

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

**Alisson Rodrigues de Sousa**

**SISTEMA DE VENDAS DROPSHIPPING - ROÇA VIRTUAL**

Belo Horizonte

2018

**Alisson Rodrigues de Sousa**

**SISTEMA DE VENDAS DROPSHIPPING - ROÇA VIRTUAL**

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

Belo Horizonte

2018

*Dedico este trabalho a meus pais, minha esposa Livia e meus  
filhos Miguel e Samuel.*

## RESUMO

Este projeto aborda uma solução de vendas na modalidade dropshipping, utilizando dispositivos móveis ou desktop. Atualmente os pequenos produtores rurais e integrantes da agricultura familiar, possuem grande dificuldade de escoar sua produção e concorrer com os grandes produtores, o que acaba obrigando as famílias e pequenos produtores a abandonar o campo e migrar para os grandes centros, o que causa um enorme problema social. Outro ponto importante é a crescente demanda por produtos orgânicos. A cada dia as pessoas estão mais atentas às questões de saúde e aceitam até pagar um pouco mais por produtos saudáveis, porém estes ainda são difíceis de encontrar. O sistema fruto deste projeto visa facilitar a venda dos produtores rurais aos consumidores finais ou revendedores por meio de uma plataforma web. Os fornecedores e produtos estarão cadastrados no sistema e o comprador realizará a compra neste, sendo os pedidos encaminhados aos fornecedores para que os mesmos fiquem responsáveis pela logística de entrega. O projeto também prevê uma camada de integração com sistemas que eventualmente os fornecedores possuam e está estruturado da seguinte forma: definição dos requisitos, modelagem e projeto arquitetural, prova de conceito funcional e avaliação da arquitetura. Espera-se ao final do projeto que os principais requisitos possam ser validados com a prova de conceito proveniente das modelagens/definições deste documento.

**Palavras-chave:** arquitetura de software, projeto de software, requisitos arquiteturais, serviços web, web responsiva, dropshipping, integração de software, agricultura familiar.

## SUMÁRIO

1. Objetivos do trabalho .....	6
<b>2. Descrição geral da solução .....</b>	<b>7</b>
2.1. Apresentação do problema .....	7
2.2. Descrição geral do software .....	7
<b>3. Definição conceitual da solução .....</b>	<b>8</b>
3.1. Requisitos Funcionais .....	8
3.2 Requisitos Não-Funcionais.....	11
3.3. Restrições Arquiteturais .....	15
3.4. Mecanismos Arquiteturais .....	15
<b>4. Modelagem e projeto arquitetural .....</b>	<b>16</b>
4.1. Modelo de casos de uso.....	16
4.2. Descrição resumida dos casos de uso .....	21
4.3. Modelo de componentes .....	26
4.4. Modelo de implantação .....	29
4.5. Modelo de dados .....	30
<b>5. Prova de conceito / protótipo arquitetural .....</b>	<b>31</b>
5.1. Implementação e implantação.....	31
5.2. Interfaces/ APIs .....	33
<b>6. Avaliação da Arquitetura .....</b>	<b>35</b>
6.1. Análise das abordagens arquiteturais .....	35
6.2. Identificação dos atributos de qualidade .....	36
6.3. Cenários .....	36
6.4. Avaliação.....	37
6.5. Resultado .....	44
7. Conclusão .....	46
<b>APÊNDICES .....</b>	<b>47</b>

## 1. Objetivos do trabalho

O objetivo geral deste projeto é apresentar uma proposta de arquitetura para um sistema de vendas na modalidade dropshipping para pequenos produtores rurais e integrantes da agricultura familiar. O projeto visa facilitar o acesso dos consumidores aos produtos agrícolas e facilitar o escoamento da produção destes produtores.

Os objetivos específicos são:

1. Criar um módulo de controle de vendas, onde os clientes possam realizar as compras pelo sistema web tanto de desktops quanto de dispositivos móveis e por onde vendedores externos que usam smartphones ou laptops também possam efetuar vendas. Acontecendo uma venda, o fornecedor deve ser comunicado para fazer a entrega.
2. Criar um módulo de geração de informações gerenciais, que deve gerar diariamente informações sobre as vendas, sobre os eventos de entrega dos produtos e sobre as avaliações dos clientes e repassar para o sistema de monitoramento - aplicação de Business Intelligence.
3. Criar um módulo de controle de devolução, reenvio e extravio integrado com o sistema do fornecedor, onde problemas nas entregas devem ser registrados e o produto deve ser encaminhado de acordo com a situação.
4. Criar um módulo de controle da logística de entrega, onde devem ser controlados todos os eventos relacionados aos processos de entrega do produto. O fornecedor despacha o produto e informa a loja. A partir deste momento a loja passa a acompanhar todo o processo.
5. Criar um módulo de controle das propagandas e promoções, onde deve ser feita a gestão das propagandas e das promoções para divulgação para os clientes.
6. Permitir a geração de relatórios de acompanhamento, como relatórios de vendas, produtos mais vendidos, rentabilidade e custos.
7. Permitir o controle das solicitações e dúvidas dos envolvidos com o sistema.

## **2. Descrição geral da solução**

### **2.1. Apresentação do problema**

Atualmente os pequenos produtores rurais repassam seus produtos a preços muito abaixo do valor de mercado devido à dificuldade de vender sua produção e isso acaba desestimulando esta categoria que responde pela produção de cerca de 70% dos alimentos consumidos no país. Facilitando o acesso dos compradores aos produtos desses produtores, os mesmos poderiam ter uma margem maior de lucro mesmo que o preço final seja mantido.

Hoje em dia também é crescente a busca por produtos saudáveis, entre os quais se destacam os produtos agrícolas orgânicos, que são alimentos produzidos de forma sustentável, respeitando e não agredindo o meio ambiente, sem a utilização de fertilizantes químicos e agrotóxicos. Muitas pessoas aceitariam pagar um pouco mais para adquirirem estes produtos, mas mesmo assim não são fáceis de encontrar.

Realizando vendas de porta em porta, mesmo que para varejistas ou feirantes, fica difícil pensar em aumentar a produção, pois o mercado acaba ficando restrito e o produtor despende muito tempo nesta tarefa, tempo este que poderia ser mais bem aplicado na produção e melhoria dos processos.

### **2.2. Descrição geral do software**

A aplicação se aplica em função de uma melhor forma de conexão entre produtores rurais e seus clientes finais, podendo ser pessoas físicas, que almejam uma alimentação mais saudável ou varejistas que precisam encontrar fornecedores de qualidade e também um melhor controle sobre suas compras.

Os produtores de posse dos pedidos podem programar melhor a colheita e evitar desperdícios. Também será facilitado o controle de expansão da produção e de quais produtos podem ter sua produção aumentada ou diminuída de acordo com a demanda.

A distribuição de produtos orgânicos será facilitada, pois os produtores conhecerão melhor a distribuição geográfica e perfis dos clientes através dos relatórios gerenciais e através destes poderão tomar melhores decisões sobre o negócio e reduzir custos planejando melhor as rotas de entregas.

Esse sistema será acessado por vários usuários: Administrador, Fornecedor, Cliente e Vendedor.

- Administrador: responsável pela administração da loja virtual, controle de acessos, fornecedores, produtos que poderão ser vendidos, clientes, promoções e propagandas, qualidade e pontualidade dos fornecedores e controle das solicitações e dúvidas dos envolvidos com o sistema.
- Fornecedor: responsável pelas atualizações de seus dados cadastrais, dos dados de seus produtos vendidos na loja virtual e pela informação da situação dos pedidos. Responsável por toda a logística de entrega dos produtos após a comunicação da venda pela loja.
- Cliente: realiza compras na loja e acompanha o andamento do pedido até a entrega dos produtos. Avalia os fornecedores, ajudando na qualificação dos mesmos.
- Vendedor: realiza vendas através de smartphones ou laptops.

### **3. Definição conceitual da solução**

#### **3.1. Requisitos Funcionais**

##### Módulo Segurança

- Cadastro de usuários

O sistema deve permitir ao administrador realizar o cadastro de usuários para o controle do sistema.

O sistema deve permitir que clientes e fornecedores se cadastrem através do site.



O sistema deve possibilitar ao administrador aprovar ou não um fornecedor.

O sistema deve permitir ao administrador realizar o controle de permissões de acesso aos módulos e funções do sistema.

- Recuperar Senha

O sistema deve possibilitar a recuperação e troca de senha pelos usuários.

- Autenticação

O sistema deve gerenciar a autenticação e autorização de acesso dos usuários aos módulos e funções do sistema.

#### Módulo de Cadastros

- Cadastro de Fornecedores

O sistema deve permitir o cadastro e gerenciamento de fornecedores.

O sistema deve permitir a aprovação ou não dos fornecedores pelos administradores.

- Cadastro de Produtos

O sistema deve permitir o cadastro e gerenciamento de produtos.

O sistema deve permitir a aprovação ou não dos produtos dos fornecedores pelos administradores.

- Cadastro de clientes

O sistema deve permitir o gerenciamento de clientes e suas compras pelos administradores.

#### Módulo de Controle de Vendas

- Vendas

O sistema deve permitir a realização de vendas diretas aos clientes.

O sistema deve permitir a realização de vendas por vendedores externos.

O sistema deve notificar o fornecedor das vendas para que o mesmo proceda com a logística de entregas.

- Pagamento

O sistema deve possibilitar a realização do pagamento das compras pelos meios comuns de mercado: boleto, cartão de crédito e débito.

#### Módulo de Controle de Logística de Entrega

- Atualização do Status da Entrega

O sistema deve permitir a atualização dos status das entregas pelos fornecedores.

O sistema deve notificar o cliente das atualizações dos status das entregas através do meio escolhido por ele: e-mail, sms ou ambos.

- Avaliação

O sistema deve possibilitar ao cliente avaliar os fornecedores para ajudar na qualificação dos mesmos.

#### Módulo de Controle das Propagandas e Promoções

- Propagandas

O sistema deve permitir ao administrador o gerenciamento de campanhas de marketing.

O sistema deve possibilitar o envio de propagandas aos clientes por e-mail e sms.

- Promoções

O sistema deve permitir ao administrador o gerenciamento de promoções.

O sistema deve possibilitar o envio de promoções aos clientes por e-mail e sms.

#### Módulo de Relatórios de Acompanhamento

- Relatórios

O sistema deve permitir a consulta e emissão de relatórios de vendas, de produtos e financeiro.

O sistema deve possibilitar a exportação dos relatórios em pdf e excel.

### Módulo de Geração de Informações Gerenciais

O sistema deve diariamente consolidar as informações gerenciais e enviá-las ao sistema de monitoramento (Business Intelligence).

### Módulo de Controle de Devolução, Reenvio e Extravio

- Devolução

O sistema deve permitir ao cliente solicitar a devolução do produto de acordo com as regulamentações do comércio eletrônico.

O sistema deve possibilitar ao administrador solicitar ao fornecedor o recolhimento do produto.

- Extravio

O sistema deve permitir ao cliente informar o extravio (não recebimento) do produto.

O sistema deve possibilitar ao administrador solicitar ao fornecedor o reenvio do produto.

### Módulo de Atendimento ao Consumidor (SAC)

- Dúvidas

O sistema deve permitir ao cliente consultar dúvidas frequentes.

O sistema deve permitir ao cliente registrar uma dúvida nova.

O sistema deve possibilitar ao administrador responder a dúvida do cliente e a resposta deve ser enviada por e-mail para o mesmo.

### Módulo de Integração com Sistemas de Fornecedores

- Atualização de Dados de Produtos

O sistema deve periodicamente solicitar dados atualizados dos produtos de cada fornecedor e atualizar a base de dados do sistema da loja virtual.

## 3.2 Requisitos Não-Funcionais

A seguir são apresentados os requisitos não funcionais do sistema:

- Usabilidade - O sistema deve prover boa usabilidade

<b>Estímulo</b>	Usuário efetuando uma compra.
<b>Fonte do Estímulo</b>	Usuário acessando uma funcionalidade de pesquisa de produtos.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo de Controle de Vendas.
<b>Resposta</b>	A camada de apresentação deve apresentar facilidade de navegação, simplicidade e objetividade.
<b>Medida da Resposta</b>	Usuário deve conseguir realizar a pesquisa de um produto em no máximo 3 minutos.

- Acessibilidade - O sistema deve suportar ambientes Web responsivos e ambientes móveis.

