99"
15    },
16    {
17      "nome": "Produto L",
18      "quantidade": 3,
19      "preco_de_venda": "15.49"
20    }
21  ],
22  "entrega": {
23    "idEntregador": 4.
```

POST http://localhost:8000/api/solicitarentregapedido

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

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

```

1  {
2    "idPedido": 1,
3    "tipoPagamento": "pago",
4    "token": "1YdMPwDttZWVORI0Li1Is46kgFJK2AKrzPSWgmb2WR7GW90MXhHq6VxUwuksp45",
5    "itens": [
6      { "nome": "Produto K", "quantidade": 2 },
7      { "nome": "Produto L", "quantidade": 300 }
8    ]
9  }
10

```

ody Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```

1  {
2    "message": "Falta de estoque para alguns itens."
3  }

```

8.4. Descrição de caso de Uso UC04

8.4 Descrição de Casos de Uso - UC04 - Consultar Pedidos

Nome:	Consultar pedidos
Objetivo:	Consultar o status de um pedido no sistema
Atores:	Cliente
Pré-condição:	Cliente logado
Pós-condição (cenário de sucesso):	Status do pedido consultado com sucesso
Pós-condição (cenário de insucesso):	
Trigger:	
Fluxo Principal:	<ol style="list-style-type: none">1. O cliente envia requisição de status pedidos2. O sistema informa uma lista de pedidos do cliente com seus respectivos status em ordem decrescente de datas de solicitação [A1][E1]3. O Cliente seleciona um pedido4. O Sistema exibe informações detalhadas do pedido selecionado[E1]
Fluxos Alternativos:	<p>[A1] Não existem pedidos associados ao cliente</p> <ol style="list-style-type: none">1. Sistema exibe mensagem2. Sistema finaliza caso de uso
Fluxos de Exceção	<p>[E1] Cliente não recebe nenhuma resposta do servidor</p> <ol style="list-style-type: none">1. Sistema informa situação2. Cliente realiza nova tentativa3. Retorna ao passo 1 do Fluxo Principal
Pontos de Extensão:	
Regras de Negócio	

8.4.1. Teste do caso de Uso UC04

Cliente informa seu id e token e recebe uma resposta contendo todos seus pedidos cadastrados no sistema

- **RESULTADO: Pedidos encontrados com SUCESSO**

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

- URL:** `http://localhost:8000/api/consultarpedidos`
- Method:** `POST`
- Body (JSON):**

```
1 {
2   "idCliente": 2,
3   "token": "1YdMPwDttZWVORI0Li1Is46kgFJK2AKzzPSWgmb2WR7GW90MXhHq6VxUwukspY3"
4 }
5
```
- Response Body (JSON):**

```
1 {
2   "message": "Pedidos encontrados com sucesso.",
3   "pedidos": [
4     {
5       "id": 9,
6       "numero": "9",
7       "data": "2023-11-24 14:00:00",
8       "idCliente": 2,
9       "idVendedor": 3,
10      "itens_de_pedido": [],
11      "entrega": {
12        "id": 9,
13        "status_entrega": "Em separacao",
14        "prev_entrega": null,
15        "idEntregador": 3
16      }
17    }
18  ]
19 }
```

- **RESULTADO: Id Cliente INVÁLIDO**

HTTP Casos de uso / UC04: consultarpedidos


POST ▼ http://localhost:8000/api/consultarpedidos

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

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

```
1 {
2   "idCliente": 22,
3   "token": "1YdMPwDttZWV0RIOLi1lIs46kgFJK2AKizPSWgmb2WR7GW90MXhHq6VxUwukspY3"
4 }
5
```

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize **JSON** ▼ 

```
1 {
2   "message": "Validação falhou.",
3   "errors": {
4     "idCliente": [
5       "The selected id cliente is invalid."
6     ]
7   }
8 }
```

- **RESULTADO: NÃO existem pedidos associados ao cliente.**

HTTP Casos de uso / UC04: consultarpedidos

POST http://localhost:8000/api/consultarpedidos

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

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

```
1 {
2   "idCliente": 6,
3   "token": "1YdMPwDttZWVORI0Li1lIs46kgFJK2AKrzPSWgmb2WR7GW90MXhHq6VxUwukspY3"
4 }
5
```

