

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

EDUARDO SILVA DOS SANTOS

SISTEMA DE LOGÍSTICA BASEADO EM MICROSERVIÇOS
(BOA ENTREGA)

Belo Horizonte

2022

EDUARDO SILVA DOS SANTOS

**SISTEMA DE LOGÍSTICA BASEADO EM MICROSERVIÇOS
(BOA ENTREGA)**

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

Orientador(a):

Belo Horizonte
2022

Primeiramente, gostaria de agradecer a Deus por me capacitar a todo momento com sabedoria para lidar com este projeto, agradecer por ter saúde e aos ensinamentos dos professores no curso. Gostaria de agradecer também a minha família que a todo momento me apoiou do início ao fim, fazendo com que não desistisse no meio do caminho.

Eduardo Santos

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a Deus por me capacitar a todo momento com sabedoria para lidar com este projeto, agradecer por ter saúde e aos ensinamentos dos professores no curso. Gostaria também, de agradecer também a minha família que a todo momento me apoiou do início ao fim, fazendo com que não desistisse no meio do caminho.

Eduardo Santos.

RESUMO

O presente projeto arquitetural visa contemplar a logística Boa Entrega, com a entrega de um sistema para controlar o fluxo da informação em um cenário onde possa ter o controle de suas entregas, financeiro, juntamente com seus clientes e fornecedores de forma transparente e deixando todos bem informados de acordo com os trâmites de processo para a entrega dos produtos cadastrados na plataforma ou alimentado por sistemas legados que já existem dentro do fluxo de trabalho da operadora logística. O projeto arquitetural consiste na criação de micro serviços para atender as demandas dos depósitos, clientes, fornecedores referentes a suas entregas, que poderão ser enviadas de um estado para outro de forma a ser acompanhada sua passagem pelos depósitos da logística até chegarem a mão de seus clientes final, processo que cresceu muito durante a pandemia e levou muitas empresas a adotarem esta prática. A aplicação foi construída e teve como base a virtualização dos ambientes, utilizando ferramentas para simular os ambientes em operação em cluster, sendo de alta escalabilidade e disponibilidade contando com os mecanismos para conter a segurança da aplicação como um todo, passando por regras de perfis específicos e end-points controlados.

Palavras-chave: arquitetura de software, projeto de software, requisitos arquiteturais.

SUMÁRIO

1. Objetivos do trabalho	8
2. Descrição geral da solução	8
2.1. Apresentação do problema.....	8
2.2. Descrição geral do software (Escopo)	9
3. Definição conceitual da solução	9
3.1. Requisitos Funcionais	9
3.2 Requisitos Não-Funcionais	11
3.3. Restrições Arquiteturais	13
3.4. Mecanismos Arquiteturais	13
4. Modelagem e projeto arquitetural.....	13
4.1. Modelo de componentes	14
4.2. Modelo de implantação	17
4.3. Modelo de dados (opcional).....	18
5. Prova de Conceito (POC) / protótipo arquitetural	20
5.1. Implementação e Implantação	20
5.2 Interfaces/ APIs.....	21
6. Avaliação da Arquitetura.....	21
6.1. Análise das abordagens arquiteturais	22
6.2. Cenários	22
6.3. Avaliação	23
6.4. Resultado.....	23
7. Conclusão.....	23
REFERÊNCIAS.....	25
APÊNDICES.....	26
CHECKLIST PARA VALIDAÇÃO DOS ITENS E ARTEFATOS DO TRABALHO..	27

1. Objetivos do trabalho

Os objetivos deste trabalho é apresentar um projeto para controlar as informações cadastrais dos clientes, fornecedores e depósito que são parceiros da Boa Entrega, ele visa manter informado, os atores que neles estarão presentes sobre todo workflow desde a preparação do pedido pelo cliente, e a entrega consumada do pedido ao cliente, em caso de extravios ou não encontrado o cliente o sistema da Boa Entrega deverá retornar o produto ao fornecedor ou ressarcir de forma financeira o fornecedor.

Os objetivos específicos são:

- Implementação de um módulo com toda parte de cadastro que será o concentrador da informação das Mercadoria, Cliente, Fornecedores e Depósitos, este proverá integrações com os sistemas de controle de workflow;
- Implementação de um módulo com o controle e acompanhamentos das entregas aos clientes, solicitações de novas entregas pelos fornecedores, cancelamento de entregas, extravio de mercadoria;
- Implementação de um sistema com perfis de usuários destinados a cada operação e manipulação da informação presente na base dados;
- Implementação de um módulo para controle do financeiro, referentes as entregas e gestão dos recursos da Boa Entrega;

2. Descrição geral da solução

2.1. Apresentação do problema

No contexto atual da pandemia, diversas empresas tiveram sua produção interrompidas de acordo com algumas regras de distanciamento social, fazendo com que as pessoas não fossem as lojas de forma física e aumentaram as compras pelo contexto da internet, utilizando os modelos de e-commerce para efetuar suas aquisições.

Com isso o crescimento da demanda para a logística aumentou, e algumas empresas que não estavam preparadas para isso, se viram recuar e perder vendas. Pois, esse aumento leva a trazer mais responsabilidade com o cliente final e uma boa operação para não ter erros

de entrega, muito menos divergências em seus estoques por motivos da entrega não ser finalizada corretamente.

2.2. Descrição geral do software (Escopo)

A solução visa entregar um sistema, onde todos os envolvidos no processo de entrega possam estar ligados em conjuntos e acompanhar do início ao fim, a entrega do produto ao cliente final.

