

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

**Ricardo Ludwig Pereira**

**SISTEMA DE COMÉRCIO ELETRÔNICO PARA O AGRONEGÓCIO**

Belo Horizonte  
2018

**Ricardo Ludwig Pereira**

**ARQUITETURA DE UM SISTEMA DE COMÉRCIO ELETRÔNICO PARA O  
AGRONEGÓCIO**

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

Orientador: Prof. Tadeu dos Reis Faria

Belo Horizonte

2018



## RESUMO

O agronegócio é um importante setor da economia brasileira, no entanto a sua presença online é pequena em relação a outros setores, como exemplo o setor varejista, segmento já consolidado na atuação em comércio eletrônico. Em pesquisa realizada na internet no período de desenvolvimento deste trabalho foi constatado que as iniciativas de comércio eletrônico voltadas para o agronegócio estão aquém do que seria esperado. Em alguns casos requisitos facilmente verificáveis como usabilidade e acessibilidade dificultam a experiência do usuário; não é raro a inviabilidade do uso do site em dispositivos móveis. Tendo esse cenário em vista, acredita-se haver um grande potencial econômico para o crescimento e desenvolvimento do comércio eletrônico de produtos relacionados ao agronegócio, com isso é proposto neste trabalho o desenvolvimento de uma plataforma de comércio eletrônico onde requisitos de arquitetura e desenvolvimento de sistemas distribuídos sejam aplicados, proporcionando tanto aos consumidores como aos vendedores uma boa experiência durante todo o fluxo de negócio para venda de um produto, desde sua disponibilização na plataforma até a sua entrega ao cliente.

**Palavras-chave:** sistemas distribuídos, arquitetura de sistemas, microserviços, integração de sistemas, usabilidade, acessibilidade, netflix oss, spring cloud, comércio eletrônico, agronegócio.

## SUMÁRIO

<b>1. Apresentação.....</b>	<b>6</b>
1.1. Problema.....	6
1.2. Objetivos do trabalho.....	7
<b>2. Requisitos.....</b>	<b>7</b>
2.1. Requisitos Funcionais.....	7
2.2. Requisitos Não-Funcionais.....	11
2.3. Restrições Arquiteturais.....	15
2.4. Mecanismos Arquiteturais.....	15
<b>3. Modelagem e projeto arquitetural.....</b>	<b>16</b>
3.1. Modelo de casos de uso.....	16
3.1.1. Histórias de usuários.....	17
3.1.2. Descrição de caso de uso crítico.....	24
3.2. Modelo de componentes.....	27
3.3. Modelo de implantação.....	27
<b>4. Prova de conceito / protótipo arquitetural.....</b>	<b>27</b>
4.1. Implementação e implantação.....	27
4.2. Interfaces.....	34
<b>5. Avaliação da Arquitetura.....</b>	<b>37</b>
5.1. Cenários.....	37
5.2. Avaliação.....	39
<b>REFERÊNCIAS.....</b>	<b>45</b>
<b>APÊNDICES.....</b>	<b>46</b>

## **1. Apresentação**

### **1.1. Problema**

Há uma baixa adesão do agronegócio ao comércio eletrônico, fatores que podem influenciar esse cenário são a resistência de produtores que preferem se manter fiéis as formas as quais já estão adaptados de negociar seus produtos, outro ponto são os pequenos produtores, esses por não possuírem uma produção em larga escala por vezes são incapazes de suprir a demanda de vendas no qual o negócio pode exigir. As alternativas a esses produtores é a inclusão de intermediários, como distribuidores ou cooperativas, mas isso torna o processo mais burocrático e financeiramente menos vantajoso uma vez que esses intermediários tendem a exigir contrapartida financeira para estabelecimento e manutenção do vínculo entre as partes, por fim, a indisponibilidade à conexões de internet nas regiões produtoras remotas inviabiliza a gestão de um site de comércio eletrônico.

Através do desenvolvimento de uma plataforma de comércio eletrônico, apresentada no decorrer desse trabalho, pretende-se impulsionar essa atividade para este segmento mitigando os principais problemas apresentados anteriormente. O sistema será capaz de lidar com vendas ao atacado do tipo Business to Business (B2B) e vendas ao varejo do tipo Business to Consumer (B2C) e para ambos modelos de negócio os produtores poderão disponibilizar seus produtos para venda em um sistema conhecido como Marketplace ou, caso queiram, criar seu próprio site. O Marketplace pode viabilizar que pequenos produtores disponibilizem seus produtos sem a necessidade de possuírem uma grande produção possibilitando a eliminação de intermediários. Acredita-se que com o desenvolvimento do sistema visando a usabilidade e a experiência do usuário possa diminuir a resistência de alguns produtores à incorporação do comércio eletrônico ao seu modelo de negócio. O sistema permitirá que cliente especiais possam enviar seus orçamentos por integração de sistemas, outra funcionalidade será a notificação aos compradores, produtores ou revendedores a respeito da necessidade de coleta seletiva de material com controle de descarte, exemplo, embalagens de agrotóxico.

## **1.2. Objetivos do trabalho**

Esse trabalho tem como objetivo apresentar o desenvolvimento de um sistema de comércio eletrônico voltado para o agronegócio, fazendo uso de técnicas de integração de sistemas, arquitetura e desenvolvimento de software, usabilidade e experiência de usuário. Um conjunto de sistemas deverão comunicar-se entre si, todas as interfaces web deverão ser responsivas para suportar as diferentes dimensões dos diversos dispositivos existentes atualmente.

O sistema será separado por módulos e cada um deles agrupará as funcionalidades referentes ao seu domínio de negócio, abaixo são listados os módulos:

- cliente;
- fornecedor;
- produto;
- venda;
- estoque;
- logística;
- marketing;
- administrativo;
- reciclagem;
- autenticação e autorização;

## **2. Requisitos**

### **2.1. Requisitos Funcionais**

Abaixo são descritos os requisitos funcionais para cada domínio da plataforma.

### **Cliente**

Os clientes poderão ser pessoas jurídicas ou físicas, as funcionalidades relacionadas aos clientes são:

- cadastro, atualização e exclusão de clientes;
- cadastro, atualização e exclusão de múltiplos endereços;
- o cliente deverá ser capaz de autenticar no site com o e-mail informado e sua senha;
- o cliente deverá ser capaz de alterar seu e-mail e senha;
- o cliente poderá, buscar por produtos, adicionar ou remover produtos do carrinho;
- o cliente deverá ser capaz de finalizar seu pedido de compra, desde que as seguintes condições sejam observadas. Clientes já cadastrados deverão informar um endereço de entrega, esse pode ser obtido de endereços cadastrados anteriormente ou um novo poderá ser informado. Clientes não cadastrados deverão antes de informar o endereço de entrega informar um e-mail de cadastro, esse deve estar ciente que o e-mail informado deverá ser válido e uma mensagem de confirmação de cadastro será enviada para esse e-mail, ou seja, o e-mail informado será o login do cliente para autenticação no site. Ao informar o e-mail o cliente deverá ser capaz de informar o endereço de entrega e continuar no processo de finalização do pedido;
- o cliente deverá informar o meio de pagamento, perante as opções disponíveis;
- o cliente poderá consultar o status do pedido, sendo os seguintes status possíveis: em aprovação, aprovado, despachado, entregue, cancelado, despachado para devolução, devolução recebida;
- o cliente poderá cancelar o pedido em até 7 dias;



– clientes especiais poderão solicitar orçamentos e buscar produtos por meio de uma API de integração de serviços.

### **Fornecedor**

Fornecedor poderá ser representado por pessoas físicas ou jurídicas, como produtores agrícolas, fabricantes de agrotóxicos ou maquinários agrícolas, cooperativas e distribuidores, o sistema deverá ser capaz de permitir o cadastro, exclusão e atualização de fornecedores, além desses deverá haver a funcionalidade de bloqueio de fornecedores, para casos onde o cadastro do fornecedor não é desejado ou para bloqueio temporário de um fornecedor já cadastrado. As razões que podem levar ao bloqueio de um fornecedor, são acordos comerciais que podem exigir exclusividade, mau uso, dívidas ou bloqueio temporário solicitado pelo fornecedor.

### **Produto**

Um produto deve ser composto obrigatoriamente por um código identificador único, identificação do produtor, identificação de origem, opcionalmente pode possuir código da unidade de manutenção de estoque (SKU), código numérico de barras (EAN), preço por unidade e características do produto (volume, tamanho, cor, etc.).

Um produto pode fazer parte de um pacote no qual conterá diversos produtos e possuirá preço próprio, diferente do preço por unidade, suas características físicas tais como tamanho e peso não necessariamente estarão relacionadas aos produtos presentes no pacote.

Produtos poderão ser cadastrados, alterados e excluídos, para isso deverá haver uma interface web para execução dessas funcionalidades por um operador do sistema, também deverá haver a possibilidade dessas atividades serem executadas diretamente por outro sistema, ou seja, sem a necessidade de intervenção humana.

### **Venda**

Pedidos para serem finalizados devem conter ao menos um produto, pedidos devem ter rastreabilidade para possibilitar o cliente acompanhar todos os seus status (em aprovação, aprovado, despachado, entregue, cancelado, despachado para devolução, devolução recebida) devem ser auditados. Pedidos poderão ser

feitos por clientes ou por assistentes de vendas em nome do cliente. Pedidos deverão ser feitos pelo site através de qualquer dispositivo com acesso à Internet e por API de integração de serviços.

O sistema de vendas deve prover os seguintes meios de pagamentos, cartão de crédito ou mecanismo de pagamento on-line, como por exemplos Paypal, Pag Seguro.

### **Estoque**

Produtos poderão ser cadastrados e excluídos do estoque, produtos cadastrados no estoque deverão possuir um código de unidade de manutenção em estoque (SKU). Usuários ou sistemas através de integrações deverão ser capazes de consultar o estoque para verificar a disponibilidade de algum produto.

### **Logística**

A logística deverá calcular a volumetria de um pedido para com essa informação acompanhada dos endereços de entrega otimizar os custos para entregas dos pedidos.

### **Marketing**

O marketing deve ser capaz de gerar campanhas que possam ser rapidamente inseridas na plataforma sem o envolvimento da equipe de T.I e sem prejudicar o correto funcionamento do sistema, uma interface web deve permitir que a equipe de marketing crie, modifique ou desabilite campanhas.

### **Administrativo**

Deve permitir obter relatórios de vendas, custos, estoque, rentabilidade, fornecedores, histórico de compras por clientes, por produtos e por fornecedores.

### **Reciclagem**

Deve integrar fabricantes ou revendedores de produtos com descarte controlado aos compradores desses. As seguintes atividades devem ser executadas:

- informar ao cliente que adquiriu um produto com descarte controlado a necessidade de fazê-lo, essa informação deve ser enviada ao cliente quando se

acreditar que o produto já tenha sido utilizado, para isso deve usar a média histórica de compras de produto pelo cliente, caso essa informação não exista usar a média histórica dos demais compradores do mesmo produto e se ainda assim não houver essa informação um intervalo padrão entre a data de compra e um possível período de uso deve ser considerado a data de validade do produto também deverá ser considerada;

– fabricantes ou revendedores devem ser informados da necessidade de retirar o descarte de seus produtos no cliente.