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Não existem pedidos associados a este cliente."
3 }
```

8.5. Descrição de caso de Uso UC05

8.5 Descrição de Casos de Uso - UC05 - Entregador confirma a entrega junto ao CD

Nome:	Entregador confirma a entrega junto ao CD
Objetivo:	Registrar a entrega de um pedido de um cliente feito ao CD
Atores:	Entregador
Pré-condição:	Tentativa de entrega em andamento
Pós-condição (cenário de sucesso):	Entrega confirmada com sucesso
Pós-condição (cenário de insucesso):	Entrega não pode ser efetuada
Trigger:	
Fluxo Principal:	<ol style="list-style-type: none">1. O entregador chega ao endereço de entrega[A1][A2]2. O entregador recolhe rubrica de cliente e informa que foi realizada com sucesso a entrega3.O entregador retorna ao CD e arquiva rubrica4. O CD registra entrega com pagamento prévio com sucesso
Fluxos Alternativos:	<p>[A1] Cliente optou por pagar na entrega</p> <ol style="list-style-type: none">1. O entregador chega ao endereço de entrega2. O entregador realiza a cobrança do pagamento do pedido durante a entrega e recolhe uma rubrica [A3]3. O entregador retorna ao CD com o comprovante do pagamento do pedido e a rubrica4. O CD registra entrega com pagamento com sucesso no ato da entrega5. Sistema finaliza Caso de Uso <p>[A2] Não foi possível realizar entrega</p> <ol style="list-style-type: none">1. O entregador retorna ao CD com as mercadorias2. O CD registra entrega a impossibilidade de entrega por erro de endereço/ausência do cliente no endereço2. Retorna ao passo 3 do Fluxo Principal <p>[A3] Cliente não realiza o pagamento</p> <ol style="list-style-type: none">1. O entregador retorna ao CD com as mercadorias

	<p>2. O CD registra entrega a impossibilidade de entrega por falta de pagamento</p> <p>3. Sistema finaliza Caso de Uso</p>
Fluxos de Exceção	
Pontos de Extensão:	
Regras de Negócio	

8.5.1. Teste do caso de Uso UC05

- **RESULTADO: Pedido ainda não saiu para entrega.**

HTTP Casos de uso / UC05: confirmarentrega

POST http://localhost:8000/api/confirmarentrega

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

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

```
1 {
2   "numero": "1",
3   "token": "1YdMPwDttZWVORIOLi1Is46kgFJK2AKrzPSWgmb2WR7GW9OMXhHq6VxUwuksp33"
4 }
5
```

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Pedido ainda não saiu para entrega."
3 }
```

- **RESULTADO: Pedido já foi entregue.**

HTTP Casos de uso / UC05: confirmarentrega

POST http://localhost:8000/api/confirmarentrega

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

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

```
1 {
2   "numero": "1",
3   "token": "1YdMPwDttZWVORI0Li1lIs46kgFJK2AKrzPSWgmb2WR7GW90MXhHq6VxUwuksp33"
4 }
5
```

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize **JSON** v

```
1 {
2   "message": "Pedido já foi entregue."
3 }
```

- **RESULTADO: Entrega confirmada com sucesso.**

HTTP Casos de uso / UC05: confirmarentrega

POST http://localhost:8000/api/confirmarentrega

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

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

```
1 {
2   "numero": "1",
3   "token": "1YdMPwDttZWVORIOLi11Is46kgFJK2AKzzPSWgmb2WR7GW90MXhHq6VxUwuksp33"
4 }
5
```

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize **JSON** ▾ ↻

```
1 {
2   "message": "Entrega confirmada com sucesso."
3 }
```

8.6. Descrição de caso de Uso UC06

8.6 Descrição de Casos de Uso - UC06 - Atualização do Status do pedido

Nome:	Atualização de status
Objetivo:	O Centro de Distribuição confirma o status do pedido
Atores:	Centro de Distribuição
Pré-condição:	Pedido registrado no sistema e em fase de separação.
Pós-condição (cenário de sucesso):	Status do pedido atualizado para "Entregando", itens separados são entregues ao entregador.
Pós-condição (cenário de insucesso):	Falha na atualização do status do pedido.
Trigger:	Requisição do CD para alteração de status do pedido.
Fluxo Principal:	1.O CD envia uma requisição informando o ID do pedido e o status a ser alterado 2. Sistema altera o status e retorna mensagem de que o status foi alterado com sucesso [E1][E2]{E3}
Fluxos Alternativos:	
Fluxos de Exceção	[E1] Sistema informa que id do pedido não foi encontrado [E2] Sistema informa que o pedido já foi entregue [E3] Sistema informa que o pedido já está em rota de entrega
Pontos de Extensão:	
Regras de Negócio	[RN1] O entregador deve ser notificado quando o pedido estiver pronto para entrega.