<b>Estímulo</b>	Cadastro de fornecedor.
<b>Fonte do Estímulo</b>	Usuário acessando o sistema de um celular.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo de Cadastros.
<b>Resposta</b>	A camada de apresentação deve se adaptar às resoluções e tamanho das telas, mudando os componentes de posição de forma a ficar melhor a navegação do usuário.
<b>Medida da Resposta</b>	Identidade visual semelhante em todas as resoluções, com objetos redimensionados de acordo com a resolução e tamanho.

- Desempenho - O sistema deve ser rápido.

<b>Estímulo</b>	Acessando a tela de pesquisa de produtos.
<b>Fonte do Estímulo</b>	Pesquisa de produto.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo de Controle de Vendas
<b>Resposta</b>	O sistema deve responder com os dados solicitados.
<b>Medida da Resposta</b>	O sistema deve responder em menos de 10 segundos.

- Manutenibilidade - O sistema deve apresentar manutenção facilitada.

<b>Estímulo</b>	Um componente de negócio responsável pela atualização de dados dos produtos de um fornecedor precisa ser atualizado.
<b>Fonte do Estímulo</b>	Um novo fornecedor possui um sistema legado que precisa ser integrado ao sistema de loja virtual.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo de Integração com fornecedores.
<b>Resposta</b>	O sistema deve se integrar com o sistema do novo fornecedor.
<b>Medida da Resposta</b>	Os dados de produtos do novo fornecedor devem ser sincronizados em até 3 minutos.

- Testabilidade - O sistema deve ser simples para testar.

<b>Estímulo</b>	Execução de testes no sistema.
<b>Fonte do Estímulo</b>	Analista desenvolvedor.
<b>Ambiente</b>	Ambiente de desenvolvimento.
<b>Artefato</b>	Módulo de Controle de Vendas.
<b>Resposta</b>	O sistema deve testar todas as funcionalidades como listagem de produtos, inclusão e retirada de produtos no carrinho de compras e pagamento.
<b>Medida da Resposta</b>	O sistema deve possibilitar efetuar os testes com scripts automatizados executando com apenas um comando.

- Interoperabilidade - O sistema deve se comunicar com os sistemas dos agentes. Alguns desses sistemas são antigos e desenvolvidos com tecnologia COBOL/CICS.

<b>Estímulo</b>	Teste de conexão.
<b>Fonte do Estímulo</b>	Sistemas de alta plataforma.
<b>Ambiente</b>	O sistema está em funcionamento com carga normal.
<b>Artefato</b>	Módulo de Integração com Sistemas de Fornecedores.

<b>Resposta</b>	O sistema do fornecedor respondeu com sucesso a solicitação.
<b>Medida da Resposta</b>	Conexão efetivada.

- Disponibilidade - O sistema deve operar em qualquer período do dia e da noite.

<b>Estímulo</b>	Desligado um dos serviços do sistema.
<b>Fonte do Estímulo</b>	Administrador do servidor de aplicações.
<b>Ambiente</b>	Diversos usuários estão utilizando o sistema.
<b>Artefato</b>	Web API.
<b>Resposta</b>	Todos os usuários logados na aplicação devem continuar utilizando o sistema sem perceber que houve uma queda em um dos serviços da aplicação.
<b>Medida da Resposta</b>	Todas as solicitações dos usuários devem ser atendidas, podendo haver um atraso de 3 segundos devido à queda de um dos nós.

- Segurança - O sistema deve apresentar altos padrões de segurança.

<b>Estímulo</b>	Acessar uma página privada pela url sem estar logado no sistema.
<b>Fonte do Estímulo</b>	Qualquer usuário.
<b>Ambiente</b>	Em funcionamento com carga normal.
<b>Artefato</b>	Módulo de Segurança, Módulo de Cadastros, Módulo de Controle de Vendas, Módulo de Controle de Logística de Entregas, Módulo de Controle das Propagandas e Promoções, Módulo de Relatórios de Acompanhamento, Módulo de Geração de Informações Gerenciais, Módulo de Controle de Devolução, Reenvio e Extravio, Módulo de Atendimento ao Consumidor (SAC).
<b>Resposta</b>	O sistema deve redirecionar o usuário para a tela de login solicitando usuário e senha.
<b>Medida da Resposta</b>	O sistema não deve permitir o acesso a páginas privadas.

### 3.3. Restrições Arquiteturais

- O sistema deve ser desenvolvido em Java;
- O sistema deve utilizar um serviço de meio de pagamentos para disponibilizar a funcionalidade de pagar pedido;
- O sistema deve abrir de forma responsiva em aparelhos menores, como celular e tablet;
- O sistema deve ser modular para facilitar a implantação;
- O sistema deve permitir a sincronização de dados com os sistemas de fornecedores periodicamente.

### 3.4. Mecanismos Arquiteturais

Mecanismos de Análise	Mecanismo de Design	Mecanismo de Implementação
Comunicação entre processos	Contêiner Web e Aplicação	Docker
Integração com outros módulos e/ou sistemas	Interfaces utilizando JSON	WebServices, WebAPI e Apache Camel
Log	Framework de Log	Log4j
Build	Geração de artefato para servidor de aplicação	Maven
Deploy	Deploy da aplicação no servidor e testes automatizados	Jenkins
Front-End	Interface de comunicação com o usuário do sistema	VueJS e Bootstrap
Versionamento	Versionamento do código-fonte da aplicação	Git
Autenticação e Autorização	Verificação das credenciais e tentativas de conexão	OAuth
Alta disponibilidade	Balanceamento de carga dos serviços	Netflix Zuul

Descoberta	Registro de serviços	Netflix Eureka
Bancos de Dados	SGBD que será utilizado pelo sistema	MySQL

## 4. Modelagem e projeto arquitetural

### 4.1. Modelo de casos de uso

Para uma melhor análise arquitetural do projeto, os casos de uso foram separados por módulos de acordo com os requisitos informados acima.

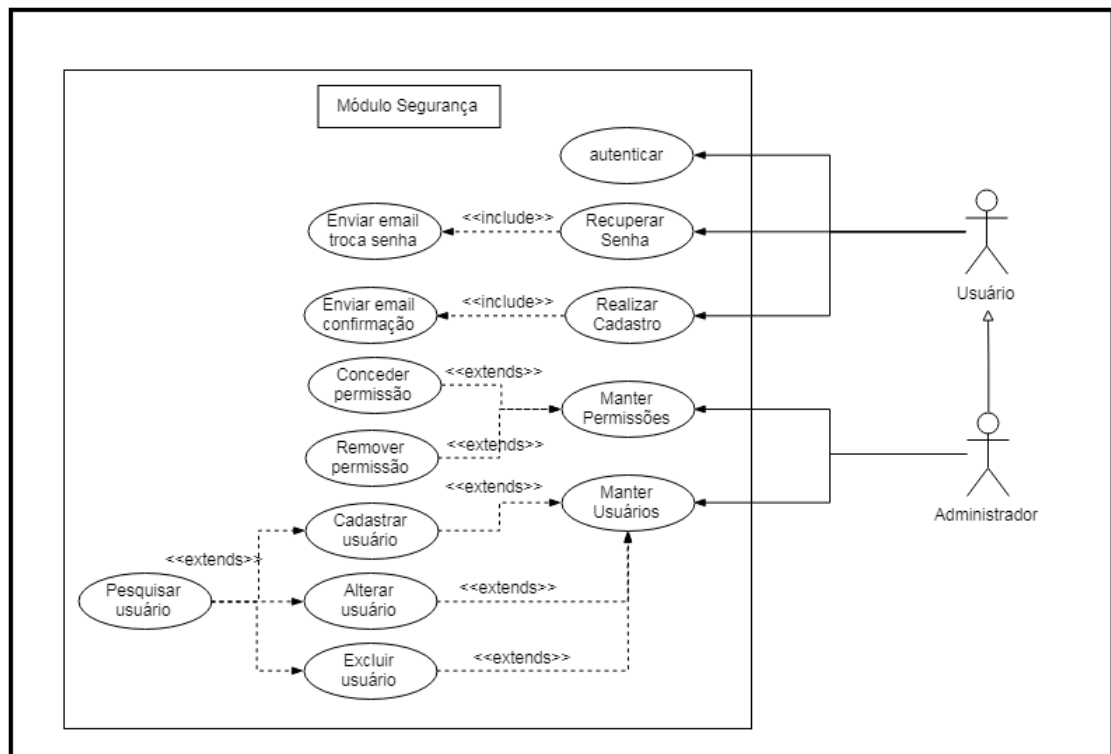


Figura 1- Diagrama de Casos de Uso: Módulo Segurança



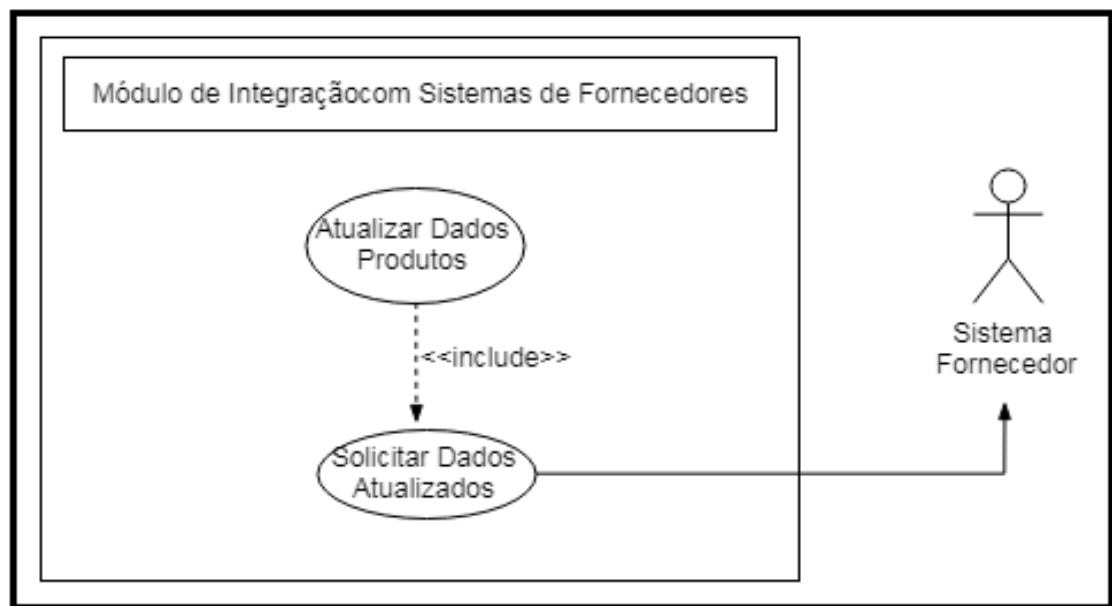


Figura 2- Diagrama de Casos de Uso: Módulo de Integração com Sistemas de Fornecedores

