

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Arquitetura de Software Distribuído

Alexandre Rorato Carneiro

SISTEMA DE GESTÃO DE SERVIÇOS DE LOGÍSTICA

Belo Horizonte

2022

Alexandre Rorato Carneiro

SISTEMA DE GESTÃO DE SERVIÇOS DE LOGÍSTICA

Trabalho de Conclusão de Curso de
Especialização em Arquitetura de Software
Distribuído como requisito parcial à obtenção
do título de especialista.

Orientador(a): Pedro A. Oliveira

Belo Horizonte

2022

*Dedico este trabalho a minha família pela paciência e
entendimento da minha ausência durante o
desenvolvimento deste trabalho.*

AGRADECIMENTOS

Agradeço a Deus por ter me dado saúde para conclusão deste árduo trabalho e aos professores de todas as disciplinas desta pós-graduação pelos ensinamentos que serão de grande valia na minha trajetória profissional.

RESUMO

A demanda dos sites de e-commerce por serviços de logística de entrega tem crescido bastante nos últimos anos e acentuou-se após o início do isolamento social provocado pela pandemia do coronavírus no início de 2020, com mais empresas se especializando em transporte e entregas de mercadoria, tornando este mercado mais competitivo e desafiador. A exigência do consumidor aumentou bastante em relação ao tempo de entrega, e por isso o atendimento ao cliente precisa ser otimizado. Esta proposta arquitetural visa: melhorar os seus processos de entrega com uma solução baseada em microsserviços; integrar seus sistemas com os de suas parceiras propiciando que a entrega possa ser realizada em parceria, em uma ou mais etapas do processo; suportar ambientes web e dispositivos móveis com responsividade; tratar dados massivos do sistema para transformá-los em informações de apoio à tomada de decisão, e que possam ser facilmente disponibilizados em dashboards ou planilhas por uma ferramenta de mercado que realiza gestão corporativa. Este documento está estruturado para apresentar os requisitos definidos, a modelagem arquitetural, uma prova de conceito, e no final uma avaliação da arquitetura através de cenários com suas respectivas evidências.

Palavras-chave: logística, microsserviços, API, CT-e

SUMÁRIO

1. Apresentação.....	7
1.1 Problema.....	7
1.2 Objetivo do trabalho.....	7
1.3 Definições e Abreviaturas.....	8
2. Especificação da Solução.....	9
2.1 Requisitos Funcionais.....	9
2.2 Requisitos Não Funcionais.....	10
2.3 Restrições Arquiteturais.....	11
2.4 Mecanismos Arquiteturais.....	12
3. Modelagem Arquitetural.....	13
3.1 Macroarquitetura.....	13
3.2 Descrição Resumida dos Casos de Uso / Histórias de Usuário.....	15
3.3 Visão Lógica.....	16
4. Prova de Conceito (POC) e Protótipo Arquitetural.....	29
4.1. Implementação.....	29
4.2 Interfaces e APIs.....	44
5. Avaliação da Arquitetura.....	45
5.1. Análise das abordagens arquiteturais.....	45
5.2. Cenários.....	46
5.3. Evidências da Avaliação.....	47
5.4. Resultados.....	60
6. Conclusão.....	62
REFERÊNCIAS.....	64
APÊNDICES.....	68

1. Apresentação

O tema deste trabalho foi proposto considerando a importância da área de logística. A demanda dos sites de e-commerce por serviços de logística de entrega já vinha em constante crescimento a alguns anos, acentuando-se abruptamente desde o início do isolamento social provocado pela pandemia do coronavírus. “O país registrou uma alavancagem de 400% na abertura de lojas virtuais desde o início da quarentena, e o número de compras on-line foi ampliado em 387% até meados de abril de 2020” (TPC, 2020).

1.1 Problema

Devido ao grande aumento de compras on-line, principalmente após o início do isolamento social, a demanda por serviços de logística de entrega ficou sobrecarregada, pois cresceu na mesma proporção. Contando também que os consumidores dos sites de e-commerce ficaram mais exigentes com relação ao tempo de entrega, tudo isso gerou mais concorrência entre as empresas de transporte do segmento de logística e mais desafios para atenderem as entregas com maior agilidade.

1.2 Objetivo do trabalho

O objetivo geral deste trabalho é apresentar uma proposta de arquitetura totalmente componentizável e baseada em microsserviços, com integração entre seus módulos e componentes para permitir que os sistemas legados desenvolvidos internamente ou contratados de fornecedores, e sistemas externos a serem acessados funcionem harmonicamente em um ambiente computacional único, visando proporcionar maior agilidade nas entregas e atender às metas gerais estabelecidas pela empresa.

Os objetivos específicos deste trabalho são:

1. Criar o módulo de informações cadastrais para obter e manter informações de clientes, fornecedores, depósitos e mercadorias. Dentre essas destacam-se: identificação, dados de localização, dados complementares e informações necessárias ao negócio da empresa. Essas informações têm como fonte os próprios fornecedores e clientes, ficando os dados armazenados em Enterprise Information Systems (EIS), com destaque para os sistemas legados SAF, SGR e SFC.

2. Criar o módulo de serviços aos clientes, pelo qual será possível desenhar, analisar e acompanhar todos os processos de atendimento aos clientes existentes na empresa melhorando o desempenho e a eficiências desses processos.

3. Criar módulo de gestão e estratégia que será responsável por gerir todas as atividades da empresa, com indicadores das entregas realizadas e a realizar, na forma de indicadores, representados na forma de planilha e/ou cockpit. Para este módulo será utilizada uma ferramenta de gestão corporativa adquirida no mercado.

4. Criar módulo de ciência de dados - DW e BI, responsável pela obtenção, guarda, recuperação e utilização dos dados corporativos pertinentes através de ferramentas adequadas, com recursos para tratamento de dados massivos (Big Data), mineração dos dados para apoio às tomadas de decisão. Todos os dados deste módulo são obtidos de planilhas e Sistemas Gerenciadores de Bancos de Dados (SGBDs), relacionais ou NoSQL. O uso de recursos de um Data Warehouse (DW) é essencial para as operações deste módulo.

1.3 Definições e Abreviaturas

GSL: abreviação de “Gestão de Serviços de Logística”

POC: abreviação de “Prova de Conceito”

API: abreviação em inglês de “Application Programming Interface”

DW: abreviação em inglês de “Data Warehouse”

BI: abreviação em inglês de “Business Intelligence”

VPS: abreviação em inglês de “Virtual Private Server”

CT-e: abreviação de “Conhecimento de Transporte Eletrônico”

CRUD: as 4 letras deste termo referem-se à primeira letra da palavra em inglês referente a cada operação básica de um sistema que utiliza banco de dados: Create, Read, Update e Delete.

CDC: abreviação em inglês de “Change Data Capture”, é “um meio para capturar mudanças nos dados de um determinado domínio” (CARDOSO, 2020).

Endpoint: no contexto desta arquitetura, significa a URL onde um determinado microserviço pode ser acessado por uma aplicação cliente.

Role: “Uma role ou papel é um agrupamento de permissões que pode ser concedida a usuários ou outras roles” (SANCHES, 2021).

2. Especificação da Solução

Esta seção descreve os requisitos contemplados neste projeto arquitetural, divididos em dois grupos: funcionais e não funcionais.

2.1 Requisitos Funcionais

ID	Descrição Resumida	Dificuldade (B/M/A)*	Prioridade (B/M/A)*
RF01	O sistema deve permitir para o cliente o cadastramento de uma solicitação de entrega a partir de dados da entrega informados, como endereços de origem e destino e conteúdo das cargas a serem transportadas, e se necessário, o sistema deve acionar uma transportadora parceira para realização da entrega.	M	A
RF02	O sistema deve permitir ao cliente estimar o valor do frete a pagar informando os dados de suas cargas e mercadorias.	B	A
RF03	O sistema deve permitir aos clientes o acompanhamento do andamento das suas entregas solicitadas.	B	A
RF04	O sistema deve permitir a emissão do CT-e (Conhecimento de Transporte Eletrônico) e o envio dele ao sistema SFC dos dados do CT-e de um pedido de entrega para apuração financeira.	M	A
RF05	As informações da solicitação de entrega devem ser enviadas ao SGE (Sistema de Gestão de Entrega)	M	A
RF06	O sistema deve permitir o cadastramento e manutenção de seus clientes, com as informações de CNPJ, razão social, nome comercial, endereço, telefone e e-mail.	B	A
RF07	O sistema deve permitir o cadastramento e manutenção dos fornecedores da Boa Entrega, com as informações de CNPJ, razão social, nome fantasia, endereço, telefone e e-mail.	B	M
RF08	O sistema deve permitir o cadastramento de empresas parceiras, com as informações de CNPJ, razão social, nome fantasia, endereço, telefone, e-mail e região do Brasil com maior atuação.	B	A
RF09	O sistema deve permitir o cadastramento e manutenção dos armazéns responsáveis pelos depósitos de mercadorias com as informações de CNPJ, razão social, nome comercial, endereço, telefone, e-mail.	M	A
RF10	O sistema deve permitir o cadastramento das mercadorias.	M	A

RF11	Os cadastros de novos clientes, fornecedores, armazéns e mercadorias ou alterações deles devem ser enviados ao Sistema Administrativo e Financeiro (SAF)	B	M
RF12	O sistema deve possibilitar a extração de dados das entregas por uma ferramenta de gestão corporativa adquirida no mercado que a partir desses dados irá prover informações estratégicas, em forma de planilha e /ou <i>cockpit</i> , com indicadores de entregas realizadas e a realizar.	A	A
RF13	O sistema deve possibilitar o tratamento via Big Data de seus dados gerados em banco de dados ou planilhas para que eles sejam obtidos por uma ferramenta de BI adquirida no mercado que fornecerá relatórios com informações de apoio a tomadas de decisão.	A	A
RF14	O sistema deve prover o seguro das mercadorias que fazem parte das cargas dos clientes a serem transportadas, por serem bens de terceiros.	M	A
RF15	O sistema deve prover entregas por transportadoras parceiras, principalmente em redutos ainda não cobertos pela Boa Entrega, como a Região Norte e outros 200 municípios de médio e pequeno porte que estão previstos para expansão da sua atuação, desta forma viabilizando o alcance de algumas metas previstas para este triênio, como convênio com 50 novos clientes e crescimento de 10% em termos de faturamento global.	M	A
RF16	O sistema deve permitir a atualização do status da entrega de acordo com a fase da jornada: solicitação, coleta de carga, roteirização, início do transporte, distribuição nos CDs, <i>last-mile</i> (última milha a percorrer antes da entrega ao cliente), e finalização.	B	A
RF17	O sistema deve permitir que o cliente efetue o pagamento do serviço após a sua solicitação de entrega.	M	A
RF18	O sistema deve permitir ao cliente efetuar o cancelamento da entrega, antes da sua finalização.	M	M

*B=Baixa, M=Média, A=Alta.

2.2 Requisitos Não Funcionais

ID	Descrição	Prioridade B/M/A
RNF01	O sistema deve realizar todas as entregas com tempo médio inferior a	A

	5 dias úteis.	
RNF02	O sistema deve apresentar bom desempenho. Por exemplo: conseguir cadastrar uma solicitação de entrega em menos de 400 ms.	A
RNF03	O sistema deve atender, de forma seletiva (por perfil) a clientes, fornecedores e colaboradores.	A
RNF04	O sistema deve apresentar boa manutenibilidade.	A
RNF05	O sistema deve ser testável em todas as suas funcionalidades.	M
RNF06	O sistema deve ser recuperável (resiliente) no caso de ocorrência de erro.	A
RNF07	O sistema deve estar disponível 24h, sete dias por semana.	A
RNF08	O sistema deve ser acessível nas plataformas <i>web</i> e móvel.	A
RNF09	O sistema deve utilizar APIs para o consumo de serviços	A
RNF10	Possuir interface responsiva em dispositivos móveis, como celulares e tablets, por exemplo.	A
RNF11	Ser desenvolvido utilizando recursos de gestão de configuração.	A
RNF12	O sistema deve ser desenvolvido utilizando recursos de integração contínua.	M
RNF13	O sistema deverá prover alta escalabilidade.	A
RNF14	O sistema deve ser de fácil implantação e utilização	M
RNF15	O sistema deve prover segurança no registro de entregas e nas transações financeiras associadas.	A

2.3 Restrições Arquiteturais

- R1: O sistema deve ser desenvolvido na linguagem JAVA na versão 11 do seu jdk;
- R2: Os microserviços desta solução devem ser desenvolvidos utilizando-se o framework Sprint Boot na sua versão 2.3.4.RELEASE e as dependências do Spring Cloud;
- R3: O sistema deve possuir características de aplicação distribuída: abertura, portabilidade, uso de recursos de rede.
- R4: O sistema deve ser modular e componentizado, utilizando orientação a serviços.
- R5: O sistema deve ser hospedado em nuvem híbrida, sendo a forma de hospedagem a utilização de um VPS (servidor virtual privado). Apesar de ser mais caro que uma hospedagem compartilhada, a utilização de um VPS traz mais segurança, garantindo privacidade das informações; e maior

escalabilidade, pois conforme haja um crescimento do sistema, é possível fazer facilmente um upgrade do plano contratado.

2.4 Mecanismos Arquiteturais

Análise	<i>Design</i>	Implementação
Persistência	ORM	Spring Data JPA
Persistência	Banco de dados relacional	H2 e MySQL
Front end	Single Page Application	Angular
Back end	Linguagem de programação	Java
Back end	Framework de desenvolvimento	Spring Boot com soluções do Spring Cloud
Integração com as APIs do sistema	JSON	Web Services
Integração com as APIs de sistemas legados	REST	Web Services
Desenvolvimento	IDE	Spring Tools Suite
Log do sistema	Framework	Log4j
Testes unitários do software	Framework	JUnit
Deploy	Servidor de Aplicação	Tomcat
Versionamento de código	Ferramenta de versionamento de código	Git
Segurança	Autenticação e Autorização	OAuth e JWT
Testes funcionais	Ferramenta para testes APIs	Postman

3. Modelagem Arquitetural

Esta seção apresenta a modelagem arquitetural da solução proposta, de forma a permitir seu completo entendimento visando à implementação da prova de conceito (seção 4).

3.1 Macroarquitetura

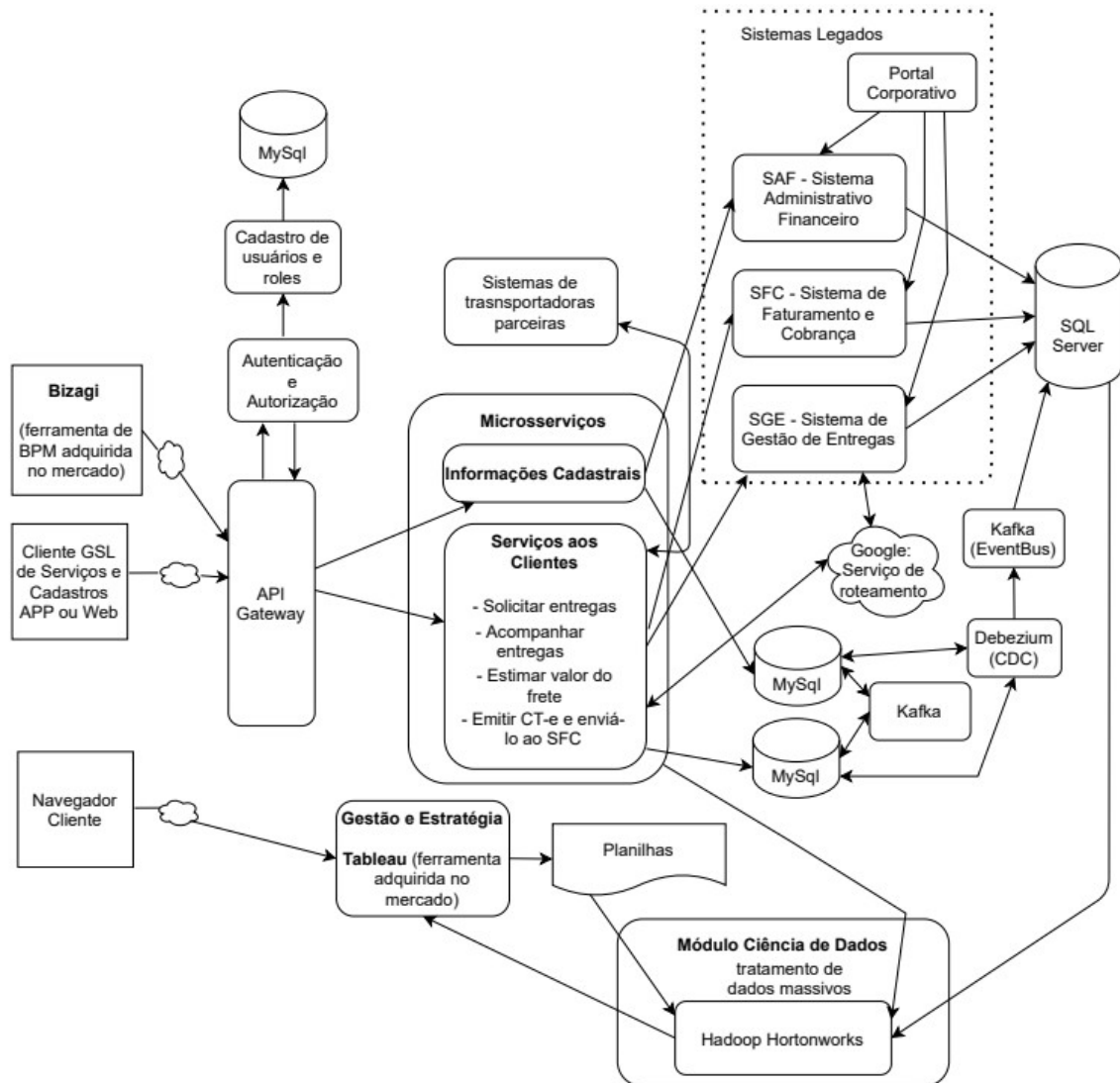


Figura 1 – Visão Geral da Solução

A figura 1 mostra o diagrama geral da solução proposta, com todos os módulos e suas interfaces. É uma arquitetura baseada em microsserviços. Para

acessar os microsserviços dos módulos de Informações Cadastrais e Serviços aos Clientes, as aplicações clientes, que podem ser soluções de front-end ou uma ferramenta de BPM, autenticam-se e após isso a API Gateway direciona sua requisição ao microsserviço desejado. Esses microsserviços integram-se com os sistemas legados SAF, SGE e SFC e com sistemas de transportadoras parceiras. Cada um dos módulos, Informações Cadastrais e Serviços aos Clientes, utilizam o seu próprio banco de dados MySql e a consistência entre eles é realizada pela troca de mensagens de eventos via Kafka. Os sistemas legados atualizam o SQL Server, e portanto este deve ser atualizado também com os eventos dos bancos MySql. Para isso foi utilizada uma ferramenta de CDC (*Change Data Capture*) para captura desses eventos e repasse para o Kafka disponibilizar o acesso a esses eventos pelo SQL Server. Para o módulo Gestão e Estratégia será adquirida no mercado a ferramenta de BI Tableau para gestão corporativa e geração de indicadores em planilhas e dashboards. Essas informações serão geradas pelo outro módulo desta arquitetura, o de Ciência de Dados, no qual será implementada uma solução de Big Data com Hadoop para tratamento dos dados corporativos oriundos de várias fontes: microsserviços de Informações Cadastrais e de Serviços aos Clientes; sistemas legados, planilhas e logs. O acesso às funcionalidades do módulo de Gestão e Estratégia é através de um navegador informado uma url.