### **Autenticação e Autorização**

Responsável pelo controle de acesso as áreas restritas do sistema por usuários e clientes.

## **2.2. Requisitos Não-Funcionais**

### **Segurança**

Todas as requisições para a plataforma devem ser através do protocolo HTTPS.

<b>Estímulo</b>	Usuário fazendo uma busca por um produto específico no site da aplicação de comércio eletrônico.
<b>Fonte do Estímulo</b>	Usuário fazendo busca de produtos na aplicação.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo de busca de produtos.
<b>Resposta</b>	Somente requisições através do protocolo HTTPS.
<b>Medida da resposta</b>	Todas as requisições do usuário deveram ter sido feitas através do protocolo HTTPS.

Requisição através do protocolo HTTPS, criptografada, autenticada.

<b>Estímulo</b>	Realização da compra de um produto.
<b>Fonte do Estímulo</b>	Usuário acessando o site da aplicação e efetuando a

	compra de um produto.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo de vendas e Módulo de Autenticação.
<b>Resposta</b>	O usuário deve ter sido autenticado e autorizado para efetuar concluir a compra.
<b>Medida da resposta</b>	Compra realizada com sucesso após o usuário ter sido autenticado.

### Desempenho

A aplicação deve suportar 100 requisições por minuto sem degradação de desempenho, ou seja, o tempo de resposta até o usuário final para cada requisição deve ser inferior a 7.5 segundos.

<b>Estímulo</b>	Usuário fazendo uma busca por um produto específico no site da aplicação de comércio eletrônico.
<b>Fonte do Estímulo</b>	Usuário fazendo busca de produtos na aplicação.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo de busca de produtos.
<b>Resposta</b>	Resposta da requisição dentro do limite de tempo estipulado.
<b>Medida da resposta</b>	O tempo de resposta de todas as requisições mensuradas não deve ser inferior a 7.5 segundos.

### Tolerância a falhas e disponibilidade

O cliente deve ser capaz de efetuar compra independentemente de falhas na plataforma.

<b>Estímulo</b>	Desligamento dos servidores do cluster responsáveis pelo fluxo de vendas.
<b>Fonte do Estímulo</b>	Administrador dos servidores da aplicação.
<b>Ambiente</b>	Usuários utilizando o sistema.
<b>Artefato</b>	Gerenciador de cluster.
<b>Resposta</b>	A falha deve ser imperceptível aos usuários que

	utilizam o sistema naquele momento.
<b>Medida da resposta</b>	Todas solicitações dos usuários devem ter sido atendidas, com atraso apenas ocasionado pela troca de servidor pelo gerenciador de cluster.

A aplicação deve suportar uma grande quantidade de requisições evitando dessa forma lentidão ou sua indisponibilidade.

<b>Estímulo</b>	Usuário fazendo uma busca por um produto específico no site da aplicação de comércio eletrônico.
<b>Fonte do Estímulo</b>	Usuário fazendo busca de produtos na aplicação.
<b>Ambiente</b>	Aumento de 80% na quantidade de requisições à aplicação.
<b>Artefato</b>	Módulo de busca de produtos.
<b>Resposta</b>	Degradação inferior a 3% no tempo de resposta do ambiente comparado ao tempo de resposta da aplicação em condições de quantidade de requisições inferior ao aumento registrado.
<b>Medida da resposta</b>	Nenhuma requisição deverá ser perdida e o tempo de resposta não será superior em 3% ao tempo usual de resposta da aplicação em condições normais de uso.

### Testabilidade

Os módulos da plataforma de comércio eletrônico deve ter cobertura de testes unitários e testes automatizados.

<b>Estímulo</b>	Execução dos testes unitários.
<b>Fonte do Estímulo</b>	Executar a suíte de testes unitários com o maven.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Suíte de testes unitários de todos os módulos da plataforma de comércio eletrônico.
<b>Resposta</b>	Todos os testes devem ter sido executados.
<b>Medida da resposta</b>	Todos os testes executados devem ter sido com sucesso.

A aplicação deve possuir testes de integração automatizados.

<b>Estímulo</b>	Execução do teste de integração.
<b>Fonte do Estímulo</b>	Executar a suíte de teste de integração para o módulo de venda efetuando um pedido.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo de venda.
<b>Resposta</b>	Teste de executado com sucesso.
<b>Medida da resposta</b>	Deve existir o pedido do teste na plataforma.

### Interoperabilidade

A plataforma de comércio eletrônico deve ser capaz de integrar-se com sistemas legados dos fornecedores, por exemplo sistemas desenvolvidos em Cobol.

<b>Estímulo</b>	Teste de conexão.
<b>Fonte do Estímulo</b>	Informar o fornecedor da necessidade da retirada de material de descarte.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo de descartes.
<b>Resposta</b>	Envio das informações para retirada do material de descarte.
<b>Medida da resposta</b>	O recebimento das informações pelo fornecedor.

### Acessibilidade e Usabilidade

O site do sistema deve ser web responsivo, suas dimensões e fluxo de navegação devem se adequar aos diversos tamanhos de telas disponíveis para acessar o site.

<b>Estímulo</b>	Realização da compra de um produto.
<b>Fonte do Estímulo</b>	Usuário acessando o site de um celular.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo de vendas.
<b>Resposta</b>	A camada de apresentação adaptou-se ao tamanho

	da tela do dispositivo usado, tornando a navegação fluída.
<b>Medida da resposta</b>	Não houve alteração funcional, o usuário do dispositivo móvel é capaz de executar a mesma atividade do que um usuário utilizando um computador de mesa.

O site da plataforma deve seguir as normas da Web Content Accessibility Guidelines (WCAG)

<b>Estímulo</b>	Realização da compra de um produto.
<b>Fonte do Estímulo</b>	Usuário deficiente visual acessando o site de um computador de mesa.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo de vendas.
<b>Resposta</b>	As cores do site devem estar adequadas para os diversos graus de deficiência visual, o fluxo de navegação deve estar adequado para utilização apenas do teclado e para o software de leitura de tela.
<b>Medida da resposta</b>	O usuário deficiente visual deve ser capaz de executar a compra sem dificuldades.

### 2.3. Restrições Arquiteturais

- O sistema deverá ser desenvolvido em Java;
- Deverão ser utilizados o frameworks Spring Boot, Spring Security e Spring Cloud;
- Deverá ser utilizada a arquitetura de microsserviços;
- As interfaces web deverão ser responsivas;

### 2.4. Mecanismos Arquiteturais

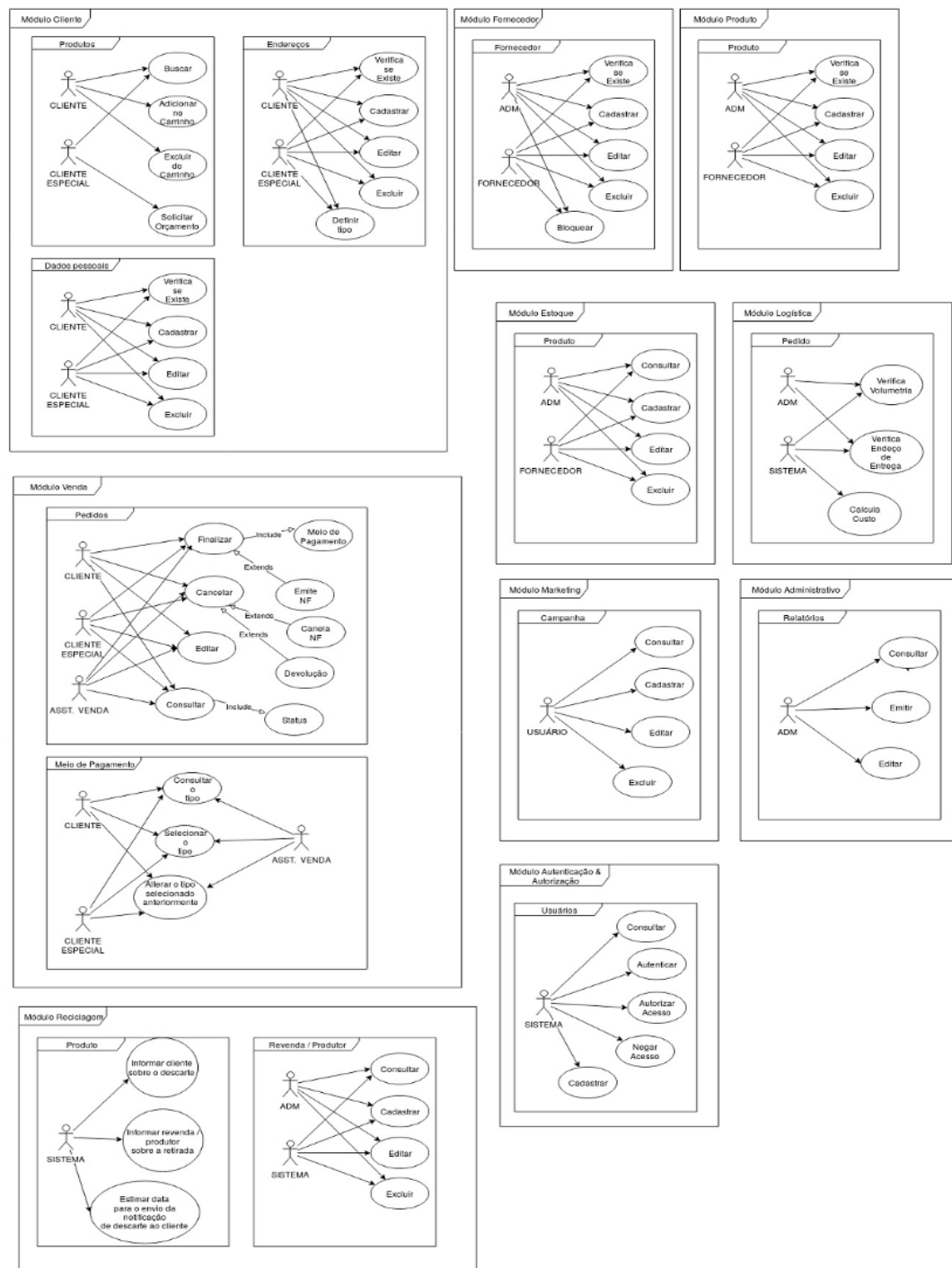
<b>Mecanismo de Análise</b>	<b>Mecanismo de Design</b>	<b>Mecanismo de Implementação</b>
Persistência	Banco de dados relacional	PostgreSQL
Persistência	Framework ORM	Hibernate
Comunicação entre processos	Contêiner	Tomcat
Log	Implementação dos recursos de log do componente.	Log4J
Build	Geração de artefato para o servidor de aplicação.	Maven
Front-End	Interface do usuário	Spring MVC Bootstrap Javascript CSS HTML
Versionamento	Versionamento dos fontes da aplicação.	GIT
Autenticação	Verificação das credenciais. Ex. Usuário e senha	Spring Security
Autorização	Verificação do perfil de usuário, autorizando ou negando acesso.	Spring Security
Criptografia	Mensagens devem ser transferidas criptografadas.	Spring Security
Alta Disponibilidade	Load Balance de Requisições Proxy e Gateway das Aplicações	Zuul
Descoberta	Registro de Serviços	Eureka
Sistema Operacional	Sistema que será executado o servidor de aplicação.	Linux

### 3. Modelagem e projeto arquitetural

#### 3.1. Modelo de casos de uso

O diagrama de casos de uso oferece uma visão global dos casos de uso e dos atores que dele participam.