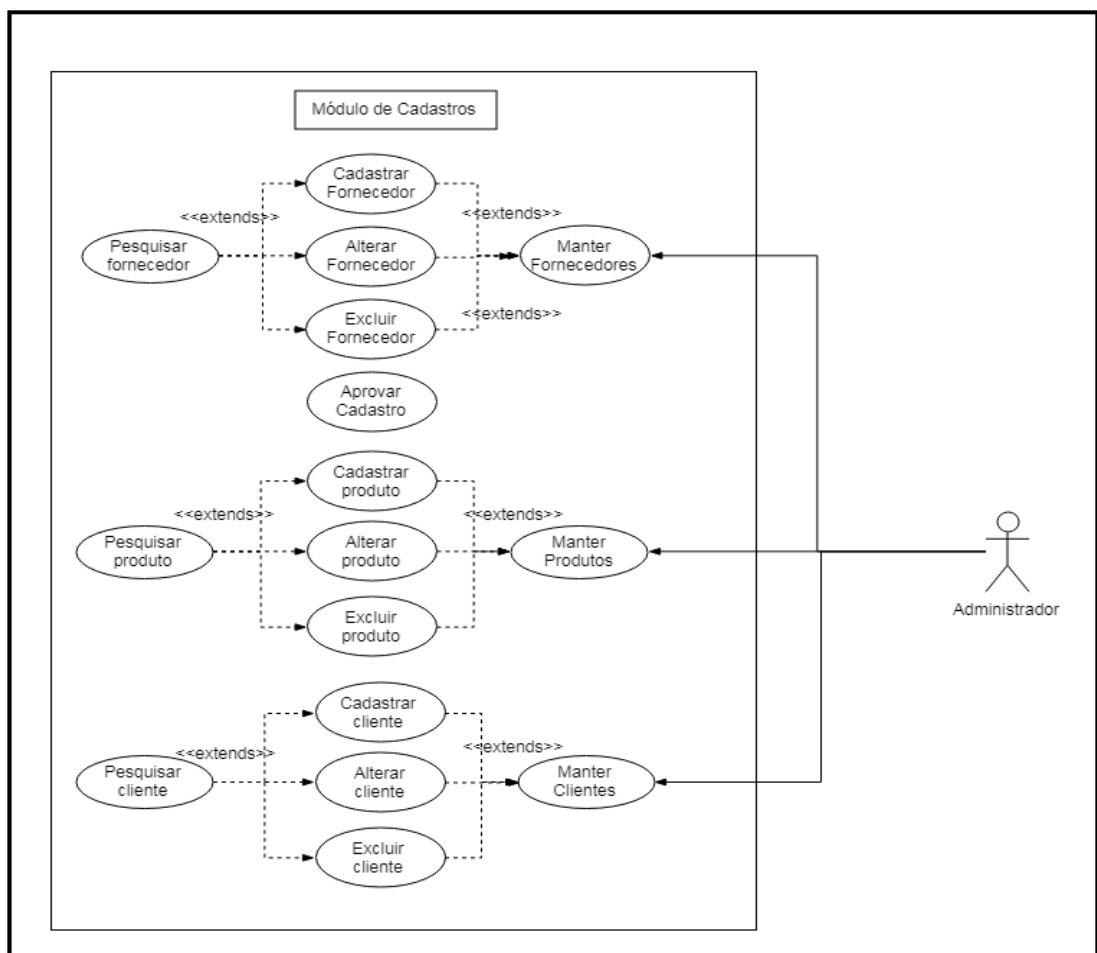


Figura 3- Diagrama de Casos de Uso: Módulo de Cadastros

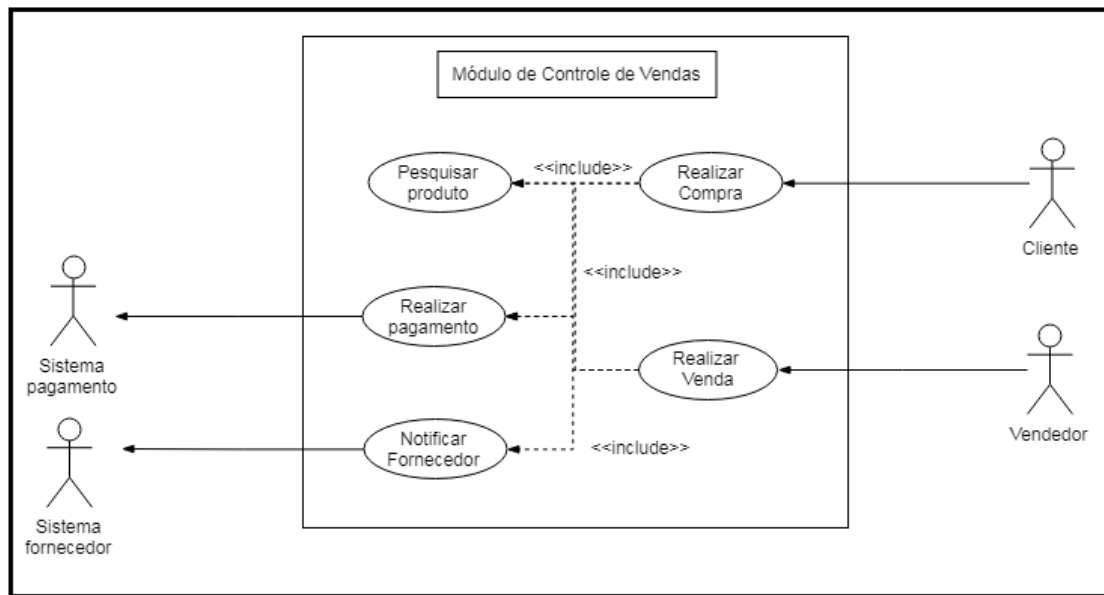


Figura 4- Diagrama de Casos de Uso: Módulo de Controle de Vendas

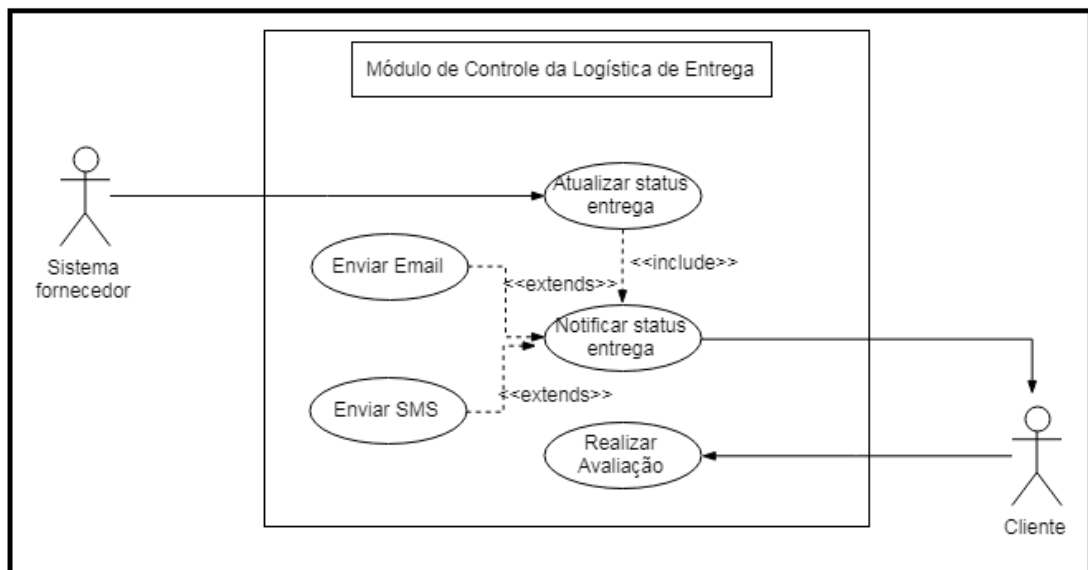


Figura 5- Diagrama de Casos de Uso: Módulo de Controle da Logística de Entrega

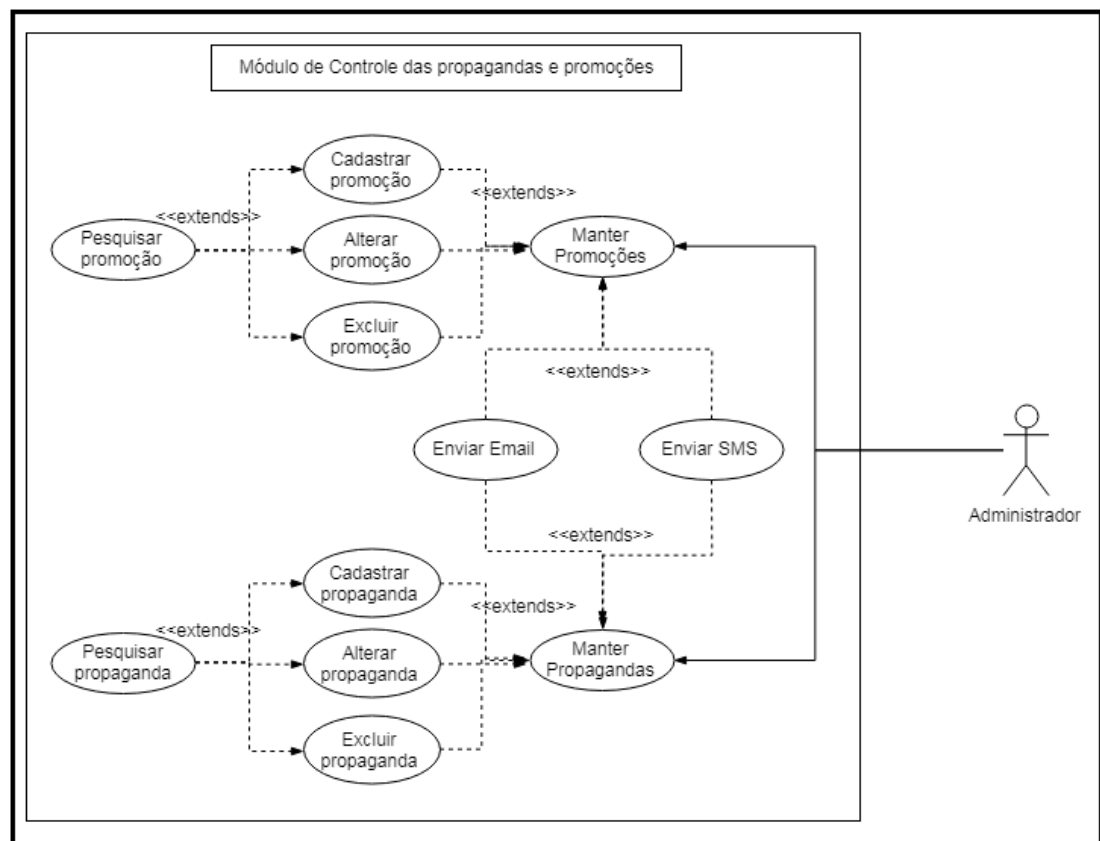


Figura 6- Diagrama de Casos de Uso: Módulo de Controle das Propagandas e Promoções