3.2 Descrição Resumida dos Casos de Uso / Histórias de Usuário

UC01 – ESTIMAR VALOR DO FRETE	
Descrição	Permitir ao cliente estimar o valor do frete a pagar antes de fazer uma solicitação de entrega.
Atores	Cliente
Prioridade	A
Requisitos associados	RF02, RNF03, RNF07 e RNF08
Fluxo Principal	O cliente da Boa entrega deve poder realizar a estimativa de frete de forma autônoma e segura, sem o compromisso de ter que efetuar a solicitação da entrega, informando os cep's de origem e destino do transporte e os dados das cargas a serem transportadas.

UC02 – CADASTRAR PEDIDO DE ENTREGA ACIONANDO TRANSPORTADORA PARCEIRA	
Descrição	Permitir à empresa cliente solicitar uma entrega de mercadorias para seu destinatário e durante o processamento o sistema aciona uma transportadora parceira para auxiliar a entrega.
Atores	Cliente da Boa Entrega e transportadora parceira

Prioridade	A
Requisitos associados	RF01, RF10, RF15, RNF02, RNF03, RNF07, RNF08, RNF09 e RNF11
Fluxo Principal	O usuário deve poder realizar sua solicitação de entrega através desta funcionalidade, de forma autônoma e segura, informando os dados essenciais para realização da entrega: cnpj do cliente, dados do destinatário, endereço de origem, endereço de destino e dados das cargas a serem transportadas.

UC03 – EMITIR O CT-e E ENVIÁ-LO AO SFC	
Descrição	Após sua emissão, o CT-e (Conhecimento de Transporte Eletrônico) deve ser enviado ao sistema legado SFC para que ele faça a devida apuração financeira.
Atores	Colaborador da Boa Entrega
Prioridade	A
Requisitos associados	RF04, RNF03, RNF07, RNF08, RNF09 e RNF11
Fluxo Principal	O CT-e, após validado e autorizado pela SEFAZ (Secretaria da Fazenda) deve ser enviado ao sistema legado SFC para que ele faça a devida apuração financeira. Nele deverá conter várias informações, dentre elas: tipo do modal de transporte, dados de autorização da SEFAZ, natureza da prestação de serviço, dados do emitente, remetente, tomador, expedidor e recebedor, dados da carga, dados do seguro da carga e componentes do valor da prestação do serviço.

3.3 Visão Lógica

Esta seção mostra a especificação dos diagramas da solução proposta, com todos os seus componentes, propriedades e interfaces. Nas subseções a seguir são apresentados os diagramas de Classes, Componentes e Implantação.

3.3.1 Diagrama de Classes

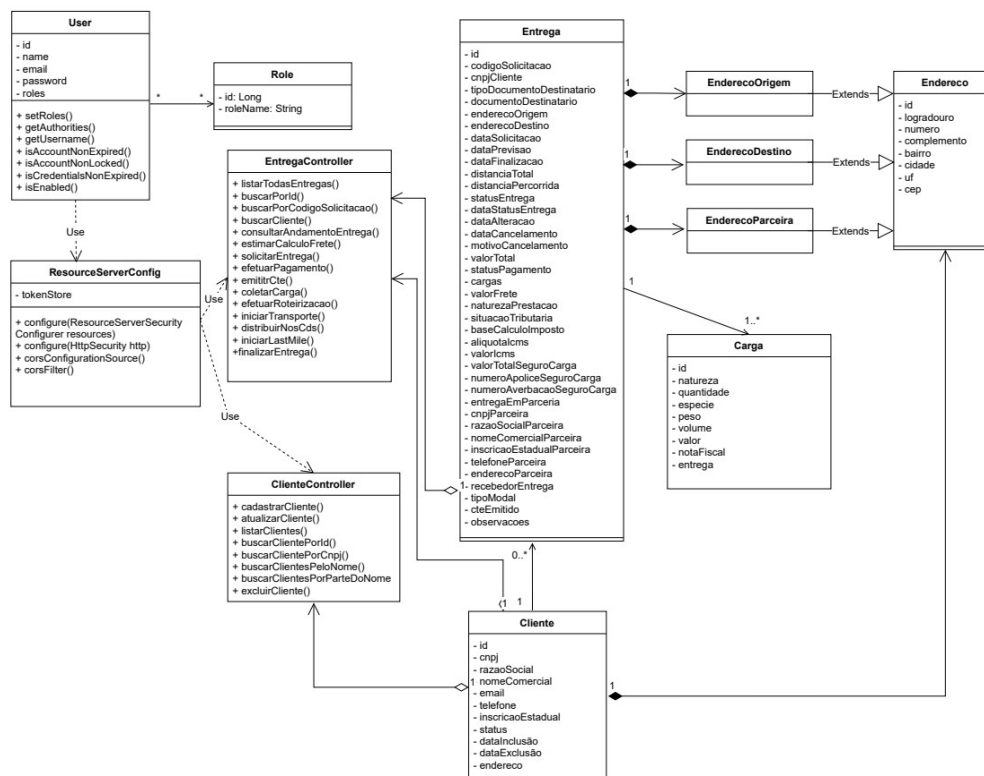


Figura 2 – Diagrama de classes.

A figura 2 apresenta todas as classes persistentes no banco de dados e algumas não persistentes com o objetivo de uma melhor contextualização. O diagrama não contempla nenhuma classe das camadas de serviço nem de repositório, apesar de algumas delas serem citadas nas descrições abaixo. Segue a descrição de cada classe deste diagrama:

- **User:** classe persistente que representa o usuário logado no sistema. Seus atributos *email* e *password* são as credenciais de acesso ao sistema. O atributo *roles* refere-se a um conjunto de roles das quais este usuário faz parte, sendo que para cada role, no microserviço de API Gateway, são atribuídos os caminhos das respectivas funcionalidades às quais seus perfis associados terão acesso.
- **Role:** classe persistente que representa um cadastro de roles (agrupamento de permissões) atribuídas aos usuários para identificação dos seus perfis.
- **ResourceServerConfig:** classe presente no microserviço do Api Gateway que recebe requisições do usuário com o token e verifica se o usuário está autorizado a acessar determinado recurso (funcionalidade) ou não.

- **EntregaController**: classe que contém todos os endpoints referentes às funcionalidades que recebem requisições para o módulo de Serviços aos Clientes (Entregas). Esta classe chama a classe de serviços *EntregaServiceImpl* para tratar as regras de negócio de cada transação, e esta chama as classes de repositório *EntregaRepository* e *CargaRepository* que chamam os métodos que persistirão no banco as classes *Entrega*, *Carga*, *EnderecoOrigem*, *EnderecoDestino* e *EnderecoParceira*.
- **ClienteController**: classe que contém todos os endpoints referentes às funcionalidades que recebem requisições para o módulo de Informações Cadastrais, especificamente relacionadas ao cadastro de clientes da Boa Entrega. Esta classe chama a classe *ClienteServiceImpl*, da camada de serviços, para tratar as regras de negócio de cada transação, e esta chama a classe *ClienteRepository* que chama os métodos que persistirão no banco as classes *Cliente* e *Endereco*.
- **Entrega**: classe que representa uma solicitação de entrega e é persistida em todas as fases da jornada de uma entrega, desde a solicitação até sua finalização ou cancelamento. Contém como parte dos seus atributos as classes *Carga*, *EnderecoOrigem*, *EnderecoDestino* e *EnderecoParceira*.
- **Carga**: classe que representa mercadorias a serem transportadas em uma jornada de entrega.
- **EnderecoOrigem**: classe que representa o endereço de origem da entrega, ou seja, é aonde a transportadora irá realizar a coleta, podendo ser o mesmo endereço do cliente ou o endereço de um galpão informado pelo cliente. Esta classe estende a classe *Endereco*.
- **EnderecoDestino**: endereço onde deverão ser entregues as mercadorias ao consumidor final. Esta classe também estende a classe *Endereco*.
- **EnderecoParceira**: endereço de uma transportadora parceira da Boa Entrega, que dependendo do destino, ela pode ser acionada para realização de transporte em uma parte do percurso ou em todo o percurso. Esta classe também estende a classe *Endereco*.
- **Endereco**: classe pai das classes *EnderecoOrigem*, *EnderecoDestino* e *EnderecoParceira*. É utilizada pelo módulo de Informações Cadastrais para representar o endereço do cliente; e no módulo de Serviços aos Clientes ela é utilizada na forma de herança para representar diferentes

endereços: os de origem e destino das mercadorias e o da transportadora parceira na realização de uma entrega.

- **Cliente:** classe que representa os dados cadastrais de um cliente da Boa Entrega. É persistida pelas operações de crud no módulo de Informações Cadastrais e também é acessada em operações do módulo de Serviços aos Clientes quando é necessário validar o cnpj do cliente informado na solicitação de uma entrega.

3.3.2 Diagrama de Componentes

Abaixo, o diagrama componentes do sistema, os quais impactaram no design da arquitetura e seleção das tecnologias. Foram organizados para serem reutilizáveis e fornecendo interfaces bem definidas de acordo com suas responsabilidades.

Figura 3 – Diagrama de Componentes.

Conforme diagrama apresentado na Figura 3, as entidades participantes da solução são:

- App Web e App Mobile – componentes utilizados para gerar o conteúdo HTML dos módulos do sistema a serem renderizados nos navegadores e dispositivos móveis, respectivamente, e podem ser utilizados tanto por clientes quanto colaboradores, fornecedores e transportadoras parceiras para acesso às funcionalidades dos módulos de Informações Cadastrais e Serviços aos Clientes.
- Navegador Cliente – componente que representa o acesso via browser às funcionalidades do Tableau, a ferramenta de gestão corporativa prevista nesta arquitetura para o módulo Gestão e Estratégia.
- Bizagi – ferramenta a ser adquirida no mercado, que utiliza solução de workflow, com o uso de Business Process Management (BPM), possibilitando desenho, análise e acompanhamento de todos os processos de atendimento ao cliente e vai integrar-se com o módulo de Serviços aos Clientes.
- API Gateway – componente utilizado para fornecer um único ponto de entrada para todos os microsserviços dos módulos de Informações Cadastrais e Serviços aos Clientes permitindo que as políticas de segurança e regras de roteamento sejam aplicadas uniformemente aos vários microsserviços desses módulos e suas instâncias (CARNELL e SÁNCHEZ, 2020). Nesta arquitetura ele também recebe as credenciais de login das aplicações clientes e a repassa para o microsserviço autorizador (componente OAuth).
- Spring Boot – framework Java que facilita o desenvolvimento de aplicações em Java, reduzindo o tempo gasto com configurações iniciais.
- Hystrix – “é o componente responsável pela implementação de modelos de resiliência em microsserviços, ou seja, interrompe falhas em cascata atuando como um circuit breaker e permite a implementação de caminhos alternativos para determinadas falhas” (CARNELL e SÁNCHEZ, 2020).
- OAuth – componente que atua como um microsserviço autorizador implementando o protocolo OAuth2. Recebe as credenciais de login no sistema, repassadas pelo API Gateway; acessa o microsserviço de Usuários e Roles (via Feign) para validar o usuário que consta nas credencias recebidas e, se válido, devolve um token de autorização para que o usuário possa então acessar as funcionalidades dos demais microsserviços cujo perfil de acesso exigido seja o mesmo.

- Usuários e Roles – microserviço responsável por manter o cadastro de usuários e suas roles, que identificam seus perfis.
- Config Server – componente que fornece uma API baseada em recursos HTTP para configuração externa e é incorporado em uma aplicação Spring Boot. Por exemplo, no arquivo `application.properties` pode ser configurado um endereço HTTP que contém configurações do sistema.
- Repositório Github – componente responsável por armazenar algumas variáveis de configuração do sistema e é referenciado pelo Config Server.
- Eureka – componente responsável por registro e descoberta, que faz com que os microserviços sejam detectáveis pelos aplicativos clientes sem que eles tenham o endereço codificado em suas implementações.
- Módulo Informações Cadastrais – contém funcionalidades que recebem e respondem a requisições no formato Json referente aos microserviços de manutenção de cadastros de clientes, fornecedores, depósitos e mercadorias. Esses microserviços farão integração com o sistema legado SAF.
- Módulo Serviços aos Clientes – Gerenciador de Entregas – contém funcionalidades que recebem requisições da ferramenta de BPM, e de aplicações Web e Mobile; e atendem às operações da jornada de entrega, como cadastramento de uma nova solicitação de entrega a realizar, acompanhamento de entregas em andamento, estimativa de cálculo de fretes e emissão do CT-e, dentre outras. Seus microserviços integram-se com os sistemas legados SFC e SGE; e também com os microserviços do módulo de Informações Cadastrais, via Feign, para consulta de dados cadastrais necessárias durante alguma transação.
- Feign: solução da lib do projeto Netflix, configurada no Spring Boot, que permite acessar funcionalidades de outros microserviços dentro do mesmo contexto do Spring Cloud a partir da criação de interfaces que devem conter: anotações com o nome do microserviço acessado e seu path; e a declaração do método referente a funcionalidade a ser acessada, anotado com o path da funcionalidade.
- Módulo Gestão e Estratégia – constituído pela ferramenta de BI Tableau, adquirida no mercado, que proverá geração de indicadores de entregas realizadas e a realizar em forma de planilhas ou cockpit, sendo que suas informações de base para geração destes indicadores serão fornecidas pelo módulo Ciência de Dados.

- Módulo Ciência de Dados – módulo responsável por gerar informações analíticas de apoio à tomada de decisão a partir dos dados corporativos oriundos de diversas fontes, através da implementação de uma solução de BigData utilizando-se de processos como ETL, Data Warehouse, Data Mining e Olap para geração destas informações, que serão acessadas pelo Tableau, no módulo Gestão e Estratégia.
- Spring Data JPA – framework do Spring que facilita a implementação da persistência de dados.
- EhCache – framework de cache responsável por armazenar em memória consultas que retornam sempre os mesmos valores para que a aplicação não precise acessar o banco de dados repetidas vezes para essa mesma consulta.
- Transportadoras Parceiras – representam as interfaces Web Services para os sistemas das transportadoras com as quais a Boa Entrega firmou contrato de parceria para entregas em regiões onde ela apresenta *marketshare* inferior a 10% ou para entregas aonde ela ainda não atua.
- Fornecedores – componente que representa qualquer fornecedor da Boa Entrega, por exemplo: a Receita Federal ou uma seguradora responsável pelo seguro de determinada carga.
- MySQL – Bancos de dados utilizados pelo microsserviços dos módulos de Informações Cadastrais, Serviços aos Clientes, e o de Usuários e Perfis.
- SQL Server – Banco de Dados utilizado pelos sistemas legados da Boa Entrega.
- Binlog – “arquivos de logs binários gerados pelo MySql que contêm todas as alterações realizadas no banco de dados, que também podem ser chamadas de eventos” (SARDINHA, 2019).
- Kafka – componente responsável pela mensageria do tipo publish/subscriber, a fim de realizar a replicação assíncrona entre os eventos dos bancos MySql de cada um dos módulos Informações Cadastrais e Serviços aos Clientes e também para realizar a replicação desses mesmos eventos (após a captura deles pelo Debezium) para o banco de dados legado SQL Server.
- Debezium – “É um novo projeto open source que implementa CDC” (YANAGA, 2017). É muito utilizada em processos de replicação de dados entre bancos de dados heterogêneos. Nesta arquitetura, será configurada para identificar as mudanças (eventos) nos bancos de dados MySQL, obtê-las dos arquivos de

Binlog e direcioná-las para um tópico do Kafka, que se encarregará de realizar a replicação desses eventos para o banco de dados SQL Server.

- SAF (Sistema Administrativo-Financeiro) – Sistema legado que consiste em uma solução de gestão administrativa completa e escalável; e que proverá informações para os microsserviços do módulo de Informações Cadastrais via Web Services.
- SGE (Sistema de Gestão de Entregas) - Sistema legado que suporta a gestão completa das entregas, permitindo a gestão de todo o processo, tanto na parte de programação e roteamento quanto na função de gestão de estoques; e que proverá as informações para os microsserviços do módulo Serviços aos Clientes via Web Services.
- SFC (Sistema de Faturamento e Cobrança) – Sistema legado que permite gerir todo o processo de faturamento e acompanhar a cobrança dos valores devidos junto aos clientes; e que receberá via Web Services informações do módulo de Serviços aos Clientes.
- Portal Corporativo – solução legada constituída por muitas funcionalidades desenvolvidas na forma de componentes, acessíveis por meio de páginas web desenvolvidas em JavaScript, HTML e CSS. Nesse portal é possível a colaboradores e clientes, após passarem por uma validação baseada em usuário/senha, acessar seletivamente os principais recursos de que necessitam, tais como: conhecer as rotas pré-existentes, escolher rotas, alterar rotas, planejar entregas, acompanhar entregas em andamento, verificar dinamicamente problemas com veículos em rota de entrega, etc.
- Google Maps – API do Google utilizada para os serviços de roteamento da Boa Entrega, consultada via Web Services tanto pelo módulo Serviços aos Clientes quanto pelo SGE.
- Servidor de FTP – contém os arquivos .csv das planilhas geradas pelo módulo Gestão e Estratégia.
- Módulo Ciência de Dados – componente macro responsável por tratar através do Hadoop da Hortonworks os dados massivos do sistema oriundos dos bancos de dados, de planilhas e de logs; e convertendo-os em informações analíticas de apoio a tomadas de decisão.
- Sqoop – nesta arquitetura, realiza a movimentação de dados do banco de dados relacional SQL Server para o HDFS do hadoop. Por ser um componente

bidirecional, o banco de dados também pode acessar dados do Hadoop via Sqoop.

