**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**NÚCLEO DE EDUCAÇÃO A DISTÂNCIA**

**Pós-graduação Lato Sensu em Arquitetura de Software Distribuído**

**Marciano José da Anunciação**

**Sistema DropShipping**

**São Paulo**

**2018**

Marciano José da Anunciação

**Sistema de Vendas DropShipping**

Trabalho de Conclusão de Curso de Especialização

em Arquitetura de Software Distribuído como requisito parcial

à obtenção do título de especialista.

Orientador(a): Tadeu Faria

São Paulo 2018

Dedicatória

Agradecimentos

**Resumo**

Este projeto aborda a solução arquitetural de um sistema integrado para controle de comércio on-line de vendas de produtos na modalidade dropshipping através de dispositivos móveis e desktops.

O dropshipping é uma modalidade de vendas on-line voltados para e-commerce e empreendedores individuais de marketplaces que vem crescendo dia no cenário atual.

O processo consiste em efetuar as vendas através do sistema e encaminhar as vendas aos fornecedores, que por sua vez realizam o envio do produto aos consumidores.

O projeto está estruturado da seguinte forma: Levantamento dos requisitos, modelagem da arquitetura que será aplicada para sustentar o modelo de negócio, uma prova de conceito para avaliar os resultados e impactos da arquitetura proposta para atender os requisitos funcionais e não funcionais. Ao final do projeto é esperado que todos os requisitos sejam atendidos através da documentação e/ou prova de conceito.

Palavras-chave: Arquitetura de Software, integração, acadêmico, Dropshipping, Web, Dispositivos Móveis, Fornecedores, Monitoramento.

Sumário

1. Objetivos do trabalho

O objetivo geral desse projeto é apresentar uma proposta arquitetural para uma loja na modalidade dropshipping que se integre com os diferentes sistemas de fornecedores, que permita a utilização por vendedores clientes através de interface Web em plataformas Desktop e Mobile. O projeto arquitetural visa atender essa necessidade através do desenho e prova de conceito do sistema em questão, possibilitando que empreendedores individuais de MarketPlace possam disponibilizar os seus produtos para venda de uma maneira rápida e eficaz para todo o mundo através da Internet.

Os objetivos específicos são:

1. Criar um módulo de Controle de vendas, onde os clientes possam realizar suas compras pelo sistema Web ou por um aplicativo móvel. Vendedores possam utilizar smartphones ou notebooks para realizar vendas.

2. Criar integração entre a loja e os fornecedores, facilitando o cadastro de produtos e promoções, a gerencia do status de cada pedido e o eventual cancelamento da venda.

3. Fornecer um módulo de monitoramento e inteligência que auxiliará a área de inteligência da loja na tomada de decisões estratégicas para o crescimento do negócio.

4. Criar um módulo de controle dos pedidos para gerenciar a vida do pedido que esteja integrado com o sistema do forcenedor que gerará detalhes sobre o status de cada pedido.

5. Criar um módulo de controle de propagandas e promoções, este módulo será o responsável pela criação de propagandas e pelo cadastro de promoções pelo forncedor através de integração.

6. Criar um módulo de atendimento ao cliente que auxiliará o serviço de atendimento ao cliente no cadastro de dúvidas e gestão de reclamações por parte dos clientes.

**2.Descrição Geral da Solução**

2.1. Apresentação do problema

Muitos Vendedores autônomos não conseguem encontrar uma maneira simples e rápida de expandir seus negócios, o sistema de dropshipping visa resolver este problema trazendo estas empresas para o mundo on-line através de um e-commerce que fará com que esses empreendedores tornem os seus produtos e suas marcars visíveis para o mercado e tenham um maior alcance sem a necessidade de aberturas de lojas físicas espalhadas pelo pais e pelo mundo, a modalidade de dropshipping tras a esses empreendedores uma solução rápida através de uma integração com seus sistemas já existentes.

Por outro lado a loja dona do sistema dropshipping também tera vários benefícios com a solução conforme abaixo:

Gastos reduzidos: A loja não terá a necessidade de investir em produtos já que o estoque é todo do forncedor.

Poucas despesas fixas: Não há a necessidade de lidar com a gestão do estoque.

Variedade de produtos: A loja pode oferecer uma variedade quase que infinita de produtos necessitando apenas realizar o cadastro dos fornecedores, com isso aumentando a variedade de produtos.

**2.2 Descrição Geral da Solução**

A proposta arquitetural visa desenvolver um sistema modular, de facil implementação e que permita facilidade nos testes e facilite a vendedores de todas as modalidades a exporem os seus produtos on-line através de integração com o sistema dropshipping tornando a integração entre clientes e fornecedores algo simples.

Publico Alvo: Fornecedores de Produtos que queiram vender seus produtos através da internet e consumidores on-line de todas as modalidades.

**3. Definição Conceitual da Solução**

**3.1 Requisitos Funcionais**

**Módulo Controle de Vendas**

O sistema deve permitir que o cliente faça a compra pelo sistema web ou aplicativo disponível para smartphone.

O sistema deve permitir que vendedores externos efetuem vendas por smartphones ou laptops.

O sistema deve permitir a pesquisa de produtos pelos clientes através de sua interface visual.

O sistema deve comunicar cada venda realizada ao fornecedor do produto para que o mesmo realize a entrega.

O sistema deve apresentar um forma simples e rápida de cadastro e log-in para finalização da compra.

**Módulo do Fornecedor**

O sistema deve permitir que o módulo de Controle de Vendas realiza a pesquisa de disponibilidade dos produtos em estoque.

O sistema deve permitir a reserva do produto.

O sistema deve permitir a pesquisa do Status de cada pedido feito pela loja.

O sistema deve permitir o cancelamento da compra pela loja.

**Módulo de Controle de Pedidos**

O sistema deve permitir Pesquisar o Status da Entrega a qualquer momento pelo cliente ou SAC da loja.

O sistema deve permitir o cancelamento da entrega pelo cliente através de integração com a loja.

**Módulo SAC**

O sistema deve permitir que o atendente do SAC pesquise pedidos pelo ID do Cliente ou número do pedido.

O sistema deve permitir que o atendente da loja possa registrar, alterar e exlcuir reclamações de clientes.

O sistema deve permitir que o atendente da loja possa registrar dúvidas, altera-las e responde-las.

**Módulo Gerencial**

O sistema deve gerar informações sobre as vendas dos produtos, cancelamentos e lista de desejo para que estas informações sejam utilizadas pela área de inteligência de negócio da loja.

O sistema deve gerar informações sobre as entregas como: extravios, quantidade de entregas realizadas dentro e for a do tempo limite, encomendas violadas.

O sistema deve gerar informações sobre as avaliações dos clientes separando as informações por produtos e forecedores para que essa informação seja utilizada pela área de inteligência da loja.

**Módulo de Inteligência de Négocios**

O sistema deve utilizar as informações sobre vendas geradas pelo módulo Gerencial para monitorar e auxiliar a área de Inteligência de negócios na tomadas de decisões ligadas as vendas para gerar promoções e propagandas para alavancar as vendas na loja.

O sistema deve utilizar as informações geradas pelo módulo Gerencial para monitorar as entregas e verificar o processo de cada pedido feito para avaliar e auxiliar cadas fornecedor na melhoria do processo de entrega.

O sistema deve gerar relatórios de vendas para auxiliar a área de Inteligência de Negócios na tomada de decisões para alavancar o crescimento da loja.