### 3.1.1. Histórias de usuários

**Cliente busca por produto**

**ID. 01**

Como cliente quero buscar produtos para verificar sua

disponibilidade e preços.	
<p>Nenhuma informação é passada no campo busca e ao clicar em buscar uma mensagem deve aparecer solicitando o critério de busca.</p> <p>Nenhum produto é encontrado, uma mensagem de produto não encontrado deve aparecer.</p> <p>O critério de busca informado pelo cliente retorna os produtos encontrados, detalhes do produto e seu preço devem ser exibidos ao cliente.</p>	
Prioridade: Alta	Estimativa: 40 Horas

<b>Cliente adiciona produto ao carrinho</b>	<b>ID. 02</b>
<p>Como cliente quero adicionar produtos ao carrinho do site e continuar navegando até que eu decida finalizar a minha compra.</p> <p>Apenas a quantidade de produto informada pelo cliente deve ser adicionada ao carrinho.</p> <p>Novo produto adicionado ao carrinho não deve subscrever os já existentes.</p> <p>Adição de produto já existente deve incrementar a quantidade.</p> <p>Ao finalizar a compra o carrinho deve ser esvaziado.</p> <p>Caso a sessão do cliente fique inativa ele deve ser capaz de ao conectar novamente recuperar o carrinho com os itens que havia adicionado.</p>	
Prioridade: Alta	Estimativa: 80 Horas

<b>Cliente exclui produto do carrinho</b>	<b>ID. 03</b>
<p>Excluir produtos que adicionei ao carrinho para não comprar um produto no qual não tenho mais interesse ou tenha adicionado inadvertidamente.</p>	
<p>Apenas o item selecionado e quantidades informadas pelo cliente</p>	

<p>devem ser excluídos.</p> <p>A opção de excluir produtos do carrinho deve estar disponível apenas se houver produtos adicionados.</p>	
Prioridade: Alta	Estimativa: 30 Horas

<b>Orçamento de cliente especial</b>	<b>ID. 04</b>
<p>Como cliente especial quero solicitar orçamentos para verificar se obtenho melhor custo.</p>	
<p>A lista de compras do cliente deve conter ao menos um produto.</p> <p>Produtos constantes da lista compras que não estiverem disponíveis para venda deveram ser listados ao cliente.</p>	
Prioridade: Alta	Estimativa: 30 Horas

<b>Cliente dados pessoais</b>	<b>ID. 05</b>
<p>Como cliente quero cadastrar, editar ou excluir meus dados cadastrais para ter controle sobre minhas informações pessoais e mantê-las sempre atualizadas.</p>	
<p>Os seguintes dados poderão ser preenchidos como dados pessoais: nome e sobrenome, e-mail, data de nascimento, telefones para contato (no máximo três), CPF, dados de cartão de crédito.</p> <p>Todos os campos deverão ter validações de preenchimento.</p> <p>A opção excluir fará com que todos os dados sejam excluídos.</p>	
Prioridade: Média	Estimativa: 40 Horas

<b>Cliente dados de endereço</b>	<b>ID. 06</b>
<p>Como cliente quero cadastrar, editar ou excluir endereços de entrega ou cobrança para poder selecionar diferentes endereços de entregas para meus pedidos e sempre manter meu endereço de</p>	

cobrança atualizado.	
<p>O cliente poderá cadastrar quantos endereços julgar necessários. O cliente poderá informar o tipo de utilização do endereço, para entrega, cobrança ou para ambos.</p> <p>Deve haver apenas um endereço de cobrança.</p> <p>Os campos permitidos para o preenchimento dos endereços são: logradouro, complemento, CEP, ponto de referência, informar se o endereço é para entrega ou cobrança.</p>	
Prioridade: Alta	Estimativa: 40 Horas

<b>Cliente pode criar uma conta de acesso ao sistema de comércio eletrônico.</b>	<b>ID. 07</b>
<p>Eu como cliente quero criar uma conta de acesso ao site para poder efetuar compras no sistema de comércio eletrônico.</p> <p>O e-mail informado durante o cadastro será o login do cliente. Não deve haver o mesmo e-mail cadastrado para mais de um cliente.</p> <p>O cliente deverá informar uma senha para o acesso e essa deverá passar por critérios de validação para identificar se é segura.</p> <p>Deverá haver verificação evitar duplicidade de cadastro.</p>	
Prioridade: Alta	Estimativa: 40 Horas

<b>Cliente pode atualizar sua conta de acesso.</b>	<b>ID. 08</b>
<p>Eu como cliente quero atualizar minha conta de acesso para poder atualizar meu e-mail utilizado como login, atualizar a minha senha ou alterar o meu método de acesso por aplicativos de terceiros.</p> <p>Uma vez cadastrado o usuário poderá atualizar o seu e-mail e esse novo e-mail será o seu novo login.</p> <p>Garantir se ao alterar login, senha ou desativar ou modificar o acesso não perderá se conectar com as credenciais anteriores, sendo possível apenas o acesso com o novo login.</p>	

Prioridade: Média	Estimativa: 60 Horas
-------------------	----------------------

<b>Cliente pode excluir sua conta de acesso.</b>	<b>ID. 09</b>
Como cliente quero excluir a minha conta de acesso para não manter uma conta no site que não tenho mais interesse em utilizar.	
Garantir que ao excluir a sua conta o cliente não terá mais acesso ao sistema de comércio eletrônico.	
Prioridade: Baixa	Estimativa: 20 Horas

<b>Métodos de autenticação utilizados pelo cliente para no sistema de comércio eletrônico</b>	<b>ID. 10</b>
Como cliente quero acessar o site através de login e senha, pela minha conta do Facebook ou pela minha conta do Google para que eu possa ter opções de tipo de acesso e mais segurança ao acessar o site.	
Garantir que um cliente com conta no Facebook ou no Google consiga utilizá-las como contas de acesso ao site.	
Garantir que um cliente consiga acesso ao site através de login e senha.	
Prioridade: Baixa	Estimativa: 40 Horas

<b>Cliente finalizando o pedido, meios de pagamento</b>	<b>ID. 11</b>
Como cliente quero ao finalizar meu pedido de compras ter opções de cartão de crédito ou pagamento por Paypal para ter opções de pagamento.	
Garantir que o cliente consiga finalizar sua compra utilizando o Paypal ou o cartão de crédito.	
Prioridade: Média	Estimativa: 50 Horas

Cliente acompanha o pedido		ID. 12
<p>Como cliente quero verificar o status do pedido para ter visibilidade do andamento do pedido, verificar se o pagamento foi aprovado, se o pedido foi despachado e data prevista para entrega.</p>		
<p>Todo o fluxo do pedido até a sua entrega deve ser visível para o cliente através do site em uma área de acompanhamento de pedidos.</p> <p>Para pedidos aprovados a nota fiscal deve estar emitida e disponível para o cliente logo após a confirmação de pagamento.</p> <p>Pedidos cancelados devem conter o fluxo de devolução do pedido.</p>		
Prioridade: Alta		Estimativa: 60 Horas

Cancelar pedido		ID. 13
<p>Como cliente quero poder cancelar pedido para caso do pedido não atenda as minhas expectativas ou eu tenha arrependimento da compra.</p>		
<p>Dever haver um link de fácil acesso ao cliente informando os procedimentos necessários para o cancelamento do pedido.</p> <p>O cliente deve ser capaz de acompanhar na área de status do pedido o andamento do cancelamento.</p>		
Prioridade: Alta		Estimativa: 40 Horas

Cadastro de produtos		ID. 14
<p>Como fornecedor quero cadastrar meus produtos para vendê-los no sistema de comércio eletrônico.</p>		
<p>Garantir que não ocorra duplicidade de cadastro de produtos. A quantidade produtos não é informada nesse momento apenas suas descrições.</p> <p>O cadastro de produto não implicará na disponibilização no catálogo de vendas do fornecedor no site, mas é o primeiro passo do fluxo. Cadastro de produtos pode ser feito manualmente através de uma</p>		

interface web ou através de integração de sistemas.	
Prioridade: Alta	Estimativa: 40 Horas

<b>Editar produtos cadastrados</b>		<b>ID. 15</b>
Como fornecedor quero editar os produtos que cadastrei para poder corrigir erros cometidos durante o cadastro.		
Garantir que não ocorra duplicidade de produtos.		
Prioridade: Alta	Estimativa: 40 Horas	

<b>Exclusão de produtos</b>		<b>ID. 16</b>
Como fornecedor quero excluir produtos para não manter no cadastro produtos que não fazem parte do meu catálogo de vendas.		
Garantir que o produto seja excluído e não conste mais do catálogo de produtos disponíveis para venda.		
Prioridade: Alta	Estimativa: 30 Horas	

<b>Cadastro, atualização e exclusão de fornecedores</b>		<b>ID. 17</b>
Como fornecedor quero ser capaz de cadastrar, editar e excluir meu acesso ao sistema de comércio eletrônico.		
Os dados inseridos pelos fornecedores devem ser validados.		
CPF e CNPJ devem ser validados junto a Receita Federal do Brasil.		
Não é permitido a alteração do CNPJ ou CPF de um cadastro existente.		
Prioridade: Alta	Estimativa: 40 Horas	

Cadastro de estoque, consulta e exclusão		ID. 18
Como fornecedor quero conseguir cadastrar, consultar e excluir produtos em estoque para gerenciá-lo.		
<p>Produto deve estar cadastrado para inclusão no estoque.</p> <p>A quantidade produtos no estoque variam de 0 à 999999999, sendo zero produto sem estoque. Excluir produto do estoque implica em removê-lo do cadastro de estoque, não alterar sua quantidade no estoque para 0. Todos produtos cadastrados em estoque devem possuir SKU. Os fornecedores devem ser capazes de gerenciar seu estoque através de uma interface web ou por integração de sistemas.</p>		
Prioridade: Alta		Estimativa: 40 Horas

### 3.1.2. Descrição de caso de uso crítico

Caso de uso	Cliente Pessoa Física efetuando uma compra
Sumário	O cliente utiliza o sistema web para realizar uma compra
Ator primário	Cliente
Atores secundários	Catálogo de produtos, sistema de pagamento, sistema de clientes, sistema de autenticação
Pré-condições	O sistema web deve estar disponível para acesso
<b>Fluxo Principal</b>	
<ol style="list-style-type: none"> <li>1. Cliente acessa a página de catálogo de acesso irrestrito</li> <li>2. Os produtos disponíveis para compra são carregados</li> <li>3. Cliente navega pelo catálogo e adiciona produto(s) ao carrinho de compras</li> <li>4. Cliente acessa seu carrinho de compras</li> <li>5. Cliente deve conseguir visualizar seu(s) produto(s) adicionado(s)</li> <li>6. Cliente clica em finalizar compra</li> <li>7. Uma página é exibida solicitando que informe o e-mail</li> </ol>	



8. O sistema identifica que o cliente não possui cadastro
9. Um e-mail para validação do cadastro é enviado ao cliente
10. Na página é solicitado ao cliente que informe o seu CPF
11. O CPF é validado pelo sistema
12. É solicitado ao cliente que informe o endereço de entrega
13. Na página é exibido as opções de pagamento e parcelamento
14. Únicas opções de pagamentos válidas são cartão de crédito ou Paypal
15. Cliente informa as opções de compra e clica em pagar
16. O sistema de pagamento irá se comunicar com o gateway de pagamento da opção escolhida pelo cliente
17. O sistema recebe a confirmação de pagamento do gateway
18. Uma mensagem de compra efetuada é exibida ao cliente acompanhada do número do pedido
19. Um e-mail de confirmação de compra é enviado ao cliente com número e dados do pedido.