Para que ocorra todo o processo de perfeita maneira, o sistema contará com um trabalho em conjunto de um sistema de mensageria para trabalhar de forma assíncrona atualizando toda informação e toda ocorrência que acontece na entrega, e que irão ser atualizados na base de dados cadastro, serviços ao cliente e gestão e estratégia.

3. Definição conceitual da solução

3.1. Requisitos Funcionais

Módulo de Cadastro

- Cadastro de Depósito
 - O módulo deve permitir o cadastro dos depósitos parceiros da Boa Entrega que poderão estar presentes em qualquer estado e região do Brasil, devendo indicar endereço, região e código externo do depósito.
 - O módulo deve informar ao módulo de serviços ao Cliente quando um novo depósito for cadastrado.
- Cadastro de Cliente
 - O módulo deve permitir o cadastro dos clientes que irão receber as mercadorias pelo Brasil, devendo indicar nome, endereço, cpf, e-mail.
 - O módulo deve informar ao módulo de serviços ao Cliente quando um novo cliente for cadastrado.
- Cadastro de Fornecedor
 - O módulo deve permitir o cadastro dos fornecedores parceiros da Boa Entrega, devendo indicar nome fantasia, cpf ou cnpj, endereço e mercadorias.

- O módulo deve informar ao módulo de serviços ao Cliente quando um novo fornecedor for cadastrado.
- Cadastro de Mercadoria
 - O módulo deve permitir o cadastro de mercadorias referentes aos fornecedores parceiros da Boa Entrega, deve ser informado a descrição, o tipo, a unidade, a quantidade disponível e quem é o fornecedor desta mercadoria.
 - O módulo deve informar ao módulo de serviços ao Cliente quando uma nova mercadoria for cadastrada.

Módulo de Serviços ao Cliente

- Iniciar Entrega
 - O módulo deve permitir o sistema iniciar uma transação de entrega para um cliente, deve ser informado o local onde está a mercadoria e local de entrega para o cliente.
 - O módulo deve informar ao módulo de cadastro a quantidade da mercadoria retirada.
- Atualizar Fluxo Entrega
 - O módulo deve permitir o sistema uma atualização no andamento da entrega do cliente e o depósito em que se encontra.
 - O módulo deve permitir ao usuário informar se a entrega foi extraviada. E o sistema informar ao módulo de gestão e estratégia uma entrega ou mercadoria extraviada.
 - O módulo deve informar o estado em que a entrega se encontra.
- Acusar Recebimento
 - O módulo deve permitir ao cliente uma opção de acusar o recebimento finalizando a entrega do produto.
 - O módulo deve informar ao módulo de gestão e estratégia que uma entrega foi efetuada com sucesso.

Módulo de Gestão e Estratégia

- Gerar Reembolso
 - O Sistema deve inserir uma entrega ou mercadoria extraviada;
 - O sistema deve permitir gerar um ticket de reembolso para o fornecedor da entrega.
- Inserir Entrega Efetuada
 - O sistema deve salvar uma entrega efetuado com sucesso.

3.2 Requisitos Não-Funcionais

- Acessibilidade – A solução deve fornecer os dados para diferentes formas de acesso por diferentes dispositivos.

Estímulo	Acessar a API de qualquer navegador utilizando os clientes de frontend, ou através de interfaces mobile.
Fonte do Estímulo	Cliente da aplicação.
Ambiente	Em funcionamento e com carga normal.
Artefato	Todos os módulos.
Resposta	Permissão de acesso aos serviços.
Medida da Resposta	O sistema deve permitir acesso as funcionalidades.

- Segurança – A solução deve apresentar implementações de segurança para micro serviços e APIs

Estímulo	Acessar um end-point sem autenticação
Fonte do Estímulo	Cliente da aplicação.
Ambiente	Em funcionamento e com carga normal.
Artefato	Todos os módulos exceto de Login.
Resposta	Usuário ganhará resposta 401.
Medida da Resposta	O sistema não deve permitir acesso as funcionalidades.

- Resiliência – A solução deve apresentar formas de integrações entre os sistemas, incluindo os legados, para integrar todos os dados da logística.

Estímulo	Aplicação legada ou novas interfaces fornecendo dados ao módulo de cadastro.
Fonte do Estímulo	Clientes da aplicação.
Ambiente	Módulo de cadastro e Módulo de Serviços ao Cliente.
Artefato	Apenas do módulo de cadastro.
Resposta	Os dados serão integrados aos outros módulos pertinentes a informação gerada no módulo de cadastro.
Medida da Resposta	O sistema deve fornecer os dados através de mensagens geradas no RabbitMQ para os outros módulos.

- Escalabilidade – A solução deve ser possível de ter uma escala horizontal

Estímulo	Realizar uma entrega.
Fonte do Estímulo	Usuário da aplicação.
Ambiente	Em funcionamento e com carga normal.
Artefato	Módulo de Serviços ao Cliente.
Resposta	O Sistema encaminhará uma entrega para um fluxo.
Medida da Resposta	Transitar toda informação entre os módulos até a entrega em residência do cliente.

3.3. Restrições Arquiteturais

- O sistema deve permitir que clientes de diferentes origens (Web, Mobile) se comuniquem através do protocolo HTTP.
- O sistema deve ser modular e de baixo acoplamento facilitando implantação e manutenção.
- O sistema deve ser desenvolvido utilizando o Java.