O sistema deve gerar relatórios de produtos mais e menos vendidos, produtos com maior avaliação e produtos com avaliações ruins.

**Módulo de Gestão de Propaganda e Promoções**

O sistema deve permitir que sejam criadas, atualizadas e excluídas propagandas que serão mostradas para os clientes de acordo com seu perfil.

O sistema deve permitir que sejam criadas, atualizadas e excluídas promoções que serão mostradas a todos os clientes da loja.

**Módulo de Administração**

O sistema deve permitir que o Administrador do sistema possa cadastrar, alterar e excluir usuários da loja.

O sistema deve permitir que o Administrador possa cadastrar, pesquisar, alterar e excluir fornecedores da loja.

O sistema deve permitir que o Administrador possa criar, pesquisar, alterar e excluir perfis referentes a loja.

**3.2 Requisitos Não-Funcionais**

A seguir são apresentadas os requisitos não funcionais do sistema:

Usabilidade – O sistema deve prover boa usabilidade

|  |  |
| --- | --- |
| **Estímulo** | Cliente realizando compra na loja |
| **Fonte de Estímulo** | Cliente realizando todo o fluxo de compra de um produto, desde a sua seleção até a finalização da compra |
| **Ambiente** | Funcionamento, carga normal |
| **Artefato** | Módulo Controle de Vendas |
| **Resposta** | A camada de apresentação é de facil uso através de uma navegação intuitiva, simples e objetiva |
| **Medidas da resposta** | Usuário consegue realizar uma compra em tempo médio de 3 minutos |

Responsivo - O sistema deve suportar ambientes Web responsívos e ambientes móveis.

|  |  |
| --- | --- |
| **Estímulo** | Cliente acessando site da loja de dispositivos moveis e desktop |
| **Fonte de Estímulo** | Cliente realizando a pesquisa de um produto em ambiente Web e Ambiente móvel |
| **Ambiente** | Funcionamento, carga normal |
| **Artefato** | Módulo Controle de Vendas |
| **Resposta** | O cliente conseguiu realizar a pesquisa nos dois dispositivos sem mudança aparente de interface |
| **Medidas da resposta** | Cliente conseguiu realizar a pesquisa do mesmo produto no ambiente Web e móvel sem detectar mudanças na interface, o sistema conseguiu se adaptar aos dois ambientes |

Performance – O sistema deve ser rápido

|  |  |
| --- | --- |
| **Estímulo** | Cliente realizando pesquisa de produtos na loja |
| **Fonte de Estímulo** | Cliente realizando pesquisa de produtos diversos na loja |
| **Ambiente** | Funcionamento, carga normal |
| **Artefato** | Módulo Controle de Vendas |
| **Resposta** | Todas as consultas feitas pelo cliente retornaram no tempo estipulado |
| **Medidas da resposta** | Todas as pesquisas realizadas retornam em menos de um segundo |

Manutebilidade - O sistema deve apresentar manutenção facilitada

|  |  |
| --- | --- |
| **Estímulo** | Manutenção do Carrinho de Compras |
| **Fonte de Estímulo** | Desenvolvedor realizando manutenção em funcionalidade do Módulo Controle de Compras |
| **Ambiente** | Funcionamento, carga normal |
| **Artefato** | Módulo Controle de Vendas |
| **Resposta** | A manutenção foi realizada pelo desenvolvedor sem afetar outras partes do sistema |
| **Medidas da resposta** | Desenvolvedor conseguiu realizar a manutenção de forma rápida e assertiva devido a componentização do sistema em módulos |

Testabilidade – O sistema deve ser simples de Testar

|  |  |
| --- | --- |
| **Estímulo** | Testes unitários e Integrados |
| **Fonte de Estímulo** | Teste do sistema ao entregar novas funcionalidades |
| **Ambiente** | Funcionamento, carga normal |
| **Artefato** | Módulo Controle de Vendas |
| **Resposta** | O sistema foi testado totalmente de forma automatizada e simples |
| **Medidas da resposta** | As rotinas de teste do sistema foram executadas de forma automatizadas a cada 30 minutos |

Interoperabilidade – O Sistema deve se comunicar com os sistemas dos fornecedores independente das tecnologias utilizadas por eles.

|  |  |
| --- | --- |
| **Estímulo** | Verificar a disponibilidade de um Produto |
| **Fonte de Estímulo** | Cliente finalizando a compra através da loja, antes de finalizar o pedido a loja realiza a integração com o fornecedor para verificar a disponibilidade do produto. |
| **Ambiente** | Funcionamento, carga normal |
| **Artefato** | Módulo Controle de Vendas |
| **Resposta** | A integração foi realizada tanto em sistemas modernos como sistemas legados |
| **Medidas da resposta** | Todas integrações foram realizadas e todas as vendas foram finalizadas pelo cliente |

Disponibilidade – O sistema deve operar a qualquer hora do dia e da noite mantendo a disponibilidade 24 horas por dia e sete dias na semana

|  |  |
| --- | --- |
| **Estímulo** | Cliente realizando log-in no sistema |
| **Fonte de Estímulo** | Cliente acessando a página inicial do sistema e realizando log-in na loja |
| **Ambiente** | Funcionamento, carga normal |
| **Artefato** | Módulo de Segurança |
| **Resposta** | O sistema foi acessado em vários horários do dia e da noite e em vários dias diferentes da semana pelo cliente e em todos os acessos foi possível realizar o log-in devido ao sistema de balanceamento e elasticidade criado para suportar quantidade de acessos altas e de se recuperar de qualquer tipo de falha de maneira automática |
| **Medidas da resposta** | Cliente conseguiu realizar o log-in na loja em todos os dias da semana e em todos os horários do dia |

Segurança – O sistema deve apresentar altos padrões de segurança

|  |  |
| --- | --- |
| **Estímulo** | Integração entre a loja e os sistemas dos fornecedores, acesso aos pedidos do cliente |
| **Fonte de Estímulo** | Todos os acessos aos modulos dos sistemas(APIs) devem ser feitos de forma segura com autenticação e autorização através da validação de cada perfil de acesso ao sistema, para cada acesso seja pelo cliente ou sistemas internos e externos, os usuários do sistema devem ter seus respectivos usuários cadastrados e ativos no sistema, a cada acesso o sistema deve gerar um token exclusivo para cada usuário, este token deve ser gerado a cada acesso autenticado dos usuários do sistema, o token deve expirar sempre que o usuário ficar inativo no sistema por mais de 5 minutos |
| **Ambiente** | Funcionamento, carga normal |
| **Artefato** | Módulo de Segurança e Módulo do Fornecedor |
| **Resposta** | Todos os acessos dos usuários só foram possíveis com autenticação |
| **Medidas da resposta** | Todos os acessos foram realizados com autenticação e todos os usuários tiveram seus tokens exclusivos gerados e ativos durante os 5 minutos definidos |

**3.3 Restrições Arquiteturais**

O sistema deve ser desenvolvido em Java

O sistema deve abrir de forma responsiva em aparelhos menores, como celular e tablet

O sistema deve ser modular para facilitar os testes, a manutenção e a implantação