#### Fluxo alternativo – Cliente cadastrado com acesso negado

1. Considerar os passos do 1 ao 7 do fluxo principal
2. O sistema identifica que o cliente possui cadastro
3. É exibido na página um campo para que o cliente informe a senha
4. Cliente informa a senha e clica em Enviar
5. O sistema de autenticação verifica que as credenciais informadas pelo cliente estão inválidas e nega o acesso
6. É enviada mensagem ao cliente de acesso negado
7. São sugeridas as seguintes opções ao cliente: cadastrar nova senha ou continuar com a compra fazendo um novo cadastro com outro e-mail.
8. Cliente clica em cadastrar uma nova senha
9. É informado ao cliente que uma mensagem foi enviada ao seu e-mail para prosseguir com o cadastro da nova senha

10. O sistema de enviar o e-mail para o recadastro da senha do usuário

11. O sistema de persistir os produtos do carrinho do cliente

#### Fluxo alternativo – Cliente cadastrado com acesso negado solicita novo cadastro

1. Considerar os passos de 1 ao 7 do fluxo alternativo anterior
2. Cliente clica em Continuar compra com fazendo novo cadastro
3. É exibido na página um campo solicitando que o usuário informe um novo e-mail
4. O fluxo continua a partir do passo 8 do fluxo principal

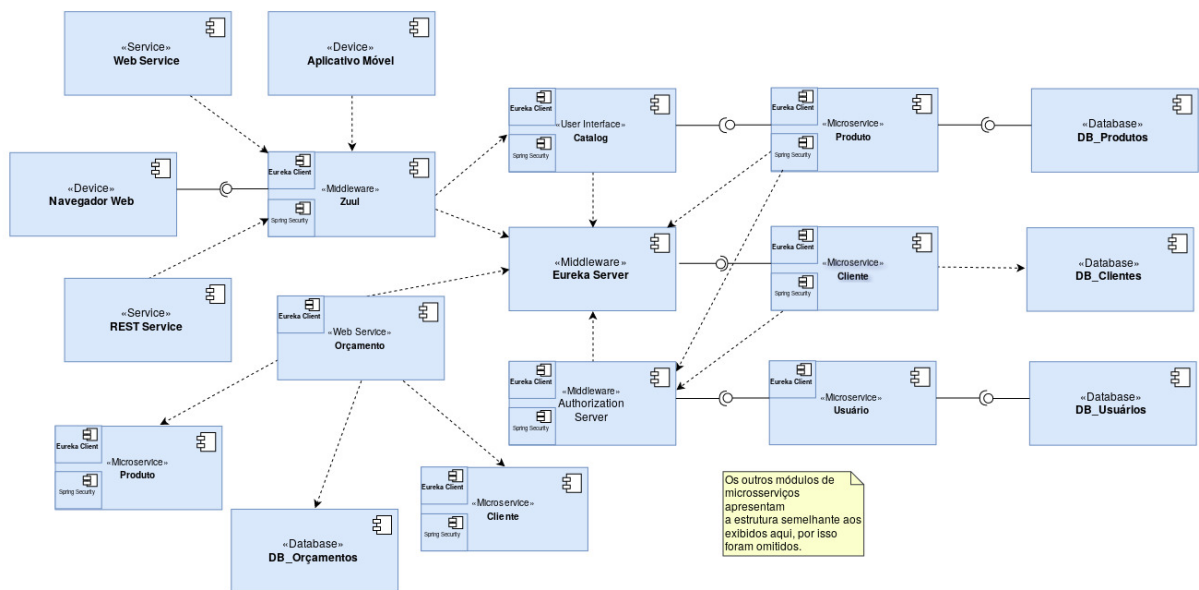
#### Fluxo alternativo – Cliente com pedido de compra negado

1. Considerar os passos de 1 ao 6 do fluxo principal
2. O sistema recebe a negação do pedido de compra do cliente pelo gateway de pagamento
3. É exibida a mensagem de negação do pedido de compra ao cliente e sugerido que escolha outra opção de pagamento.
4. O gateway de pagamento que negou a compra não deve estar mais disponível ao cliente. Exemplo: Caso o Paypal negue a compra ele não deverá estar disponível.

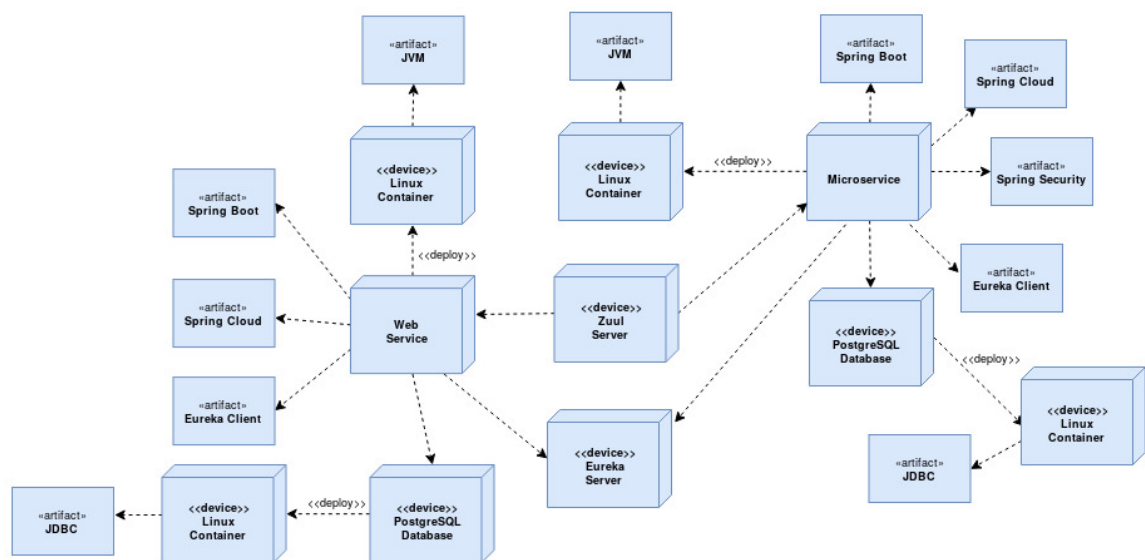
#### Fluxo de exceção

1. O gateway de pagamento ou o sistema de pagamento estão indisponíveis
2. É enviado ao cliente a mensagem que o pedido está aguardando aprovação
3. Os dados do pedido do cliente devem ser persistidos
4. Assim que o(s) sistema(s) estiver(em) disponível(eis) é enviado um e-mail ao cliente informando que o pedido não foi processado e se o cliente deseja refazer o pedido clicando em um link.
5. O link redirecionará o cliente para página de conclusão de compra aonde serão exibidos todos o(s) produto(s) selecionado(s) anteriormente para que o cliente envie o pagamento novamente.

### 3.2. Modelo de componentes



### 3.3. Modelo de implantação



## 4. Prova de conceito / protótipo arquitetural

### 4.1. Implementação e implantação

Com essa implementação se pretende apresentar conceitos de desenvolvimento de aplicações de sistemas distribuídos com o enfoque na arquitetura de microsserviços, para tal são utilizadas nessa prova de conceito parte das tecnologias frequentemente encontradas nesse tipo de implementação, principalmente quando a plataforma de desenvolvimento é Java. Com o auxílio do framework Spring Cloud foram desenvolvidos servidores utilizando a pilha de tecnologias da Netflix (Zuul, Eureka, Ribbon). Ainda foram utilizados o Spring Security, Spring Boot, Spring MVC e Spring Data, o banco de dados utilizado foi o PostgreSQL. Para implantação foi utilizada a plataforma de cloud Heroku.

## Eureka

Eureka é um sistema baseado em REST para descoberta de serviços, balanceamento de cargas (round-robin) e suporte para recuperação de falhas.

Diagrama da arquitetura em alto nível.

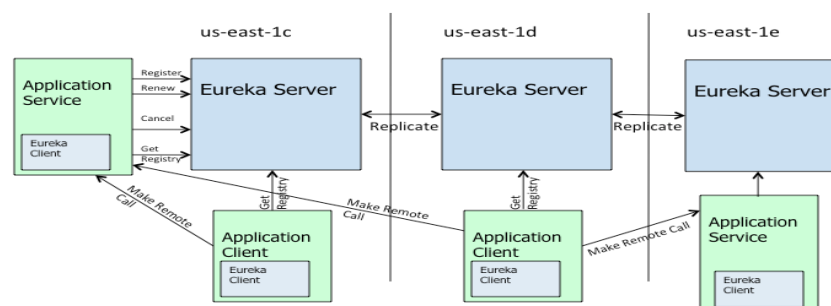


Imagem original:

[https://raw.githubusercontent.com/Netflix/eureka/master/images/eureka\\_architecture.png](https://raw.githubusercontent.com/Netflix/eureka/master/images/eureka_architecture.png)

Para essa aplicação o Eureka é utilizado com registro e descoberta de serviços.

Abaixo temos um imagem do dele configurado para essa prova de conceito.

The screenshot shows the Spring Eureka web interface. At the top, there's a navigation bar with the Spring Eureka logo and links for HOME and LAST 1000 SINCE STARTUP. Below this, the 'System Status' section displays environment details (test, default) and system metrics (Current time, Uptime, Lease expiration enabled, Renewal threshold, Renewal count). The 'DS Replicas' section shows 'localhost'. The 'Instances currently registered with Eureka' section contains a table with columns: Application, AMIs, Availability Zones, and Status.