3.4. Mecanismos Arquiteturais

Mecanismo de Análise	Mecanismo de Design	Mecanismo de Implementação
Persistência	Banco de dados relacional	MySQL
Versionamento	Ferramenta para controlar códigos fontes	Git
Barramento de Mensagens	Integrar os módulos por mensagens	RabbitMQ
Serviço de Registro	Registra os serviços para o identificar as rotas no gateway	Eureka
Framework Módulos	Framework dos módulos	Spring 2.6.2
Aplicação Gateway	Solução para centralizar requisições	Zull Server
Autenticação e Autorização	Protocolo de autenticação	JWT
Ambiente execução	Solução para simular a imagem rodando no cloud.	Docker
Orquestração dos containers	Solução para orquestrar todos os containers.	Docker Compose

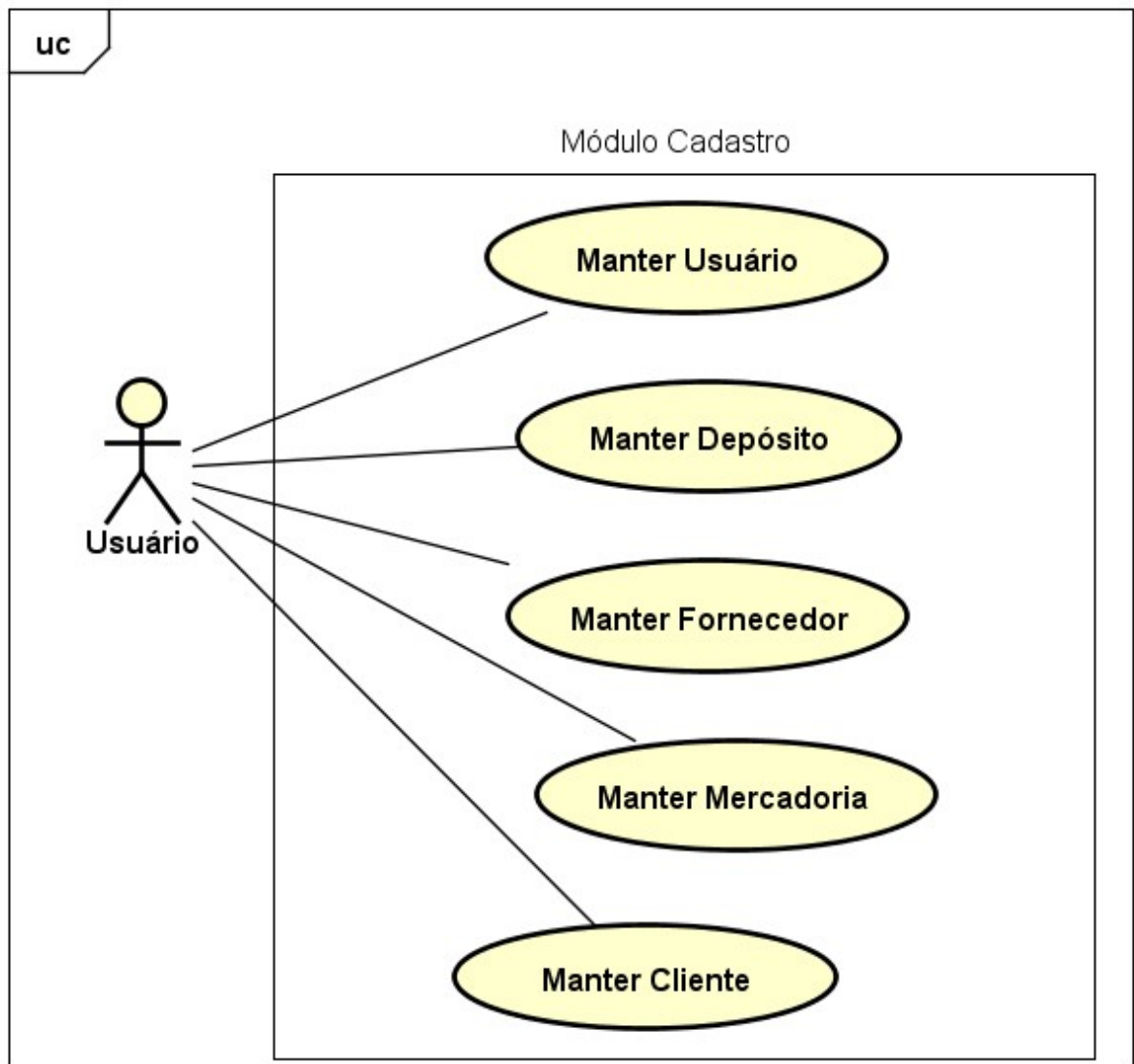
4. Modelagem e projeto arquitetural

Nesta seção são apresentados os diagramas que permitem entender a arquitetura da aplicação, detalhando-a suficientemente para viabilizar sua implementação.

4.1. Casos de uso

Abaixo estão os diagramas de caso de uso que permitem entender de maneira macro a parte de cada módulo referente a arquitetura proposta.