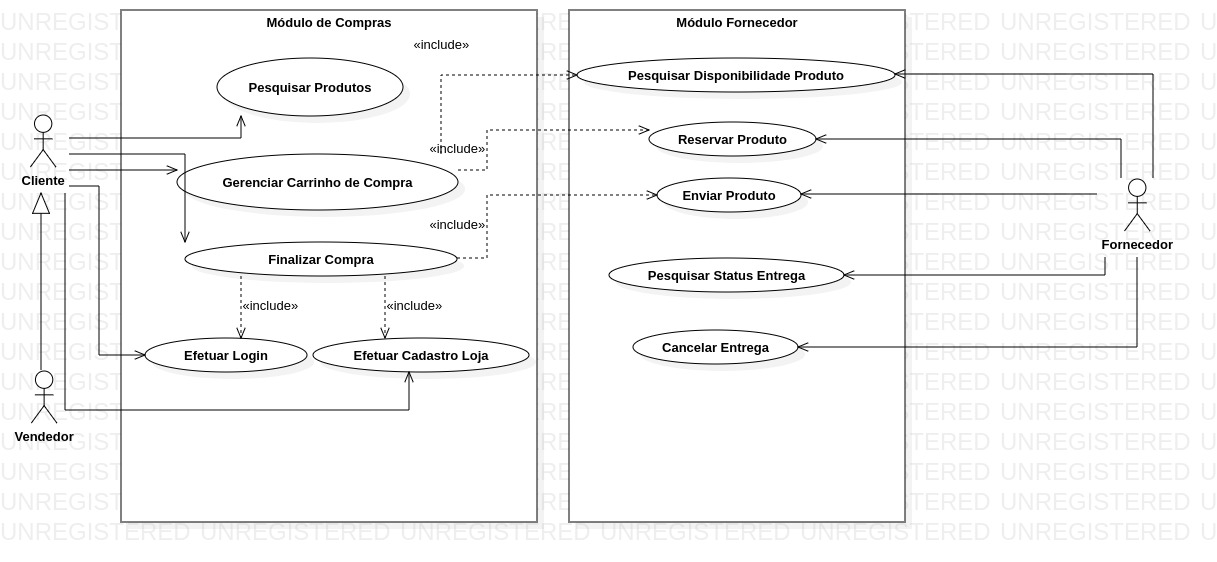
**3.4 Mecanismos Arquiteturais**

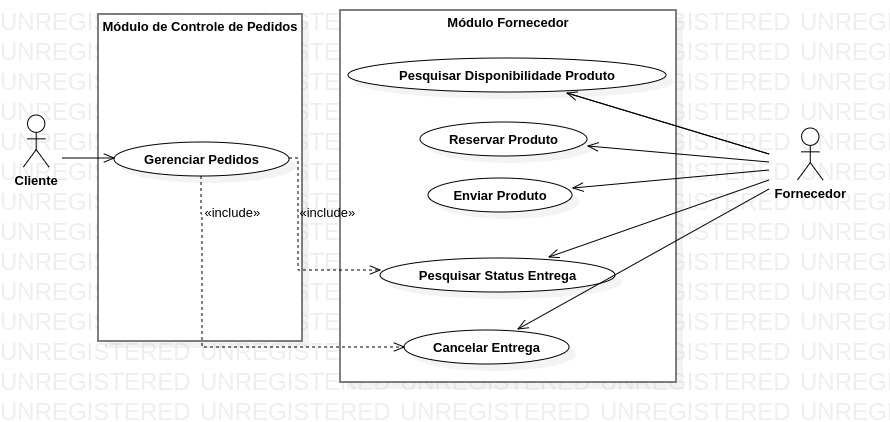
|  |  |  |
| --- | --- | --- |
| **Mecanismo de Análise** | **Mecanismod de Design** | **Mecanismo de Implementação** |
| Comunicação entre Processos | Contêiner Web e Aplicação | Docker |
| Integração com outros módulos e/ou sistemas | Interfaces Utilizando XML ou JSON | WebServices e WebAPI |
| ESB | Integração através de Mensagens | Spring Integration e ActiveMQ |
| Log | Framework de Log | SLF4J |
| Cache | Framework de Cache | Redis e Hibernate |
| Build | Geração de Artefatos para servidores de aplicação | Maven |
| Deploy | Deploy da Aplicação no servidor e testes automatizados e de integração | Jenkins |
| Front-End | Interface de Comunicação com o usuário do sistema | Angular e Bootstrap |
| Versionamento | Versionamento do código fonte da aplicação | GIT/GitHub |
| Autenticação e Autorização | Verificação de credenciais e tentativas de acesso a funcionalidades restritas e validação de perfis | Oauth e JWT |
| Alta Disponibilidade | Balanceamento de carga dos serviços | Ribbon(Spring Cloud Netflix OSS) |
| Descoberta de Serviços | Registro e Descoberta de serviços | Eureka(Spring Cloud Netflix OSS) |
| Tolerancia a Flahas | Failover para chamadas entre micro serviços | Hystrix(Spring Cloud Netflix OSS) |
| Segurança | Segurança de Apis | Spring Cloud Security |
| Sistema Operacional | Sistema que será executado nos servidores | Docker |