Application	AMIs	Availability Zones	Status
AUTHORIZATION-SERVER	n/a (1)	(1)	UP (1) - f5410b02-f659-4aaf-8f8c-ccc9336b477f.prvt.dyno.rt.heroku.com:authorization-server:18680
CATALOG-UI-SERVER	n/a (1)	(1)	UP (1) - d946cae2-584c-4cfd-a4be-edabd262eaf0.prvt.dyno.rt.heroku.com:catalog-ui-server:56773
CUSTOMER-DOMAIN	n/a (1)	(1)	UP (1) - 2df72cb2-d9a1-4cae-8ed0-1a3eebc9a67d.prvt.dyno.rt.heroku.com:customer-domain:44112
USER-DOMAIN	n/a (1)	(1)	UP (1) - 444e8171-587f-4076-b7da-dc661d90c083.prvt.dyno.rt.heroku.com:user-domain:50171
ZUULSERVER	n/a (1)	(1)	UP (1) - 3cd3cb47-41f9-4941-a102-4ce60bb89659.prvt.dyno.rt.heroku.com:zuulserver:35165

## Zuul

Zuul é um API gateway utilizado para roteamento dinâmico, monitoramento e segurança.

Nessa aplicação ele é usado como porta de entrada para todas as requisições externas mapeando essas para as aplicações de destino. Além disso ele também é utilizado como servidor de autenticação dos usuários da aplicação.

### Os seguintes requisitos não funcionais serão avaliados:

#### Segurança

A escolha desse requisito funcional é devido a sua relevância para sistemas de comércio eletrônico.

Para atendimento desse requisito foi desenvolvido um servidor de autorização que utiliza o protocolo OAuth2. Quando é feita uma requisição REST para esse servidor informando usuário e senha um token é gerado. Somente com o token será possível acessar serviços seguros configurados para permitir acesso através dele.

Os critérios de aceite são:

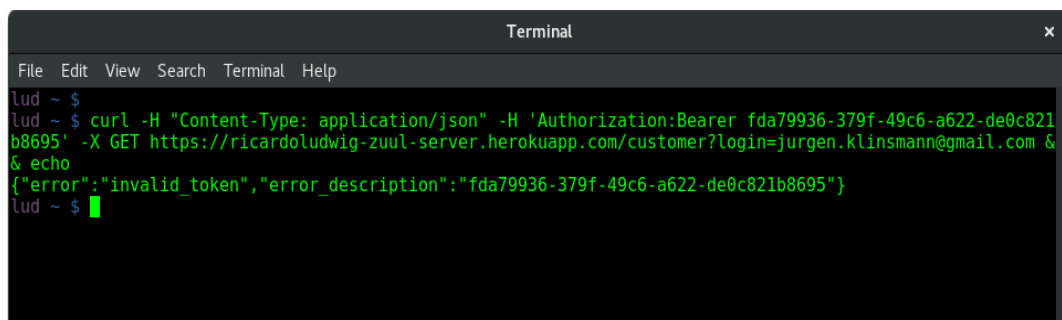
- o acesso ao serviço REST seguro com um token inválido deve ser recusado;
- com usuário e senha válidos um token deverá ser gerado quando requisitado ao servidor;

- de posse do token o acesso ao serviço REST seguro deve ser validado;

Exemplos de requisições:

- tentativa de obter recurso do servidor com token inválido;

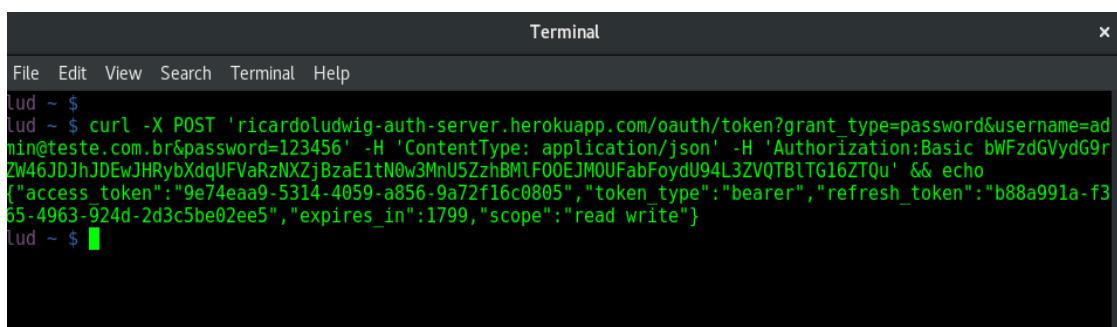
```
curl -H "Content-Type: application/json" -H 'Authorization:Bearer fda79936-379f-49c6-a622-de0c821b8695' GET https://ricardoludwig-zuul-server.herokuapp.com/customer?login=jurgen.klinsmann@gmail.com && echo
```



```
Terminal
File Edit View Search Terminal Help
lud ~ $
lud ~ $ curl -H "Content-Type: application/json" -H 'Authorization:Bearer fda79936-379f-49c6-a622-de0c821b8695' -X GET https://ricardoludwig-zuul-server.herokuapp.com/customer?login=jurgen.klinsmann@gmail.com && echo
{"error":"invalid_token","error_description":"fda79936-379f-49c6-a622-de0c821b8695"}
lud ~ $
```

- obtendo o token.

```
curl -X POST 'ricardoludwig-auth-server.herokuapp.com/oauth/token?grant_type=password&username=admin@teste.com.br&password=123456' -H 'ContentType: application/json' -H 'Authorization:Basic bWFzdGVydG9rZW46JDJhJDEwJHRybXdqUFVaRzNXZjBzaE1tN0w3MnU5ZzhBMlFOOEJMQUFabFoydU94L3ZVQTB1TG16ZTQu' && echo
```



```
Terminal
File Edit View Search Terminal Help
lud ~ $
lud ~ $ curl -X POST 'ricardoludwig-auth-server.herokuapp.com/oauth/token?grant_type=password&username=admin@teste.com.br&password=123456' -H 'ContentType: application/json' -H 'Authorization:Basic bWFzdGVydG9rZW46JDJhJDEwJHRybXdqUFVaRzNXZjBzaE1tN0w3MnU5ZzhBMlFOOEJMQUFabFoydU94L3ZVQTB1TG16ZTQu' && echo
{"access_token":"9e74eaa9-5314-4059-a856-9a72f16c0805","token_type":"bearer","refresh_token":"b88a991a-f365-4963-924d-2d3c5be02ee5","expires_in":1799,"scope":"read write"}
lud ~ $
```

Outro item de segurança adicionado na aplicação foi a configuração de um servidor de autenticação no gateway Zuul, com usuário e senha é possível o acesso páginas restritas.

Critérios de aceite:

- ao tentar acessar página com acesso restrito o usuário é redirecionado para a página de login;
- usuário e senha inválidos não conseguem acesso e mensagem de erro é exibida;
- usuário e senha válido tem acesso á página.

## Testabilidade

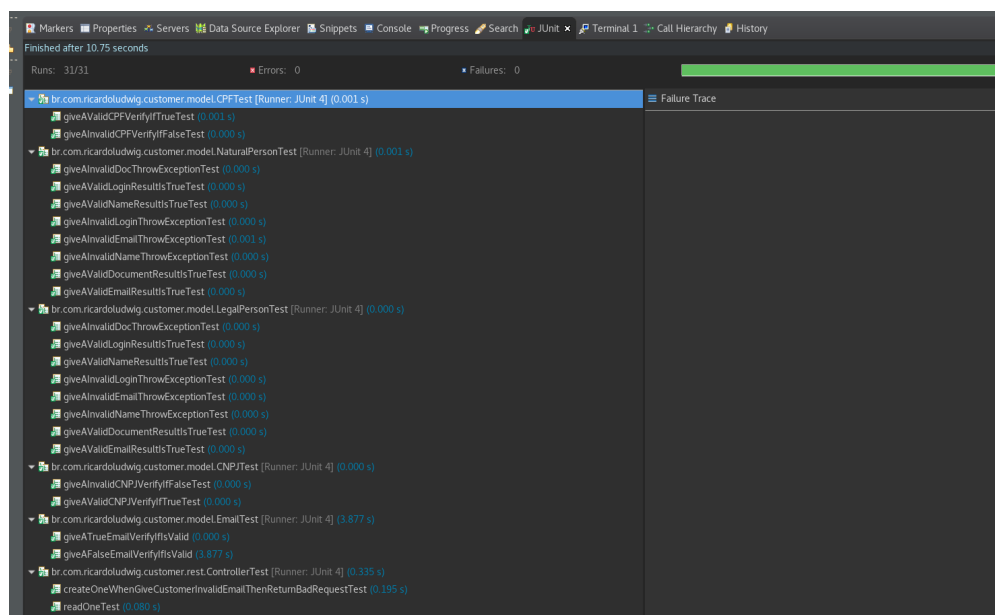
Com o objetivo de atender o requisito de qualidade de software foram feitos testes unitários. Para os testes foram utilizados os frameworks JUnit, Mockito e a suíte de testes do Spring.

Os critérios de aceite são:

- a execução dos testes deve ter 100% de sucesso;
- os testes devem ser executados pela ferramenta de build maven.

Exemplo da execução dos testes.

Através do Eclipse IDE.



Através da ferramenta de build Maven.

[illegible]