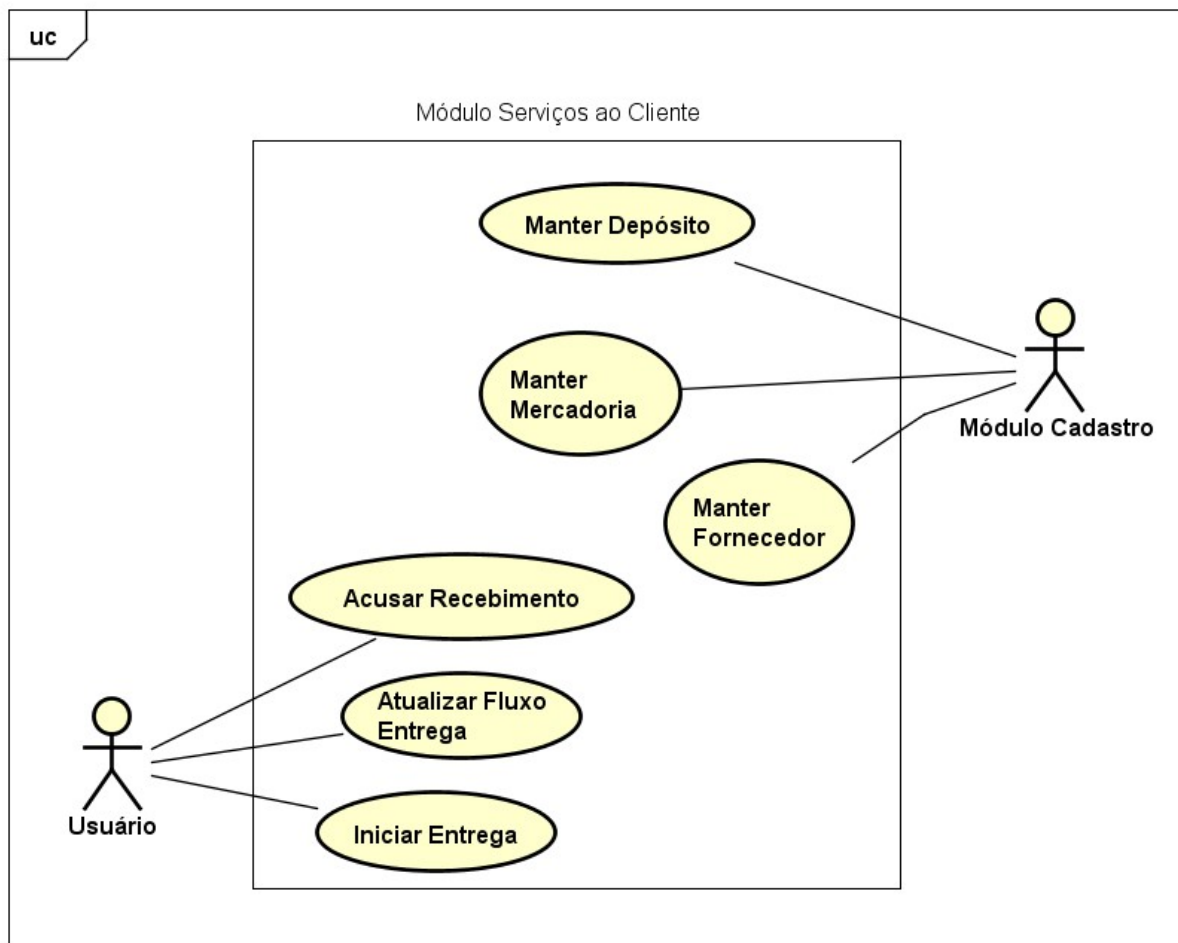
Módulo de Cadastro



powered by Astah

Este caso de uso mostra o modelo macro das funcionalidades do módulo de cadastro, onde o usuário será responsável por imputar as informações que iram gerar o workflow no módulo de serviços ao Cliente.