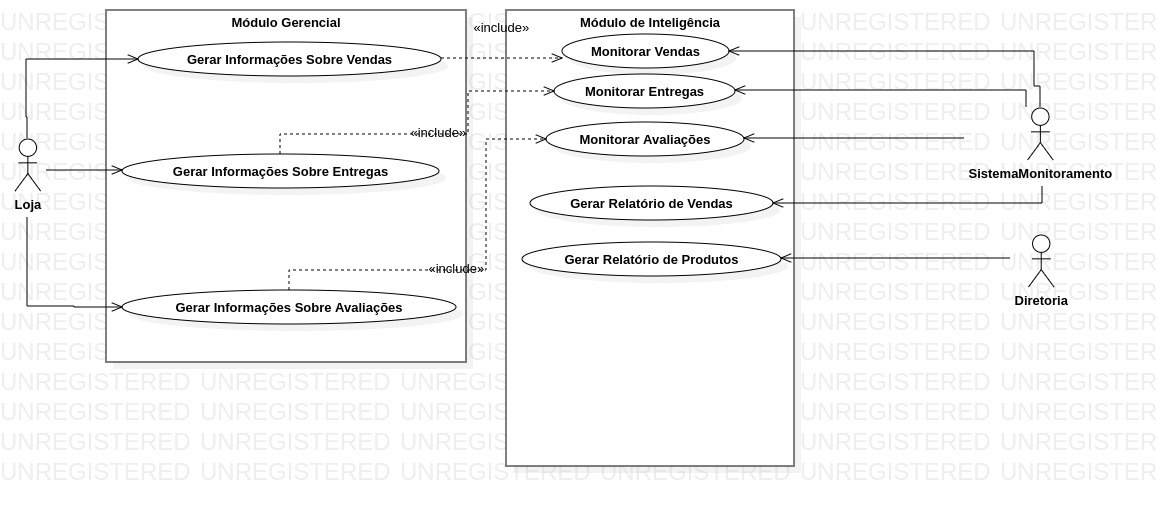
**4. Modelagem e Projeto Arquitetural**

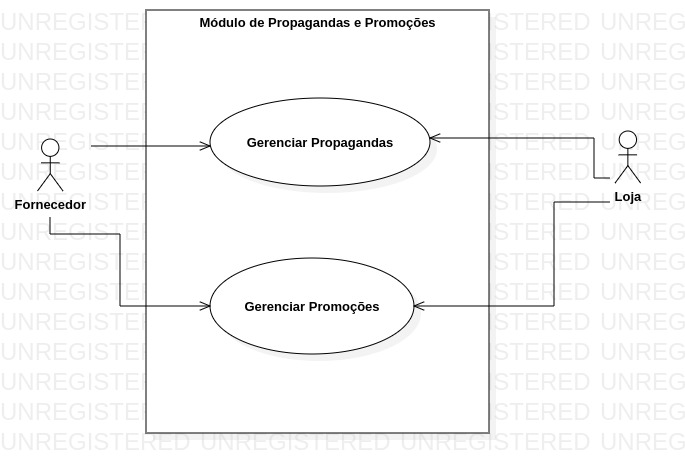
**4.1 Modelo de Casos de Uso**

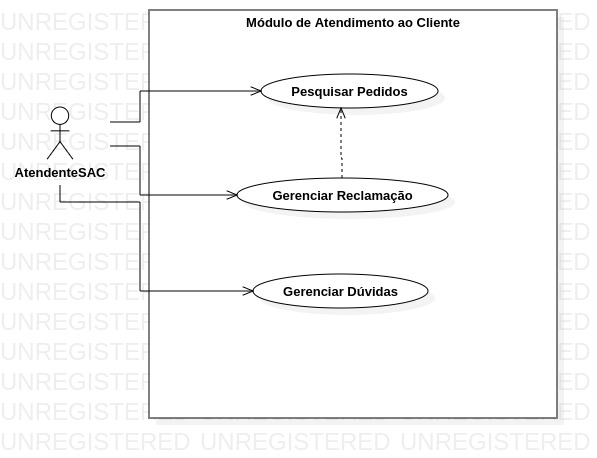
Os diagramas de caso de uso abaixo visam demonstrar as funções do sistema e as informações pertinentes a a análise e design da solução, os diagramas estão separados por módulos de acordo com seus requisitos para facilitar o entendimento e análise do design arquitetural.

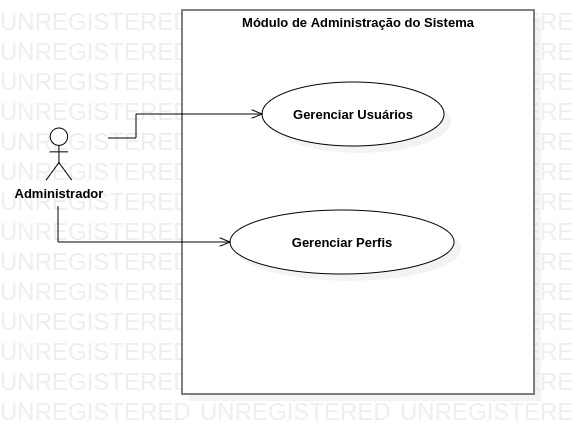












**4.2. Descrição resumida dos casos de uso**

**Estória**

|  |  |
| --- | --- |
| **ID** | 01 |
| **Estória do usuário** | Pesquisar Produtos |
| **Criador** | Como cliente da loja |
| **Descrição** | Eu quero consultar produtos pelo nome ou tipo |
| **Valor do negócio** | Para visualizar produtos disponíveis para venda na loja |
| **Prioridade** | Média |
| **Estimativa** | 2 |

|  |  |
| --- | --- |
| **ID** | 02 |
| **Estória do usuário** | Gerenciar Carrinho de Compras |
| **Criador** | Como cliente da loja |
| **Descrição** | Eu quero gerenciar meu carrinho de compras incluindo, atualizando quantidade e excluindo pedidos |
| **Valor do negócio** | Para selecionar itens para compra na loja |
| **Prioridade** | Alta |
| **Estimativa** | 8 |

|  |  |
| --- | --- |
| **ID** | 03 |
| **Estória do usuário** | Finalizar Compra |
| **Criador** | Como cliente da loja |
| **Descrição** | Eu quero realizar o pagamento na loja escolhendo a forma de pagamento e realizando a compra dos itens no meu carrinho de compra |
| **Valor do negócio** | Realizar compras na loja |
| **Prioridade** | Alta |
| **Estimativa** | 10 |

|  |  |
| --- | --- |
| **ID** | 04 |
| **Estória do usuário** | Efetuar Login |
| **Criador** | Cliente da Loja |
| **Descrição** | Eu quero realizar login com meu usuário e senha previamente cadastrados no sistema, caso esqueça a senha quero que a loja tenha um mecanismo de recuperação da senha caso necessário |
| **Valor do negócio** | Realizar autenticação na loja |
| **Prioridade** | Alta |
| **Estimativa** | 08 |

|  |  |
| --- | --- |
| **ID** | 05 |
| **Estória do usuário** | Efetuar Cadastro na loja |
| **Criador** | Cliente da loja |
| **Descrição** | Eu quero realizar o cadastro na loja com meus dados como nome, endereço, e-mail e senha |
| **Valor do negócio** | Realizar o cadastro na loja |
| **Prioridade** | Média |
| **Estimativa** | 5 |

|  |  |
| --- | --- |
| **ID** | 06 |
| **Estória do usuário** | Pesquisar disponibilidade do produto |
| **Criador** | Sistema de controle de compras |
| **Descrição** | Eu quero checar a disponibilidade dos produtos sempre que um cliente se interessar pelo produto |
| **Valor do negócio** | Realizar a checagem da disponibilidade do produto antes de vende-lo ao cliente |
| **Prioridade** | Média |
| **Estimativa** | 04 |

|  |  |
| --- | --- |
| **ID** | 07 |
| **Estória do usuário** | Reservar Produto |
| **Criador** | Sistema de controle de compras |
| **Descrição** | Eu quero reservar o produto sempre que um cliente inseri-lo em seu carrinho de compras |
| **Valor do negócio** | Reservar o produto para que a entrega para o cliente seja garantida |
| **Prioridade** | Média |
| **Estimativa** | 6 |

|  |  |
| --- | --- |
| **ID** | 08 |
| **Estória do usuário** | Enviar Produto |
| **Criador** | Sistema de controle de compras |
| **Descrição** | Eu quero enviar o produto ao cliente sempre que uma venda seja finalizada no sistema |
| **Valor do negócio** | Enviar o produto ao cliente |
| **Prioridade** | Média |
| **Estimativa** | 06 |

|  |  |
| --- | --- |
| **ID** | 09 |
| **Estória do usuário** | Pesquisar status da entrega |
| **Criador** | Sistema de controle de compras |
| **Descrição** | Eu quero pesquisar o status de cada compra feita na loja para acompanhar o pedido e informa o cliente de cada atualização |
| **Valor do negócio** | Acompranha o status do pedido para informar o cliente |
| **Prioridade** | Média |
| **Estimativa** | 5 |

|  |  |
| --- | --- |
| **ID** | 10 |
| **Estória do usuário** | Cancelar entrega |
| **Criador** | Sistema de controle de compras |
| **Descrição** | Eu quero realizar o cancelamento da entrega sempre que o pagamento do cliente não for aprovado ou o cliente resolver cancelar o produto na loja |
| **Valor do negócio** | Cancelar a entrega do produto ao cliente |
| **Prioridade** | Média |
| **Estimativa** | 5 |

|  |  |
| --- | --- |
| **ID** | 11 |
| **Estória do usuário** | Gerenciar Pedidos |
| **Criador** | Cliente da Loja |
| **Descrição** | Eu quero realizar a pesquisa, alteração ou cancelamento do pedido.  Também quero realizar a pesquisa de status de dos meus pedidos para me manter atualizado sobre a entrega |
| **Valor do negócio** | Gerenciar os pedidos realizados na loja |
| **Prioridade** | Média |
| **Estimativa** | 7 |