8.6.1. Teste do caso de Uso UC06

CD informa número do pedido e token de identificação único e o status a ser alterado, recebe uma resposta confirmando alteração e informações do pedido

- **RESULTADO:** status da entrega atualizado com sucesso.

The screenshot displays a REST client interface for testing the 'UC06: atualizarstatus' endpoint. The request is a POST to 'http://localhost:8000/api/atualizarstatus' with a JSON body containing 'numero': '5', 'status_entrega': 'Entregando', and a long 'token'. The response is a JSON object with a success message and detailed delivery information.

Request:

```
1 {
2   "numero": "5",
3   "status_entrega": "Entregando",
4   "token": "1YdMPwDttZWVORIOLi1lIs46kgFJK2AKrzPSWgmb2WR7GW90MXhHq6VxUwukspYA"
5 }
6
```

Response:

```
1 {
2   "message": "Status da entrega atualizado com sucesso.",
3   "entrega": {
4     "id": 5,
5     "status_entrega": "Entregando",
6     "prev_entrega": "2023-11-28 09:30:00",
7     "idEntregador": 5
8   }
9 }
```

8.7. Descrição de caso de Uso UC07

8.7 Descrição de Casos de Uso - UC07 - Pedido de abastecimento à matriz

Nome:	Pedido de abastecimento no CD de produtos
Objetivo:	Solicitar abastecimento de estoque no Centro de Distribuição
Atores:	Vendedor(CD)
Pré-condição:	O token de autorização do CD é necessário para realizar o pedido.
Pós-condição (cenário de sucesso):	Abastecimento solicitado.
Pós-condição (cenário de insucesso):	Não foi possível registrar abastecimento
Trigger:	
Fluxo Principal:	<ol style="list-style-type: none">1. O Vendedor (CD) envia uma solicitação de abastecimento incluindo os detalhes dos itens (nome, quantidade, preço) e o token de autorização.2. O sistema valida o token e os dados da requisição.3. O sistema registra a solicitação de abastecimento e retorna uma confirmação.
Fluxos Alternativos:	
Fluxos de Exceção	[E1] Falha na confirmação <ol style="list-style-type: none">1.Sistema não envia confirmação de registro de abastecimento2.Retorna ao passo 2 do fluxo principal
Pontos de Extensão:	
Regras de Negócio	

8.7.1. Teste do caso de Uso UC07

Vendedor informa token de identificação pessoal e itens a serem abastecidos

- **RESULTADO: pedido de abastecimento realizado com sucesso.**

HTTP Casos de uso / UC07: pedirabastecimento

POST http://localhost:8000/api/pedirabastecimento

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

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

```
1 {
2   "token": "1YdMPwDttZWVORI0Li1Is46kgFJK2AKrzPSWgmb2WR7GW90MXhHq6VxUwuksp14",
3   "itens": [
4     {
5       "produto": "Produto HA",
6       "quantity": 5,
7       "price": 10.99
8     },
9     {
10      "produto": "Produto IL",
11      "quantity": 3,
12      "price": 7.50
13    }
14  ]
15 }
16
```