- Microsoft JDBC Driver – driver para conexão do Sqoop com o SqlServer.
- Flume – componente responsável por movimentar dados não estruturados (logs e arquivos) para o HDFS do Hadoop.
- HDFS – “Sistema de arquivos distribuídos para dados estruturados, semiestruturados e não estruturados. Arquivos são divididos em bloco e armazenados com redundância no cluster” (LUCIO, 2020).
- Map Reduce – componente responsável por transformar dados em estruturas de chave valor (tarefa Map), podendo utilizar processamentos paralelos; e depois realizar a agregação destes dados por chave (tarefa Reduce) através de algum critério.
- HBase – banco de dados NoSql do tipo coluna que armazena em suas tabelas as informações geradas pelo Map Reduce e pode ser acessado em tempo real.
- Hcatalog – componente responsável por criar e prover acesso no Hadoop aos metadados Hive para ferramentas internas ou externas. Possui interoperabilidade com Pig, Map Reduce e Hive e oferece interface de acesso Rest para metadados Hive (LUCIO, 2020).
- Pig – componente executado sobre o Hadoop e que fornece uma linguagem (Pig Latin) para escrita de scripts para realização de extrações, transformações e carregamentos (ETL). Por ser uma linguagem de alto nível, é uma alternativa à escrita de programas do tipo Map Reduce.
- Hive – “interface SQL para o Hadoop que possibilita sumarização de dados, consultas ad-hoc e análise de grandes volumes de dados” (LUCIO, 2020). Nesta arquitetura, a ferramenta de BI, Tableau, conecta-se com o Hive para obtenção das informações tratadas pelo Hadoop. O Hive também faz “acesso a arquivos residentes no HDFS e a informações do HBase” (LUCIO, 2020). Nesta arquitetura, o Hive pode ser considerado um data warehouse porque “facilita a consulta e gerenciamento de grandes massas de dados residentes em sistema de armazenamento distribuído” (LUCIO, 2020).
- Hive ODBC Driver – driver que permite a conexão entre a ferramenta Tableau e o Hive do Hadoop da Hortonworks.

3.3.3 Diagrama de Implantação

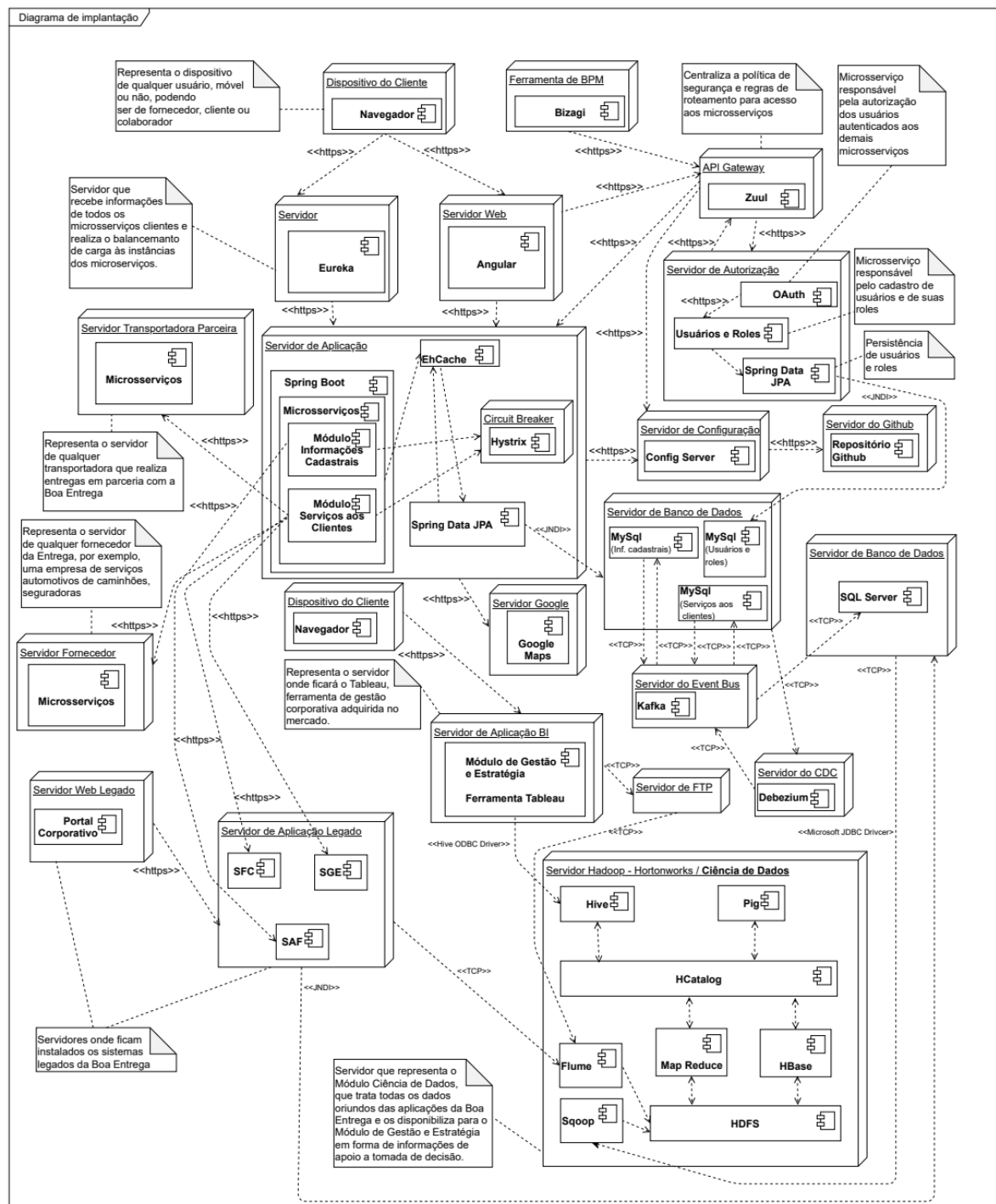


Figura 4 – Diagrama de Implantação.

Conforme diagrama apresentado na Figura 4, as entidades participantes da solução são:

- Navegador – Componentes com esse nome representam os browsers que irão acessar o sistema, podendo ser de clientes, fornecedores ou colaboradores.
- Ferramenta de BPM – servidor onde será instalada a solução de workflow adquirida no mercado (para esta arquitetura foi escolhida a Bizagi) que permite desenhar, analisar e acompanhar todos os processos de atendimento aos clientes da Boa Entrega. Ela irá integrar-se com os microsserviços do módulo de Serviços aos Clientes (no servidor de aplicação) via acesso ao API Gateway.
- Eureka – servidor onde ficará o serviço de registro e descoberta, que faz com que os microsserviços sejam detectáveis pelos aplicativos clientes sem que eles tenham o endereço codificado em seus aplicativos. Também efetua o balanceamento de carga às instâncias dos microsserviços.
- API Gateway – servidor que conterà o microsserviço Zuul (Api Gateway) responsável por fornecer um único ponto de entrada para todos os microsserviços para que as políticas de segurança e regras de roteamento sejam aplicadas uniformemente aos vários microsserviços e suas instâncias (CARNELL e SÁNCHEZ, 2020).
- Zuul – nome dados ao microsserviço que realiza o papel do API Gateway, solução responsável por efetuar o roteamento das requisições recebidas dos diversos dispositivos para seus respectivos microsserviços de forma dinâmica.
- Servidor de Autorização – acessado via Api Gateway e responsável pela autorização dos usuários autenticados aos diversos microsserviços. Nele estão os microsserviços OAuth, de autorização; e o microsserviço que mantém o cadastro dos usuários e suas roles.
- Servidor Web – servidor onde será instalado o portal das ferramentas adquiridas pela Boa Entrega, sendo elas o BPM, uma ferramenta de gestão corporativa e uma ferramenta de BI, além das soluções de front-end desenvolvidas.
- Servidor de Aplicação – contém todos os módulos da Boa Entrega com seus respectivos microsserviços, mais a camada de persistência com Spring Data JPA, o serviço de cache e o Circuit Break com Hystrix para tratar a interrupção de falhas em cascata. Os microsserviços dos módulos de Informações

Cadastrais e de Serviços aos Clientes são gerenciados pelo Spring Boot e acessados pelos usuários via Api Gateway.

- Servidor de Aplicação BI – servidor onde fica instalado a ferramenta Tableau, adquirida no mercado, que faz parte do módulo de Gestão e Estratégia, e poder acessada pelo browser do navegador do cliente para geração de indicadores de entregas.
- Servidor Hadoop (Hortonworks) – servidor onde fica instalada toda a estrutura de Hadoop que faz o tratamento de dados referente ao módulo Ciência de Dados. Os componentes presentes nela são: Sqoop, Flume, HDFS, HBase, Map Reduce, Hcatalog, Pig e Hive.
- Servidor de Configuração – servidor onde fica instalado o componente Config Server, responsável pela obtenção via https de configurações do Sistema armazenadas externamente, no repositório Github.
- Servidor do Github – repositório onde ficam armazenadas externamente algumas configurações do sistema. É acessado pelo Config Server.
- Servidor Web legado – servidor onde fica instalado o Portal Corporativo da Boa entrega.
- Servidor de Aplicação Legado – servidor onde ficam instaladas as aplicações legadas da Boa Entrega: SAF, SFC e SGE.
- Servidor Transportadora Parceira – servidor de terceiro, que representa qualquer transportadora parceira ou que venha a ser parceira da Boa Entrega e que será acessado sempre que houver necessidade de entregas em regiões que a Boa Entrega não atende, ou por qualquer outro motivo que vise ganho de agilidade na entrega.
- Servidor Fornecedor – servidor de terceiro, que representa qualquer fornecedor de serviços, por exemplo, uma oficina automotiva especializada em veículos de carga, uma seguradora e a Receita Federal, dentre vários possíveis.
- Servidor de Banco de Dados – nesta arquitetura há dois servidores de banco de dados: em um deles ficam os bancos MySQL dos microsserviços dos módulos de Informações Cadastrais, de Serviços aos Clientes e o microsserviço de cadastro de usuários e roles; e no outro fica o SQL Server, banco acessado pelos sistemas legados.
- Servidor do Event Bus – servidor onde fica o sistema de mensageria Kafka.
- Servidor do CDC – servidor onde fica a ferramenta Debezium, que captura os eventos do MySQL e os publica em um tópico do Kafka.

- Servidor de FTP – servidor onde ficam os arquivos gerados pelo módulo Gestão e Estratégia
- Servidor de Big Data / Ciência de Dados – servidor que atende ao módulo Ciência de Dados, onde as informações do banco de dados e de arquivos são transformadas, mineradas e disponibilizadas em modo analítico para a ferramenta Tableau no módulo de Gestão e Estratégia. Contém os componentes de ETL, DataWarehouse, Data Mining e OLAP para transformação dos dados oriundos de diversas fontes em informações de apoio a tomadas de decisão.
- Servidor Google – servidor que fornece serviços de geolocalização via Google Maps. É acessado tanto pelo sistema legado SGE quanto pelos microserviços do módulo Serviços aos Clientes.

4. Prova de Conceito (POC) e Protótipo Arquitetural

A prova de conceito desta arquitetura foi desenvolvida com o intuito de atender as necessidades da Boa Entrega, que são os objetivos descritos no item 1.2.

Os casos de uso implementados nesta prova de conceito são os seguintes, já descritos no item 3.2 deste documento:

- UC01 – Estimar valor do Frete.
- UC02 – Cadastrar solicitação de entrega acionando transportadora parceira.
- UC03 – Emitir o CT-e (Conhecimento de Transporte Eletrônico) e enviá-lo ao SFC.

4.1. Implementação

Nesta implementação foi utilizado o framework Spring Boot para todos os microserviços desta solução, sendo que os microserviços que fazem parte do Sistema GSL da Boa Entrega utilizaram também as dependências do Spring Cloud, pois elas facilitam a integração entre os microserviços numa arquitetura distribuída. Dentre seus vários recursos, foram implementados: configuração distribuída; roteamento com o componente API Gateway; registro de serviço e descoberta (componente Eureka); chamadas entre os microserviços de módulos diferentes; e implementações de segurança, como autenticação e autorização com OAuth2. O Spring Boot foi utilizado por ser um framework que facilita o desenvolvimento sem necessidade de muitas configurações iniciais. Em cada microserviço foram implementadas

as funcionalidades que receberão as requisições no formato Json utilizando o padrão REST. Também foram implementados microserviços fora do contexto do Spring Cloud para representarem serviços externos ou sistema do legado da Boa Entrega e que retornam informações mockadas quando acionados.

Os serviços dos módulos de Serviços aos Clientes e de Informações Cadastrais foram implementados com as seguintes camadas:

- Controller – realiza a interface com a aplicação chamadora da funcionalidade; define o mapeamento da uri de acesso; recebe as requisições e devolve as repostas de acordo com o layout; e possuem o mapeamento do Swagger.

- Service – camada que implementa os métodos responsáveis exclusivamente pelas regras de negócio da aplicação, e que são chamados pela camada controller.

- Repository – camada que possui métodos responsáveis por prover informações do banco de dados ou métodos para persistência, através do Spring Data JPA.

Na figura abaixo consta a estrutura da implementação desta POC:

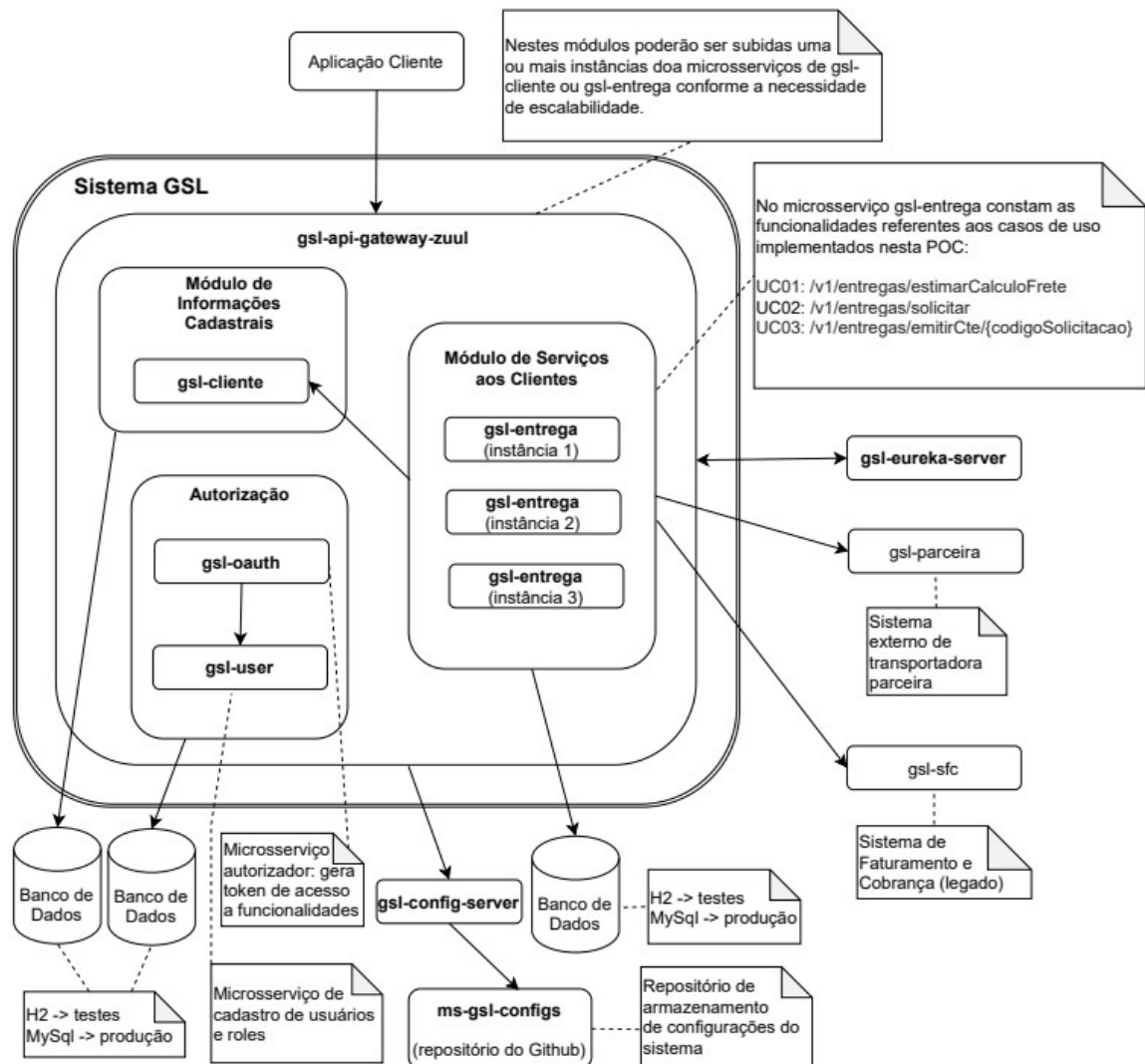


Figura 5 – Estrutura da implementação da POC.

Na figura acima são mostrados os microsserviços do Sistema GSL da Boa Entrega e também os microsserviços implementados para representarem sistemas externos e de legado.