|  |  |
| --- | --- |
| **ID** | 12 |
| **Estória do usuário** | Gerar informações sobre vendas |
| **Criador** | Sistema gerencial |
| **Descrição** | Eu quero gerar informações sobre vendas do sistema como: produtos mais vendidos, produtos menos vendidos e produtos com maior e menor avaliação |
| **Valor do negócio** | Gerar informações gerenciais sobre vendas |
| **Prioridade** | Média |
| **Estimativa** | 5 |

|  |  |
| --- | --- |
| **ID** | 13 |
| **Estória do usuário** | Gerar informações sobre entregas |
| **Criador** | Sistema Gerencial |
| **Descrição** | Eu quero gerar informações sobre todas as entregas da loja como: encomendas entregues no prazo, encomendas extraviadas, encomendas devolvidas e fornecedores com maior e menor taixa de sucesso nas entregas |
| **Valor do negócio** | Gerar informações gerenciais sobre entregas |
| **Prioridade** | Média |
| **Estimativa** | 5 |

|  |  |
| --- | --- |
| **ID** | 14 |
| **Estória do usuário** | Gerar informações sobre avaliações |
| **Criador** | Sistema Gerencial |
| **Descrição** | Eu quero gerar informações sobre as avaliações feitas na loja pelos clientes sobre fornecedores com melhor e pior avaliação. |
| **Valor do negócio** | Gerar informações gerencias sobre avaliações |
| **Prioridade** | Média |
| **Estimativa** | 5 |

|  |  |
| --- | --- |
| **ID** | 15 |
| **Estória do usuário** | Monitorar vendas |
| **Criador** | Sistema de monitoramento |
| **Descrição** | Eu quero monitorar todas as informações geradas sobre vendas no dia, no mês e no ano |
| **Valor do negócio** | Monitorar vendas no sistema |
| **Prioridade** | Média |
| **Estimativa** | 5 |

|  |  |
| --- | --- |
| **ID** | 16 |
| **Estória do usuário** | Monitorar entregas |
| **Criador** | Sistema de monitoramento |
| **Descrição** | Eu quero monitorar todas as informações geradas sobre entregas feitas no dia, no mês e no ano |
| **Valor do negócio** | Monitorar entregas no sistema |
| **Prioridade** | Média |
| **Estimativa** | 5 |

|  |  |
| --- | --- |
| **ID** | 17 |
| **Estória do usuário** | Monitorar avaliações |
| **Criador** | Sistema de monitoramento |
| **Descrição** | Eu quero monitorar todas as avaliações feitas na loja por dia, por mês e por ano para tomar decisões relacionadas aos fornecedores da loja |
| **Valor do negócio** | Monitorar avaliações no sistema |
| **Prioridade** | Média |
| **Estimativa** | 5 |

|  |  |
| --- | --- |
| **ID** | 18 |
| **Estória do usuário** | Gerar relatório de vendas |
| **Criador** | Sistema de Monitoramento |
| **Descrição** | Eu quero gerar relatórios sobre as vendas realizadas na loja, esses relatórios serão diário, mensais e anuais |
| **Valor do negócio** | Gerar relatórios sobre as vendas na loja |
| **Prioridade** | Média |
| **Estimativa** | 6 |

|  |  |
| --- | --- |
| **ID** | 19 |
| **Estória do usuário** | Gerar relatório de produtos |
| **Criador** | Mesa Diretora |
| **Descrição** | Eu quero gerar relatórios sobre os produtos como: produtos mais e menos vendidos, produtos com melhor e pior avalização e produtos com mais e menos rentabilidade na loja, esses relatórios nos auxiliarão na tomada de decisões estratégicas na compania |
| **Valor do negócio** | Gerar relatórios de produtos para a mesa diretora |
| **Prioridade** | Média |
| **Estimativa** | 6 |

|  |  |
| --- | --- |
| **ID** | 20 |
| **Estória do usuário** | Gerenciar propagandas |
| **Criador** | Loja e Fornecedor |
| **Descrição** | Eu quero realizar a inclusão, atualização e exclusão de propagandas sobre novos fornecedores e novos produtos |
| **Valor do negócio** | Realizar propagandas na loja |
| **Prioridade** | Baixa |
| **Estimativa** | 4 |

|  |  |
| --- | --- |
| **ID** | 21 |
| **Estória do usuário** | Gerenciar promoções |
| **Criador** | Loja e Fornecedor |
| **Descrição** | Eu quero realizar a inclusão, a atualização e a exclusão de promoções sobre produtos |
| **Valor do negócio** | Realizar promoções na loja |
| **Prioridade** | Baixa |
| **Estimativa** | 4 |

|  |  |
| --- | --- |
| **ID** | 22 |
| **Estória do usuário** | Pesquisar Pedidos |
| **Criador** | Atendente SAC |
| **Descrição** | Eu quero realizar a pesquisa de pedidos pelo número do pedido ou CPF/CNPJ do cliente |
| **Valor do negócio** | Pesquisar pedidos para auxiliar os clientes da loja |
| **Prioridade** | Média |
| **Estimativa** | 4 |

|  |  |
| --- | --- |
| **ID** | 23 |
| **Estória do usuário** | Gerenciar reclamação |
| **Criador** | Atendente SAC |
| **Descrição** | Eu quero realizar a inclusão, atualização e exclusão de reclamações feitas pelos clientes através dos canais de comunicação da loja |
| **Valor do negócio** | Gerenciar as reclamações feitas na loja |
| **Prioridade** | Média |
| **Estimativa** | 5 |