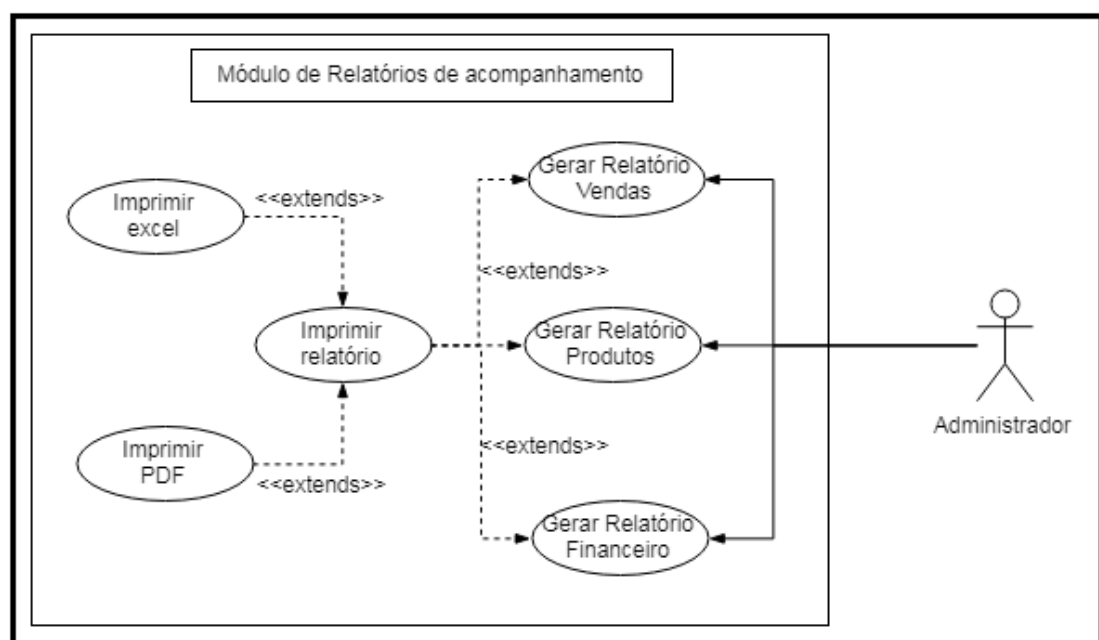


Figura 7- Diagrama de Casos de Uso: Módulo de Relatórios de Acompanhamento

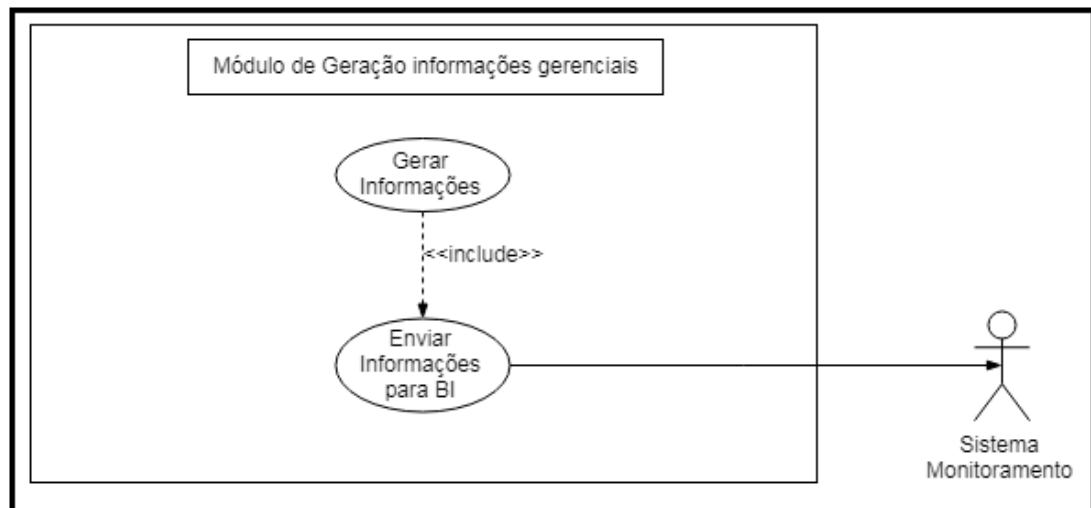


Figura 8- Diagrama de Casos de Uso: Módulo de Geração de Informações Gerenciais

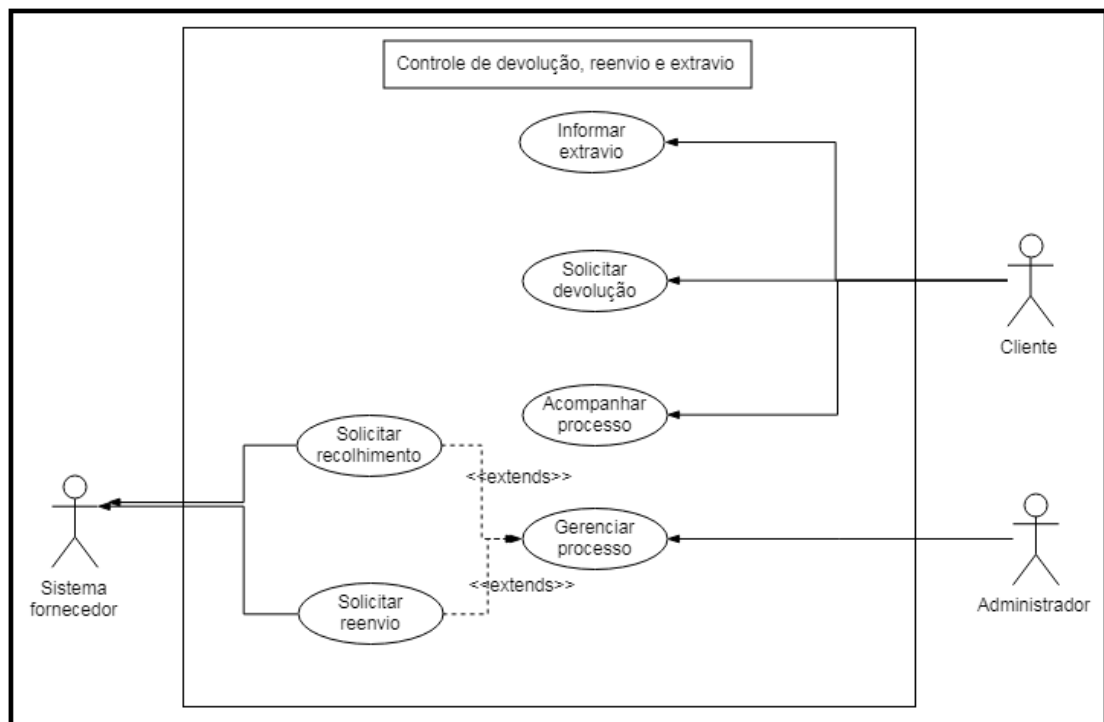


Figura 9 - Diagrama de Casos de Uso: Módulo de Controle de Devolução, Reenvio e Extravio

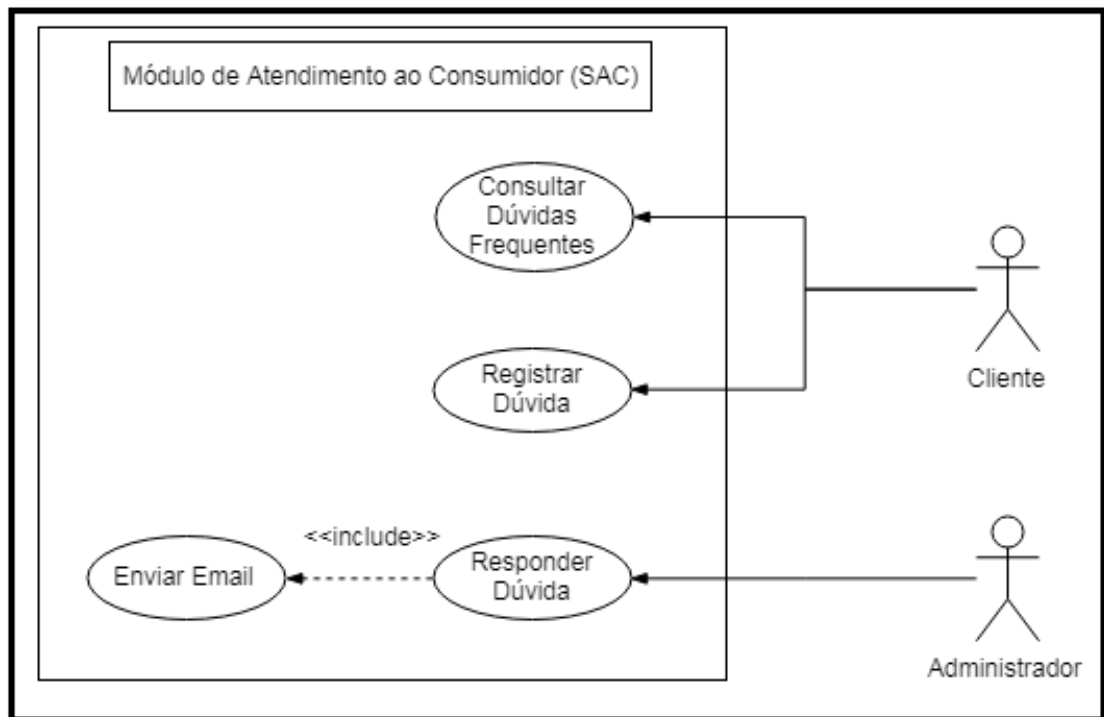


Figura 10 - Diagrama de Casos de Uso: Módulo de Atendimento ao Consumidor (SAC)

## 4.2. Descrição resumida dos casos de uso

### Módulo de Segurança

**Autenticar:** este caso de uso deve permitir ao usuário realizar a autenticação na aplicação. Deve ser disponibilizado um formulário com os campos usuário e senha. Caso o usuário tente acessar o sistema sem estar autenticado ele deve ser redirecionado para este formulário de autenticação. Também deve ser disponibilizado um serviço de autenticação para acesso às API's de integração do sistema, por onde, por exemplo, os sistemas de fornecedores serão autenticados e autorizados a utilizar as API's.

**Recuperar Senha:** este caso de uso deve permitir ao usuário recuperar sua senha caso a tenha esquecido. Deve ser disponibilizado um formulário para entrada do e-mail do usuário. Após validado o e-mail, deve ser enviado para este um link de recuperação da senha.

**Realizar Cadastro:** este caso de uso deve permitir ao usuário realizar o cadastro para utilizar o sistema. Caso o cadastro seja realizado como fornecedor o

mesmo deve ser aprovado pelo administrador do sistema. Caso seja como consumidor o usuário já deverá estar apto a realizar compras na loja virtual.

**Manter Usuários:** este caso de uso deve permitir ao administrador gerenciar os usuários do sistema. Deverá ser exibida uma lista de usuários e as opções pesquisar, cadastrar, alterar e excluir usuário. A exclusão deve ser lógica.

**Manter Permissões:** este caso de uso deve permitir ao administrador gerenciar as permissões de acesso ao sistema. Deve ser possível ao administrador conceder e remover permissões aos módulos e funções do sistema.

#### Módulo de Cadastros

**Manter Fornecedores:** este caso de uso deve permitir ao administrador gerenciar fornecedores no sistema. Deverá ser exibida uma lista de fornecedores e as opções pesquisar, cadastrar, alterar e excluir fornecedor. A exclusão deve ser lógica. Também deve ser possível aprovar o cadastro de um fornecedor previamente cadastrado.

**Manter Produtos:** este caso de uso deve permitir ao administrador gerenciar os produtos do sistema. Deverá ser exibida uma lista de produtos cadastrados e as opções pesquisar, cadastrar, alterar e excluir produto. A exclusão deve ser lógica.

**Manter Clientes:** este caso de uso deve permitir ao administrador gerenciar os clientes do sistema. Deverá ser exibida uma lista de clientes e as opções pesquisar, cadastrar, alterar e excluir cliente. A exclusão deve ser lógica.

**Manter Vendedores:** este caso de uso deve permitir ao administrador gerenciar os vendedores do sistema. Deverá ser exibida uma lista de vendedores e as opções pesquisar, cadastrar, alterar e excluir vendedor. A exclusão deve ser lógica.

#### Módulo de Controle de Vendas

**Realizar Compra:** este caso de uso deve permitir ao cliente realizar compras na loja virtual. Deverá ser possível pesquisar um produto por descrição, categoria e produtos em promoção. Ao acessar a tela de compras deverá ser exibido em

destaque produtos em promoção, e se o usuário estiver logado estes produtos deverão ser filtrados de acordo com a preferência do cliente, obtida através do histórico de compras e pesquisas do mesmo na loja.

**Realizar Venda:** este caso de uso deve permitir ao vendedor realizar vendas pela loja virtual. Deverá ser possível pesquisar um produto por descrição, categoria e produtos em promoção. Ao acessar a tela de vendas deverá ser exibido em destaque produtos em promoção. Para que seja realizada uma venda por um vendedor, o cliente deverá estar cadastrado no sistema.

**Realizar Pagamento:** este caso de uso deve permitir ao cliente e ao vendedor realizar o pagamento dos pedidos de compra realizados na loja virtual. O pagamento deverá ser possível por boleto bancário e cartão de crédito ou débito.

**Notificar Fornecedor:** este caso de uso deve ser disparado automaticamente após a aprovação do pagamento do pedido de compras ou vendas. Os sistemas dos fornecedores dos itens dos pedidos de cada compra ou venda deverão ser notificados para que procedam o processo de logística de entrega.

**Acompanhar Pedido:** este caso de uso deve permitir ao cliente, vendedor ou administrador acompanhar a situação dos pedidos realizados na loja virtual. Deve ser possível filtrar os pedidos por produto, situação e data do pedido. Caso o usuário seja vendedor ou administrador também deverá ser possível filtrar por cliente.

#### Módulo de Controle da Logística de Entrega

**Atualizar Status Entrega:** este caso de uso deve permitir ao sistema do fornecedor atualizar o status dos pedidos realizados na loja virtual. A cada mudança de status o mesmo deve ser atualizado para que o cliente possa acompanhar o mesmo.

**Notificar Status Entrega:** este caso de uso deve ser disparado automaticamente na atualização dos status de entrega. O cliente deve ser notificado por e-mail ou SMS de acordo com a configuração realizada por ele no seu cadastro.

**Realizar Avaliação:** Este caso de uso deve permitir ao cliente realizar uma avaliação sobre sua satisfação ao utilizar o sistema da loja virtual e sobre a qualidade e a entrega dos produtos, o que será utilizado para avaliar os fornecedores.

#### Módulo de Controle das Propagandas e Promoções

**Manter Promoções:** este caso de uso deve permitir ao administrador gerenciar as promoções de produtos da loja virtual. Deverá ser exibida uma lista de promoções e as opções pesquisar, cadastrar, alterar e excluir promoção. A exclusão deve ser lógica. Deverá ser possível montar listas de e-mails e SMS e realizar o envio de mensagens para estas listas.

**Manter Propagandas:** este caso de uso deve permitir ao administrador gerenciar as propagandas da loja virtual. Deverá ser exibida uma lista de propagandas e as opções pesquisar, cadastrar, alterar e excluir propaganda. A exclusão deve ser lógica. Deverá ser possível montar listas de e-mails e SMS e realizar o envio de propagandas para estas listas.