Antes da realização de uma transação é necessário o login do usuário na aplicação, através do envio pela aplicação cliente das credenciais da aplicação e do usuário. Estes dados tem como destino o microsserviço autorizador `gsl-auth` passando antes pelo microsserviço de API Gateway, o `gsl-api-gateway-zull`. O `gsl-auth` acessará o microsserviço `gsl-users` para verificar o cadastro de usuários e roles e autenticá-lo (se suas credenciais estiverem corretas). E então, gerar um token de autorização para o aplicativo cliente. Seguem duas imagens, sendo que a primeira refere-se às variáveis de ambiente definidas no Postman; e a segunda, um exemplo de login, utilizando-se destas variáveis:

POST `{{api-gateway}}/gsl-oauth/oauth/token` o path /oauth/token é uma rota padrão do OAuth2, portanto não está explícita no código

Params Authorization Headers (9) **Body** Pre-request Script Tests Settings Cookies

	VARIABLE	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ
<input checked="" type="checkbox"/>	api-gateway	http://localhost:8765	http://localhost:8765
<input checked="" type="checkbox"/>	config-host	http://localhost:8888	http://localhost:8888
<input checked="" type="checkbox"/>	client-name	gslappnamexyz	gslappnamexyz
<input checked="" type="checkbox"/>	client-secret	gslappsecret\$y%Y!15&	gslappsecret\$y%Y!15&
<input checked="" type="checkbox"/>	username	jose@gmail.com	jose@gmail.com
<input checked="" type="checkbox"/>	password	123456	123456

```

1
2
3
4
5
6
7
"access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE2NDA3MjA2MTcsInZlcnFtZSI6ImNvc2VhZ21hbnV5Y29tIiwiaXN0aG9yaXRpZXMiOiJ1b3R5bGV9DTE1FTIRI",
"token_type": "bearer",
"expires_in": 86399,
"scope": "read write",
"jti": "df5085bb-1e89-49f8-aa08-18c63f0ab986"

```

token de autorização gerado

Com o token obtido, basta acessar com ele as funcionalidades de negócio desejadas, cujas transações serão executadas somente se o usuário tiver perfil de acesso a ela; caso contrário, retornará mensagem de erro. No item 5.3, nas evidências do cenário do atributo de qualidade segurança, constam imagens de exemplo de acesso autorizado e não autorizado a uma funcionalidade.

As funcionalidades de negócio estão disponibilizadas nos microsserviços dos módulos de Serviços aos Clientes e de Informações Cadastrais, conforme a figura 5:

- **gsl-entrega:** contém todas as funcionalidades referentes a uma jornada de entrega, com destaque para as que estão relacionadas aos casos de uso desta POC e citadas no quadro de comentário anexo ao Módulo de serviços aos clientes, na

figura 5. Conforme a necessidade de escalabilidade, pode-se subir mais instâncias deste microsserviço.

- **gsl-cliente:** microsserviço de cadastro de clientes, responsável por manter atualizada a base de dados de clientes da Boa Entrega, sendo que suas funcionalidades são somente operações básicas de CRUD. Nesta POC, ele é bastante acessado por algumas funcionalidades do microsserviço gsl-entrega, que em algum momento de suas transações necessitam consultar um cliente.

Por se tratar de uma POC com um número pequeno de funcionalidades, foram desenvolvidos somente um microsserviço dentro de cada módulo. No entanto, o Módulo de Serviços aos clientes poderia ter tantos microsserviços quanto fossem necessários caso houvesse uma quantidade muito maior de funcionalidades, para uma melhor separação de responsabilidades e mais facilidade para manutenções. E quanto ao Módulo de Informações Cadastrais, ele poderia comportar também outros microsserviços, que seriam responsáveis pelos cadastros de fornecedores, armazéns e mercadorias, por exemplo.

Nesta POC também foram implementados os microsserviços gsl-parceira e gsl-sfc, que representam, respectivamente, um sistema externo e um sistema legado, sendo que qualquer um deles poderá ser acionado em alguma etapa da jornada de uma entrega. Eles estão explicados na seção 4.2 (Interfaces e APIs) deste documento.

Nas imagens abaixo, extraídas do Swagger, constam os endpoints referentes às funcionalidades do Módulo de Serviços aos Clientes (gsl-entrega) referentes aos casos de uso implementados nesta POC:

- UC01 – Estimar valor do frete:

GET /v1/entregas/estimarCalculoFrete Estimar cálculo do frete

Parameters Try it out

Name	Description
calculoFreteRequest * required object (body)	calculoFreteRequest Example Value Model <pre>{ "cargas": [{ "especie": "alimentos", "natureza": "granel", "notaFiscal": "NF0005", "peso": 3, "quantidade": 5, "valor": 57, "volume": 2 }], "cepDestino": "string", "cepOrigem": "string" }</pre>

Parameter content type
application/json

Responses Response content type: application/json

Code	Description
200	Estimativa de cálculo realizada

Example Value Model

```
{
  "fatorCubagemKgPorM3": 300,
  "observacao": "No cálculo foi considerado o peso cubado da carga",
  "pesoCubado": 4500,
  "pesoTotal": 100,
  "precoPorKg": 1.5,
  "valorFrete": 6750,
  "volumeTotal": 15
}
```

- UC02 – Cadastrar solicitação de entrega:

POST /v1/entregas/solicitar Solicitar entrega Try it out

Parameters

Name	Description
entregaRequest <small>required</small> object (body)	entregaRequest Example Value <small>Model</small>

```
{
  "cargaRequest": [
    {
      "especie": "alimentos",
      "natureza": "granel",
      "notaFiscal": "NF0005",
      "peso": 3,
      "quantidade": 5,
      "valor": 57,
      "volume": 2
    }
  ],
  "cnpjCliente": 88200356000150,
  "documentoDestinatario": 27365467096,
  "enderecoDestino": {
    "bairro": "Centro",
    "cep": 69000001,
    "cidade": "Manaus",
    "complemento": "esquina",
    "logradouro": "Rua XV de Novembro",
    "numero": 100,
    "uf": "AM"
  },
  "enderecoOrigem": {
    "bairro": "Centro",
    "cep": 69000001,
    "cidade": "Manaus",
    "complemento": "esquina",
    "logradouro": "Rua XV de Novembro",
    "numero": 100,
    "uf": "AM"
  },
  "naturezaPrestacao": "16556 - Transporte a estabelecimento comercial",
  "observacoes": "Carga contém produtos perecíveis",
  "tipoDocumentoDestinatario": "CPF"
}
```

Parameter content type: application/json

Responses		Response content type	application/json
Code	Description		
200	OK		
	Example Value Model		
	<pre>{ "aliquotaIcms": 12, "baseCalculoImposto": 4500, "cargas": [{ "especie": "alimentos", "id": 1, "natureza": "granel", "notaFiscal": "NF0005", "peso": 3, "quantidade": 5, "valor": 57, "volume": 2 }], "cnpjCliente": "80200396000150", "cnpjParcelara": "32614634000120", "codigoSolicitacao": "116922963214", "cteEmitido": true, "dataAlteracao": "06-11-2021 19:15:50", "dataCancelamento": "06-11-2021 15:17:12", "dataFinalizacao": "06-11-2021 19:08:45", "dataPagamento": "06-11-2021 16:22:31", "dataPrevisao": "06-11-2021", "dataSolicitacao": "05-11-2021 12:09:29", "dataStatusEntrega": "06-11-2021 19:08:45", "distanciaPercorrida": 16,}</pre>		
201	Solicitação de entrega cadastrada		
404	Cliente não encontrado		

● UC03 – Emitir o CT-e:

POST

/v1/entregas/emitirCte/{codigoSolicitacao}

Emitir o CT-e do transporte a ser realizado ao sistema legado SFC para que ele realize os devidos controles.

Parameters

Try it out

Name

Description

codigoSolicitacao

*

required

integer(\$int64)

(path)

Código de solicitação da entrega

codigoSolicitacao - Código de solicitação da

Responses

Response content type

application/json

Code

Description

200

OK

Example Value

Model

```
{  "cte": {    "componentesValorPrestacaoServico": {      "componenteValor": {        {          "nome": "Frete Valor",          "valor": 6750        }      }    },    "dacte": {      "chaveAcesso": "33.4444.55.555.444/1111-88-22-000-000.000.111-100.005.111-2",      "dataHoraEmissao": "06-11-2021 10:43:43",
```

	<pre> "dataHoraGeracaoProtocolo": "06-11-2021 10:43:43", "folha": "1/1", "inscricaoSuframa": "90068714", "modelo": "S7", "numero": 1, "numeroProtocoloAutorizacaoUso": "328448663319735", "serie": 1, "textoChaveAcesso": "Consulta de autenticidade no portal nacional de CT-e, no site da Sefaz Autorizadora, ou em http://cte.fazenda.gov.br", "titulo": "Documento Auxiliar do Conhecimento de Transporte Eletrônico" }, "dadosCarga": { "cubagemM3": 300, "outrasCaracteristicasCarga": "granel", "pesoAferido": 4500, "pesoBruto": 100, "produtoPredominante": "frutas", "valorTotalMercadoria": 6750, "volume": 15 }, "dadosDestinatario": { "cpfCnpj": "13814889000192", "endereco": { "bairro": "Centro", "cep": "01058-000", "cidade": "São Paulo", "complemento": "Galpão 2", "logradouro": "Rua Duque de Caxias", </pre>		
--	--	--	--

```

    "numero": 225,
    "uf": "sp"
  },
  "inscricaoEstadual": "518.208.406.485",
  "nomeRazaoSocial": "BOA ENTREGA LTDA.",
  "telefone": "(11)5773-8532",
  "tipoDocumento": "CNPJ"
},
"dadosEmittente": {
  "cpfCnpj": 13814889000192,
  "endereco": {
    "bairro": "Centro",
    "cep": "01958-050",
    "cidade": "São Paulo",
    "complemento": "galpão 2",
    "logradouro": "Rua Duque de Caxias",
    "numero": 225,
    "uf": "SP"
  },
  "inscricaoEstadual": "518.208.406.485",
  "nomeRazaoSocial": "BOA ENTREGA LTDA.",
  "telefone": "(11)5773-8532",
  "tipoDocumento": "CNPJ"
},
"dadosExpedidor": {
  "cpfCnpj": 13814889000192,
  "endereco": {

```

```

    "bairro": "Centro",
    "cep": "01958-050",
    "cidade": "São Paulo",
    "complemento": "galpão 2",
    "logradouro": "Rua Duque de Caxias",
    "numero": 225,
    "uf": "SP"
  },
  "inscricaoEstadual": "518.208.406.485",
  "nomeRazaoSocial": "BOA ENTREGA LTDA.",
  "telefone": "(11)5773-8532",
  "tipoDocumento": "CNPJ"
},
"dadosRecebido": {
  "cpfCnpj": 13814889000192,
  "endereco": {
    "bairro": "Centro",
    "cep": "01958-050",
    "cidade": "São Paulo",
    "complemento": "galpão 2",
    "logradouro": "Rua Duque de Caxias",
    "numero": 225,
    "uf": "SP"
  },
  "inscricaoEstadual": "518.208.406.485",

```

```

    "nomeRazaoSocial": "BOA ENTREGA LTDA.",
    "telefone": "(11)5773-8532",
    "tipoDocumento": "CNPJ"
  },
  "dadosEmittente": {
    "cpfCnpj": 13814889000192,
    "endereco": {
      "bairro": "Centro",
      "cep": "01958-050",
      "cidade": "São Paulo",
      "complemento": "galpão 2",
      "logradouro": "Rua Duque de Caxias",
      "numero": 225,
      "uf": "SP"
    },
    "inscricaoEstadual": "518.208.406.485",
    "nomeRazaoSocial": "BOA ENTREGA LTDA.",
    "telefone": "(11)5773-8532",
    "tipoDocumento": "CNPJ"
  },
  "dadosSeguroCarga": {
    "nomeResponsavel": "Beatriz Fernandes Carvalho",
    "nomeSeguradora": "Porto Seguro S.A.",
    "numeroApolice": 7989457344,
    "numeroAverbacao": 1606264
  }
}

```

```

    "nomeRazaoSocial": "BOA ENTREGA LTDA.",
    "telefone": "(11)5773-8532",
    "tipoDocumento": "CNPJ"
  },
  "dadosTomador": {
    "cpfCnpj": 13814889000192,
    "endereco": {
      "bairro": "Centro",
      "cep": "01958-050",
      "cidade": "São Paulo",
      "complemento": "galpão 2",
      "logradouro": "Rua Duque de Caxias",
      "numero": 225,
      "uf": "SP"
    },
    "inscricaoEstadual": "518.208.406.485",
    "nomeRazaoSocial": "BOA ENTREGA LTDA.",
    "telefone": "(11)5773-8532",
    "tipoDocumento": "CNPJ"
  },
  "formaPagamento": "A pagar",
  "naturezaPrestacao": {
    "codigoNomeNaturezaPrestacao": "16556 - Transporte a estabelecimento comercial",
    "localInicioPrestacao": "São Paulo - SP",
    "localTerminoPrestacao": "Fortaleza - CE"
  },
  "tipoCte": "Normal",
  "tipoModal": "rodoviário",

```

```

    "tipoServico": "Normal",
    "tipoTomadorServico": "Destinatário"
  },
  "emissaoOk": true,
  "mensagemDoSfc": "Cte recebido com sucesso. Apuração financeira iniciada. Protocolo de recebimento pelo SFC: 208911820879169"
}

```

201 CT-e emitido ao SFC

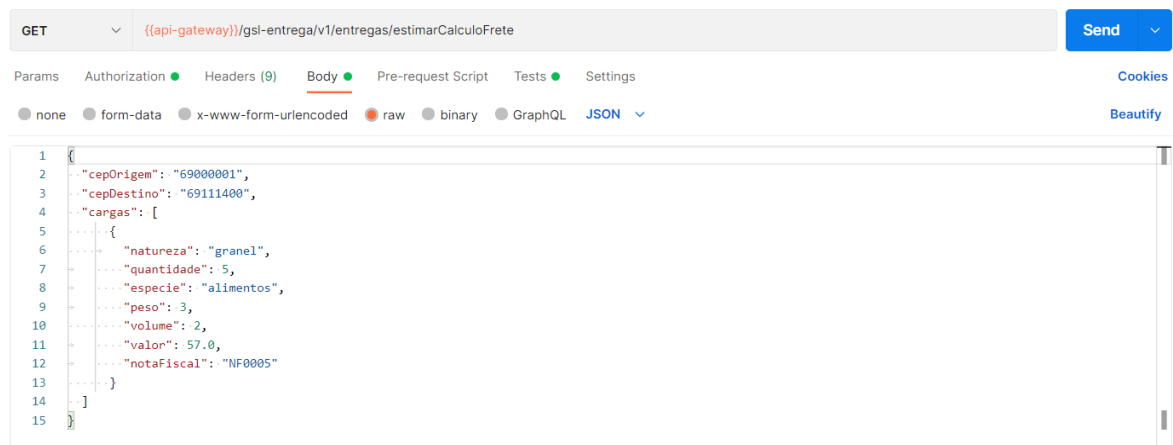
404 Entrega não encontrada

422 Operação não efetuada

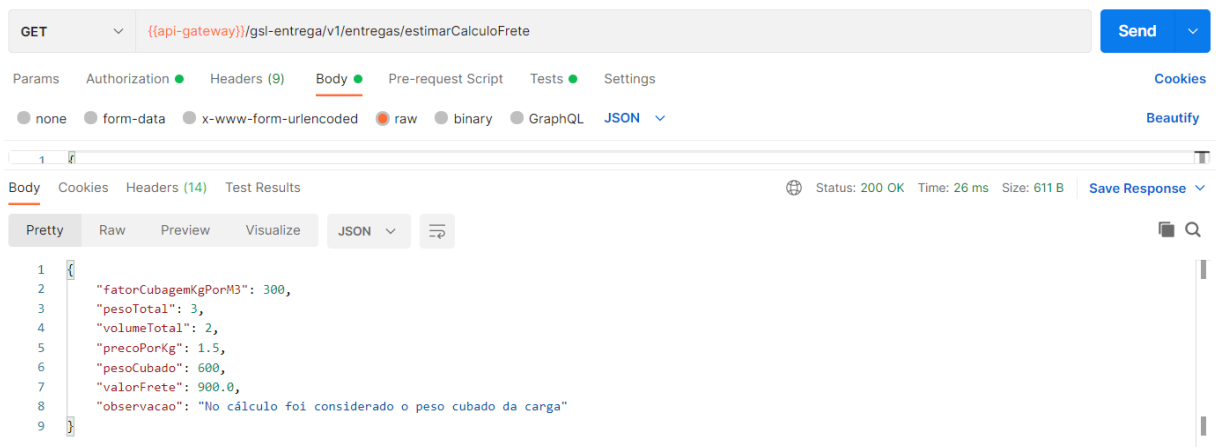
Os casos de uso desta POC apresentam as seguintes interfaces, exemplificadas com chamadas via Postman:

- UC01 – Estimar valor do Frete.

- Requisição:



- Resposta:



- UC02 – Cadastrar pedido de entrega acionando transportadora parceira.

- Requisição:

POST ▼ {{api-gateway}}/gsl-entrega/v1/entregas/solicitar ... Send ▼

Params Authorization ● Headers (9) ● Body ● Pre-request Script Tests ● Settings Cookies Beautify

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

```

1  {
2    "cnpjCliente": "80200396000150",
3    "tipoDocumentoDestinatario": "CPF",
4    "documentoDestinatario": "27365467096",
5    "enderecoOrigem": {
6      "logradouro": "Rua XV de Novembro",
7      "numero": "111",
8      "complemento": "esquina",
9      "bairro": "Centro",
10     "cidade": "Manaus",
11     "uf": "AM",
12     "cep": "69000001"
13   },
14   "enderecoDestino": {
15     "logradouro": "Rua Correa Neto",
16     "numero": "999",
17     "complemento": "casa 2",
18     "bairro": "Alvorada",
19     "cidade": "Parintins",
20     "uf": "AM",
21     "cep": "69111400"
22   },
23   "cargaRequests": [

```

```

24     {
25       "natureza": "granel",
26       "quantidade": 5,
27       "especie": "alimentos",
28       "peso": 3,
29       "volume": 2,
30       "valor": 57.0,
31       "notaFiscal": "NF0005"
32     },
33     {
34       "natureza": "geral",
35       "quantidade": 1,
36       "especie": "brinquedos",
37       "peso": 1,
38       "volume": 2,
39       "valor": 78.0,
40       "notaFiscal": "NF0006"
41     },
42     {
43       "natureza": "perigosa",
44       "quantidade": 1,
45       "especie": "substancias oxidantes",
46       "peso": 4,

```

```

47       "volume": 2,
48       "valor": 150.0,
49       "notaFiscal": "NF0007"
50     },
51     {
52       "natureza": "frigorificada",
53       "quantidade": 11,
54       "especie": "frutas",
55       "peso": 3,
56       "volume": 2,
57       "valor": 127.0,
58       "notaFiscal": "NF0008"
59     },
60     {
61       "natureza": "geral",
62       "quantidade": 2,
63       "especie": "alimentos",
64       "peso": 3,
65       "volume": 2,
66       "valor": 57.0,
67       "notaFiscal": "NF0009"
68     }
69   ],

```

```

70   "naturezaPrestacao": "16556 - Transporte a estabelecimento comercial",
71   "observacoes": "contem produtos pereciveis"
72 }

```

• Resposta: o Json de resposta tem vários atributos iguais aos da requisição e por isso foram omitidas as imagens de algumas linhas para que esta evidência não fique muito extensa. Portanto, as imagens abaixo são das linhas que possuem informações geradas pela funcionalidade, como exemplo o código de solicitação gerado, na primeira imagem; e nas duas últimas o atributo *entregaEmParceria* com o valor *true*, indicando que na transação foi acionada uma transportadora parceira, e na sequência dos atributos, dados desta transportadora parceira.