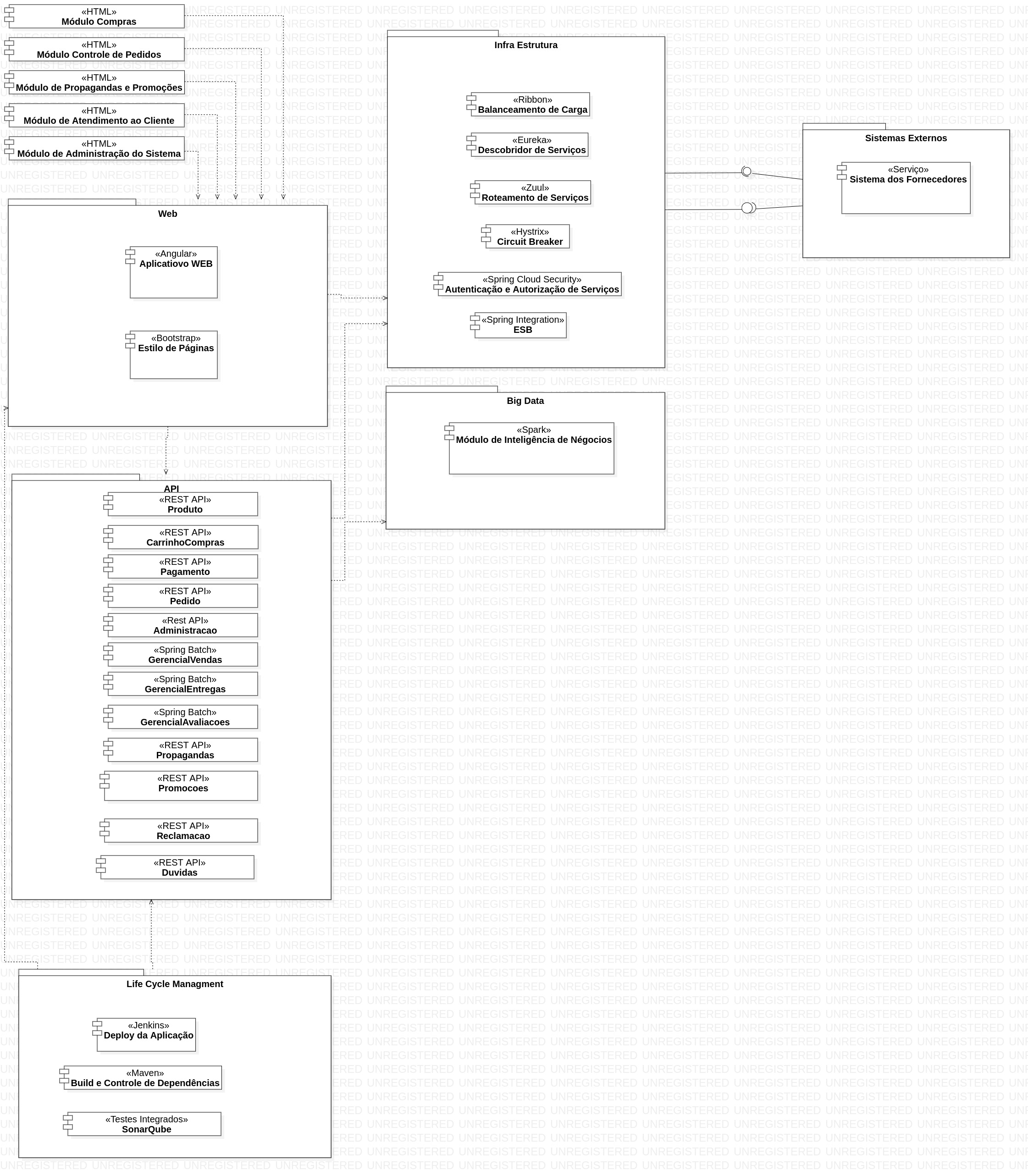
|  |  |
| --- | --- |
| **ID** | 24 |
| **Estória do usuário** | Gerenciar dúvidas |
| **Criador** | Atendente SAC |
| **Descrição** | Eu quero realizar a inclusão, atualização e resolução de dúvidas feitas pelos clientes nos canais da loja |
| **Valor do negócio** | Auxiliar clientes da loja |
| **Prioridade** | Média |
| **Estimativa** | 5 |

|  |  |
| --- | --- |
| **ID** | 25 |
| **Estória do usuário** | Gerenciar usuários |
| **Criador** | Administrador |
| **Descrição** | Eu quero incluir, atualizar e excluir usuários no sistema da loja através de interface WEB |
| **Valor do negócio** | Gerenciar usuários do sistema da loja |
| **Prioridade** | Alta |
| **Estimativa** | 6 |

|  |  |
| --- | --- |
| **ID** | 26 |
| **Estória do usuário** | Gerenciar perfis |
| **Criador** | Administrador |
| **Descrição** | Eu quero realizar a inclusão, atualização e exclusão de perfis de autorização do sistema criando e decidindo quais niveis de informação cada perfil poderá acessar através de interface WEB |
| **Valor do negócio** | Administrar perfis no sistema |
| **Prioridade** | Alta |
| **Estimativa** | 7 |

4.3. Modelo de componentes

Abaixo diagramas de componentes do sistema que proporcionam uma visão de cada componente arquitetural e sua organização mostrando suas interfaces e definições de cada componente.



Descrição dos componentes:

Web: Componente responsável por toda a camada de apresentação e interface com o usuário.

API: Componentes que agregam todos os serviços expostos pela aplicação através de Interfaces REST.

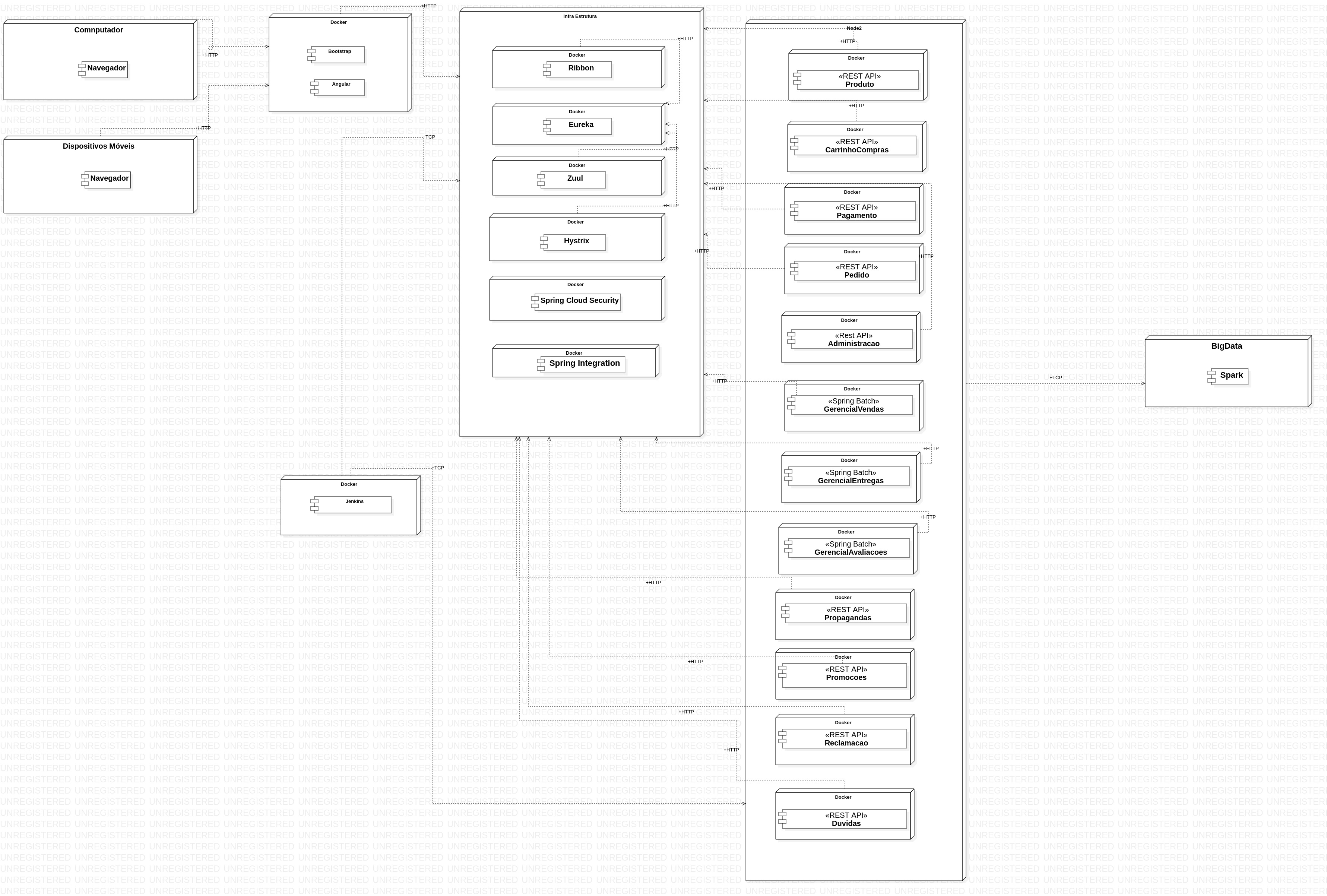
Infra Estrutura: Camada que agrega toda a infra estrutura do sistema como exemplo Gateway de APIs, Balanceamento de Carga, Circuit Breaker, Camada de Integração, Segurança, Roteador de Serviço e Descobridor de Serviços.