#### Módulo de Relatórios de Acompanhamento

**Gerar Relatório de Vendas:** este caso de uso deve permitir ao administrador emitir relatórios de vendas. Deverá ser possível gerar relatórios por produtos, categorias, períodos, vendedores, fornecedores e clientes. Os relatórios poderão ser salvos ou impressos em PDF e Excel.

**Gerar Relatório de Produtos:** este caso de uso deve permitir ao administrador emitir relatórios de produtos. Deverá ser possível gerar relatórios por fornecedores, categorias e disponibilidade em estoques. Os relatórios poderão ser salvos ou impressos em PDF e Excel.

**Gerar Relatório Financeiro:** este caso de uso deve permitir ao administrador emitir relatórios financeiros. Deverá ser possível gerar relatórios de pedidos a receber, fornecedores a pagar e outros custos. Os relatórios poderão ser salvos ou impressos em PDF e Excel.



### Módulo de Geração de Informações Gerenciais

Gerar Informações Gerenciais: este caso de uso deve executar periodicamente consolidando estatísticas do sistema como dados de vendas, cadastros de usuários e vendedores e avaliações de fornecedores. Os dados consolidados devem ser armazenados em banco de dados para posterior análise através de sistema de Business Intelligence.

Enviar Informações Gerenciais para Sistema de Business Intelligence: Após a execução do caso de uso Gerar Informações Gerenciais, este caso de uso deve ser executado para que os dados consolidados sejam enviados para o Sistema de Monitoramento existente.

### Módulo de Controle de Devolução, Reenvio e Extravio

Informar Extravio: este caso de uso deve permitir ao cliente informar o extravio do produto durante o processo de entrega do mesmo.

Solicitar Devolução: este caso de uso deve permitir ao cliente solicitar a devolução do produto conforme as regulamentações de comércio eletrônico.

Acompanhar Processo: este caso de uso deve permitir ao cliente acompanhar o status da sua solicitação de devolução ou do reenvio dos produtos pelo extravio dos mesmos.

Solicitar Recolhimento: este caso de uso deve permitir ao administrador solicitar ao fornecedor o recolhimento do produto em devolução pelo cliente.

Solicitar Reenvio: este caso de uso deve permitir ao administrador solicitar ao fornecedor o reenvio do produto em caso de extravio do mesmo.

Gerenciar Processo: este caso de uso deve permitir ao administrador gerenciar os processos de solicitações dos clientes nos casos de informações de extravio e solicitações de devolução de produtos.

### Módulo de Atendimento ao Consumidor (SAC)

Consultar Dúvidas Frequentes: este caso de uso deve permitir ao cliente consultar dúvidas previamente cadastradas no sistema. Deve ser possível filtrar por categoria, assunto e texto livre.

Registrar Dúvida: este caso de uso deve permitir ao cliente registrar uma dúvida. Caso o cliente não encontre uma resposta para suas dúvidas na sessão dúvidas frequentes, deverá existir um formulário para cadastro de uma nova dúvida, a qual deverá ser encaminhada por e-mail para o administrador para que possa ser respondida o mais breve possível. O e-mail deverá conter um link para um formulário de resposta.

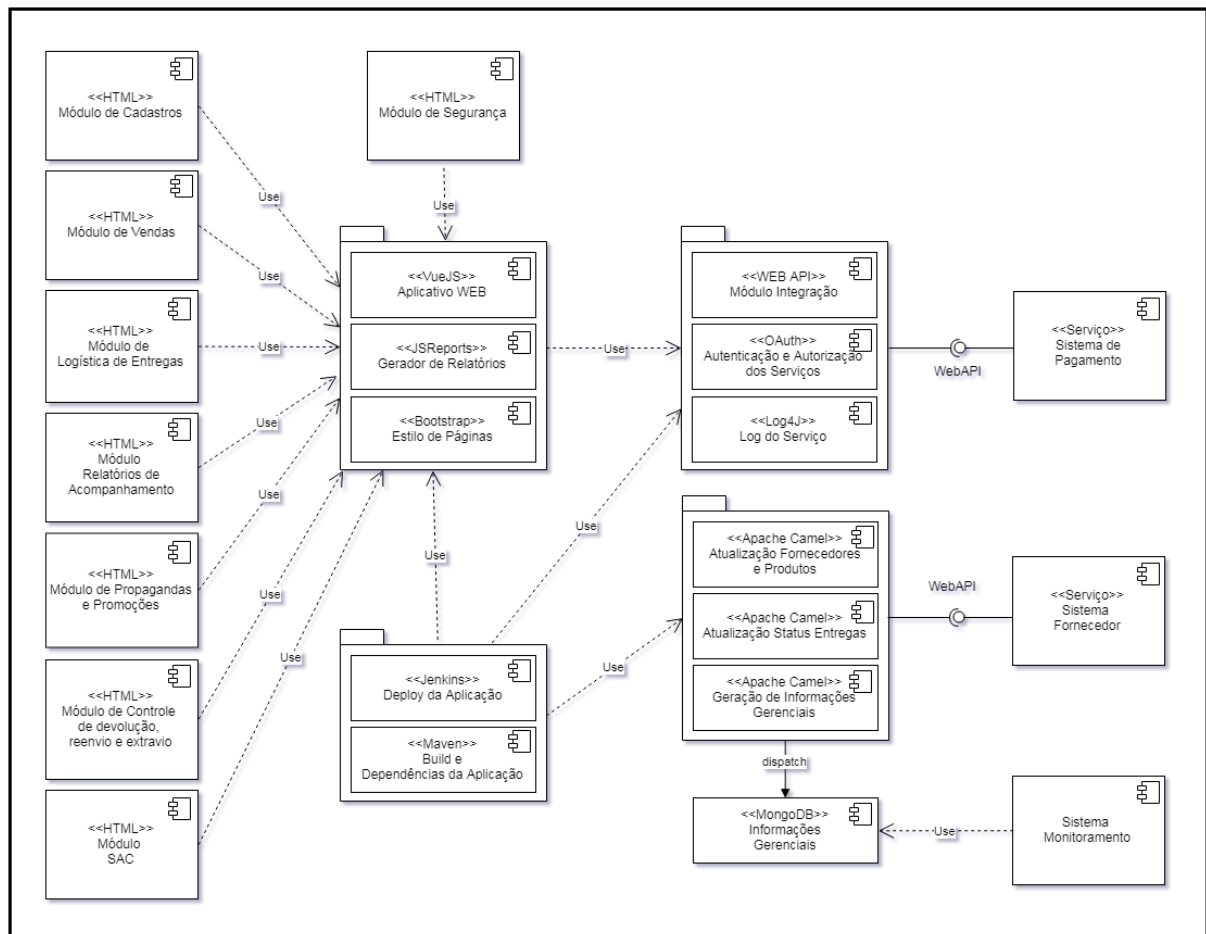
Responder Dúvida: este caso de uso deve permitir ao administrador responder a uma dúvida do cliente. O administrador deverá clicar no link enviado por e-mail e acessar o formulário de resposta. Após o administrador preencher o formulário e submeter o mesmo, um e-mail deverá ser enviado ao cliente com os dados da solicitação e a resposta do administrador.

### Módulo de Integração com Sistemas de Fornecedores

Atualizar Dados Produtos: este caso de uso deve executar periodicamente atualizando os dados dos produtos de todos os fornecedores cadastrados no sistema. Este caso de uso também deve permitir aos sistemas dos fornecedores atualizarem os dados dos produtos através de uma API, sendo responsabilidade neste caso, do fornecedor atualizar os dados.

## **4.3. Modelo de componentes**

Os componentes foram organizados para serem reutilizáveis e fornecendo interfaces bem definidas de acordo com suas responsabilidades. No diagrama de componentes, foram inseridos os componentes que impactaram no design da arquitetura e seleção das tecnologias.



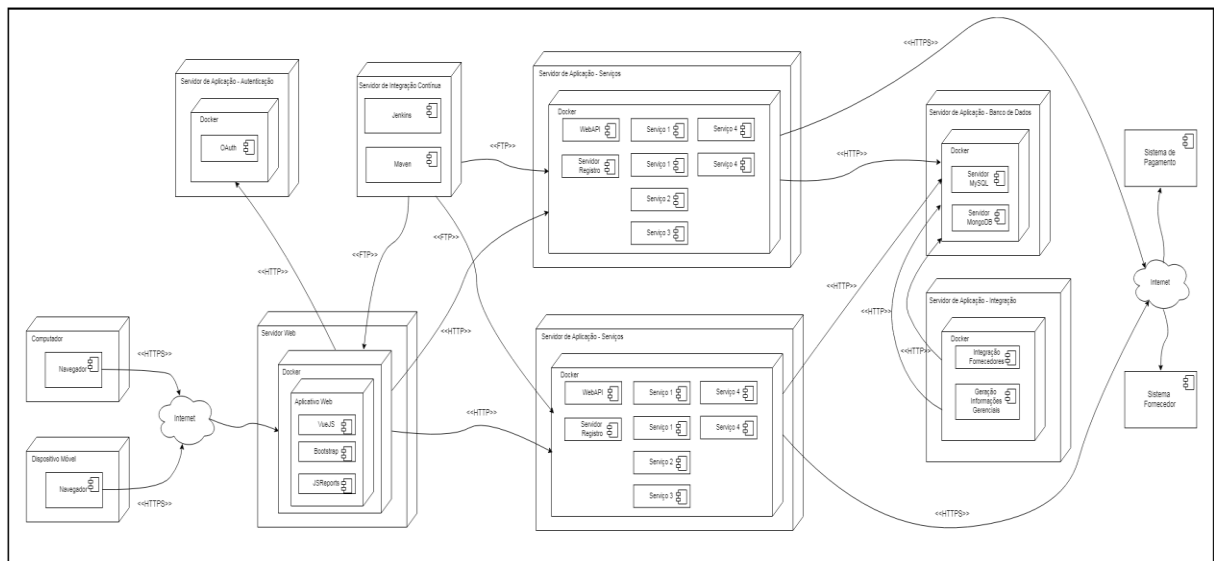
**Figura 11 - Diagrama de Componentes**

A seguir é mostrado um detalhamento dos componentes e os módulos envolvidos no modelo de componentes. Nessa arquitetura devemos considerar a divisão do sistema em aplicativo Web, composto por VueJS, JSReports, Bootstrap, que foi escolhido pensando no RNF Acessibilidade, e uma WebAPI, que irá fornecer os dados necessários para exibir em uma página acessada pelo navegador. Para garantir o RNF de Segurança, foi implantado um componente de autenticação OAuth. Para acesso a todos os outros componentes é necessário estar autenticado neste servidor. Pensando no requisito acessibilidade, o aplicativo web e a WebAPI são construídos e implantados pelos componentes de integração contínua, garantindo os RNF de Manutenibilidade e Testabilidade. Com relação aos RNF de Desempenho e Disponibilidade, o componente WebAPI foi construído utilizando balanceamento de carga com registro e descoberta de serviços.

<b>Componente</b>	<b>Descrição</b>
Aplicativo Web (Módulos de Cadastro, Controle de Vendas, Logística de Entregas, Relatórios de Acompanhamento, Propagandas e Promoções, Controle de Devolução, Reenvio e Extravio, Segurança e Atendimento ao Consumidor - SAC)	Componentes utilizados para gerar o conteúdo HTML dos módulos do sistema a serem renderizados nos navegadores.
WebAPI (Módulo de Integração)	Componentes utilizados para expor as operações que serão invocadas por outros módulos do sistema.
Integração com Sistemas de Fornecedores	Componentes utilizados para integração com os sistemas dos fornecedores.
Geração de Informações Gerenciais	Componente utilizado para consolidação de informações gerenciais e integração com o Sistema de Monitoramento.
Integração contínua da aplicação	Componentes utilizados para construir, gerenciar as dependências e implantar os módulos da aplicação nos respectivos containers.
Sistema de Pagamento	Interface WebAPI REST, responsável por prover os serviços do sistema do meio de pagamento.
Sistema dos Fornecedores	Interface responsável por prover os serviços dos sistemas dos fornecedores.
Sistema de Monitoramento (Business Intelligence)	Interface responsável por prover os serviços do Sistema de Monitoramento (Business Intelligence).

#### 4.4. Modelo de implantação

Abaixo é apresentado o diagrama de implantação, mostrando a relação dos componentes com seus respectivos servidores e mostrando como a aplicação deve ser implantada.