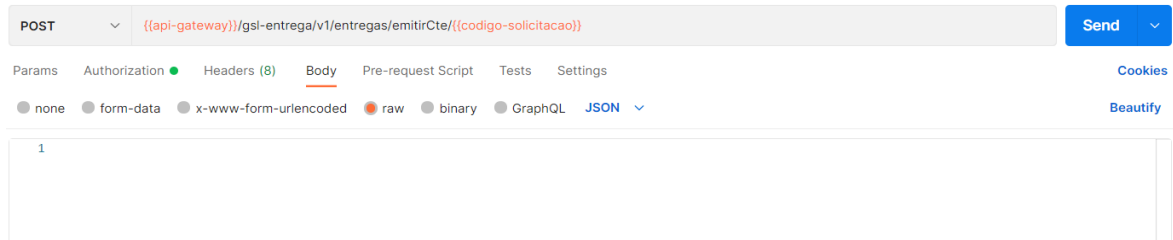
```

POST ((api-gateway))/gsi-entrega/v1/entregas/solicitar...
Send

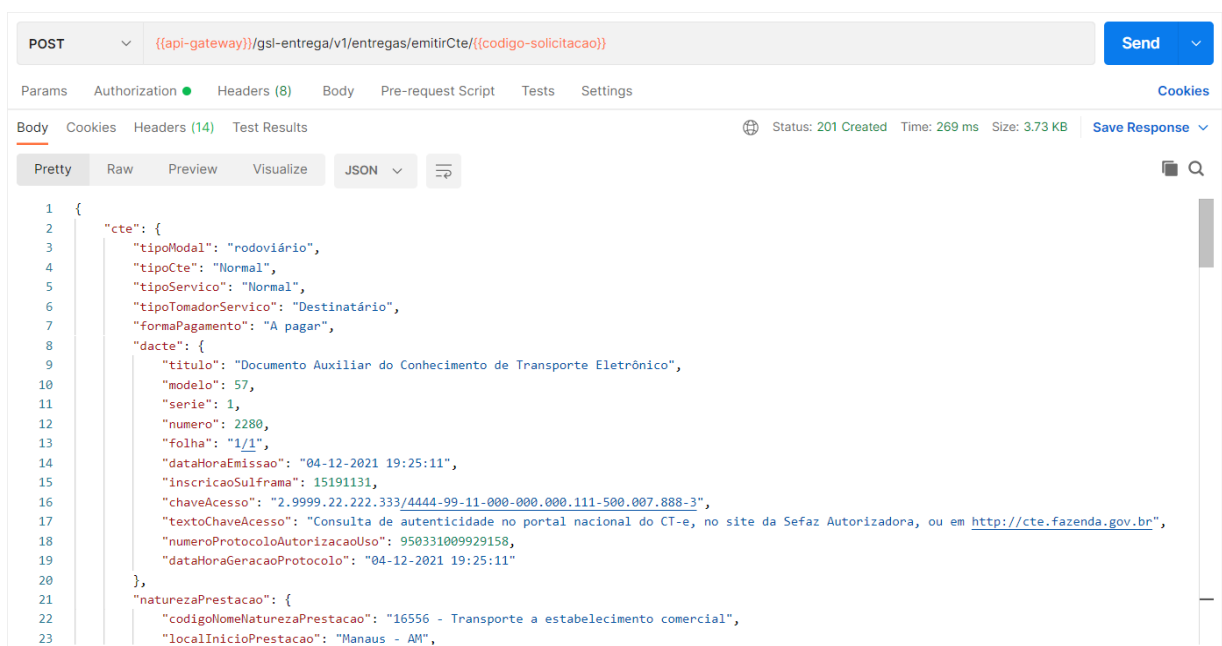
Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies
Body Cookies Headers (14) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "id": 3,
3   "codigoSolicitacao": 895068484643,
4   "cnpjCliente": 80200396000150,
5   "tipoDocumentoDestinatario": "CPF",
6   "documentoDestinatario": 27365467096,
7   "enderecoOrigem": {
8     "id": 8,
9     "logradouro": "Rua XV de Novembro",
10    "numero": "111",
11    "complemento": "esquina",
12    "bairro": "Centro",
13    "cidade": "Mauaus",
14    "uf": "AM",
15    "cep": "69000001"
16  },
17  "enderecoDestino": {
18    "id": 7,
19    "logradouro": "Rua Correa Neto",
20    "numero": "999",
21    "complemento": "casa 2",
22    "bairro": "Alvorada",
23    "cidade": "Parintins",
24  },
25  },
26  "dataSolicitacao": "04-12-2021 18:23:08",
27  "dataPrevisao": "06-12-2021",
28  "dataFinalizacao": null,
29  "distanciaTotal": 250.57,
30  "distanciaPercorrida": 0,
31  "statusEntrega": "SOLICITACAO",
32  "dataStatusEntrega": "04-12-2021 18:23:08",
33  "dataAlteracao": null,
34  "dataCancelamento": null,
35  "motivoCancelamento": null,
36  "valorTotal": 4500.0,
37  "statusPagamento": "PENDENTE",
38  "dataPagamento": null,
39  },
40  "valorFrete": 4500.0,
41  "naturezaPrestacao": "16556 - Transporte a estabelecimento comercial",
42  "situacaoTributaria": "00 - Tributação normal ICMS",
43  "baseCalculoImposto": 4500.0,
44  "aliquotaIcms": 12,
45  "valorIcms": 540.0,
46  "valorTotalSeguroCarga": 30.22,
47  "numeroApoliceSeguroCarga": 5578685073,
48  "numeroAverbacaoSeguroCarga": 5537301,
49  "entregaEmParceria": true,
50  "cnpjParceira": 32614634000120,
51  "razaoSocialParceira": "Transportadora Entrega Rápida Ltda.",
52  "nomeComercialParceira": "Transportadora Entrega Rápida.",
53  "inscricaoEstadualParceira": "67.375.056-6",
54  "telefoneParceira": "(11)5773-8532",
55  "enderecoParceira": {
56    "id": 9,
57    "logradouro": "Rua Paraíso do Norte",
58    "numero": "817",
59    "complemento": "Galpão",
60    "bairro": "CORADO",
61    "cidade": "MANAUS",
62    "uf": "AM",
63    "cep": "69080-350"
64  },
65  "recebedorEntrega": null,
66  "tipoModal": "RODOVIARIO",
67  "cteEmitido": false,
68  "observacoes": "contem produtos perecíveis"
69  }

```

- UC03 – Emitir o CT-e e enviá-lo ao SFC.
 - Requisição: é o código da solicitação passado como parâmetro na URL. Neste exemplo ele está na variável dinâmica {{codigo-solicitacao}} configurada no Postman. Para emissão do CT-e de outra entrega, basta digitar o código numérico da solicitação desejada no lugar da variável.



- Resposta:




```

24      "localTerminoPrestacao": "Parintins - AM"
25    },
26    "dadosEmitente": {
27      "tipoDocumento": "CNPJ",
28      "cpfCnpj": 13814889000192,
29      "nomeRazaoSocial": "BOA ENTREGA LTDA.",
30      "inscricaoEstadual": "518.208.406.485",
31      "telefone": "(11)5773-8532",
32      "endereco": {
33        "logradouro": "Rua Duque de Caxias",
34        "numero": "225",
35        "complemento": "Galpão",
36        "bairro": "CENTRO",
37        "cidade": "SÃO PAULO",
38        "uf": "SP",
39        "cep": "01958-050"
40      }
41    },
42    "dadosRemetente": {
43      "tipoDocumento": "CNPJ",
44      "cpfCnpj": 80200396000150,
45      "nomeRazaoSocial": "Supermercado Manaus Ltda.",

```

```

46      "inscricaoEstadual": "475.209.639.488",
47      "telefone": "(92)97878-8989",
48      "endereco": {
49        "logradouro": "Rua Leonardo Malcher",
50        "numero": "888",
51        "complemento": "esquina",
52        "bairro": "Centro",
53        "cidade": "Manaus",
54        "uf": "AM",
55        "cep": "69010170"
56      }
57    },
58    "dadosDestinatario": {
59      "tipoDocumento": "CPF",
60      "cpfCnpj": 27365467096,
61      "nomeRazaoSocial": "Fábio Correia Araujo",
62      "inscricaoEstadual": null,
63      "telefone": "(12)93827-9748",
64      "endereco": {
65        "logradouro": "Rua Correa Neto",
66        "numero": "999",
67        "complemento": "casa 2",
68        "bairro": "Alvorada",

```

```

69      "cidade": "Parintins",
70      "uf": "AM",
71      "cep": "69111400"
72    }
73  },
74  "dadosExpedidor": {
75    "tipoDocumento": "CNPJ",
76    "cpfCnpj": 80200396000150,
77    "nomeRazaoSocial": "Supermercado Manaus Ltda.",
78    "inscricaoEstadual": "475.209.639.488",
79    "telefone": "(92)97878-8989",
80    "endereco": {
81      "logradouro": "Rua Leonardo Malcher",
82      "numero": "888",
83      "complemento": "esquina",
84      "bairro": "Centro",
85      "cidade": "Manaus",
86      "uf": "AM",
87      "cep": "69010170"
88    }
89  },
90  "dadosRecebedor": {
91    "tipoDocumento": "CNPJ",

```

```

92      "cpfCnpj": 32614634000120,
93      "nomeRazaoSocial": "Transportadora Entrega Rápida Ltda.",
94      "inscricaoEstadual": "67.375.056-6",
95      "telefone": "(11)5773-8532",
96      "endereco": {
97        "logradouro": "Rua Paraíso do Norte",
98        "numero": "817",
99        "complemento": "Galpão",
100       "bairro": "COROADO",
101       "cidade": "MANAUS",
102       "uf": "AM",
103       "cep": "69080-350"
104     }
105   },
106   "dadosTomador": {
107     "tipoDocumento": "CNPJ",
108     "cpfCnpj": 80200396000150,
109     "nomeRazaoSocial": "Supermercado Manaus Ltda.",
110     "inscricaoEstadual": "475.209.639.488",
111     "telefone": "(92)97878-8989",
112     "endereco": {
113       "logradouro": "Rua Leonardo Malchen",
114       "numero": "888",

```

```

115       "complemento": "esquina",
116       "bairro": "Centro",
117       "cidade": "Manaus",
118       "uf": "AM",
119       "cep": "69010170"
120     }
121   },
122   "dadosCarga": {
123     "produtoPredominante": "frutas",
124     "outrasCaracteristicasCarga": "frigorificada",
125     "valorTotalMercadoria": 4500.00,
126     "pesoBruto": 14.00,
127     "pesoAferido": 3000.00,
128     "cubagemM3": 300,
129     "volume": 10.00
130   },
131   "dadosSeguroCarga": {
132     "nomeSeguradora": "Sul America Seguros",
133     "nomeResponsavel": "Matheus Melo Santos",
134     "numeroApolice": 4929605038,
135     "numeroAverbacao": 623356
136   },
137   "componentesValorPrestacaoServico": {

```

```

138     "componenteValor": [
139       {
140         "nome": "Frete Valor",
141         "valor": 4500.00
142       }
143     ]
144   },
145 },
146 "emissaoOk": true,
147 "mensagemDoSfc": "Ct-e recebido com sucesso. Apuração financeira iniciada. Protocolo de recebimento pelo SFC: 171343878657606"
148 }

```

Nesta prova de conceito pretende-se avaliar os seguintes requisitos não funcionais:

1) Desempenho – O sistema deve processar rapidamente as solicitações de entrega recebidas.

Critério de aceite:

- O sistema deve ser capaz de cadastrar uma solicitação de entrega em menos 400 ms.

2) Segurança – o sistema deve atender de forma seletiva (por perfil) a clientes, fornecedores, colaboradores e administradores.

Critérios de aceites:

- O usuário de perfil administrador deve ter acesso a todas as funcionalidades do sistema.
- Os usuários de perfis cliente, colaborador e fornecedor devem ter permissão de acesso somente às suas respectivas funcionalidades pertinentes.

3) Interoperabilidade – o sistema deve ser capaz de comunicar-se com outros serviços e utilizar APIs ou outro recurso adequado para o consumo deles.

Critério de aceite:

- A comunicação com os diversos serviços deverá ser no padrão REST e os dados de requisição e resposta serão trafegados no formato JSON.

4) Manutenibilidade – o sistema deve prover facilidades para manutenções.

Critérios de aceites:

- O desenvolvedor deve conseguir alterar uma funcionalidade mantendo ativa a versão anterior dela para não impactar o cliente que não deseja utilizar a nova.
- Informações passíveis de mudança em qualquer momento devem estar configuradas no repositório do Github ou em arquivos properties dos microsserviços, para que não haja necessidade de intervenção no código ou que ela seja mínima. Exemplos: endereços de acesso aos serviços externos e endereços dos bancos devem constar em um arquivo do repositório de configurações; novos microsserviços implementados devem ter suas rotas cadastradas no arquivo application.properties do microsserviço de API Gateway.

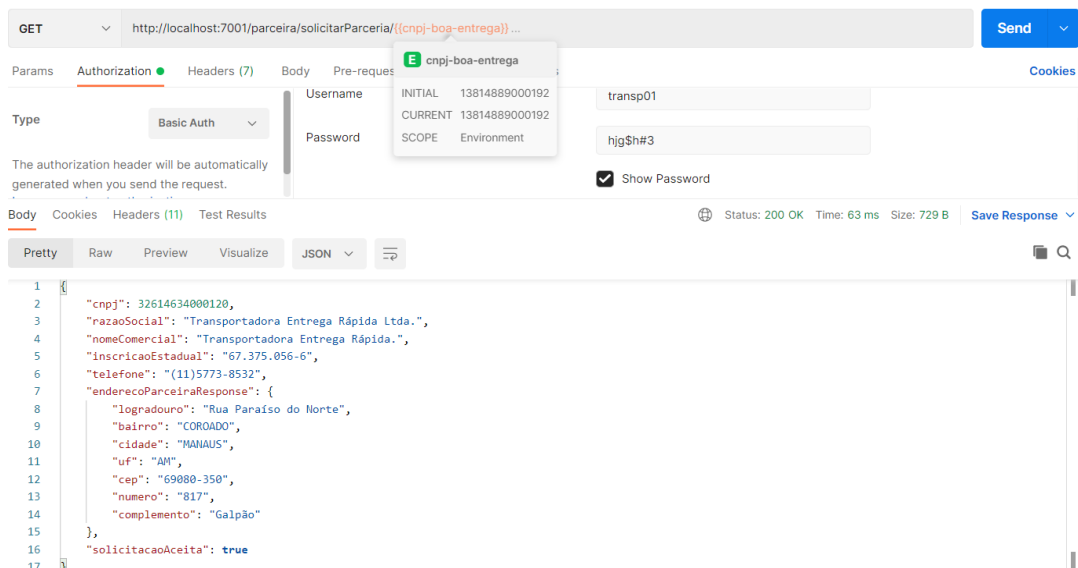
No apêndice deste documento encontra-se o link do repositório onde está o código fonte desta POC.

4.2 Interfaces e APIs

Nesta seção serão mostradas as interfaces de duas APIs que representam serviços externos e uma API que representa um dos sistemas legados da Boa Entrega. Elas foram implementadas nesta POC para retornarem informações *mockadas* e são acessadas por requisições no padrão Rest e com dados trafegados no formato Json. Seguem as interfaces e APIs:

- **gsi-parceira** – representa uma API externa de uma transportadora parceira.

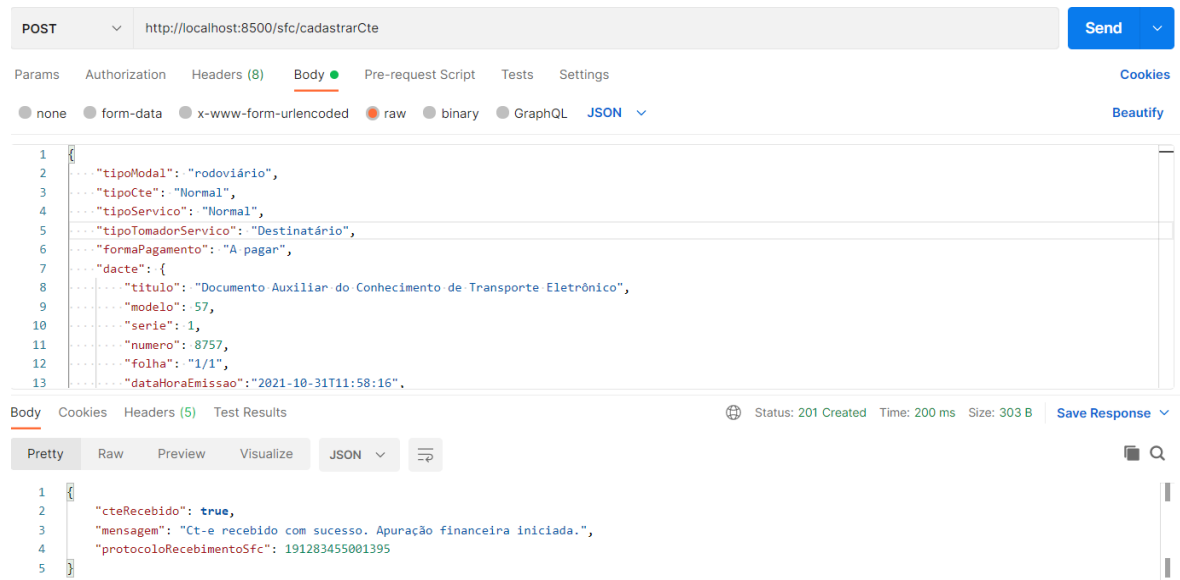
Esta API é acessada pela funcionalidade que cadastra uma solicitação de entrega sempre quando o destino da entrega for a região Norte, que não tem cobertura direta pela Boa entrega. Na requisição é informado o cnpj da Boa Entrega na URL. E na resposta a Boa Entrega recebe os dados cadastrais da transportadora parceira e um indicador de aceite do serviço (“true” ou “false” significando “sim” ou “não” respectivamente) representado pelo atributo “solicitacaoAceita”. Segue exemplo de requisição e resposta:



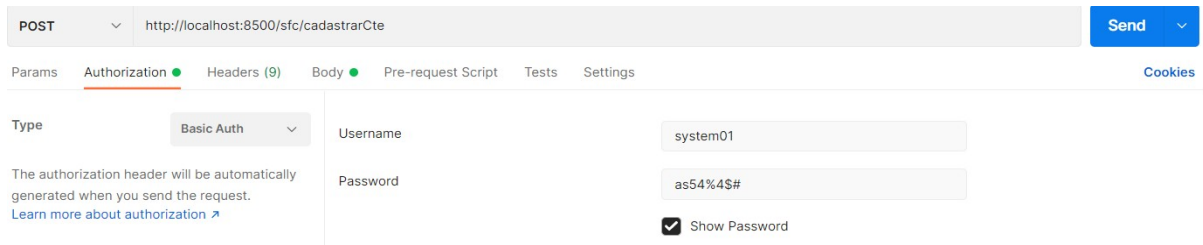
- **gsi-sfc** – representa uma API do sistema legado de faturamento e cobrança (SFC).