Life Cycle Management: Camada que agrega todos os componentes responsáveis pela Integração Continua do sistema.

Big Data: Camada responsável por tratar com os dados históricos gerados por todas as aplicações do sistema.

Sistemas Externos: Sistemas dos fornecedores do sistema de dropshipping.

4.4. Modelo de implantação



Abaixo a relação de componentes do sistema e a explicação de cada um deles, não há relação com servidores fisicos devido a utilização do docker para conteinirazação das máquinas garantindo o funcionamento de cada componente do sistema independente de hardware e sistema operacional disponibilizado pelo provedor de nuvem a ser contratado.

O sistema é baseado em Micro Serviços que possibilitam a facil implantação, manutebilidade pois cada serviço tem sua responsabilidade e facil entendimento devido o tamanho de cada serviço.

Os Micro Serviços serão implantados na nuvem para proporcionar alta escalabilidade e alta disponibilidade proporcionando um ambiente totalmente disponivel para compras durante o dia e noite e com alta performance devido o escalonamento que permite que nenhum serviço fique sobrecarregado pela quantidade de requisições.

|  |  |
| --- | --- |
| Componente | Descrição |
| Navegador | Browser Utilizado pelos usuários para realizar a conexão com o sistema de dropshipping, será utilizado em um computador ou dispositivo móvel. |
| Servidor de Integração Contínua | O Jenkins será o responsável por realizar o gerenciamento de dependência, contrução e implantação das APIs do sistema. |
| Container | O Docker será responsável por toda a virtualização e infraestrutura necessária para que cada componente tenha seu funcionamento garantido independente do ambiente que será instalado. |

5. Prova de conceito / protótipo arquitetural

A prova de conceito visa testar se a arquitetura proposta atende todas as necessidades e requisitos não funcionais do sistema de dropshipping, visando gerar provas que todas as necessidades dos clientes estão sendo atendidas.

A prova de conceito testará os seguintes requisitos:

Segurança – O sistema deve apresentar altos padrões de segurança.

Requisito extremamente importante pois o sistema lida com dados sensiveis de usuários, o sistema deve garantir que estes dados não sejam expostos a terceiros e deve garantir também que todo o acesso a esses dados seja feito por pessoas autenticadas e autorizadas no sistema.

Os Critérios são:

Não permitir acesso a páginas privadas sem a autorização e autenticação prévia.

Não permitir a navegação em qualquer área segura do sistema sem a prévia geração de um token de acesso.

Obrigar a atualização do token ao usuário após um período de tempo inativo no sistema.

Usabilidade - O sistema deve prover boa usabilidade.

Requisito que garante a facil navegação e facil entendimento do sistema, através de uma navegação simples e objetiva.

Os critérios são:

A tela deve apresentar facilidade de navegação

Cada tela deve mostrar ao usuário de forma clara qual funcionalidade é representada por ela.

O fluxo de navegação deve ser simples e objetivo.

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

O requisito visa a garantir que os cleintes e vendedores consigam utilizar o sistema de forma

transparente em dispositivos móveis e computadores.

Os critérios são:

A tela do sistema deve adaptar todos os seus componentes a navegação em todos os dispositivos móveis e computadores.

O sistema deve ser compatível com todos os browsers de mercado.

O sistema deve manter o padrão independente do dispositivo utiliado.

• Performance - O sistema deve ser rápido.

Esse requisito visa garantir um sistema com performance adequado através da customização de cada serviço e a separação de responsabilidades e escalabilidade dos serviços, com serviços menores e de facil escalonamento nenhum serviço chamado enfrentará gargalos ou concorrência em suas chamadas.

Os critérios de aceite são:

• Nenhum serviço chamado deve ultrapassar o tempo de 5 segundos.

• As telas não devem demorar mais que 5 segundos para serem carregadas.

5.2. Interfaces

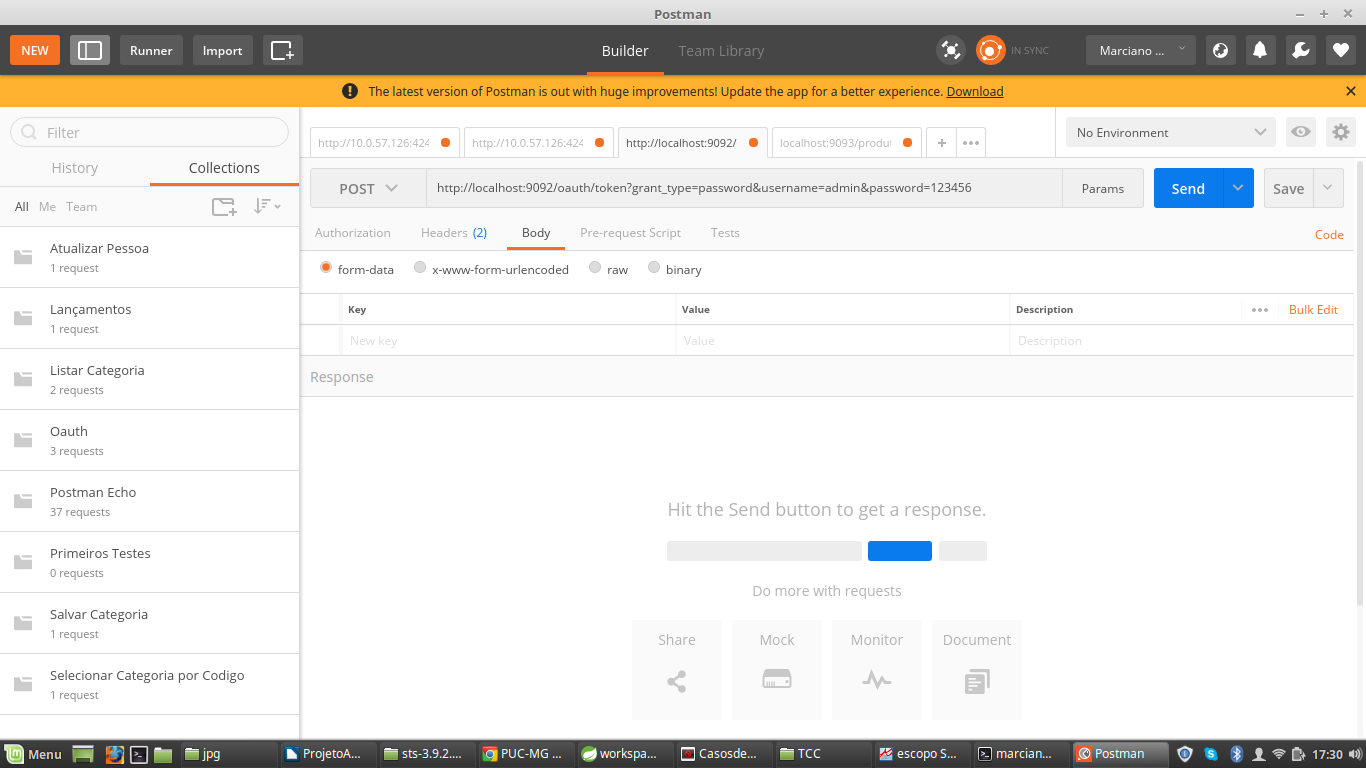
Sessão 1: Interface Serviço de Segurança.