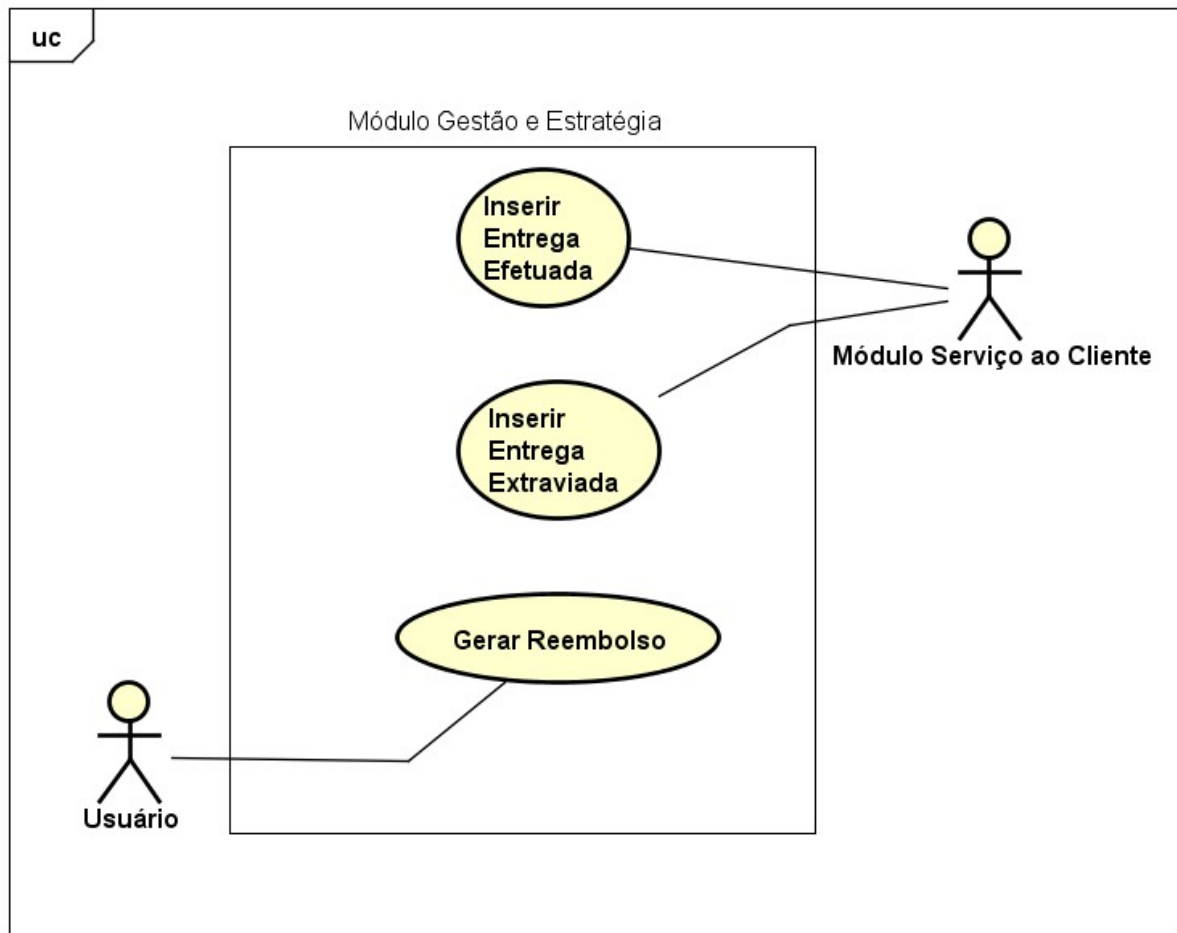
Módulo de Serviços ao Cliente



powered by Astah

Este caso de uso mostra o modelo macro das funcionalidades que irão comportar o módulo de serviços ao cliente, nele estarão todo fluxo da informação pertinentes as entregas dos fornecedores, atendendo também a eventuais problemas no fluxo da entrega da mercadoria.

Módulo Gestão e Estratégia

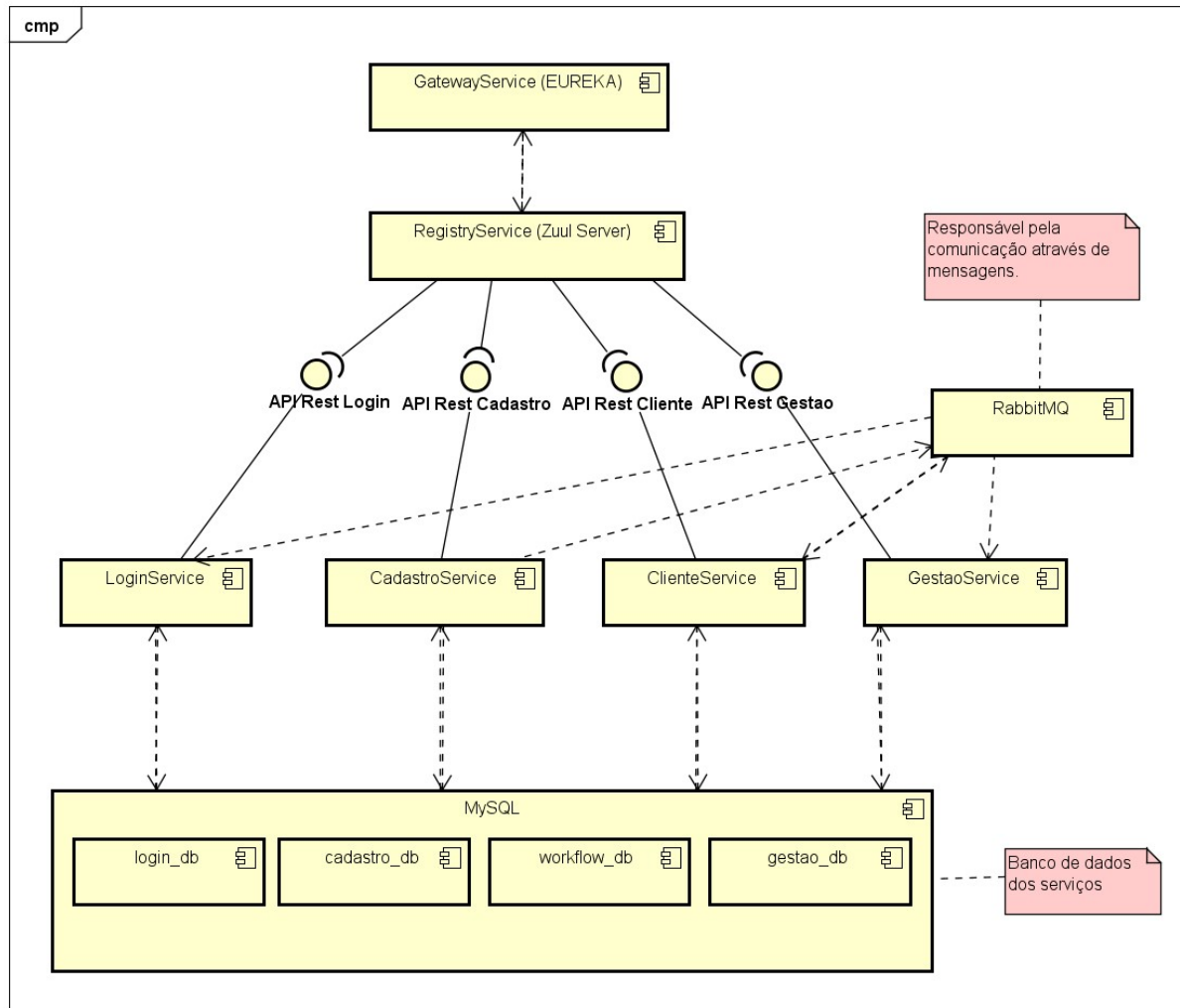


powered by Astah

Este caso de uso mostra o modelo macro das funcionalidades que irão comportar o módulo de gestão e estratégia, nele será permitido quando houver um extravio identifiico no workflow da entrega, o usuário gerar um reembolso que será emitido pela Boa Entrega ao fornecedor do produto. Neste módulo também é possível acompanhar todas as entregas efetuadas com sucesso.

4.2. Modelo de componentes

Abaixo a representação dos componentes e suas ligações, mostrando como os serviços se comunicam e persistem na base de dados.