Esta API é chamada pela funcionalidade que emite o CT-e. A imagem abaixo é um exemplo de acesso a esta funcionalidade, contendo a URL, o request body (Json contendo os dados do CT-e), e o Json de resposta com os seguintes atributos: `cteRecebido`, que contém “true” se o SFC recebeu o CT-e, ou “false” caso contrário; mensagem contendo texto avisando do recebimento

do CT-e e que a apuração financeira será iniciada; e `protocoloRecebimentoSfc` contendo um número *mockado* gerado randomicamente.



A imagem abaixo contém os dados de autenticação desta requisição.



5. Avaliação da Arquitetura

Nesta seção serão mostrados uma análise das abordagens arquiteturais, cenários, evidências da avaliação e resultados.

5.1. Análise das abordagens arquiteturais

No quadro baixo constam as principais características desta proposta arquitetural, que serão avaliadas considerando o método ATAM (Architecture Tradeoff Analysis Method), no qual são utilizados cenários para esta análise:

Atributos de Qualidade	Cenários	Importância	Complexidade
Desempenho	Cenário 1: O sistema deve apresentar bom desempenho.	A	M
Segurança	Cenário 2: O sistema deve atender de forma seletiva (por perfil) a clientes, fornecedores e colaboradores.	A	B
Interoperabilidade	Cenário 3: O sistema de comunicar-se com outros serviços e utilizar APIs ou outro recurso adequado para o consumo deles.	A	M
Manutenibilidade	Cenários 4 e 5: O sistema deve ser de fácil manutenção.	M	B

5.2. Cenários

Seguem os cenários utilizados na realização dos testes da aplicação que demonstram os requisitos não funcionais sendo satisfeitos:

Cenário 1 – desempenho: ao receber uma requisição de solicitação de entrega, o sistema deve apresentar uma boa performance, processando-a rapidamente.

Cenário 2 – segurança: para cada funcionalidade há perfis específicos de usuários autorizados a acessá-las. O sistema deve negar o acesso a uma funcionalidade na qual o usuário autenticado não esteja autorizado.

Cenário 3 – interoperabilidade: dependendo da funcionalidade, durante a execução da transação pode ser necessário o acesso a um sistema externo ou a um sistema legado e o sistema deve utilizar APIs ou outro recurso adequado para consumo do serviço desejado. Ao acessar o serviço da transportadora parceira, os dados de requisição e resposta devem ser trafegados no formato Json.

Cenário 4 – manutenibilidade: a aplicação foi desenvolvida com base em microsserviços e cada um deles implementados com as camadas controller, service e repository, desta forma possuindo baixo acoplamento entre seus módulos e uma forte separação de responsabilidades.

Cenário 5 – manutenibilidade: a aplicação utiliza-se de muitos dados parametrizados nos arquivos properties e no repositório do Github, como rotas de microserviços, endereços de acesso a sistemas externos ou legados, desta forma não sendo necessária alteração em código fonte quando há mudanças nesses parâmetros.

5.3. Evidências da Avaliação

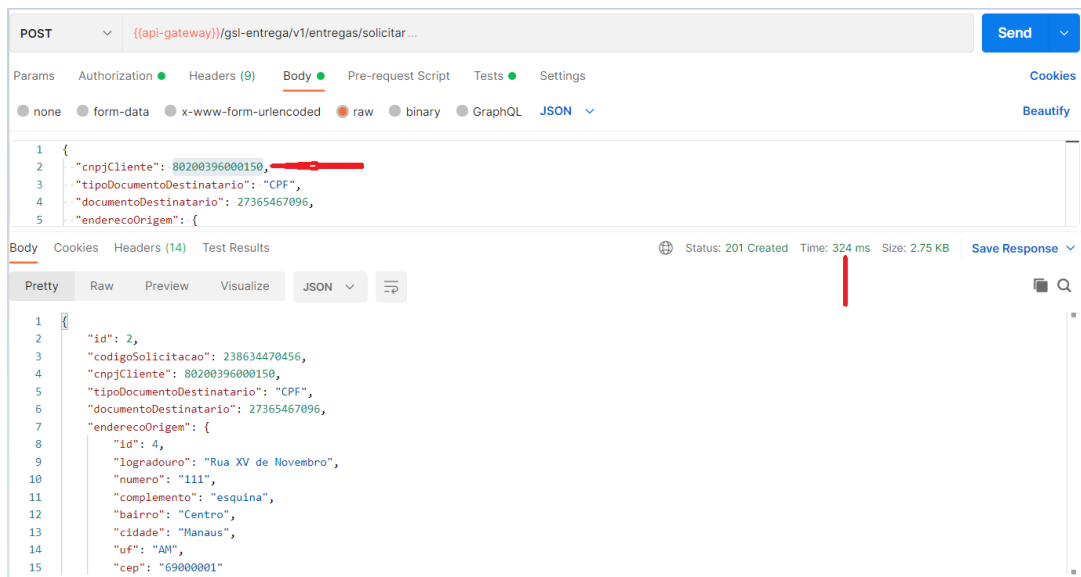
- **Cenário 1:**

Atributo de Qualidade:	Desempenho
Requisito de Qualidade:	O sistema deve apresentar boa performance de processamento.
Preocupação:	
O sistema estar apresentando alguma lentidão.	
Cenário(s):	
Cenário 1	
Ambiente:	
Sistema em operação normal	
Estímulo:	
API de solicitação de entrega recebe uma requisição.	
Mecanismo:	
Criar um serviço REST para atender a requisição de solicitação de entrega.	
Medida de resposta:	
As requisição será gravadas no banco de dados.	
Considerações sobre a arquitetura:	
Riscos:	Instabilidade na rede ou no banco de dados, e sobrecarga de acessos ao servidor de aplicação podem deixar as transações mais lentas.
Pontos de Sensibilidade:	O API Gateway é o ponto único de entrada de todas as requisições na aplicação e ele é responsável pelo roteamento delas até a API

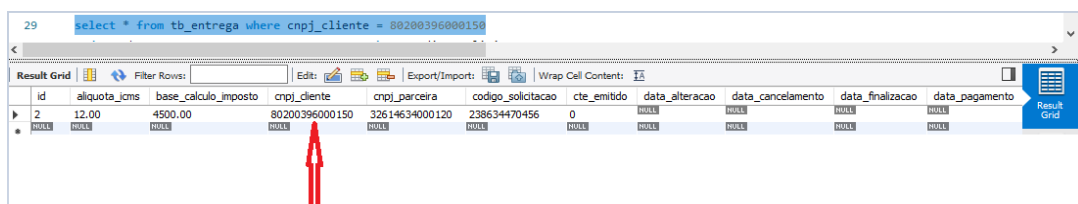
	referente a funcionalidade de negócio.
Tradeoff:	A disponibilidade do sistema seria afetada caso o microserviço do API Gateway apresentasse algum problema que demorasse para ser resolvido.

● Evidências do cenário 1:

Foi realizado um teste via ferramenta Postman na API de solicitação de entregas, e o tempo de processamento foi de 324 ms conforme imagem abaixo:



A transação foi registrada no banco de dados MySQL conforme imagem abaixo, onde mostra a coluna `cnpj_cliente`, que refere-se ao mesmo cnpj que consta no body da requisição no Postman:



Foram disparadas mais requisições de solicitação de entrega e o tempo delas manteve-se dentro do esperado, conforme imagem abaixo:

• POST http://localhost:8766/gsl-entrega/v1/entregas/solicitar	201 324 ms
• POST http://localhost:8766/gsl-entrega/v1/entregas/solicitar	201 288 ms
• POST http://localhost:8766/gsl-entrega/v1/entregas/solicitar	201 139 ms
• POST http://localhost:8766/gsl-entrega/v1/entregas/solicitar	201 132 ms
• POST http://localhost:8766/gsl-entrega/v1/entregas/solicitar	201 171 ms
• POST http://localhost:8766/gsl-entrega/v1/entregas/solicitar	201 146 ms
• POST http://localhost:8766/gsl-entrega/v1/entregas/solicitar	201 161 ms
• POST http://localhost:8766/gsl-entrega/v1/entregas/solicitar	201 167 ms
• POST http://localhost:8766/gsl-entrega/v1/entregas/solicitar	201 136 ms
• POST http://localhost:8766/gsl-entrega/v1/entregas/solicitar	201 331 ms
• POST http://localhost:8766/gsl-entrega/v1/entregas/solicitar	201 133 ms
• POST http://localhost:8766/gsl-entrega/v1/entregas/solicitar	201 301 ms
• POST http://localhost:8766/gsl-entrega/v1/entregas/solicitar	201 202 ms
• POST http://localhost:8766/gsl-entrega/v1/entregas/solicitar	201 212 ms

● Cenário 2:

Atributo de Qualidade:	Segurança
Requisito de Qualidade:	O sistema deve atender de forma seletiva (por perfil) a clientes, fornecedores e colaboradores.
Preocupação:	
O sistema não deve autorizar o acesso de um usuário às funcionalidades para as quais ele não possua perfil para acessá-las.	
Cenário(s):	
Cenário 2	
Ambiente:	
Sistema em operação normal	
Estímulo:	
O usuário já realizou o login e vai tentar acessar uma funcionalidade do sistema.	
Mecanismo:	
No sistema foram configuradas as atribuições de roles (agrupamento de permissões) para as funcionalidades do sistema.	
Medida de resposta:	
Se o perfil do usuário tiver acesso à funcionalidade, ela será executada; caso contrário será retornado o erro 403 do http que significa que o usuário não tem direito de acesso ao recurso que ele tentou acessar.	
Considerações sobre a arquitetura:	
Riscos:	Ocorrer falha de segurança em um momento em que haja quantidade de acessos ao sistema acima da média, podendo resultar em

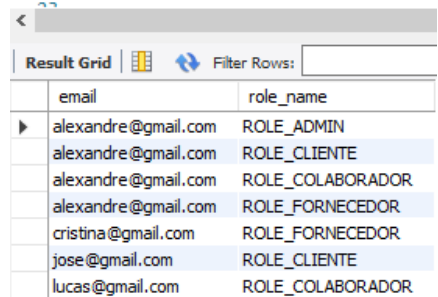
	prejuízos financeiros.
Pontos de Sensibilidade:	Servidor de aplicação operando em modo https
Tradeoff:	Não há

Nesta POC, o microserviço gsl-user carrega no banco MySQL alguns usuários e suas respectivas roles durante a sua inicialização, conforme exibido abaixo:

```

16  select * from tb_user_role
17  -- Lista usuarios e suas role
18  select u.email, r.role_name
19         from tb_user u, tb_role r,
20         where ur.user_id = u.id
21               and ur.role_id = r.id
22         order by 1, 2
23

```



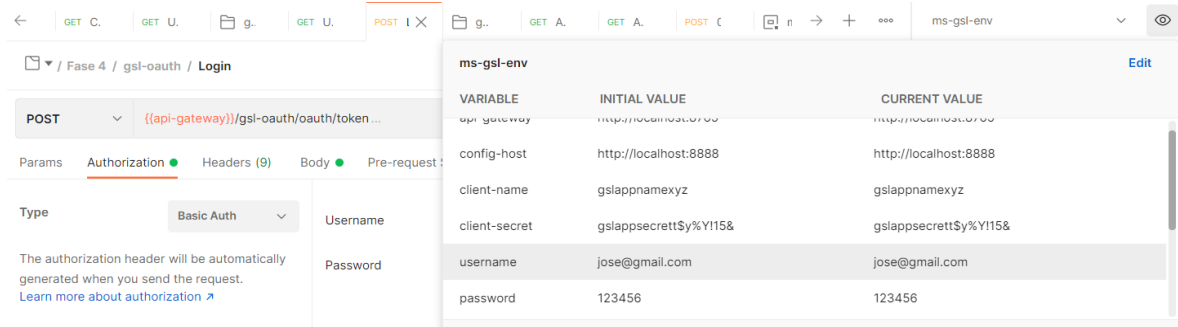
The screenshot shows a database query result grid with two columns: 'email' and 'role_name'. The results are as follows:

email	role_name
alexandre@gmail.com	ROLE_ADMIN
alexandre@gmail.com	ROLE_CLIENTE
alexandre@gmail.com	ROLE_COLABORADOR
alexandre@gmail.com	ROLE_FORNECEDOR
crisina@gmail.com	ROLE_FORNECEDOR
jose@gmail.com	ROLE_CLIENTE
lucas@gmail.com	ROLE_COLABORADOR

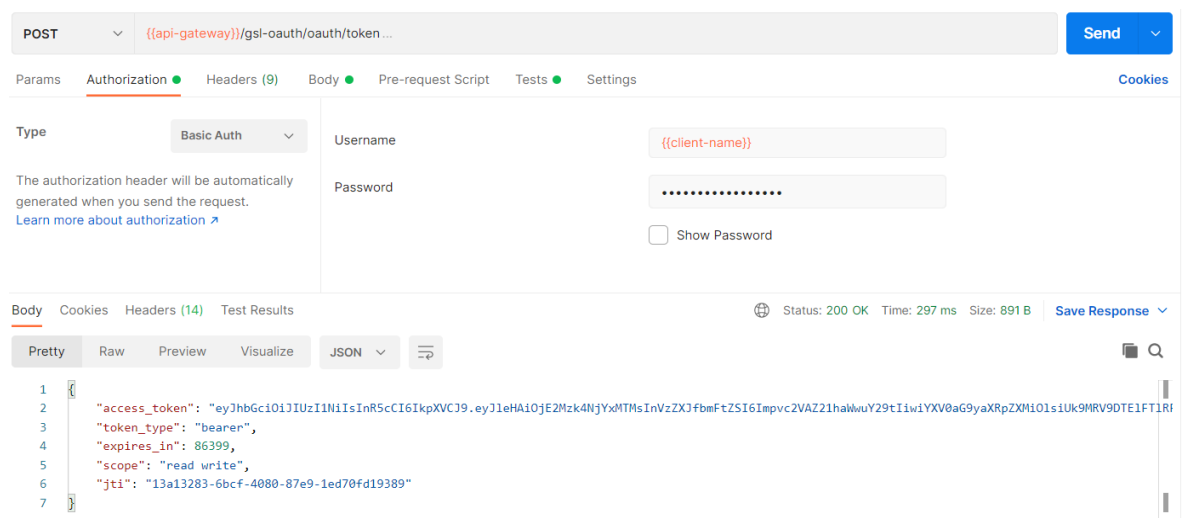
● Evidências do cenário 2:

- **Evidência 1:** Usuário fez login no sistema e tentou acessar uma funcionalidade para a qual seu perfil não possui autorização de acesso. Para esta evidência o login foi feito com o usuário *jose@gmail.com* cujo perfil é de cliente (*ROLE_CLIENTE*).

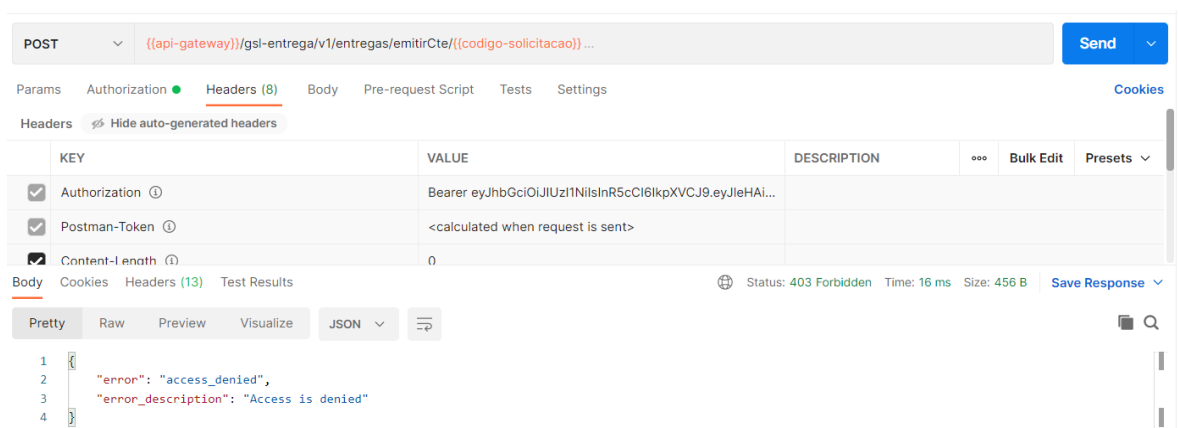
As credenciais de login foram informadas na janela de configuração de ambiente do postman:



Usuário autenticou-se e foi gerado o token no atributo de resposta *access_token*:

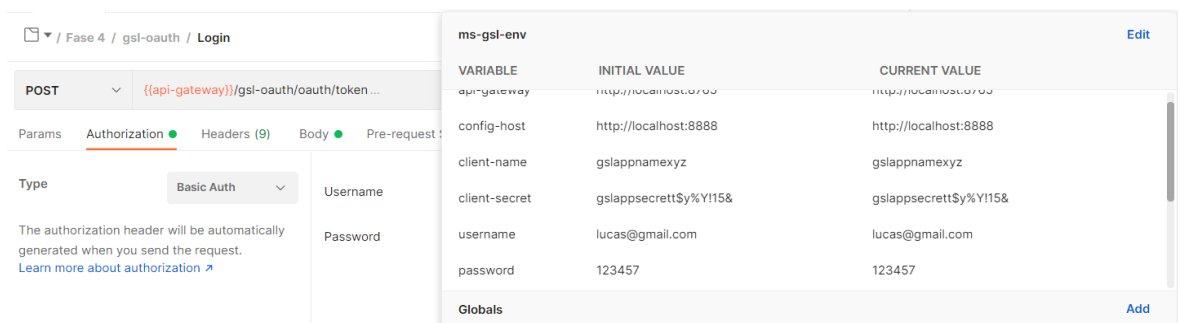


Após logar-se, o usuário tentou acessar a funcionalidade para emissão de CT-e, a qual não está autorizada para o seu perfil:

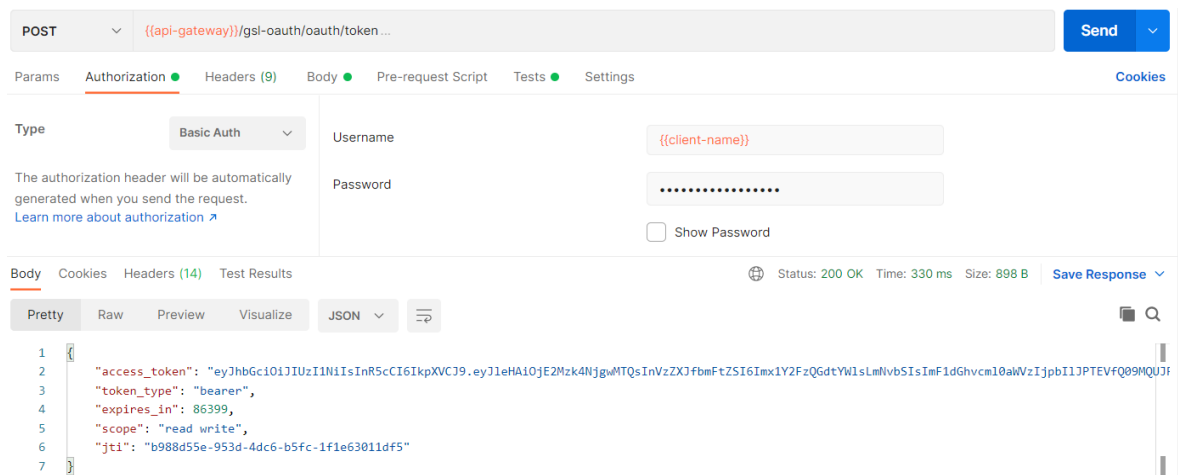


- **Evidência 2:** Usuário fez login no sistema e tentou acessar uma funcionalidade para a qual o seu perfil está autorizado a acessá-la. Para esta evidência o login foi feito com o usuário *lucas@gmail.com* cujo perfil é de colaborador (*ROLE_COLABORADOR*).