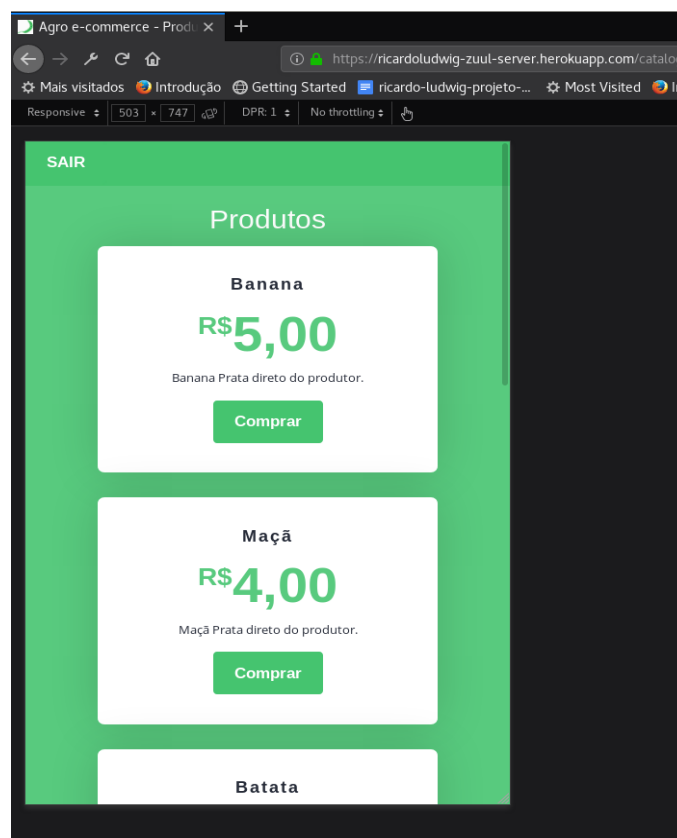
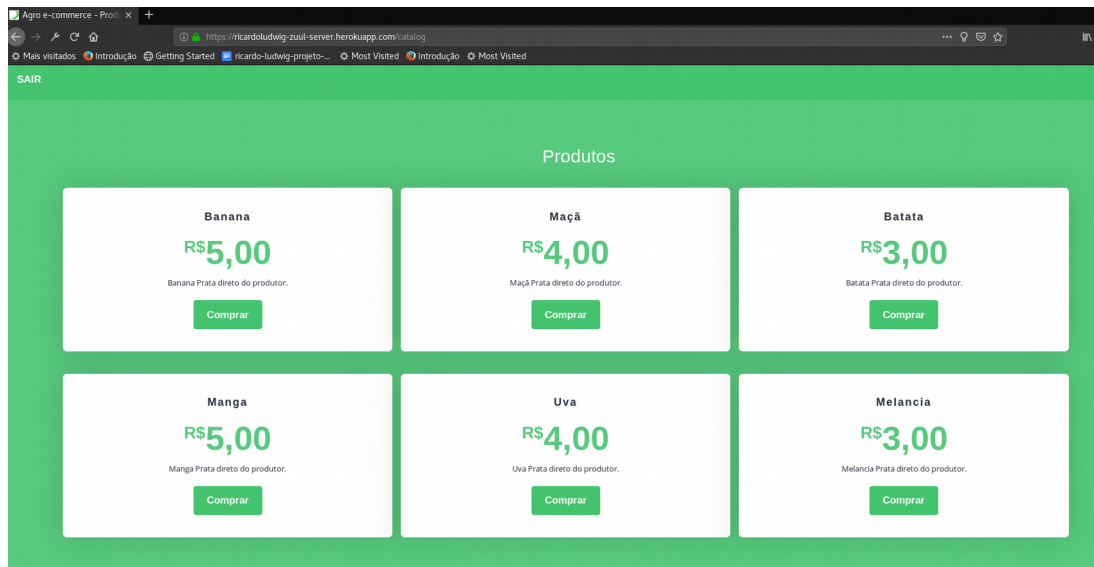
## Acessibilidade e Usabilidade

A relevância desse requisito funcional é possibilitar que qualquer página da aplicação seja responsiva, ou seja, adapte-se aos variados tipos de telas existentes hoje, sem degradação significativa de cores, perda da facilidade de navegação ou de informação.

Critérios de aceite:

- adequação da página para diferentes formatos e tamanhos de telas;
- não dever haver perda de informação nas diferentes telas.





## Desempenho

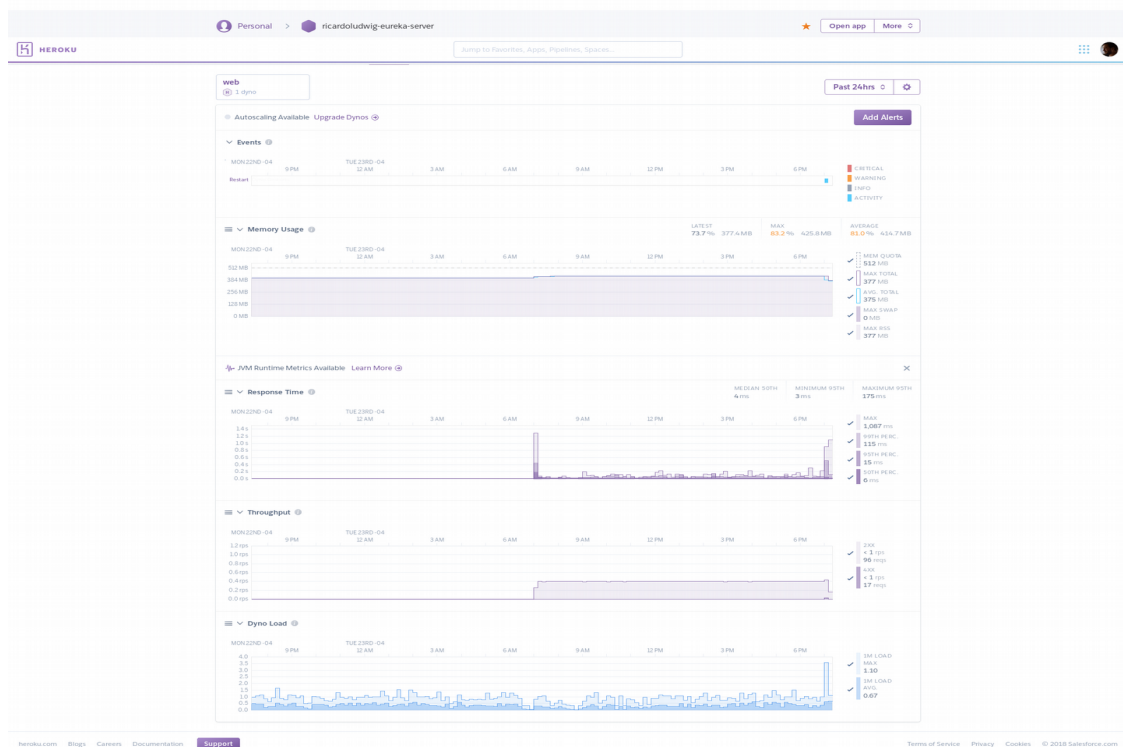
Resposta rápidas em sistema de comércio eletrônico são importantes tendo em vista que a demora na navegação ou no processo de finalização de uma compra pode fazer com que o cliente abandone sua compra. Portanto para garantir um bom tempo de resposta nas requisições é necessário desenvolver pensando em otimizações, garantir uma boa escalabilidade das aplicações e um bom throughput

de rede, respostas rápidas à falhas ou proporcionar fallbacks para requisições que estiverem apresentando alto tempo de resposta.

Critérios de aceito:

- as aplicações devem ser otimizadas para possuírem um bom tempo de resposta;
- as aplicações devem ser monitoradas para mensurar os tempos de respostas;
- devem ser facilmente escaláveis;
- devem existir estratégias para correção rápida de falhas ou lentidão.

Abaixo a imagem do sistema de monitoração utilizado no servidor de Cloud:



## 4.2. Interfaces

Abaixo uma breve demonstração de como utilizar os serviços REST da aplicação.

Obtendo um token.

Requisição:

```
curl -X POST 'ricardoludwig-auth-server.herokuapp.com/oauth/token?
grant_type=password&username=admin@teste.com.br&password=123456' -H 'ContentType:
application/json' -H 'Authorization:Basic
bWFzdGVydG9rZW46JDJhJDEwJHRybXdqUFVaRzNXZjBzaE1tN0w3MnU5ZzhBMlFOOEJMOUFab
FoydU94L3ZVQTBITG16ZTQu' && echo
```

Resposta:

```
{ "access_token": "9e5dfe16-dbb2-4f65-bebc-
72d54e9afd0f", "token_type": "bearer", "refresh_token": "b88a991a-f365-4963-924d-
2d3c5be02ee5", "expires_in": 1234, "scope": "read write" }
```

Criando um cliente, observe a necessidade do token.

Requisição:

```
curl -i -d
'{"name": "Flavia", "email": "flavia@teste.com", "login": "flavia@teste.com", "birthDate": "30/06/1994", "docu
ment": "69569178078", "typeOfDocument": "CPF", "creditCard": "9999999-9", "phones": "+49 391392939"}'
-H "Content-Type: application/json" -H 'Authorization: Bearer 9e5dfe16-dbb2-4f65-bebc-
72d54e9afd0f' POST https://ricardoludwig-customer-domain.herokuapp.com/customer -v && echo
```

Resposta:

```
{ "timestamp": "2018-10-23T23:39:03.745+0000", "message": "customer.created", "content":
{ "login": { "valid": true }, "document": { "valid": true, "number": "69569178078", "email": { "valid": true }, "phones":
["+49 391392939"], "creditCards": [""], "name": "Flavia", "_links": { "self": { "href": "https://ricardoludwig-
customer-domain.herokuapp.com/customer" } } } }
```

Criando um usuário do sistema.

Requisição:

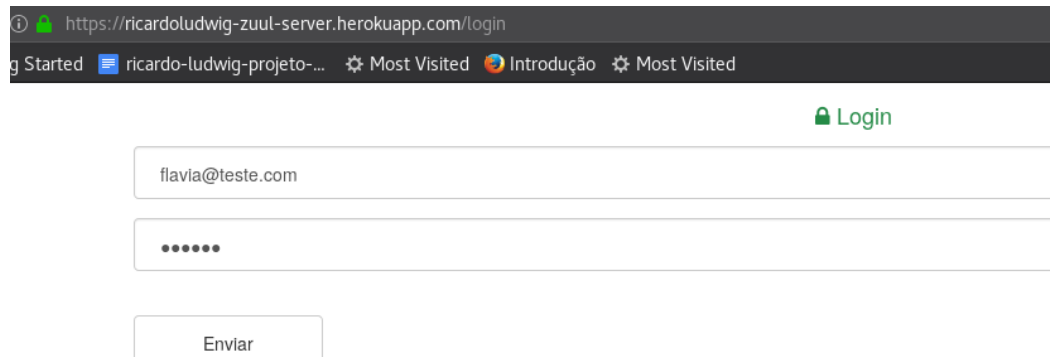
```
curl -i -d '{"username": "flavia@teste.com", "password": "123456"}' -H "Content-Type:
application/json" POST ricardoludwig-user-domain.herokuapp.com/user && echo
```

Resposta:

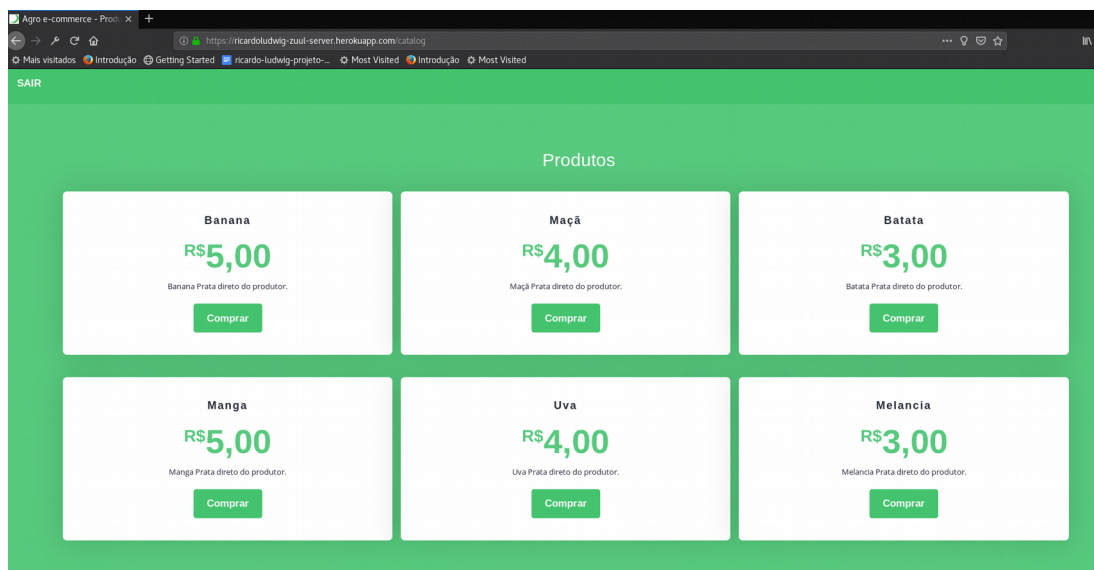
```
{ "timestamp": "2018-10-24T00:11:10.442+0000", "message": "user.created", "content":
{ "activated": true, "password": "$2a$10$mOXcKmFx3Gr30KB.MXUDfe5.IS7MY7fMvwDrwirs6avKo4Z.v
YkO", "username": "flavia@teste.com", "_links": { "self": { "href": "http://ricardoludwig-user-
domain.herokuapp.com/user" } } }
```

Utilizando o usuário criado anteriormente para autenticar no sistema web.