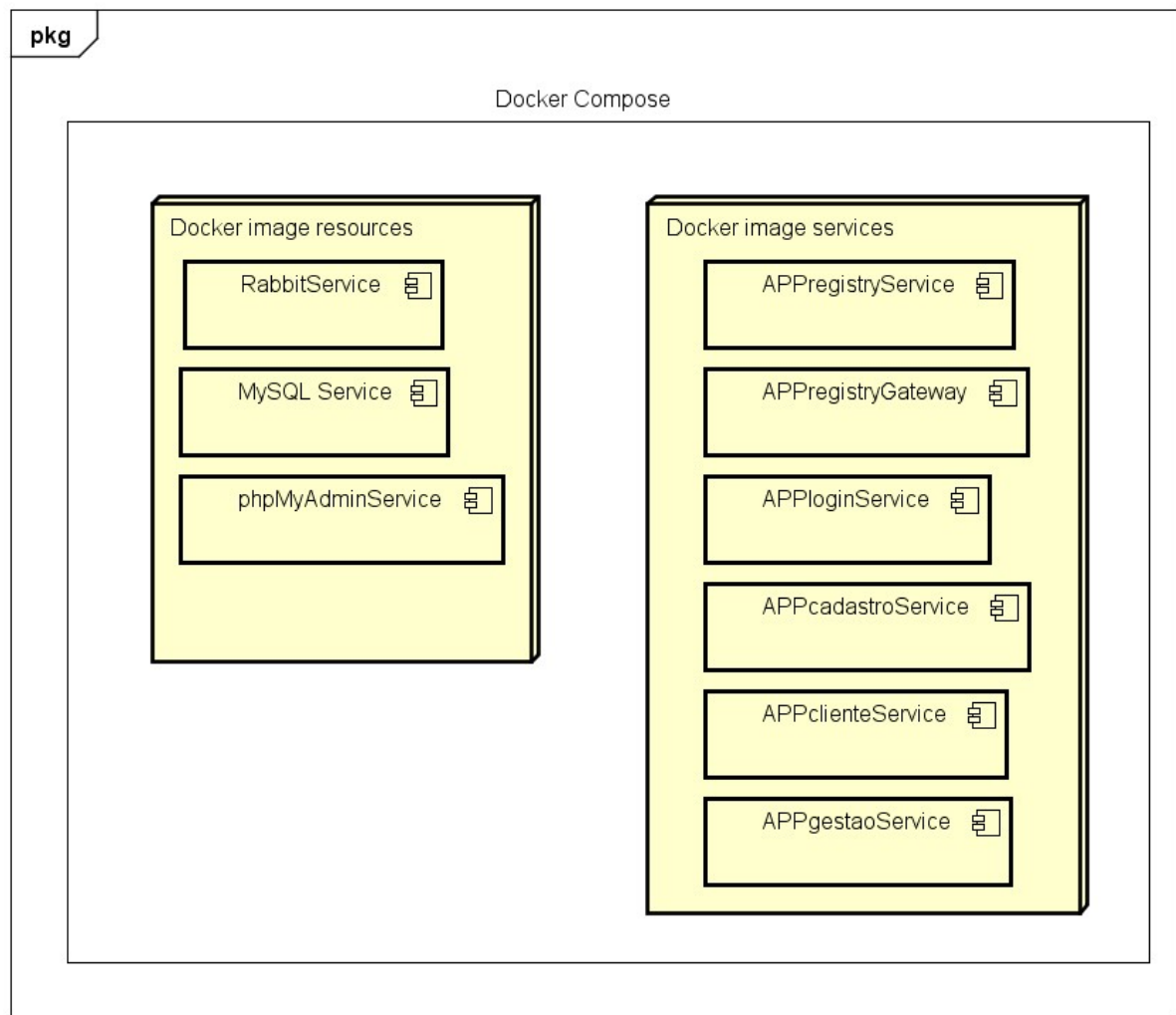
Descrição dos componentes:

Componente	Descrição
RabbitMQ	Message Broker que exerce o papel de barramento de eventos. Os microserviços SCA publicam eventos que são consumidos pelos outros componentes, buscando sempre o menor acoplamento possível entre os microserviços.

Eureka	Eureka permite com que serviços sejam registrados através do Eureka e descobertos através do Eureka Client, facilitando esse controle da aplicação distribuída.
Zuul Server	Zuul Server é uma aplicação gateway que trata de todas as requisições e faz o roteamento dinâmico da aplicação de microsserviços.
MySQL	MySQL é um sistema de gerenciamento de banco de dados, que utiliza a linguagem SQL como interface.

4.3. Modelo de implantação

Neste modelo será demonstrado como estão as aplicações orquestradas nos containers e que irão com o arquivo Docker-compose, para serem iniciadas em algum cluster que possua o Docker instalado, somente será necessário a liberação das portas para comunicação com o lado externo das aplicações.



powered by Astah

Componente	Descrição
Docker image resources	Imagem com todos os recursos e volumes que a aplicação irá necessitar para guardar os dados de mensagens e persistência.
Docker image services	Imagem com todas as aplicações e micro serviços orquestrados para iniciar uma instância em um cluster que tenha o Docker instalado.

5. Prova de Conceito (POC) / protótipo arquitetural

5.1. Implementação e Implantação

O conteúdo deste trabalho está preparado para atender aos problemas de comunicação do software legado, criando uma solução para otimizar o sistema de gestão e cadastro de entrega da logística Boa Entrega. Abaixo seguem os requisitos não funcionais que irão contemplar o projeto, visando a qualidade e sustentabilidade do sistema.