As credenciais de login foram informadas na janela de configuração de ambiente do postman:



Usuário autenticou-se e foi gerado o token no atributo de resposta *access_token*:



Após logar-se, o usuário tentou acessar a funcionalidade de emissão de CT-e, para a qual seu perfil tem autorização, e conseguiu executá-la:

POST {{api-gateway}}/gsl-entrega/v1/entregas/emitirCte/{{codigo-solicitacao}} ... Send

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

Headers Hide auto-generated headers

	KEY	VALUE	DESCRIPTION	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization ^①	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAi...				Go to authorization
<input checked="" type="checkbox"/>	Postman-Token ^①	<calculated when request is sent>				

Body Cookies Headers (14) Test Results Status: 201 Created Time: 798 ms Size: 3.73 KB [Save Response](#)

Pretty Raw Preview Visualize JSON ≡

```

134      "numeroApolice": 8860513159,
135      "numeroAverbacao": 9214095
136    },
137    "componentesValorPrestacaoServico": {
138      "componenteValor": [
139        {
140          "nome": "Frete Valor",
141          "valor": 4500.00
142        }
143      ]
144    }
145  },
146  "emissaoOk": true,
147  "mensagemDoSfc": "Ct-e recebido com sucesso. Apuração financeira iniciada. Protocolo de recebimento pelo SFC: 329716019010671"
148 }

```

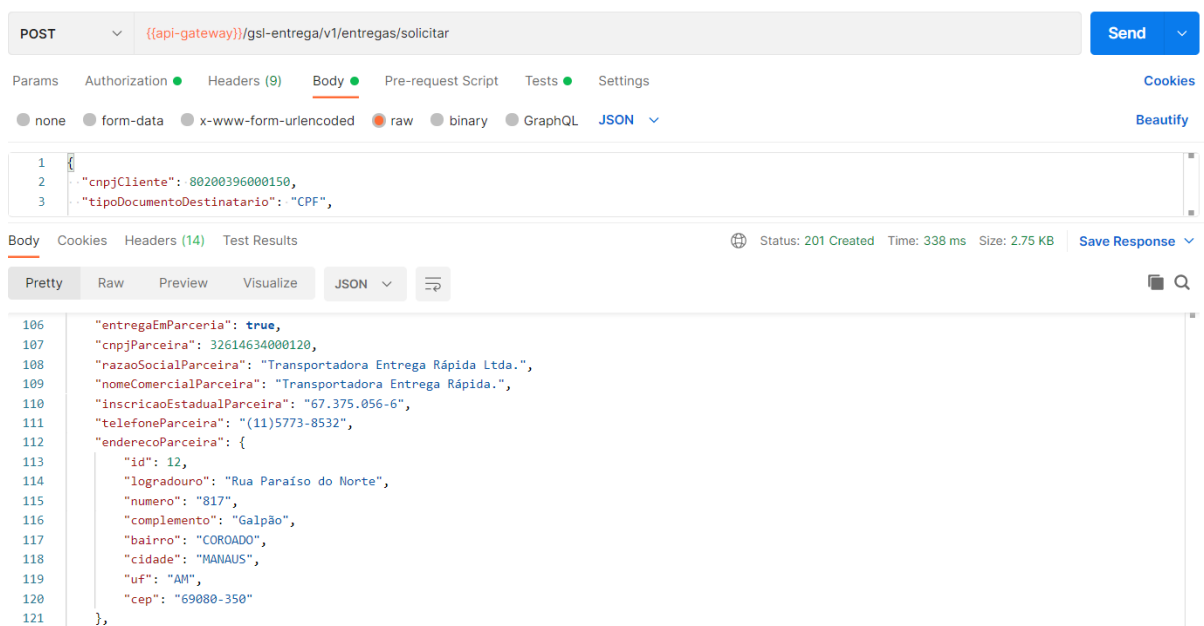
● Cenário 3:

Atributo de Qualidade:	Interoperabilidade
Requisito de Qualidade:	O sistema deve se comunicar com sistemas externos de transportadoras parceiras.
Preocupação:	
O sistema deve acessar outros sistemas de forma transparente ao usuário obtendo as informações necessárias e agregando-as na resposta da transação em formato Json.	
Cenário(s):	
Cenário 3	
Ambiente:	
Sistema em operação normal	
Estímulo:	
O sistema recebe uma solicitação de entrega para uma região ainda sem cobertura pela Boa Entrega.	
Mecanismo:	
Efetuar uma chamada ao serviço REST da transportadora parceira durante a processamento da solicitação de entrega, a fim de solicitar parceria para atendimento desta demanda. Os dados devem ser trafegados no formato Json.	
Medida de resposta:	
Retornar na resposta da solicitação de entrega os atributos retornados pelo serviço da transportadora parceira.	

Considerações sobre a arquitetura:	
Riscos:	Instabilidade no sistema da transportadora parceira poderá inviabilizar a operação.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

● Evidências do cenário 3:

Foi efetuada uma requisição de solicitação de entrega onde durante a transação ocorreu um acionamento ao sistema da transportadora parceira para solicitação de parceria para realização da entrega. Na imagem abaixo constam os atributos “entregaEmParceria”, que indica que a entrega será realizada pela transportadora acionada, e os demais atributos que são os dados cadastrais desta transportadora:



Segue trechos extraídos do log evidenciando a requisição ao sistema externo e a resposta dele:

- Requisição:

```
[2m2021-12-17 00:43:00.020[0;39m [32m INFO[0;39m [35m42292[0;39m [2m---[0;39m [2m[o-auto-1-exec-1][0;39m [36mc.d.g.-
service.EntregaServiceImpl [0;39m [2m:[0;39m Chamando sistema da transportadora parceira: http://localhost:7001/parcei-
ra/solicitarParceria/13814889000192
```

- Resposta:

[2m2021-12-17 00:43:00.027[0;39m [32m INFO[0;39m [35m42292[0;39m [2m---[0;39m [2m[o-auto-1-exec-1][0;39m [36mc.d.g.-service.EntregaServiceImpl [0;39m [2m:[0;39m Resposta do sistema da transportadora parceira: [ParceiraResponse(cnpj=32614634000120, razaoSocial=Transportadora Entrega Rápida Ltda., nomeComercial=Transportadora Entrega Rápida., inscricaoEstadual=67.375.056-6, telefone=(11)5773-8532, enderecoParceiraResponse=EnderecoParceiraResponse(logradouro=Rua Paraíso do Norte, bairro=CORADO, cidade=MANAUS, uf=AM, cep=69080-350, numero=817, complemento=Galpão), solicitacaoAceita=true)]

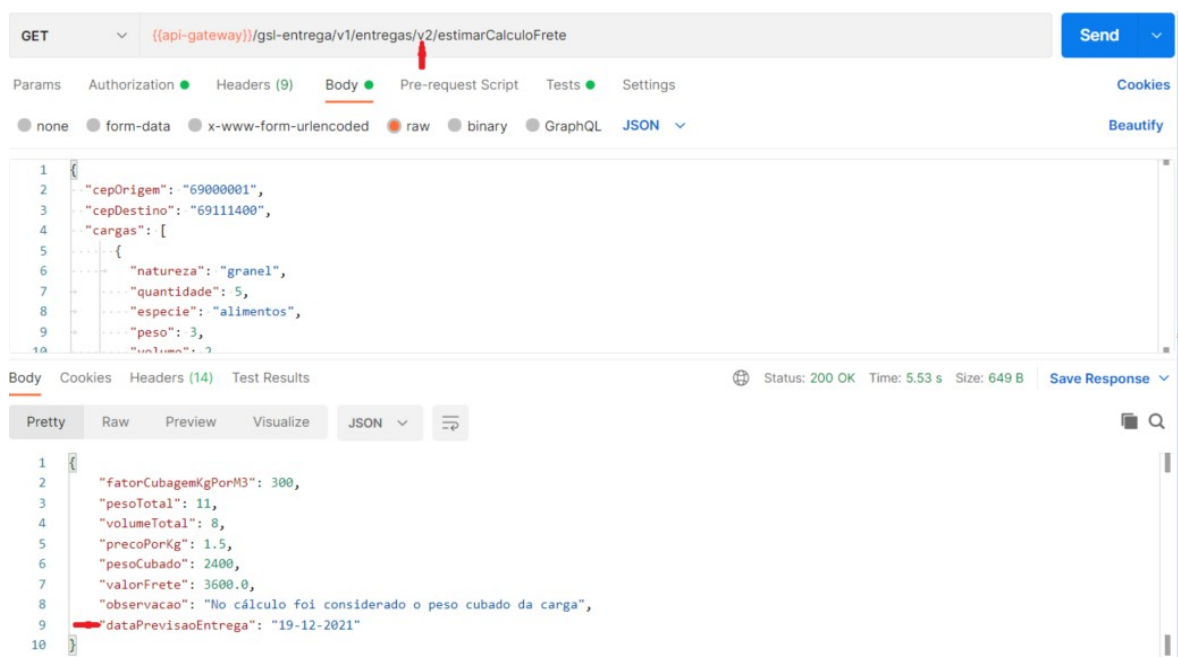
● Cenário 4:

Atributo de Qualidade:	Manutenibilidade
Requisito de Qualidade:	O sistema deve ser de fácil manutenção.
Preocupação:	
As novas alterações em APIs, ao serem implantadas, não devem impossibilitar a jornada do usuário que optar por não utilizar a nova versão.	
Cenário(s):	
Cenário 4	
Ambiente:	
Sistema em operação normal	
Estímulo:	
A área de Gestão da Boa Entrega recebeu uma solicitação de um grande e estratégico cliente para que fosse incluída uma <i>data de previsão de entrega</i> das mercadorias na resposta da funcionalidade de estimativa de cálculo do frete e a redirecionou ao time de desenvolvimento para implementá-la.	
Mecanismo:	
Criar novo método de endpoint na classe de controller do microserviço gsl-entrega para calcular o frete e também estimar uma data de entrega; mapear a url do recurso com um versionamento (/v2); criar nova classe de response herdando a atual para somente incluir o novo atributo; e na camada service implementar novo método para trazer a nova informação, mas reaproveitando o método antigo para obter as informações do cálculo do frete.	
Medida de resposta:	
A funcionalidade alterada deverá retornar no corpo da resposta uma data de previsão de entrega além dos atributos referentes ao cálculo do frete e se ela for acessada por qualquer cliente que ainda esteja apontando para a url referente a versão anterior, deve continuar funcionando normalmente.	
Considerações sobre a arquitetura:	
Riscos:	Desenvolvedor “esquecer” de versionar a url do <i>endpoint</i> referente a funcionalidade.

Pontos de Sensibilidade:	Modificações em várias funcionalidades dentro do mesmo microserviço.
Tradeoff:	Caso haja muitas modificações em vários métodos e ele possuírem na url um número de versionamento diferentes, isto pode dificultar um pouco a usabilidade para o cliente que quiser acessar sempre a última versão de cada funcionalidade.

● Evidências do cenário 4:

● **Evidência 1:** a funcionalidade alterada foi disponibilizada em uma nova url (a diferença para a anterior é o apontamento de versão “/v2”. Ao ser executada gerou na sua resposta o atributo dataPrevisaoEntrega, além dos demais atributos que já eram respondidos na versão anterior.



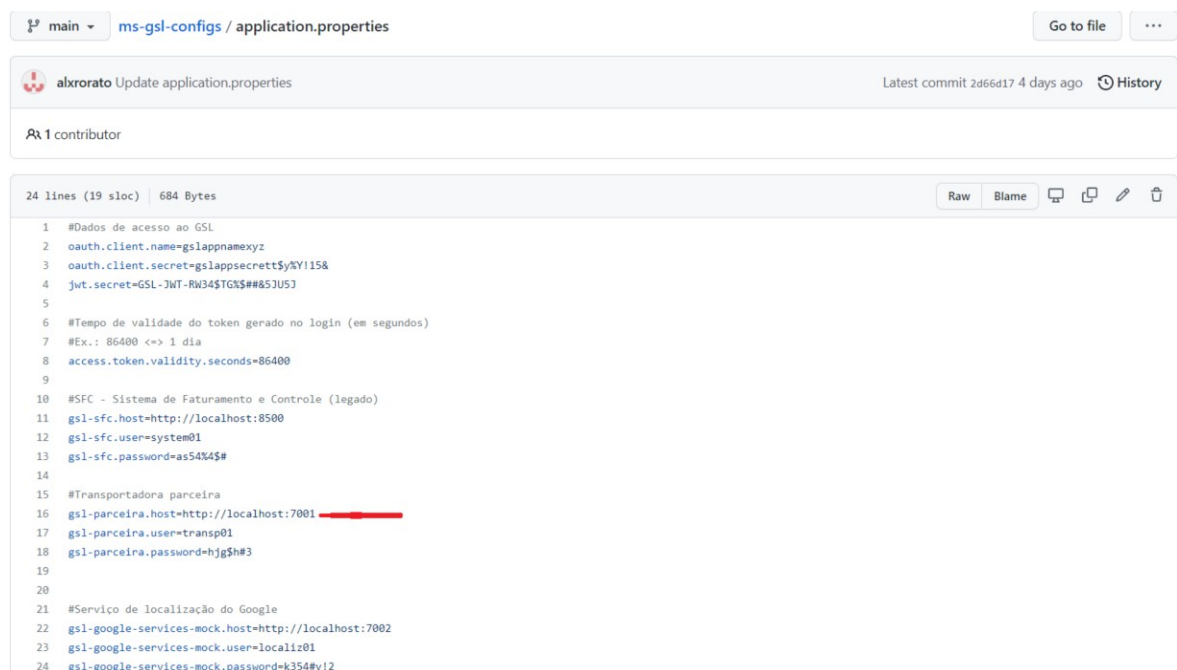
● **Evidência 2:** a funcionalidade sendo acessada na sua versão anterior continua respondendo normalmente, conforme figura abaixo onde vemos a url anterior e na resposta não consta o atributo dataPrevisaoEntrega porque ele só está disponível na versão mais recente:

Atributo de Qualidade:	Manutenibilidade
Requisito de Qualidade:	O sistema deve ser de fácil manutenção.
Preocupação:	
Alterações em endereços de acesso a sistemas de terceiros devem ser realizadas o mais rápido possível para que não haja indisponibilidade de funcionalidades que utilizam-se destes sistemas.	
Cenário(s):	
Cenário 5	
Ambiente:	
Sistema em operação normal	
Estímulo:	
Mudança urgente do endereço de acesso ao sistema da transportadora parceira, que não avisou a mudança antecipadamente.	
Mecanismo:	
Realizar a alteração do endereço no arquivo application.properties do repositório ms-gsl-configs no GitHub e fazer um restart no microserviço de Serviços aos Clientes para reconhecer o novo endereço.	
Medida de resposta:	
O sistema da transportadora parceira deverá ser acessado normalmente pela funcionalidade que o utiliza.	
Considerações sobre a arquitetura:	

Riscos:	Instabilidade no sistema da transportadora parceira, mesmo acessando-o com o endereço atualizado.
Pontos de Sensibilidade:	No arquivo de configuração pode constar alguma informação sensível.
Tradeoff:	Segurança, devido a utilização de um arquivo de configuração para expor informações que podem ser consideradas sensíveis.