Acessando a url <https://ricardoludwig-zuul-server.herokuapp.com/catalog> se não estiver logado no sistema será redirecionado para a página de login.



Após autenticado o Zuul se encarrega de redirecionar a requisição para a página com exibição dos produtos.



Demonstrando a utilização do Web Service para recebimento do orçamento de cliente especiais.

Supondo que algum desses clientes tenham um sistema legado e para expô-lo faça uso de uma arquitetura SOA, com isso para integrar nosso sistema ao dele foi desenvolvido um Web Service.

Contrato WSDL:

<https://ricardoludwig-quotation-domain.herokuapp.com/ws/products.wsdl>

**Requisição:**

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:gs="http://ricardoludwig-quotation.com.br/web-service">
  <soapenv:Header/>
  <soapenv:Body>
    <gs:getProductRequest>
      <gs:product>
        <gs:name>banana</gs:name>
        <gs:amount>12</gs:amount>
      </gs:product>
      <gs:product>
        <gs:name>maca</gs:name>
        <gs:amount>2</gs:amount>
      </gs:product>
    </gs:getProductRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

**Resposta:**

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header />
  <SOAP-ENV:Body>
    <ns2:getProductResponse xmlns:ns2="http://ricardoludwig-quotation.com.br/web-service">
      <ns2:response>Orçamento recebido com sucesso</ns2:response>
    </ns2:getProductResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

**5. Avaliação da Arquitetura****5.1. Cenários**

Os cenários de qualidade descritos abaixo implementam alguns dos requisitos não funcionais apresentados anteriormente nesse documento, o método ATAM foi utilizado para definição desses.

A priorização dos cenários foi obtida através do método de Árvore de Utilidade, abaixo temos o quadro com a priorização:

Atributo de qualidade	Cenário	Importância	Complexidade
Segurança	Serviço REST com acesso restrito	Alta	Média
Segurança	Acesso restrito ao conteúdo web	Alta	Média
Desempenho	Explorar o sistema o estressando por múltiplas requisições	Alta	Média
Acessibilidade e Usabilidade	Conteúdo web responsivo	Alta	Baixa

### **Cenário: Serviço REST com acesso restrito**

Requisições para serviços REST protegidos devem obrigatoriamente conter o token obtido através das credenciais do requisitante junto ao servidor de Autorização responsável pelo controle de acesso dos recursos protegidos.

As respostas possíveis para requisição são token inválido, seja por de fato ser inválido ou haver expirado e ainda a outra resposta possível é o recebimento do recurso solicitado após a autorização do acesso.

### **Cenário: Acesso restrito ao conteúdo web**

Acesso ao conteúdo web restrito deve ser controlado através da autenticação de usuário e senha, uma vez que o usuário tenha obtido acesso a sessão da sua conexão deve permanecer ativa até que o usuário efetua a desconexão (logout) ou sua sessão expire.

As únicas respostas possíveis são a liberação do acesso ou a sua negação essa devidamente informada através de mensagem exibida ao usuário.

### **Cenário: Explorar o sistema o estressando por múltiplas requisições**

Como um cenário exploratório para expor os limites do sistema devem ser executadas respectivamente 100, 300, 700 requisições ao serviço e monitorado a sua degradação.

### **Cenário: Conteúdo web responsivo**

No acesso ao conteúdo web do sistema as páginas devem se adequar aos diferentes tamanhos e tipos de telas para os variados tipos de dispositivos existentes

e navegadores é importante que o redimensionamento do conteúdo não provoque a perda de informações relevantes para o usuário garantindo dessa forma a acessibilidade.

## 5.2. Avaliação

Dando continuidade a utilização do método ATAM serão validados os cenários descritos acima para principalmente serem identificados os principais tradeoffs da solução arquitetural assim como os riscos e os pontos de sensibilidade.

### Cenário: Serviço REST com acesso restrito

<b>Atributo de qualidade</b>	Segurança
<b>Requisito de qualidade</b>	Os serviços expondo dados críticos para o negócio ou de clientes devem possuir mecanismo de segurança.
<b>Preocupação</b>	Não expor informações críticas para acessos não autorizados.
<b>Ambiente</b>	Em estado normal de operação.
<b>Estímulo</b>	Requisição para um serviço protegido.
<b>Mecanismo</b>	Implementar a autorização via token utilizando OAuth2.
<b>Medida de resposta</b>	Ao tentar acesso com token inválido ou expirado o acesso ao recurso deve ser negado caso contrário o é acesso autorizado.
<b>Considerações sobre a arquitetura</b>	
<b>Riscos</b>	Escalonar o Servidor de Autorização e sua base de dados pode implicar no aumento de complexidade da arquitetura e no aumento de pontos de falhas.
<b>Pontos de Sensibilidade</b>	Garantir a disponibilidade do Servidor de Autorização.
<b>Tradeoffs</b>	<p>A característica distribuída dos microsserviços levou ao desenho do sistema no qual o componente de autorização é isolado em outro servidor por esse fato a indisponibilidade desse provocaria a indisponibilidade de todos os serviços que dependem dele.</p> <p>O fato de serem necessárias duas requisições para obter um recurso pode implicar no desempenho do serviço.</p>

**Evidências:**

Esse cenário está evidenciado em 4.1. Implementação e implantação, item Segurança, nesse documento.

**Cenário: Acesso restrito ao conteúdo web**

<b>Atributo de qualidade</b>	Segurança
<b>Requisito de qualidade</b>	Garantir que somente acessos autenticados tenham acesso aos sistemas web da aplicação.
<b>Preocupação</b>	Impedir o acesso não autenticado a sistemas da aplicação com acesso restrito.
<b>Ambiente</b>	Em estado normal de operação.
<b>Estímulo</b>	Cliente tenta acessar página web restrita sem estar autenticado.
<b>Mecanismo</b>	Implementar solução de segurança para identificar usuários que possuem acesso ou não ao sistema.
<b>Medida de resposta</b>	Ao tentar o acesso sem estar autenticado o cliente deve ser direcionado para a página de autenticação.
<b>Considerações sobre a arquitetura</b>	
<b>Riscos</b>	O ciclo de vida da sessão e seu compartilhamento entre os microserviços não possuem a robustez necessária.
<b>Pontos de Sensibilidade</b>	A correta codificação do mecanismo de segurança para evitar o vazamento de dados sensíveis.
<b>Tradeoffs</b>	Aumento na complexidade de desenvolvimento da aplicação, necessidade de auditoria e testes de segurança implicam maior custo em pessoa-ano.

**Evidências:**

Esse cenário está evidenciado em 4.2. Interfaces - Criando um usuário, nesse documento.

**Cenário: Explorar o sistema o estressando por múltiplas requisições**

<b>Atributo de qualidade</b>	Desempenho
------------------------------	------------



<b>Requisito de qualidade</b>	Expor os limites do sistema em relação ao seu desempenho quando exposto a quantidades altas de requisições.
<b>Preocupação</b>	Entender os limites do sistema para prevenir futuras sobrecargas.
<b>Ambiente</b>	Em estado normal de operação.
<b>Estímulo</b>	Envio de um grande número de requisições.
<b>Mecanismo</b>	<p>O código da aplicação e consulta a recursos externos devem ser automatizados. Exemplos:</p> <ul style="list-style-type: none"> <li>– consultas aos bancos de dados automatizadas;</li> <li>– a troca de informações, quando em grande volume, entre os microserviços em formato binário em vez de texto;</li> <li>– aumento do throughput de rede;</li> <li>– escalonar a aplicação.</li> </ul>
<b>Medida de resposta</b>	É esperado que apesar do aumento de requisições não existe uma degradação muito alta no tempo de resposta ou indisponibilidade do serviço.
<b>Considerações sobre a arquitetura</b>	
<b>Riscos</b>	Um protocolo de comunicação binário, como o BJson, para serviços que troquem grande volume de mensagens em formato texto pode impulsionar a velocidade de comunicação uma vez que a quantidade de dados trafegados diminuirá.
<b>Pontos de Sensibilidade</b>	O throughput da rede e capacidade de processamento dos serviços.
<b>Tradeoffs</b>	A necessidade de aumentar o desempenho da aplicação pode privilegiar a codificações obscuras em detrimento códigos mais claros e melhor legibilidade. Isso pode prejudicar a manutenibilidade e testabilidade da aplicação.

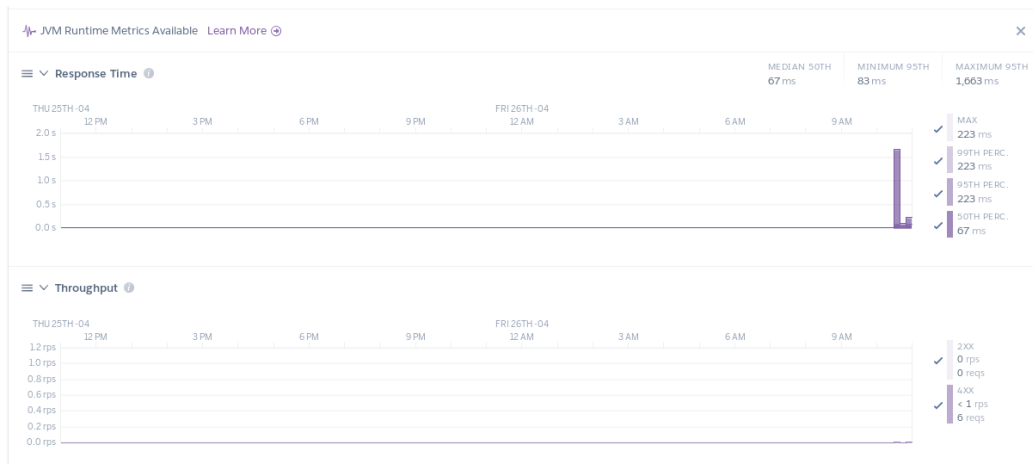
### Evidências:

O serviço que sob teste é a busca de cliente cadastrado da aplicação Customer Domain.