Body Cookies Headers (9) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "message": "Pedido de abastecimento registrado com sucesso!",
3   "abastecimento": {
4     "data_solicitacao": "2023-12-26T18:44:19.238937Z",
5     "data_recebimento": null,
6     "idVendedor": 7,
7     "idCdVendedor": 3,
8     "isRecebimento": false,
9     "itens_solicitados": [{"produto": "Produto HA", "quantity": 5, "price": 10.99}, {"produto": "Produto IL", "quantity": 3, "price": 7.5}],
10    "id": 10
11  }
12 }
```

8.8. Descrição de caso de Uso UC08 - Informa localização

8.8 Descrição de Casos de Uso - UC08 - Informa localização do pedido

Nome:	Informa a localização do pedido
Objetivo:	Permitir que o entregador atualize a localização atual do pedido em entrega.
Atores:	Entregador e Serviço de Localização
Pré-condição:	Pedido marcado como "entregando".
Pós-condição (cenário de sucesso):	Localização atual do pedido é registrada no sistema.
Pós-condição (cenário de insucesso):	Não foi possível registrar a localização do pedido.
Trigger:	Entregador obtém a localização atual através de um serviço de localização externo e envia para a API.
Fluxo Principal:	<ol style="list-style-type: none">1. O Entregador (dispositivo GPS) obtém a localização atual (latitude e longitude) através de um serviço de localização externo.2. O Entregador envia de forma automática, uma requisição para a API, incluindo o ID do pedido, o ID do entregador, e a localização atual obtida automaticamente através de um sistema de localização.3. O sistema valida as informações recebidas.4. O sistema atualiza o status de localização do pedido com os dados recebidos.5. O sistema retorna uma confirmação de sucesso para o entregador.
Fluxos Alternativos:	
Fluxos de Exceção	<p>[E1] Dados de localização inválidos ou incompletos:</p> <ol style="list-style-type: none">1. O sistema identifica dados inválidos ou incompletos e retorna uma mensagem de erro ao entregador.2. O caso de uso é finalizado sem atualização do status de localização. <p>[E2] Pedido não encontrado ou não está em status de entrega:</p>

	<ol style="list-style-type: none"> 1. O sistema identifica que o pedido não é elegível para atualização de localização e retorna uma mensagem de erro ao entregador. 2. O caso de uso é finalizado sem atualização do status de localização.
Pontos de Extensão:	
Regras de Negócio	<p>[RN1] O sistema deve validar se o pedido está em status de entrega antes de permitir a atualização da localização.</p> <p>[RN2] As informações de localização devem estar no formato correto (latitude e longitude).</p> <p>[RN3] O ID do entregador deve corresponder a um entregador válido no sistema alocado ao pedido do cliente sendo entregue</p>

8.8.1. Teste do caso de Uso UC08

Entregador informa seu, sua localização e o seu pedido assim como seu token de identificação pessoal, recebe uma resposta confirmando atualização de localização

- **RESULTADO: Localização informada com sucesso.**

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

- URL:** `http://localhost:8000/api/informarlocalizacao`
- Method:** `POST`
- Body (Request):**

```
1 {
2   "idEntregador": 1,
3   "latitude": -28.55052,
4   "longitude": -46.633308,
5   "pedido": 1,
6   "token": "1YdMPwDttZWVORI0Li1Is46kgFJK2AKrzPSWgmb2WR7GW90MXhHq6VxUwuksp33"
7 }
8
```
- Body (Response):**