● Evidências do cenário 5:

Imagem do arquivo `application.properties` do repositório do Github antes da alteração, quando o sistema da transportadora parceira atendia na porta 7001 do seu servidor:



```

1 #Dados de acesso ao GSL
2 oauth.client.name=gslappnamexyz
3 oauth.client.secret=gslappsecret$y$Y!15&
4 jwt.secret=GSL-JWT-RW34$TG$##85JUSJ
5
6 #Tempo de validade do token gerado no login (em segundos)
7 #Ex.: 86400 <=> 1 dia
8 access.token.validity.seconds=86400
9
10 #SFC - Sistema de Faturamento e Controle (legado)
11 gsl-sfc.host=http://localhost:8500
12 gsl-sfc.user=system01
13 gsl-sfc.password=as54$4$#
14
15 #Transportadora parceira
16 gsl-parceira.host=http://localhost:7001
17 gsl-parceira.user=transp01
18 gsl-parceira.password=hjg$h#3
19
20
21 #Serviço de localização do Google
22 gsl-google-services-mock.host=http://localhost:7002
23 gsl-google-services-mock.user=localiz01
24 gsl-google-services-mock.password=k354#y!2

```

O endereço da transportadora parceira foi alterado para atender na porta 7003, e então o arquivo `application.properties` foi alterado para contemplar o novo endereço, conforme imagem abaixo:

alxratoro Update application.properties Latest commit 57fad55 7 minutes ago History

1 contributor

24 lines (19 sloc) | 684 Bytes

```

1 #Dados de acesso ao GSL
2 oauth.client.name=gslappnamexyz
3 oauth.client.secret=gslappsecret$yY115&
4 jwt.secret=GSL-JWT-Rw345TG$##85JU5J
5
6 #Tempo de validade do token gerado no login (em segundos)
7 #Ex.: 86400 <=> 1 dia
8 access.token.validity.seconds=86400
9
10 #SFC - Sistema de Faturamento e Controle (legado)
11 gsl-sfc.host=http://localhost:8500
12 gsl-sfc.user=system01
13 gsl-sfc.password=as54$4$#
14
15 #Transportadora parceira
16 gsl-parceira.host=http://localhost:7003
17 gsl-parceira.user=transp01
18 gsl-parceira.password=hjg$h#3
19
20
21 #Serviço de localização do Google
22 gsl-google-services-mock.host=http://localhost:7002
23 gsl-google-services-mock.user=localiz01
24 gsl-google-services-mock.password=k354#y12

```

O microserviço gsl-entrega, do módulo de Serviços aos Clientes foi reiniciado e foi executada a funcionalidade de solicitação de entrega conforme imagem abaixo, onde na resposta mostra a confirmação de que a entrega será realizada em parceira com outra transportadora:

POST (api-gateway)/gsl-entrega/v1/entregas/solicitar ... Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies Beautify

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

```

1 {
2   "cnpjCliente": 80200396000150,
3   "tipoDocumentoDestinatario": "CPF",

```

Body Cookies Headers (14) Test Results Status: 201 Created Time: 1743 ms Size: 2.75 KB Save Response

Pretty Raw Preview Visualize JSON

```

106 "entregaEmParceria": true,
107 "cnpjParceira": 32614634000120,
108 "razaoSocialParceira": "Transportadora Entrega Rápida Ltda.",
109 "nomeComercialParceira": "Transportadora Entrega Rápida.",
110 "inscricaoEstadualParceira": "67.375.056-6",
111 "telefoneParceira": "(11)5773-8532",
112 "enderecoParceira": {
113   "id": 3,
114   "logradouro": "Rua Paraíso do Norte",
115   "numero": "817",
116   "complemento": "Galpão",
117   "bairro": "COROADO",
118   "cidade": "MANAUS",
119   "uf": "AM",
120   "cep": "69080-350"
121 },

```

Segue log gerado pela funcionalidade de solicitação de entrega (referente a execução evidenciada pelo print acima), registrando o momento do acesso ao servidor da transportadora parceira:

- Requisição:

[2m2021-12-19 00:41:54.194[0;39m [32m INFO[0;39m [35m34320[0;39m [2m---[0;39m [2m[o-auto-1-exec-3][0;39m [36mc.d.g.-service.EntregaServiceImpl [0;39m [2m:[0;39m Chamando sistema da transportadora parceira: <http://localhost:7003/parceira/solicitarParceria/13814889000192>

- Resposta:

[2m2021-12-19 00:41:54.323[0;39m [32m INFO[0;39m [35m34320[0;39m [2m---[0;39m [2m[o-auto-1-exec-3][0;39m [36mc.d.g.-service.EntregaServiceImpl [0;39m [2m:[0;39m Resposta do sistema da transportadora parceira:
[ParceiraResponse(cnpj=32614634000120, razaoSocial=Transportadora Entrega Rápida Ltda., nomeComercial=Transportadora Entrega Rápida., inscricaoEstadual=67.375.056-6, telefone=(11)5773-8532, enderecoParceiraResponse=EnderecoParceiraResponse(logradouro=Rua Paraíso do Norte, bairro=COROADO, cidade=MANAUS, uf=AM, cep=69080-350, numero=817, complemento=Galpão), solicitacaoAceita=true)]

5.4. Resultados

Considerando os atributos de qualidade analisados nesta POC, foi verificado que a arquitetura atende às necessidades do projeto, embora possua limitações. A análise dos requisitos listados na tabela abaixo foi importante para conseguir definir pontos fortes e fracos na avaliação.

Requisitos Não Funcionais	Testado	Homologado
RNF02: O sistema deve apresentar bom desempenho.	SIM	SIM
RNF03: O sistema deve atender de forma seletiva (por perfil) a clientes, fornecedores e colaboradores.	SIM	SIM
RNF04: O sistema deve apresentar boa manutenibilidade.	SIM	SIM
RNF09: O sistema deve utilizar APIs para o consumo de serviços.	SIM	SIM

As implementações realizadas nesta POC referem-se a camada back-end da arquitetura referente aos módulos de Serviços aos Clientes e de Informações Cadastrais. Foi utilizado o padrão de API Rest para o desenvolvimento das funcionalidades nos microsserviços destes módulos, com cada uma possuindo a sua própria url de acesso e desta forma havendo um baixo acoplamento e separação de responsabilidades. Isto facilita as chamadas por uma solução de workflow com uso de BPM, ou por outras soluções de front-end. Considerando que a Boa Entrega deseja evoluir esta POC, o padrão API Rest permitirá que os desenvolvimentos no front-end e back-end sejam possíveis de serem implementados em paralelo por times diferentes, e também a evolução do back-end facilitada.

Outro ponto referente a implementação, é que, por se tratar de uma POC com um número pequeno de funcionalidades, foram desenvolvidos somente um microserviço dentro de cada módulo. No entanto, o módulo de Serviços aos Clientes poderia ter tantos microserviços quanto fossem necessários caso houvesse uma quantidade muito maior de funcionalidades, para uma melhor separação de responsabilidades e manter a facilidade para manutenções, mesmo com o crescimento do sistema. E quanto ao Módulo de Informações Cadastrais, ele poderia comportar também outros microserviços, para prover cadastros de fornecedores, armazéns e mercadorias, por exemplo.

No requisito de desempenho o sistema se comportou bem ao processar requisições seguidas com tempo de resposta inferior a 400 ms. No entanto, algumas transações podem demorar um tempo maior que este, o que seriam exceções. E caso ocorra uma situação de stress no sistema, com grandes lotes de requisições e muitos disparos paralelos, pode ser que sua performance caia ou não consiga efetivar 100% das requisições, precisando ganhar escalabilidade e sugerindo-se como solução a implementação de mensageria para permitir que as requisições não processadas num primeiro momento fossem para uma fila para posterior reprocessamento. Para este requisito há um trade-off de disponibilidade, caso o microserviço de API Gateway falhasse ou necessitasse de manutenção ficando indisponível, pois ele centraliza o recebimento de todas as requisições ao sistema e faz o roteamento delas até o microserviço de negócio.

No requisito de interoperabilidade o sistema interagiu normalmente com o sistema externo acessado no acionamento da transportadora parceira para efetuar a entrega. Quanto a este requisito pode ser também acrescentado que o sistema da Boa Entrega também possuirá fácil interoperabilidade com aplicações que o chamarão, pelo fato de suas funcionalidades serem expostas para acesso através do padrão API Rest e com o tráfego de dados no padrão Json.

No requisito de segurança a arquitetura se mostrou segura ao não permitir acesso às funcionalidades para usuários sem o perfil de acesso adequado.

No requisito de manutenibilidade, o sistema mostrou-se de fácil manutenção evolutiva, facilitada pela adoção do padrão de desenvolvimento API Rest, versionamento de APIs e desenvolvimento dos microserviços em camadas de

controller, service e repository. E para mudanças mais simples, como uma alteração no endereço de acesso de um sistema externo ou mudanças referentes a configurações, algumas informações ficam em um arquivo de configuração no repositório do Github, e outras em arquivos properties nos microsserviços. Desta forma não há alteração em código fonte. Mas isto gera um trade-off em relação ao atributo de segurança, pois nestes arquivos podem constar informações sensíveis, como dados de login, por exemplo.

Como resultado desta avaliação, considero que esta arquitetura possui muito mais pontos fortes do que limitações, como o desempenho, que pode ser melhorado com a implementação de fila de mensagens para as requisições, e a segurança, pelo fato dos arquivos de configuração conterem dados sensíveis nesta POC, podendo isso ser melhorado, por exemplo, deixando os dados de configurações mais sensíveis em tabelas de banco de dados e gravados de forma criptografada. Esta arquitetura mostrou ser de fácil evolução e é facilmente integrável com outras aplicações devido à sua implementação no padrão de API Rest e em camadas.

6. Conclusão

Este trabalho apresentou uma proposta arquitetural para uma solução de gestão de serviços de logística baseada em microsserviços visando atender a jornada de entregas de uma transportadora. Nesta POC foram avaliados apenas uma parte dos requisitos não funcionais. Foram apresentadas neste trabalho algumas possíveis limitações, como uma provável perda de desempenho em alguma situação de stress no sistema. Na verificação de desempenho foi apontado um tradeoff de disponibilidade devido a utilização de um API Gateway para recebimento e roteamento de todas as requisições ao sistema, pois caso este serviço necessitasse de manutenção, ela precisaria ser rápida para não causar muita indisponibilidade no sistema. Foi verificado um trade-off envolvendo segurança em uma das evidências da avaliação da manutenibilidade. Mesmo assim, conclui-se que os objetivos foram atingidos porque esta arquitetura é de fácil evolução e integração com aplicações clientes, fornecedoras e ferramentas de BMP. A facilidade de integração pode gerar ganhos financeiros no médio prazo, por permitir, por exemplo,

a substituição de um fornecedor ou solução de BPM por outros que atendam melhor com menor custo; ou até mesmo o desenvolvimento interno de soluções pelo time interno para comunicação com esta aplicação.

REFERÊNCIAS

ALVES, Nelio. **Microserviços Java com SpringBoot e Spring Cloud**. <<https://www.udemy.com/course/microservicos-java-spring-cloud/?src=sac&kw=Microservi%C3%A7os+Java+com+Spring+Boot+e+Spring+Cloud>>, 2021

RV, Rajesh. **Spring Microservices**. Birmingham: Pack Publishing, 2016

CARNELL, John; SÁNCHEZ, Illary Huaylupo. **Spring Microservices In Action**. Manning Publications, 2020

BOAGLIO, Fernando. **Spring Boot: Acelere o desenvolvimento de microserviços**. São Paulo: Casa do Código, 2018

TPC. **Logística no Brasil: principais previsões e desafios do setor**. <<https://www.grupotpc.com/blog/logistica-no-brasil/>>, 2020

MENDES, Marco. **Arquitetura de Sistemas Web: Princípios, Prática e Tecnologias**. Atribuição-Compartilhada 4.0 Internacional, 2017

SILVEIRA, Paulo; TURINI, Rodrigo. **Java 8 Prático: Lambdas, Streams e os novos recursos da linguagem**. São Paulo: Casa do Código, 2019

YANAGA, Edson. **Migrating to Microservice Databases: From Relational Monolith to Distributes Data**. Sebastopol: O'Really, 2017

COELHO, Hérbert. **JPA Eficaz: As melhores práticas de persistência de dados em Java**. São Paulo: Casa do Código, 2018

Guia do TRC. **Como funciona o Conhecimento de Transporte Eletrônico (Ct-e)?** <<http://www.guiadotrc.com.br/noticias/noticialID.asp?id=34413>>, 2018

Diálogo, Logística Inteligente. **Conheça todas as etapas do processo de entrega da Diálogo Logística**. <<https://dialogologistica.com.br/noticias/etapas-processo-de-entrega/>>, 2021

REIS, Marco. **Arquitetura de Referência para Soluções de Big Data.** <<http://blog.marcoreis.net/arquitetura-de-referencia-para-solucoes-de-big-data/>>, 2018

Microsoft. **Visão geral do OLAP (Online Analytical Processing).** <<https://support.microsoft.com/pt-br/office/vis%C3%A3o-geral-do-olap-online-analytical-processing-15d2cdde-f70b-4277-b009-ed732b75fdd6>>, 2021

ILHE, Galleger. **Last-Mile: o que é e o quanto isso impacta em sua operação ?** <<https://www.ecommercebrasil.com.br/artigos/last-mile-o-quanto-isso-impacta-em-sua-operacao/>>, 2018

LIMA, Daniele. **Funções dos atores do Cte: Recebedor, expedidor e tomador.** <<https://blog.oobj.com.br/funcao-atores-do-cte/>>, 2021

ANDRADE, Carolini. **Quem é o Recebedor, Expedidor e Tomador no CT-e? Conheça todos os participantes.** <<https://bsoft.com.br/blog/recebedor-expedidor-e-tomador-no-cte>>, 2020

Gestran. **Saiba o que é Cte.** <<https://gestran.com.br/2018/08/saiba-o-que-e-cte/>>, 2018

LIMA, Julio. **Como fazer testes de performance em API Rest? (Aula 15).** <https://www.youtube.com/watch?v=Wjig_fVSL6w>, 2021

Cetax. **O que é OLAP – Online Analytical Processing ?** <<https://www.cetax.com.br/blog/o-que-e-olap/>>, 2020

Cetax. **Data Warehouse / Armazém de Dados: O que é, conceito e definição.** <<https://www.cetax.com.br/blog/o-que-e-data-warehouse/>>, 2020

ActiveCorp. **Como calcular as principais taxas que compõem o frete.** <<https://www.activecorp.com.br/Marketing/inbound/como-calculas-as-principais-taxas-que-compoem-o-frete.pdf>>, 2021

Interlog Seguros. **Como calcular o valor do seguro da carga ?** <<https://interlogseguros.com.br/2020/03/23/como-calculas-o-valor-do-seguro-da-carga/>>, 2020

ALBUQUERQUE, Danieli. **Guia Completo para emitir Cte (Conhecimento de Transporte eletrônico).** <<https://www.hivecloud.com.br/post/como-emitir-nota-conhecimento-transporte-eletronico-cte/>>, 2021

LORÊDO, Heliomar. **Replicação Assíncrona entre Bancos de Dados Heterogêneos: uma abordagem prática.**

<http://www.ufrgs.br/niee/eventos/CBCOMP/2004/pdf/Banco_Dados/t170100218_3.pdf>, 2004

Geekhunter. **Como fazer a integração de dados com Apache Kafka.** <<https://blog.geekhunter.com.br/apache-kafka/>>, 2020

SARDINHA, Renato. **Introdução ao Change Data Capture (CDC).** <<https://blog.elo7.dev/cdc-parte-1/>>, 2019

CARDOSO, Carlos Alberto Rocha. **CDC e streaming de eventos na prática.** <<https://medium.com/dataengineerbr/cdc-e-streaming-de-eventos-na-pr%C3%A1tica-d0e240c4e6a6>>, 2020

GONÇALVES, Marcelo M. **Event-streaming (Apache Kafka _ Kafka Streams) em uma Arquitetura de Microsserviços.** <<https://medium.com/@marcelomg21/event-streaming-apache-kafka-kafka-streams-em-uma-arquitetura-de-micro-servi%C3%A7os-136c28d7c9c8>>, 2020

Cetax. **SSIS – Sql Server Integration Services.**

<<https://www.cetax.com.br/blog/ssis-sql-server-integration-services/>>, 2019

Nerv Informática. **MySQL: Binlog Assíncrono.**

<<http://nervinformatica.com.br/blog/index.php/2017/01/18/mysql-binlog-assincrono/>>, 2017

SOUZA, Clayton. **Para que serve o SQL Server Integration Services?** <<https://bdasolutions.com.br/2019/04/para-que-serve-o-sql-server-integration-services/>>, 2019

ESTRELLA, Carlos. **O que é VPS ? Conheça as vantagens e como escolher um Servidor VPS.** <https://www.hostinger.com.br/tutoriais/o-que-e-vps-como-escolher-um-servidor-vps?ppc_campaign=google_search_generic_vps&bidkw=cloud>

%20vps&gclid=EAlaIQobChMI2NDnha729AIVEwWRCh3n4AhaEAAYASAAEgLCQ_D_BwE#Pros-e-Contras-da-Hospedagem-VPS>, 2021

SANCHES, Renato. **Como verificar usuários, privilégios e roles no banco de dados Oracle.** <<https://www.webmundi.com/banco-de-dados/oracle/como-verificar-usuarios-privilegios-e-roles-no-banco-de-dados-oracle/>>, 2021

Bizagi. **Modelagem de processos de negócios com Bizagi.** <<https://www.bizagi.com/pt/modelagem-de-processos>>, 2021

Tableau. **Tableau e ODBC.** <https://help.tableau.com/current/pro/desktop/pt-br/odbc_tableau.htm>, 2021

Tableau. **Hortonworks Hadoop Hive.** <https://help.tableau.com/current/pro/desktop/pt-br/examples_hortonworkshadoop.htm>, 2021

Hortonworks. **Hive ODBC Driver.** <<http://hortonworks.com/wp-content/uploads/2013/04/Hortonworks-Hive-ODBC-Driver-User-Guide.pdf>>, 2013

FLORIANO, Rodrigo. **[Sqoop] Importando bases SQL Server para o Hadoop.** <<https://churrops.io/2018/01/19/sqoop-importando-bases-sql-server-para-o-hadoop/>>, 2018

LUCIO, Claudio. **Download GUIA AULAS – 4.0 – Hadoop.pdf.** <https://pucminas.instructure.com/courses/49122/files/2466892/download?download_frd=1>, 2020

Sas. **Hadoop O que é e qual sua importância ?** <https://www.sas.com/pt_br/insights/big-data/hadoop.html>, 2021

APÊNDICE

URL do Github onde está o código fonte desta POC:

<https://github.com/alxrorato/ms-gsl>

URL do Github onde está o arquivo de configurações gerais desta POC:

<https://github.com/alxrorato/ms-gsl-configs>

URL da apresentação da POC: <https://youtu.be/6J9Ze5T26g8>

Observação: no arquivo README.md localizado no repositório desta POC no Github encontram-se as instruções de como subir localmente seus microsserviços e executar as funcionalidades referentes aos casos de uso implementados.