- Tecnologias utilizadas para implementar o projeto estão descritas no apêndice 3.4 com o nome de Mecanismos Arquiteturais.
- Os casos de usos implementados para demonstrar os módulos e as interações estão exibidos e descritos no apêndice 4.1 com o nome de Casos de Uso.

Segue abaixo os requisitos não funcionais para validação dos serviços:

Acessibilidade

Requisito não funcional escolhido, devido ao sistema atender a todos os tipos de dispositivos.

- O sistema deve permitir que os endpoints que permitam acessos de diferentes tipos de navegadores e clientes.

Segurança

Requisito não funcional escolhido, que visa atender a necessidade de manter dados sigilosos em segurança e com restrições de visibilidade.

- O usuário deve estar autenticado no serviço para Login para utilizar os serviços providos pelos sistemas.
- O usuário deve ter um acesso munido de senha pessoal e um perfil para visualizar e utilizar o sistema.
- O sistema deve autenticar e identificar o usuário através do token gerado com um tempo para expirar.

Resiliência

O requisito foi escolhido pois, o sistema deve possuir baixo acoplamento entre os componentes, e caso algum destes serviços tenha uma parada técnica, não interfira no processo do outro. Sendo assim, quando retornar da indisponibilidade nenhum dado seja perdido.

- O envio de mensagem dos módulos e serviços não deve ser interrompido por nenhum módulo que estiver indisponível, e nem interferir no funcionamento de outros módulos.
- Após o reestabelecimento do serviço que estava fora do ar, o sistema deve processar as mensagens que ficaram na fila de espera e emitir alerta de funcionamento.

Escalabilidade

Requisito não funcional que foi escolhido devido a preocupação com a implantação da solução, não é passível que gere conflitos ou exceções entre os dados e nem com as aplicações que estarão instaladas de forma horizontal.

- O sistema deve processar todas as mensagens dos módulos e os fluxos da entrega, sem que haja conflitos, exceções entre os dados que tramitam na infraestrutura entre um serviço e outro.

5.2 Interfaces/ APIs

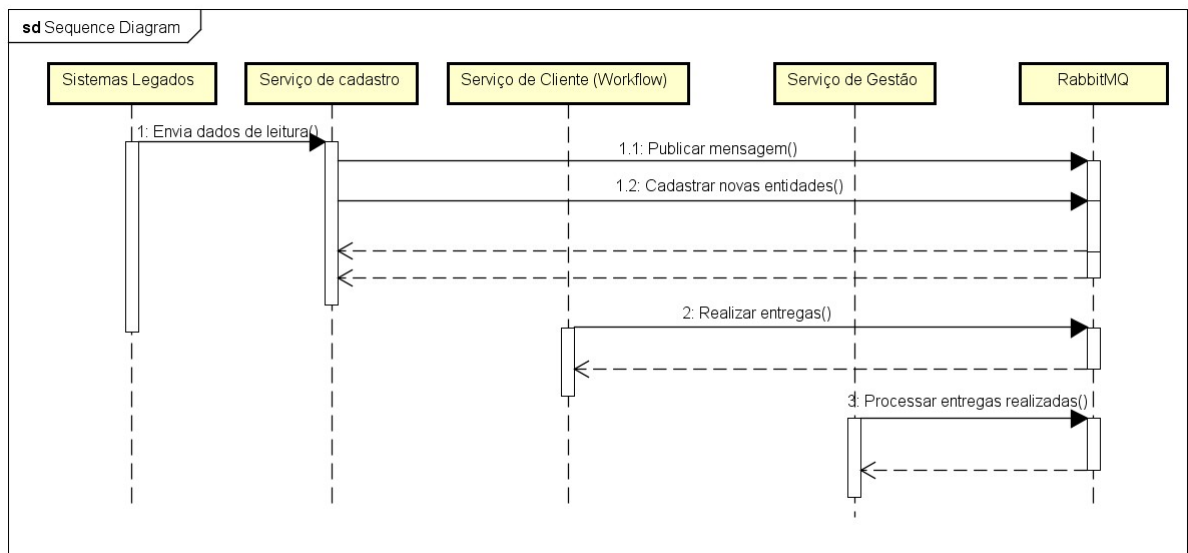
Caso exista algum componente na arquitetura da sua aplicação que é genérico e pode ser utilizado em outras aplicações semelhantes, documente a interface desse componente seguindo um modelo de documentação de interfaces.

6. Avaliação da Arquitetura

A avaliação da arquitetura desenvolvida neste trabalho é abordada nesta seção, visando avaliar se atende ao que foi proposto.

6.1. Análise das abordagens arquiteturais

As características desta abordagem arquitetural é proporcionar um sistema integro com baixo acoplamento entre os módulos, para facilitar a manutenção e aumentar a escalabilidade entre os serviços, proporcionando mais eficiência nas transações e operações realizadas nas interfaces do sistema. O RabbitMQ foi utilizado como recurso para prover as mensagens de comunicação entre os microserviços, onde buscamos seguir uma arquitetura orientada a eventos, a partir das operações que são efetuadas entre as comunicações com os sistemas legados e os novos serviços. Abaixo será demonstrado o diagrama de sequência explicando como funciona este trâmite e as mensagens.



powered by Astah

No projeto arquitetural os consumidores estão dentro de cada microserviço, e a partir de cada requisição realizada de acordo com a regra definida irá produzir as mensagens a serem consumidas pelos serviços a quais são destinadas, seguindo o plano de escalabilidade sem afetar os outros processamentos, e outras requisições que são feitas aos serviços, sendo assim de acordo com a projeto arquitetural proposto, nenhuma requisição irá impactar em outros processamentos.