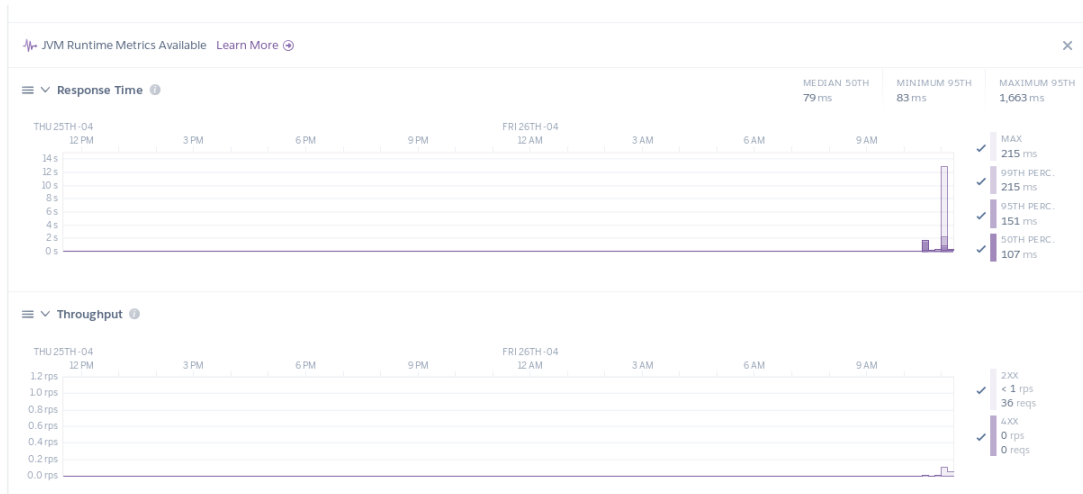
### Requisição

```
curl -H "Content-Type: application/json" -H 'Authorization:Bearer 29c9460d-3d68-4fd7-8176-34cf297f9e23' -X GET https://ricardoludwig-customer-domain.herokuapp.com/customer?login=jurgen.klinsmann@gmail.com
```

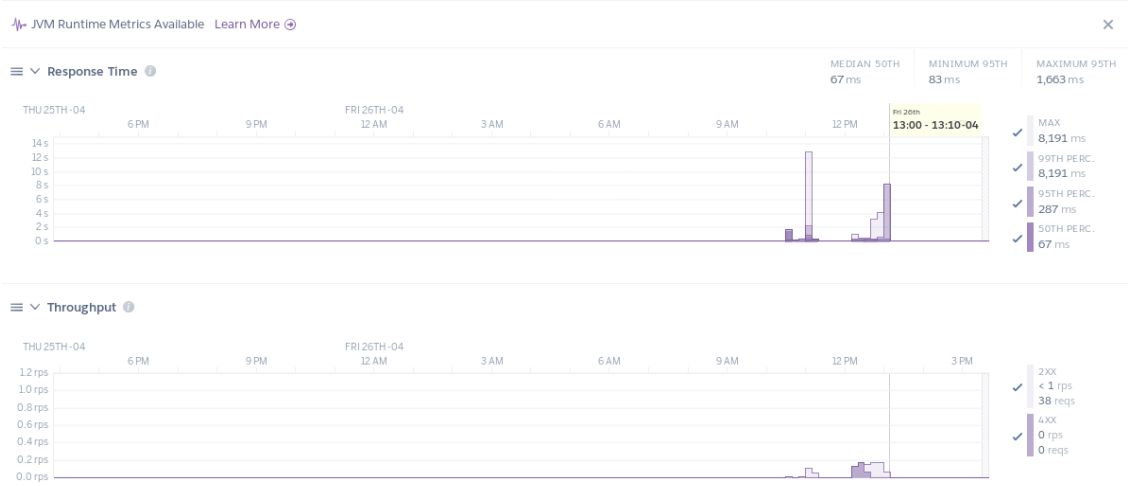
Abaixo temos as métricas de tempo de resposta e throughput do serviço em seu estado de operação normal.



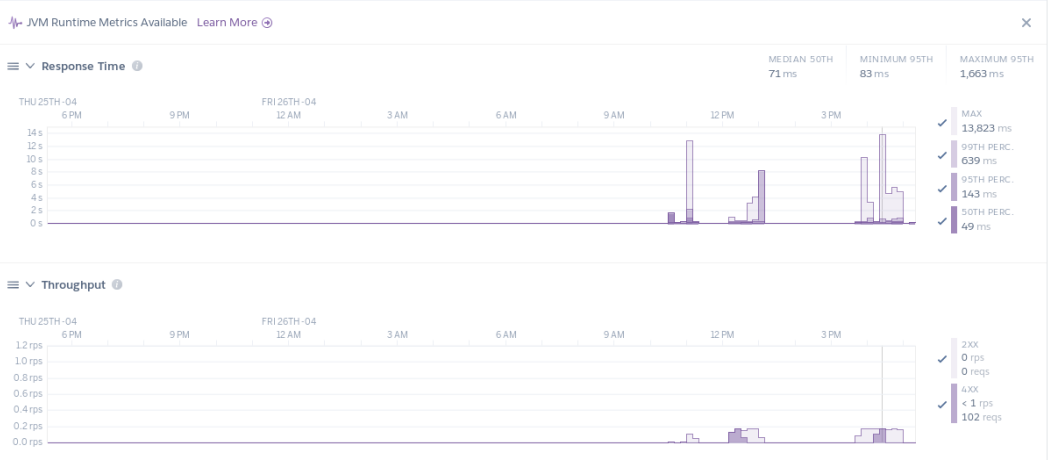
Métricas após o processamento de 100 requisições.



Métricas após o processamento de 300 requisições.



Métricas após o processamento de 700 requisições.



Baseado nas métricas obtidas observamos que apesar picos no tempo de resposta de quase 14 segundos aproximadamente 95% das requisições ficaram em torno de 2 segundos. O serviço está com um tempo satisfatório mesmo em condições estressantes.

Cenário: Conteúdo web responsivo

Atributo de qualidade	Acessibilidade e Usabilidade
Requisito de qualidade	O sistema web deve ser capaz de adequar a exibição de seu conteúdo aos diferentes tipos de dispositivos, tamanhos de telas e navegadores.
Preocupação	Garantir a correta visualização do conteúdo web sem a perda de informação ao usuário nos diferentes tipos de dispositivos, navegadores e tamanho de telas.

<b>Ambiente</b>	Em estado normal de operação.
<b>Estímulo</b>	Usuário acessando uma página web do sistema de um dispositivo móvel.
<b>Mecanismo</b>	Desenvolver as páginas web de forma responsiva.
<b>Medida de resposta</b>	O conteúdo exposto deve se adequar ao tamanho da tela sem que exita a perda de informações.
<b>Considerações sobre a arquitetura</b>	
<b>Riscos</b>	A impossibilidade de testes serem feitos em todos os tipos e dispositivos com seus variados tamanhos de tela pode ocasionar em alguns cenários no qual algum dispositivo não atenda ao atributo de qualidade.
<b>Pontos de Sensibilidade</b>	Frameworks podem facilitar a tarefa de desenvolvimento de páginas web responsivas no entanto o mau uso pode acarretar em um aumento significativo na quantidade dados trafegados principalmente em redes móveis aonde há restrições de consumo de dados e throughput de rede.
<b>Tradeoffs</b>	Garantir que a página web não seja pesada, não precise carregar um grande volume de dados ou necessite de uma quantidade considerável processamento para ser renderizada. Em condições aonde exista baixa capacidade de processamento do dispositivo ou limitações de throughput de rede impactará no desempenho e na usabilidade.

### **Evidências:**

Esse cenário está evidenciado em 4.1. Implementação e implantação, item Acessibilidade e Usabilidade, nesse documento.

## REFERÊNCIAS

- NEWMAN, Sam. **Building Microservices**. Sebastopol: O'Reilly, 2015.
- ERIC, Evans. **Domain-Driven Design**. Westford: Pearson Education, 2011.
- LARMAN, Graig. **Utilizando UML e Padrões: Uma introdução a análise e ao projeto orientados a objetos. 3º Edição**. Porto Alegre: Bookman, 2011.
- SOMMERVILLE, Ian. **Engenharia de Software. 9º edição**. São Paulo: Pearson, 2011.
- BECK, Kent. **TDD Desenvolvimento Guiado Por Testes**. Porto Alegre: Bookman, 2010.
- SPRING, **Spring Cloud Netflix**. <<http://cloud.spring.io/spring-cloud-netflix>>, 2018
- HEROKU, **Metrics Response Time**.<<https://devcenter.heroku.com/articles/metrics#response-time>>, 2018
- NETFLIX, **Netflix OSS**. <<https://netflix.github.io/>>, 2018
- RICHARDSON, **Chris**. <<http://microservices.io>>, 2018
- FOWLER, **Martin**. <<https://martinfowler.com/articles/microservices.html>>, 2018
- LELIS, **Gabriel**. <<http://revistagloborural.globo.com/Tecnologia-no-Campo/noticia/2016/01/e-commerce-de-produtos-agropecuarios-e-lancado-no-brasil.html>>, 2018
- MONTEIRO, **Ana Victoria Martins**. <<http://www.iea.sp.gov.br/out/LerTexto.php?codTexto=2360>>, 2018
- LIMA, **Rafael**. <<http://aprendendogestao.com.br/o-que-e-sku-stock-keeping-unit/>>, 2018
- Goodin, **Dan**. <<https://arstechnica.com/information-technology/2015/09/once-seen-as-bulletproof-11-million-ashley-madison-passwords-already-cracked/>>, 2018

## APÊNDICES

### Vídeo do sistema em execução

<https://www.youtube.com/watch?v=D9cfABUk-wo>

### Código Fonte

[https://github.com/ricardoludwig/agro\\_ecommerce](https://github.com/ricardoludwig/agro_ecommerce)

### Estrutura simplificada do projeto

```
|— middleware
|   |— authorization-server
|   |— eureka-server
|   └─ zuul-server
|— resources
|   |— customer-domain
|   |— quotation
|   └─ user-domain
└─ ui
    └─ catalog
```

### Links do local aonde as aplicações estão implantadas

#### Eureka

<https://ricardoludwig-eureka-server.herokuapp.com>

#### Zuul

<https://ricardoludwig-zuul-server.herokuapp.com>

#### Authorization Server

<https://ricardoludwig-auth-server.herokuapp.com>

Customer Domain

<https://ricardoludwig-customer-domain.herokuapp.com>

User Domain

<https://ricardoludwig-user-domain.herokuapp.com>

Quotation

<https://ricardoludwig-quotation-domain.herokuapp.com/ws/products.wsdl>

Catalog

<https://ricardoludwig-zuul-server.herokuapp.com/catalog>