```
1 {
2   "message": "Localização informada com sucesso!",
3   "posicao": {
4     "localizacao_gps": "-28.55052,-46.633308",
5     "localizacao": "2023-12-26T18:46:51.992691Z",
6     "idEntrega": 1,
7     "id": 6
8   }
9 }
```

8.9. Requisitos não funcionais.

Disponibilidade: O sistema foi projetado com ênfase na disponibilidade contínua, permitindo que os usuários acessem a API a qualquer momento sem interrupções significativas.

Usabilidade: A interface do usuário da API foi criada para ser intuitiva e de fácil navegação, garantindo que os usuários possam realizar suas tarefas com eficiência e sem complicações desnecessárias.

Segurança: Reconhecemos a importância da segurança, mas a API foi desenvolvida com foco prioritário na disponibilidade e usabilidade para atender às necessidades imediatas do projeto.

9. Estratégia de execução e especificação dos Requisitos Funcionais a serem implementados pela API

9.1. Introdução

Este relatório detalha o desenvolvimento de um projeto de software utilizando a linguagem PHP com o framework Laravel e o banco de dados MySQL. O projeto segue o padrão de design MVC (Model-View-Controller) e foi desenvolvido com o objetivo de tornar o sistema para entrega de mercadorias adquiridas em sites de venda de organizações de varejo, de forma ágil e dinâmica, seja diretamente em domicílio ou pela retirada em 'lojas físicas', bem como gerenciar pedidos e clientes de forma eficaz.

9.2. Ferramentas e Tecnologias

- **Linguagem de Programação:** PHP
- **Framework:** Laravel, versão 10.33
- **Banco de Dados:** MySQL
- **Ambiente de Desenvolvimento:** Visual Studio Code (VSCode)
- **Servidor Local:** XAMPP (Apache versão 2.4.58, PHP 8.2, phpMyAdmin versão 5.2.1)
- **Testes de API:** Postman

9.3. Desenvolvimento

9.3.1. Banco de Dados

O desenvolvimento começou com a implementação do banco de dados. As tabelas foram criadas utilizando migrations no Laravel, em vez de serem criadas diretamente no phpMyAdmin. Isso permitiu um controle de versão mais eficiente e integração direta com o framework Laravel.

9.3.2. Criação do Projeto Laravel

Após a configuração inicial do XAMPP e a criação do ambiente de desenvolvimento, um novo projeto Laravel foi iniciado. A conexão com o banco de dados MySQL foi estabelecida, garantindo a comunicação entre a aplicação e o banco de dados.

9.3.3. Models

Com o banco de dados implementado, os models correspondentes a cada tabela foram criados no Laravel. Estes models são fundamentais para o padrão MVC, permitindo a interação entre a aplicação e o banco de dados.

9.3.4. Rotas (Endpoints) da API

As rotas da API foram definidas no arquivo `api.php` do Laravel. Estas rotas são os endpoints pelos quais a aplicação interage com os usuários e outros sistemas, facilitando operações como adição e rastreamento de pedidos.

8.3.5. Desenvolvimento do Controller

O desenvolvimento do controller foi focado inicialmente no caso de uso de registros de abastecimento. O controller atua como intermediário entre a view (usuário) e o model (dados), processando as requisições e retornando as respostas adequadas.

8.3.6 Testes Realizados

Os testes foram realizados em um servidor local (`localhost:8000`) utilizando o Postman. Requisições do tipo POST foram enviadas para os endpoints definidos, e as respostas obtidas confirmaram o funcionamento esperado das funcionalidades implementadas. Além disso, a integração com o phpMyAdmin permitiu verificar diretamente no banco de dados os resultados das operações realizadas pela aplicação.

9. Considerações Finais

A implementação da API proposta neste projeto permite melhorar a eficiência e confiabilidade das entregas expressas e no rastreamento de pedidos, uma vez evidenciadas as falhas no sistema atual de entrega e logística que demora muito para atualizar o status do pedido e, muitas vezes, não dando uma perspectiva exata de onde se encontra o produto e sua estimativa de entrega na residência do cliente. A proposta desse projeto foi criar um sistema mais uniformizado e eficiente, com tecnologia real time de rastreamento de pedidos bem como a ampliação dos armazéns de mercadorias em diferentes pontos dos centros urbanos, possibilitando uma redução da distância de frete e aumentando a dinâmica e eficácia das entregas.

A API proposta tem o potencial de expandir o mercado de logística atual, considerando as tendências e necessidades emergentes do setor. A proposta do sistema permite a redução dos meios de transporte com a proximidade de grandes centros urbanos, possibilitando a realização de entregas 'a pé', via bicicletas/motocicletas, veículos autônomos, otimizando as entregas expressas e respeitando seus prazos restritos de alta demanda. Desta forma, o projeto ainda contribui com práticas mais sustentáveis e responsáveis de entrega, visando melhor sustentabilidade e aumento de responsabilidade social, especialmente nas áreas urbanas.