6.2. Cenários

Apresente os cenários utilizados na realização dos testes da sua aplicação. Escolha cenários de testes que demonstrem os requisitos não funcionais (atributos de qualidade) sendo satisfeitos. Priorize os cenários para a avaliação segundo critérios quantitativos ou qualitativos.

6.3. Avaliação

Apresente as evidências dos testes de avaliação. Apresente as medidas registradas na coleta de dados. Para aquilo que não for possível quantificar apresente uma justificativa baseada em evidências qualitativas que suportem o atendimento aos requisitos não-funcionais. As evidências das avaliações neste item são fundamentais.

6.4. Resultado

Considerando todos os atributos e requisitos não funcionais, foram realizados os testes de aceitação e após verificado a prova de conceito foi atingida com sucesso, sendo assim os microserviços irão entregar uma solução de qualidade, seguindo os preceitos de segurança, escalabilidade, resiliência e acessibilidade.

Requisito não funcional	Testado	Homologado
Segurança	Sim	Sim
Acessibilidade	Sim	Sim
Escalabilidade	Sim	Sim
Resiliência	Sim	Sim

Os principais pontos de testes que foram testados, foi a questão do baixo acoplamento entre os serviços, seguindo um nível granular por aplicação. Alguns pontos de melhorias devem ser implementados no projeto final.

O Docker serviu para criar os ambientes de forma virtual, simulando e orquestrados para atender as necessidades do sistema, os banco de dados dos serviços são independentes um dos outros e seguindo uma boa prática para implementação de microserviços.

7. Conclusão

Com este projeto arquitetural foi possível demonstrar a comunicação entre microserviços utilizando a arquitetura orientada a eventos, este projeto foi de extrema aprendizagem para por em prática todos os conceitos aprendidos durante o decorrer do curso.

O objetivo final é atender a logística da Boa Entrega de forma eficiente a gerir seus clientes, fornecedores e garantir a entrega de qualidade dos produtos aos clientes finais. Visando uma arquitetura de transparência, tanto para quem fornece quanto para quem compra um produto com os parceiros da Boa Entrega.

REFERÊNCIAS

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

APÊNDICES

Repositório do projeto:

➤ <https://github.com/edusil16/microservices-puc>

Inclua o URL do vídeo mostrando uma apresentação da POC, contendo de 3 a 5 minutos.
Utilize preferencialmente o formato avi.

CHECKLIST PARA VALIDAÇÃO DOS ITENS E ARTEFATOS DO TRABALHO

Nº	Item a ser cumprido	Sim	Não	Não se aplica
Completeza do documento				
1	Todos os elementos iniciais do documento (capa, contracapa, resumo, sumário...) foram definidos?	x		
2	Os objetivos do trabalho (objetivos gerais e pelo menos três específicos) foram especificados?	x		
3	Os requisitos funcionais foram listados e priorizados?	x		
4	Os requisitos não funcionais foram listados e identificados usando o estilo estímulo-resposta?	x		
5	As restrições arquiteturais foram definidas?	x		
6	Os mecanismos arquiteturais foram identificados?	x		
7	Um diagrama de caso de uso foi apresentado junto com uma breve descrição de cada caso de uso?	x		
8	Um modelo de componentes e uma breve descrição de cada componente foi apresentada?	x		
9	Um modelo de implantação e uma breve descrição de cada elemento de hardware foi apresentada?	x		
10	Prova de conceito: uma descrição da implementação foi feita?	x		
11	Prova de conceito: as tecnologias usadas foram listadas?	x		
12	Prova de conceito: os casos de uso e os requisitos não funcionais usados para validar a arquitetura foram listados?	x		
13	Prova de conceito: os detalhes da implementação dos casos de uso (telas, características, etc) foram apresentadas?	x		
14	Prova de conceito: foi feita a implantação da aplicação e indicado como foi feita e onde está disponível?	x		
15	As interfaces e/ou APIs foram descritas de acordo com um modelo padrão?	x		
16	Avaliação da arquitetura: foi feita uma breve descrição das características das abordagens da proposta arquitetural?	x		
17	Avaliação da arquitetura: Os atributos de qualidade e os cenários onde eles seriam validados foram apresentados?	x		
18	Avaliação da arquitetura: uma avaliação com as evidências dos testes foi apresentada?	x		
19	Os resultados e a conclusão foram apresentados?	x		
20	As referências bibliográficas foram listadas?	x		
21	As URLs com os códigos e com o vídeo da apresentação da POC foram listadas?	x		

Nº	Item a ser cumprido	Sim	Não	Não se aplica
Consistência dos itens do documento				
1	Todos os requisitos funcionais foram mapeados para casos de uso?	x		
2	Todos os casos de uso estão contemplados na lista de requisitos funcionais?	x		
3	Os requisitos não funcionais, mecanismos arquiteturais e restrições c arquiteturais estão coerentes com os modelos de componentes e implantação?	x		
4	Os modelos de componentes e implantação estão coerentes com os requisitos não funcionais, mecanismos arquiteturais e restrições arquiteturais?	x		
5	As tecnologias listadas na implementação estão coerentes com os requisitos não funcionais, mecanismos arquiteturais e restrições arquiteturais?	x		
6	Os casos de uso e os requisitos não funcionais listados na implementação estão coerentes com o que foi listado nas seções anteriores?	x		
7	Os atributos de qualidade usados na avaliação estão coerentes com os requisitos não funcionais na sessão 3?	x		
8	Os cenários definidos estão no contexto dos casos de uso implementados?	x		
9	O apresentado no item resultado está coerente com o que foi mostrado no item avaliação?	x		