**Figura 12 - Diagrama de Implantação**

Os módulos dos sistemas foram implantados em containers de aplicação nos seus respectivos servidores, a fim de prover todos os recursos necessários para que a aplicação funcione independente do ambiente. Os servidores acima podem estar em alocados em um Data Center ou podem estar na nuvem, para garantir a disponibilidade durante o dia e a noite. Os containers facilitam uma futura migração, no caso de optar por implantar inicialmente a aplicação em um ambiente e querer migrar para outro, pois provém toda a infraestrutura necessária para que a aplicação funcione no servidor, independente de ambiente. A utilização de containers também visa facilitar a escalabilidade, pois com a utilização do módulo de registro de serviços é possível incluir quantas instancias forem necessárias de cada serviço para atender a demanda de forma transparente para o usuário.

<b>Componente</b>	<b>Descrição</b>
Navegador	Browser a ser utilizado pelos usuários para interagir com as funcionalidades do sistema. Pode ser utilizado através de um computador ou dispositivo móvel.
Servidor Web	É o responsável por armazenar os componentes acessados pelos navegadores.
Servidor de Aplicação - Autenticação	É o componente responsável pelo processo de autenticação e autorização da aplicação. Neste servidor será implantado o serviço de autenticação OAuth.
Servidor de Aplicação - Serviços	É o componente do modelo de implantação responsável pelos serviços de negócio da aplicação. Deverá ser escalado de acordo com a demanda e conterá os serviços necessários e acessados pela aplicação Web.
Servidor de Aplicação – Banco de Dados	É o componente responsável pelos bancos de dados da aplicação.
Servidor de Aplicação – Integração	Componente responsável pelas integrações com outros sistemas.
Servidor de integração contínua	Responsável pelo gerenciamento de dependências, construção e implantação da aplicação nos servidores.

#### 4.5. Modelo de dados

Abaixo é apresentado o diagrama entidade relacionamento. O modelo de dados foi pensado de forma a permitir o cadastro de produtos tanto pela própria loja quanto de parceiros, denominados Fornecedores, na modalidade dropshipping.

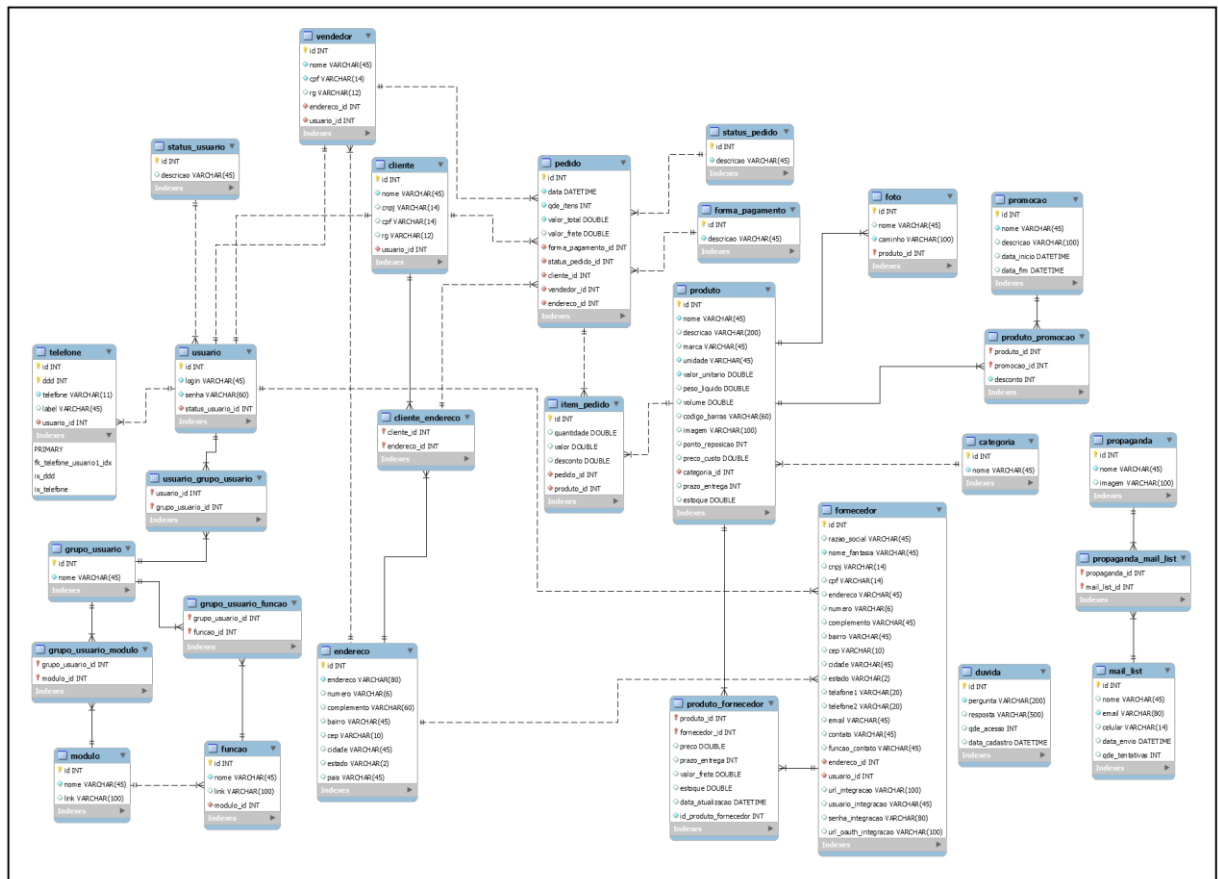


Figura 13 - Diagrama Entidade Relacionamento (ER)

## 5. Prova de conceito / protótipo arquitetural

### 5.1. Implementação e implantação

A prova de conceito deste projeto visa avaliar a viabilidade e se a arquitetura proposta atende as necessidades do projeto. Para isto, foram escolhidos três casos de uso bem distintos para que o protótipo abrangesse um maior número de requisitos de qualidade e funcionais. Os três casos de uso escolhidos foram o Autenticar, Manter Fornecedores e Atualizar Dados Produtos. As telas da aplicação podem ser visualizadas nas evidências da avaliação.

As tecnologias utilizadas são descritas no item de mecanismos arquiteturais.

Os seguintes requisitos não funcionais serão avaliados nesta POC:

- Segurança – O sistema deve apresentar altos padrões de segurança.

Esse requisito foi escolhido devido à importância em manter os dados seguros e evitar falhas de segurança no projeto.

Os critérios de aceite são:

- Não permitir que usuários possam acessar páginas privadas sem estar autenticados no sistema.
- Ao identificar que o usuário não está autenticado, o sistema deve redirecionar o usuário para tela de autenticação.
- O sistema deve permitir que o usuário navegue em telas públicas sem precisar estar autenticado.
- Acessibilidade – O sistema deve suportar ambientes Web responsivos e ambientes móveis.

Esse requisito foi escolhido para garantir que atenda todas as exigências da arquitetura em ter um sistema responsivo e que se adapte em celulares, tablets e desktops.

Os critérios de aceite são:

- A tela do sistema deve apresentar facilidade de navegação e os objetos da tela devem se adaptar de acordo com a resolução identificada, tanto para celulares como desktops.
- O sistema deve se manter com o mesmo padrão de cores e objetos.
- O sistema deve ser compatível com os principais browsers do mercado como: Internet Explorer, Google Chrome e Firefox.
- Disponibilidade – O sistema deve operar em qualquer período do dia e da noite.

Esse requisito foi escolhido visando avaliar o comportamento do sistema em caso de falha de um dos serviços da aplicação.



Os critérios de aceite são:

- Todos os usuários logados na aplicação devem continuar utilizando o sistema sem perceber que houve uma queda em um dos serviços da aplicação.
- O deve ser possível subir novos serviços ou desligar outros dependendo da demanda de forma transparente ao usuário.

## 5.2. Interfaces/ APIs

### Interface do Serviço de Autenticação

Visando centralizar a autenticação e autorização de acesso a todos os serviços da aplicação, foi criado um servidor de autenticação OAuth que pode ser acessada no seguinte endereço:

<https://contexto-servidor/oauth>

Os métodos e elementos necessários para utilização do Web Service serão descritos e exemplificados abaixo:

Método token (<https://contexto-servidor/oauth/token>)

Este método deve ser utilizado para solicitar um token de acesso às aplicações e serviços que utilizam o servidor de autenticação como meio de proteção.

Este método espera receber os seguintes parâmetros:

Parâmetro	Tipo	Descrição	Obrigatório
grant_type	String	Tipo de autenticação. Valores aceitos: "password", utilizado para formulários de autenticação e "client_credentials", utilizado por aplicações clientes.	Sim

client_id	String	Id da aplicação cliente. Utilizado para o tipo de autenticação "client_credentials".	Sim, para o respectivo tipo de autenticação
client_secret	String	Senha da aplicação cliente. Utilizado para o tipo de autenticação "client_credentials".	Sim, para o respectivo tipo de autenticação
username	String	Nome do usuário. Utilizado para o tipo de autenticação "password".	Sim, para o respectivo tipo de autenticação
password	String	Senha do usuário. Utilizado para o tipo de autenticação "password".	Sim, para o respectivo tipo de autenticação

O cabeçalho da requisição deve conter o seguinte par chave-valor:

Chave: Authorization

Valor: Basic base64\_token

Onde o base64\_token deve ser obtido pela codificação Base64 da seguinte combinação: client\_id:client\_secret

Exemplos:

- Requisição

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

- Method:** POST
- URL:** http://localhost:9092/oauth/token?grant\_type=password&username=oauth\_admin&password=user
- Headers:**

Key	Value
<input checked="" type="checkbox"/> Authorization	Basic Y3Vybf9jbGllbnQ6dXNlcg==
New key	Value

- Resposta

```
{
  "access_token": "82302aa5-ca85-4638-bf73-8f6793658074",
  "token_type": "bearer",
  "expires_in": 7199,
  "scope": "read write"
}
```

- Requisição

POST <span>▼</span> http://localhost:9092/oauth/token?grant_type=client_credentials&client_id=curl_client&client_secret=user			
Authorization   Headers (1)   Body   Pre-request Script   Tests			
	Key	Value	Descriptio
<input checked="" type="checkbox"/>	Authorization	Basic Y3Vybf9jbGllbnQ6dXNlcg==	
	New key	Value	Descript

- Resposta

```
{
  "access_token": "5e27af80-e838-4c63-9f17-633f98eeb90a",
  "token_type": "bearer",
  "expires_in": 6730,
  "scope": "read write"
}
```

## 6. Avaliação da Arquitetura

### 6.1. Análise das abordagens arquiteturais

A arquitetura proposta é baseada na modularização e separação de responsabilidades. Está sendo proposta a utilização de serviços independentes agrupados por responsabilidades. Estes serviços serão embutidos em containers que possuem toda a infraestrutura necessária para executar independentemente da plataforma. A partir destes containers também será possível criar quantas imagens forem necessárias para atender a demanda da aplicação.

A integração entre módulos será feita através de interfaces bem definidas em serviços rest e o acesso centralizado em uma WebAPI. A integração com outros sis-

temas poderá ser feita utilizando a WebAPI ou através do Apache Camel, se necessário.

Na implementação da WebAPI, é utilizado o Netflix Zuul e o Netflix Eureka, possibilitando também o balanceamento de carga através do registro e descoberta de serviços.

Para a interface de usuário, está sendo proposta a utilização do VueJS, por sua simplicidade e objetividade. Também é utilizado o Bootstrap, que torna mais simples a criação de páginas elegantes e responsivas.

O versionamento de código-fonte sugerido nesta proposta é o Git, por ser o mais utilizado atualmente e pela equipe o conhecer, reduzindo o tempo de aprendizado de novas tecnologias e os riscos do projeto. O mecanismo de deploy da aplicação e realização de testes automatizados escolhido foi o Jenkins. Para o controle de dependências e build escolhido foi o Maven.

## **6.2. Identificação dos atributos de qualidade**

Os principais atributos de qualidade considerados foram segurança, acessibilidade e disponibilidade.

## **6.3. Cenários**

**Cenário 1:** Neste cenário procura-se garantir o atendimento do atributo de qualidade segurança. Ao acessar uma URL ou página privada, o sistema deve garantir a segurança e confidencialidade das informações, permitindo apenas a usuários autorizados este acesso. Caso um usuário tente acessar uma página privada sem estar autenticado, o sistema deve redirecionar o mesmo para a tela de login. O sistema também deve garantir que as páginas públicas possam ser acessadas sem autenticação.