Este projeto me auxiliou em implementar na prática o que aprendemos em aulas de banco e modelagem de dados, engenharia de requisitos, assim como a de programação para web.

10. Desafios encontrados

Durante a fase de testes, diversos desafios foram encontrados, o que enriqueceu significativamente o aprendizado. Um dos principais desafios enfrentados foi a decisão sobre onde realizar as validações dos dados: se deveriam ser implementadas no próprio programa, utilizando as funcionalidades do Laravel, ou se deveriam ser realizadas diretamente no banco de dados, por meio de regras estabelecidas no MySQL.

Optamos por uma abordagem mista, onde algumas validações críticas, como integridade e consistência dos dados, foram implementadas no nível do banco de dados. Isso se mostrou eficaz para evitar dados inválidos, independentemente das operações realizadas pela aplicação. Já as validações relacionadas à lógica de negócios e regras de aplicação foram implementadas no Laravel, aproveitando a sua robusta camada de validação, o que facilitou a manutenção e a implementação de mudanças nas regras de negócio. Além disso, enfrentei dificuldades na implementação de JSON Web Tokens, optando no final por um token permanente sem tempo de expiração, algo que reconheço como uma área a ser aprimorada e implementada corretamente no futuro.

11. Aprendizados

Em resumo, esta experiência proporcionou um crescimento significativo em termos práticos e teóricos. A aplicação dos conceitos aprendidos em sala de aula, combinada com os desafios e soluções encontrados no desenvolvimento desta API, solidificou um entendimento profundo sobre a importância de um planejamento flexível, uma execução cuidadosa e a capacidade de adaptar-se rapidamente a novas informações e exigências.

12. Trabalhos Futuros

Como já mencionado anteriormente, a API proposta permite uma expansão na qualidade em diferentes tipos de mercado, bem como a integração com novas tecnologias de entrega emergentes, o que contribui para o avanço na estratégia de entregas mais dinâmicas e eficazes. O projeto pode ser aplicado em conjunto com diferentes parcerias estratégicas, como empresas de logística, varejo online e até startups de tecnologia para teste e expansão de sua aplicabilidade. Com isso, poderíamos dizer que as aplicações futuras do projeto são inúmeras, isso sem contar a adaptação do sistema para diferentes segmentos de mercado, como entregas de medicamentos e pequenas entregas em geral.

Visando dar melhor segurança seria desejável que os tokens ‘expirassem’ e que fosse utilizado o padrão JWT

Visando a melhoria e implementação do projeto de maneira mais ampla é possível no futuro desenvolver recursos que permitam maior personalização e flexibilidade nas opções de entrega, atendendo às necessidades específicas de cada cliente usuário.

Além disso, seria interessante programar estudos de caso detalhados em ambientes reais de entrega e implementação de um sistema testes automatizados e de feedback contínuo para melhorar e adaptar a API as evoluções da sociedade.

13. Referências Bibliográficas:

[1] What Is Outbound Logistics and Why Is It So Important to Modern Business? Disponível em:

<<https://leadgrowdevelop.com/what-is-outbound-logistics-and-why-is-it-so-important-to-modern-business/>> Acessado em 09/10/2022

[2] Inbound and Outbound Logistics: Differences, Benefits, KPIs & Challenges Disponível em: < <https://www.shipbob.com/blog/inbound-and-outbound-logistics/>> Acessado em 09/10/2022

[3] SAIBA COMO EVITAR ATRASOS DE ENTREGAS NO TRANSPORTE DE CARGAS Disponível em: < <https://diavanti.com.br/atrasos-no-transporte-de-cargas/>> Acessado em 09/10/2022

[4] Atraso na entrega: as principais causas e como evitá-las. Disponível em:<<https://cargox.com.br/blog/atraso-na-entrega/>> Acessado em 09/10/2022

[5] Atrasos na entrega de encomendas: quais as causas e o que fazer. Disponível em: <<https://www.eurosender.com/blog/pt/atraso-de-encomenda/>> Acessado em 09/10/2022

[6] CORNERSHOP BY UBER E GRUPO CARREFOUR ENTRAM NO SEGMENTO DE ENTREGAS RÁPIDAS. Disponível em : <<https://www.ecommercebrasil.com.br/noticias/cornershop-by-uber-grupo-carrefour-entregas-rapidas>> Acessado em 09/10/2022

[7] A logística do Magazine Luiza. Disponível em : <<https://saclogistica.com.br/logistica-magazine-luiza/>> Acessado em 09/10/2022

[8] Applying UML and patterns - Craig Larman - 2a edição. Disponível em:<<https://personal.utdallas.edu/~chung/SP/applying-uml-and-patterns.pdf>> Acessado em: 19/12/2023

[9] Documentação oficial Laravel, versão 10.x . Disponível em: <<https://laravel.com/docs/10.x>> Acessado em: 10/01/2024

[10] Documentação oficial PHP. Disponível em: <https://www.php.net/manual/pt_BR/> Acessado em: 11/01/2024

[11] Documentação da IDE(vscode) sobre PHP. Disponível em:

<<https://code.visualstudio.com/docs/languages/php>> Acessado em: 11/01/2024