**Cenário 2:** O sistema deve suportar ambientes Web responsivos e ambientes móveis. Ao realizar o acesso à aplicação através de um dos principais browsers do mercado, como Google Chrome, Firefox e IE, em um dispositivo móvel ou desktop

com resolução reduzida, o sistema deverá se adaptar automaticamente, permitindo uma experiência agradável ao usuário.

**Cenário 3:** O sistema deve operar em qualquer período do dia e da noite, atendendo ao requisito disponibilidade. Havendo a queda de um dos serviços da aplicação, todos os usuários logados devem continuar utilizando o sistema normalmente. O sistema deve garantir a disponibilidade redirecionando a requisição para outro serviço ativo.

Foi utilizado o método de árvore de utilidade reduzida e com prioridades para priorização dos cenários para avaliação. Os cenários foram categorizados de acordo com os atributos de qualidade a que estão relacionados e então classificados em função da sua importância e complexidade, considerando a percepção de negócio e arquitetura. As variáveis de priorização “Importância”, (IMP) e “Complexidade” (COM), foram classificadas de acordo com o requisito em alta (A), média (M) e baixa (B).

Atributos de Qualidade		Cenários	IMP.	COM.
Usabilidade e Funcionalidade	Segurança	Cenário 1: O sistema deve apresentar altos padrões de segurança.	A	M
	Acessibilidade	Cenário 2: O sistema deve suportar ambientes Web responsivos e ambientes móveis.	M	A
	Disponibilidade	Cenário 3: O sistema deve operar em qualquer período do dia e da noite.	A	A

#### 6.4. Avaliação

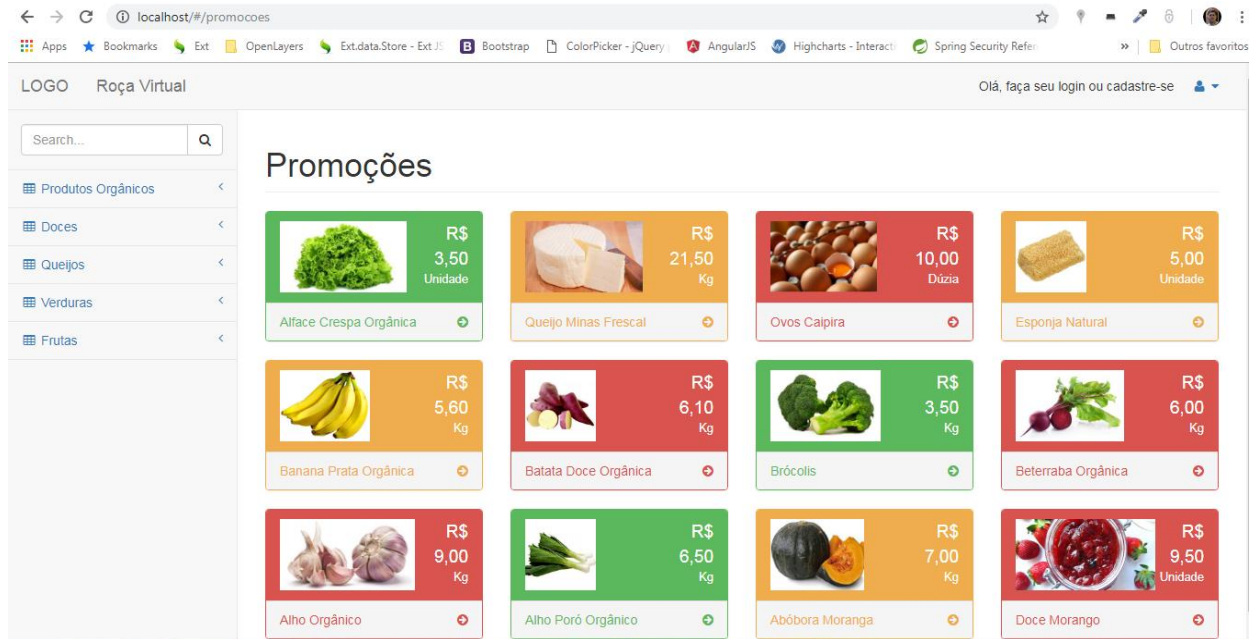
Abaixo serão apresentadas as evidências dos testes de avaliação dos cenários identificados no item 6.3.

- **Cenário 1:**

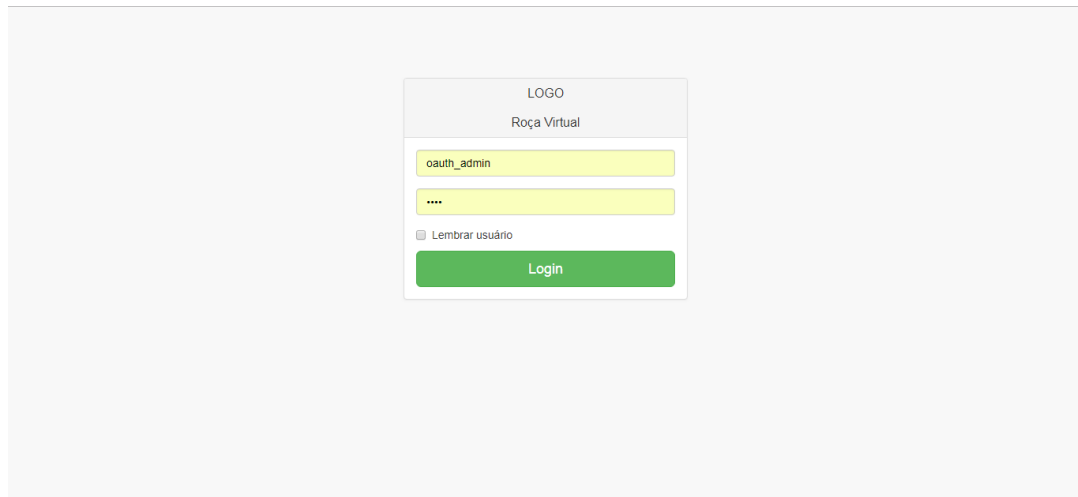
Atributo de Qualidade:	Segurança
Requisito de Qualidade:	O sistema deve apresentar altos padrões de segurança
Preocupação:	Não permitir o acesso a páginas privadas do sistema sem autenticação no sistema.
Ambiente:	Sistema em operação normal
Estímulo:	Usuário tentando acessar uma página privada do sistema sem estar autenticado no sistema.
Mecanismo:	Criar um mecanismo de autorização que impeça o acesso a paginas privadas sem autorização.
Medida da Resposta:	O usuário deve ser redirecionado para a tela de autenticação.
Riscos:	O controle de acesso à aplicação é crítico, pois acessos não autorizados podem trazer enormes prejuízos financeiros e denegrir a imagem da empresa.
Pontos de Sensibilidade:	Servidor de aplicação operando sobre protocolo SSL (HTTPS)

- **Evidências do Cenário 1**

Usuário acessando uma página pública da loja virtual.



Usuário realizando autenticação para acesso às páginas privadas do sistema. Ao tentar acessar uma página privada sem estar autenticado o usuário é direcionado para a página de login.



- **Cenário 2:**

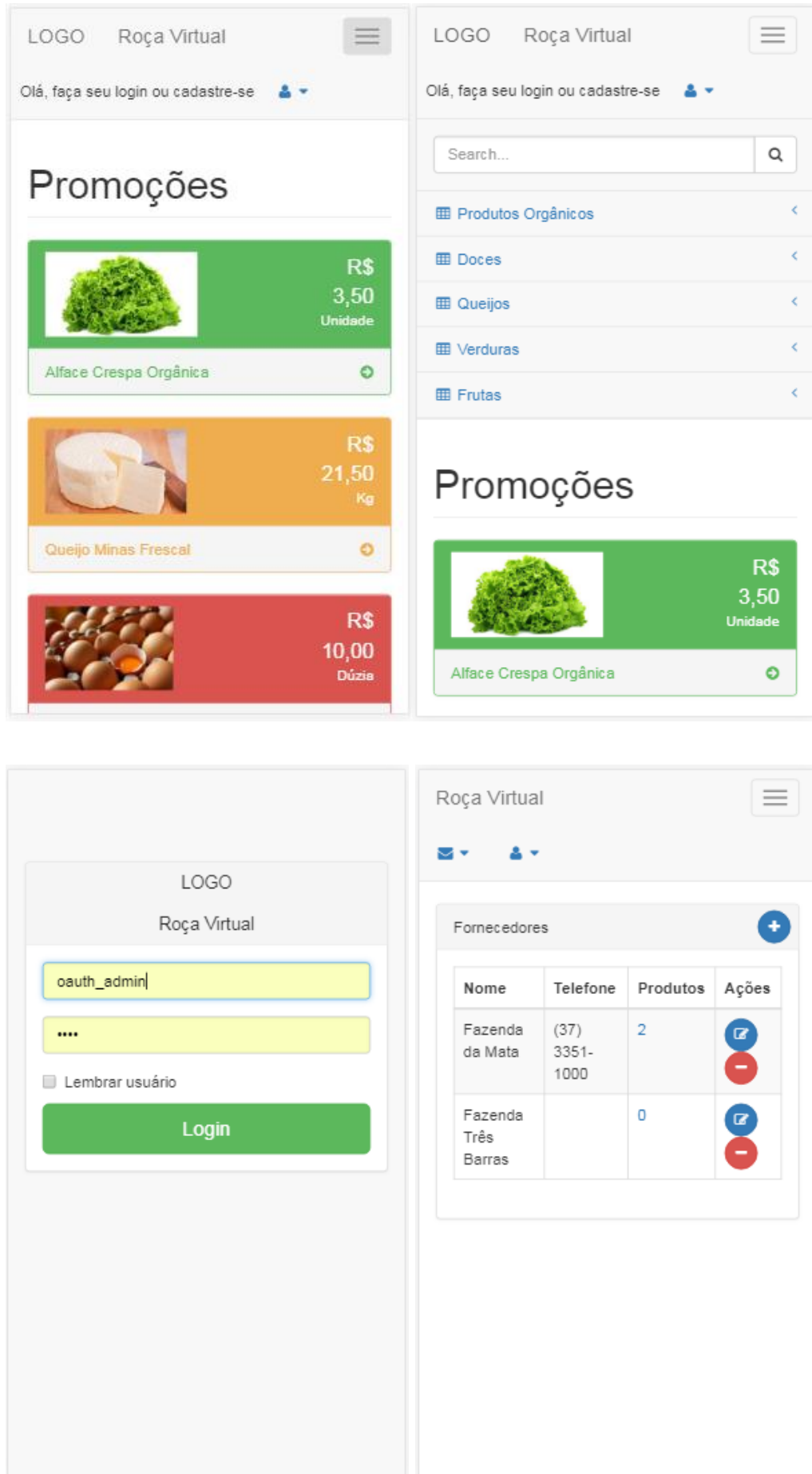
Atributo de Qualidade:	Acessibilidade
Requisito de Qualidade:	O sistema deve suportar ambientes Web responsivos e ambientes móveis.
Preocupação:	Garantir que o sistema atenda todas as exigências da arquitetura em ter um sistema responsivo e que se adapte

	em celulares, tablets e desktops.
Ambiente:	Funcionamento, carga normal.
Estímulo:	Usuário acessando o sistema de um celular.
Mecanismo:	Desenvolver a camada de apresentação com componentes responsivos, que devem se adaptar às resoluções e tamanho das telas, mudando os componentes de posição de forma a ficar melhor a navegação do usuário.
Medida da Resposta:	Identidade visual semelhante em todas as resoluções, com objetos redimensionados de acordo com a resolução e tamanho.
Riscos:	Caso o usuário não tenha uma boa experiência ao acessar o site não retornará ao mesmo.
Pontos de Sensibilidade:	Diversidade de browsers no mercado.

- **Evidências do Cenário 2**

Usuário acessando o site de um dispositivo móvel Galaxy S5.






- **Cenário 3:**

Atributo de Qualidade:	Disponibilidade
Requisito de Qualidade:	O sistema deve operar em qualquer período do dia e da noite.
Preocupação:	Havendo a queda de um dos serviços da aplicação, todos os usuários logados devem continuar utilizando o sistema normalmente. O sistema deve garantir a disponibilidade redirecionando a requisição para outro serviço ativo.
Ambiente:	Funcionamento, carga normal.
Estímulo:	Falha em um dos serviços da aplicação.
Mecanismo:	Utilizar recursos de balanceamento de carga e redundância de serviços.
Medida da Resposta:	Ao ocorrer a falha em um dos serviços o usuário deve continuar utilizando o sistema.
Riscos:	Caso o sistema fique fora do ar e o usuário não consiga acessá-lo ou realizar uma compra usuário poderá não acessar mais o site.
Pontos de Sensibilidade:	Não há.

- **Evidências do Cenário 3**

Página do servidor de registro de serviços exibindo as instâncias de serviços registradas. Neste momento podemos observar duas instâncias do serviço produto-service, executando nas portas 8096 e 8097.


HOME
LAST 1000 SINCE STARTUP

### System Status

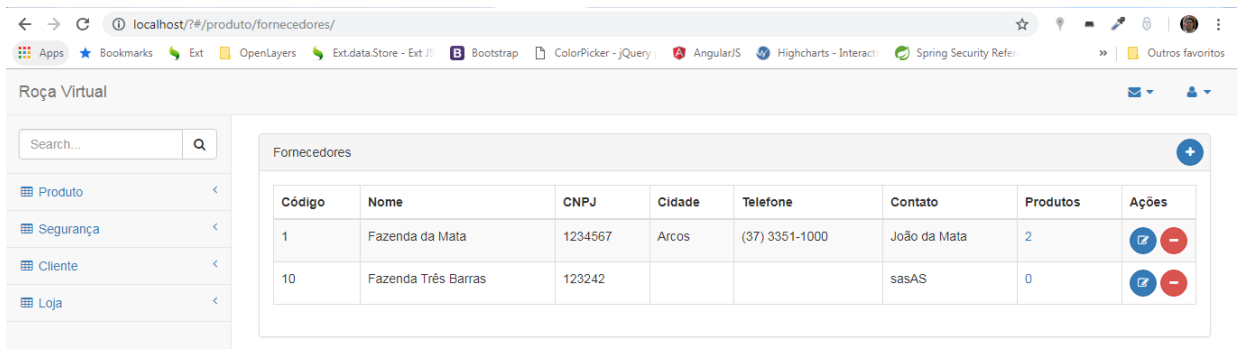
Environment	test	Current time	2018-09-27T22:23:59 -0300
Data center	default	Uptime	23:13
		Lease expiration enabled	true
		Renews threshold	6
		Renews (last min)	8

### DS Replicas

#### Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - Alisson-PC:api-gateway:8080
AUTHORIZATION_SERVER	n/a (1)	(1)	UP (1) - Alisson-PC:authorization_server:9092
EUREKA-SERVER	n/a (1)	(1)	UP (1) - Alisson-PC:eureka-server:9091
PRODUTO-SERVICE	n/a (2)	(2)	UP (2) - Alisson-PC:produto-service:8096 , Alisson-PC:produto-service:8097

Usuário acessando uma página que utiliza o serviço produto-service.

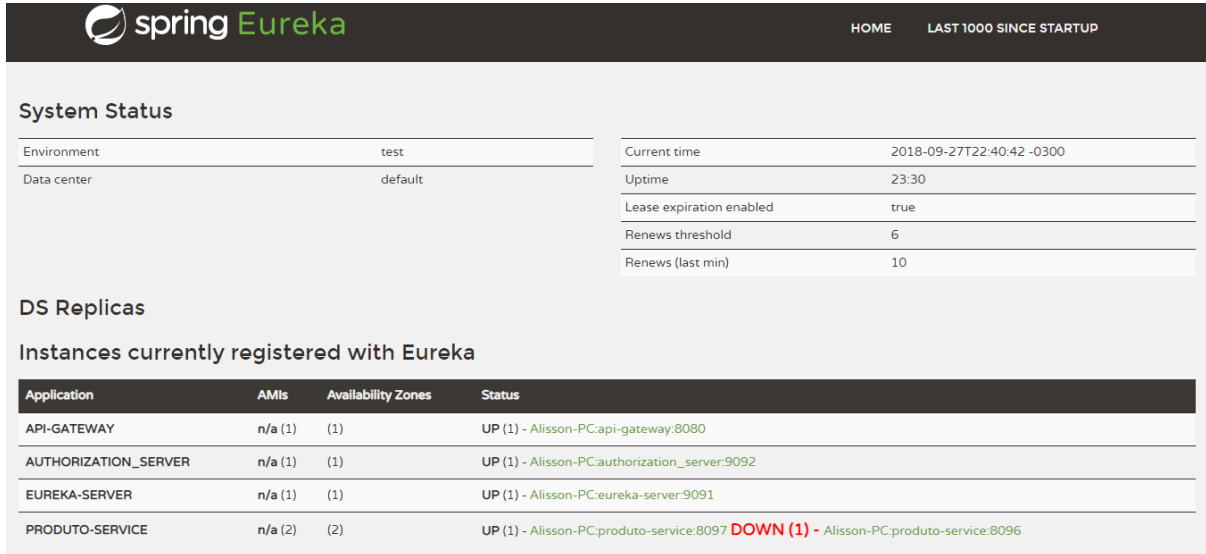


Log do servidor da API Gateway. Todas as requisições de serviços são enviadas à API Gateway, que é responsável por redirecionar as requisições aos serviços solicitados, realizando também o balanceamento de carga.

```

2018-09-27 22:29:43.936 INFO 10648 --- [nio-8080-exec-1] c.a.apigateway.filter.SimpleFilter : GET request to http://localhost:8080/produto-service/produto/fornecedores
2018-09-27 22:30:06.650 INFO 10648 --- [nio-8080-exec-4] c.a.apigateway.filter.SimpleFilter : GET request to http://localhost:8080/produto-service/produto/fornecedores
  
```

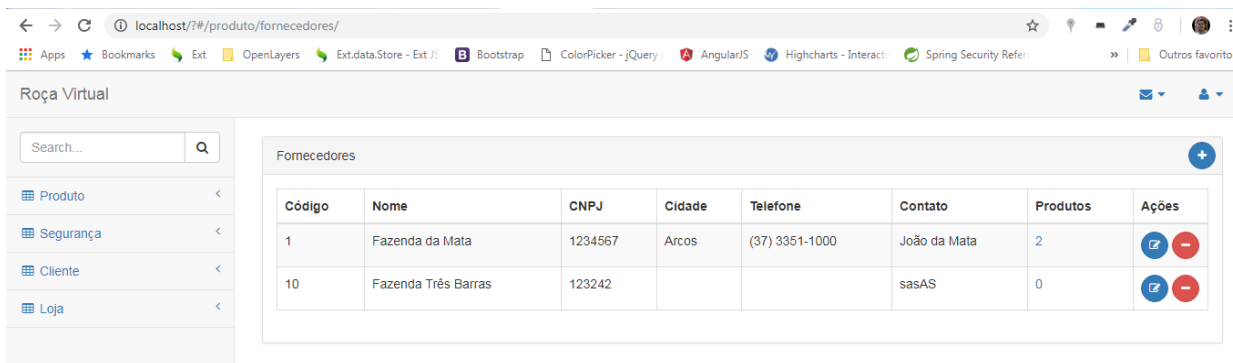
Página do servidor de registro de serviços exibindo as instâncias de serviços registradas após forçada a interrupção de uma das instâncias do serviço produto-service. Neste momento podemos observar uma instância ativa (porta 8097) e uma instância parada (porta 8096) do serviço.



The image shows the Spring Eureka web interface. At the top, there's a navigation bar with 'HOME' and 'LAST 1000 SINCE STARTUP'. Below this, the 'System Status' section displays a table with environment details (test, default) and system metrics (Current time, Uptime, Lease expiration enabled, Renew threshold, Renew last min). The 'DS Replicas' section shows 'Instances currently registered with Eureka' in a table with columns: Application, AMIs, Availability Zones, and Status. The table lists four applications: API-GATEWAY, AUTHORIZATION\_SERVER, EUREKA-SERVER, and PRODUTO-SERVICE. The PRODUTO-SERVICE application shows a status of 'DOWN (1)' for one instance.

Application	AMIs	Availability Zones	Status
API-GATEWAY	n/a (1)	(1)	UP (1) - Alisson-PC:api-gateway:8080
AUTHORIZATION_SERVER	n/a (1)	(1)	UP (1) - Alisson-PC:authorization_server:9092
EUREKA-SERVER	n/a (1)	(1)	UP (1) - Alisson-PC:eureka-server:9091
PRODUTO-SERVICE	n/a (2)	(2)	UP (1) - Alisson-PC:produto-service:8097 <b>DOWN (1)</b> - Alisson-PC:produto-service:8096

Usuário acessando uma página que utiliza o serviço produto-service após a parada de uma das instâncias.



The image shows a web browser window displaying a web application. The address bar shows 'localhost/?#/produto/fornecedores/'. The application has a sidebar with a search bar and a list of menu items: Produto, Segurança, Cliente, and Loja. The main content area is titled 'Fornecedores' and contains a table with columns: Código, Nome, CNPJ, Cidade, Telefone, Contato, Produtos, and Ações. The table lists two suppliers: 'Fazenda da Mata' and 'Fazenda Três Barras'.

Código	Nome	CNPJ	Cidade	Telefone	Contato	Produtos	Ações
1	Fazenda da Mata	1234567	Arcos	(37) 3351-1000	João da Mata	2	[Edit] [Delete]
10	Fazenda Três Barras	123242			sasAS	0	[Edit] [Delete]

Log do servidor da API Gateway após a parada de uma das instâncias.

```
2018-09-27 22:44:53.771 INFO 10648 --- [io-8080-exec-10] c.a.apigateway.filter.SimpleFilter : GET request to http://localhost:8080/produto-service/produto/fornecedores
2018-09-27 22:48:49.800 INFO 10648 --- [trap-executor-0] c.n.d.s.r.aws.ConfigClusterResolver : Resolving eureka endpoints via configuration
2018-09-27 22:48:56.875 INFO 10648 --- [nio-8080-exec-3] c.a.apigateway.filter.SimpleFilter : GET request to http://localhost:8080/produto-service/produto/fornecedores
```

## 6.5. Resultado

Considerando os requisitos não funcionais e os principais atributos de qualidade, é possível verificar que a arquitetura proposta atende as necessidades do projeto. Como o objetivo da prova de conceito era apenas validar a arquitetura, não foi investido muito tempo na criação de interfaces de usuário, mas foi possível verificar que com as tecnologias propostas, VueJS e Bootstrap, é possível criar

interfaces que proveem boa usabilidade e acessibilidade de forma produtiva e rápida, além de possibilitar a componentização e reaproveitamento de código.

A separação de código entre front-end e back-end, possibilita a paralelização de atividades o que pode reduzir consideravelmente o tempo de desenvolvimento. Outro fator que auxilia nesta divisão do trabalho de desenvolvimento é a divisão dos serviços do back-end em módulos. Diferentes desenvolvedores poderão trabalhar em paralelo nestes serviços, que ficarão também mais fáceis de testar e realizar manutenções.

A utilização de serviços coesos e com responsabilidades bem definidas facilita a reutilização dos mesmos por outras aplicações na organização, e a separação do serviço de autenticação em um servidor separado cria uma independência ainda maior, pois a autenticação e autorização passam a ser centralizadas, o que pode facilitar o controle da segurança.

A containerização dos serviços facilita a migração da aplicação para diferentes plataformas e o registro e descoberta de serviços, juntamente com a redundância de serviços, facilitam muito a manutenção e escalagem dos mesmos, aumentando a disponibilidade do sistema.

Na avaliação foi possível testar e homologar os principais atributos de qualidade considerados e analisando a arquitetura foi possível perceber alguns pontos de melhoria, como a modularização da interface de usuários. Foi considerada a separação dos serviços em módulos reutilizáveis em outras aplicações, como módulo de produtos e clientes, mas seria interessante separar também a interface de usuários de forma a facilitar sua reutilização. Mesmo apresentando pontos de melhoria a arquitetura proposta se mostrou interessante e com capacidade de facilitar o crescimento do projeto.

## **7. Conclusão**

O objetivo geral deste projeto era apresentar uma proposta arquitetural para um sistema de vendas na modalidade dropshipping para pequenos produtores rurais e integrantes da agricultura familiar. Acredito que o objetivo do projeto foi atendido, pois através da prova de conceito foi possível avaliar e validar os principais atributos de qualidade. O trabalho também possibilitou o entendimento de novas tecnologias e sua viabilidade para futuros projetos.

## **APÊNDICES**

URL do Github: <https://github.com/alissonrsousa/tcc-puc-asd>

URL da apresentação da POC no Youtube: <https://youtu.be/32Xm9wrXhDc>