Para o acesso da interface é necessário os dados abaixo:

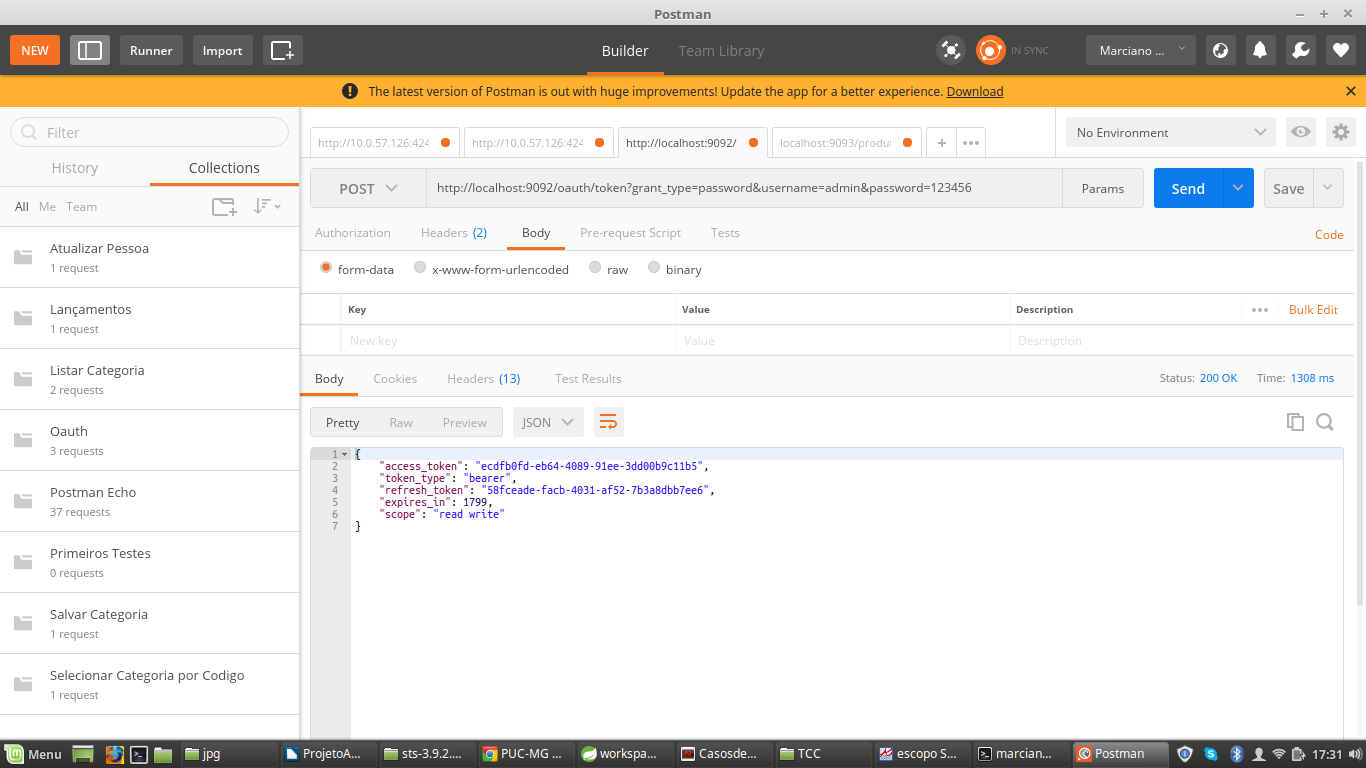
|  |  |  |
| --- | --- | --- |
| Usuário | Senha | Autorização |
| admin | 123456 | Basic Y29kZXJlZjokMmEkMTAkcDlQazBmUU5BUVNlc0k0dnV2S0EwT1phbkREMg== |

Ao realizar a chamada com os dados acima um token é retornado, esse token é necessário para realizar o acesso e chamada as demais APIs do sistema.

Exemplo:



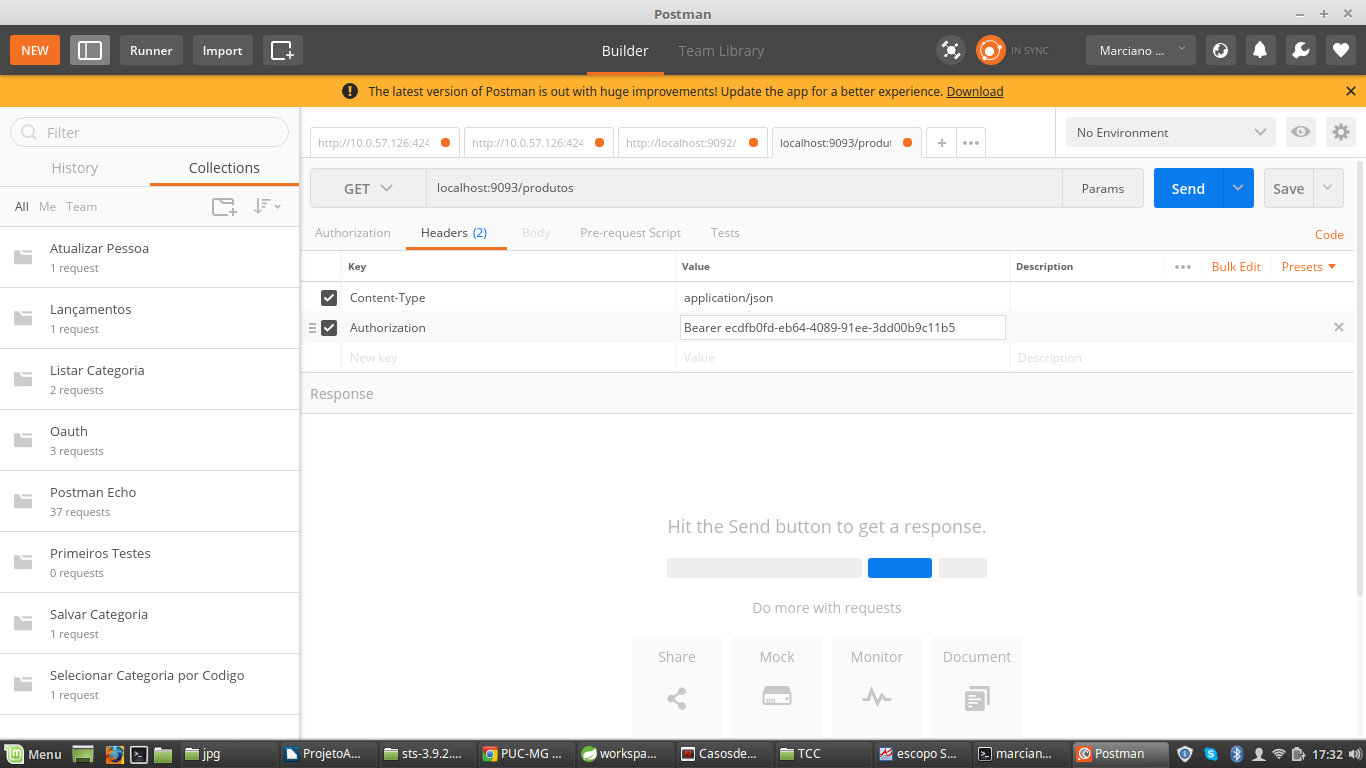
Token sendo retornado no ResponseBody:

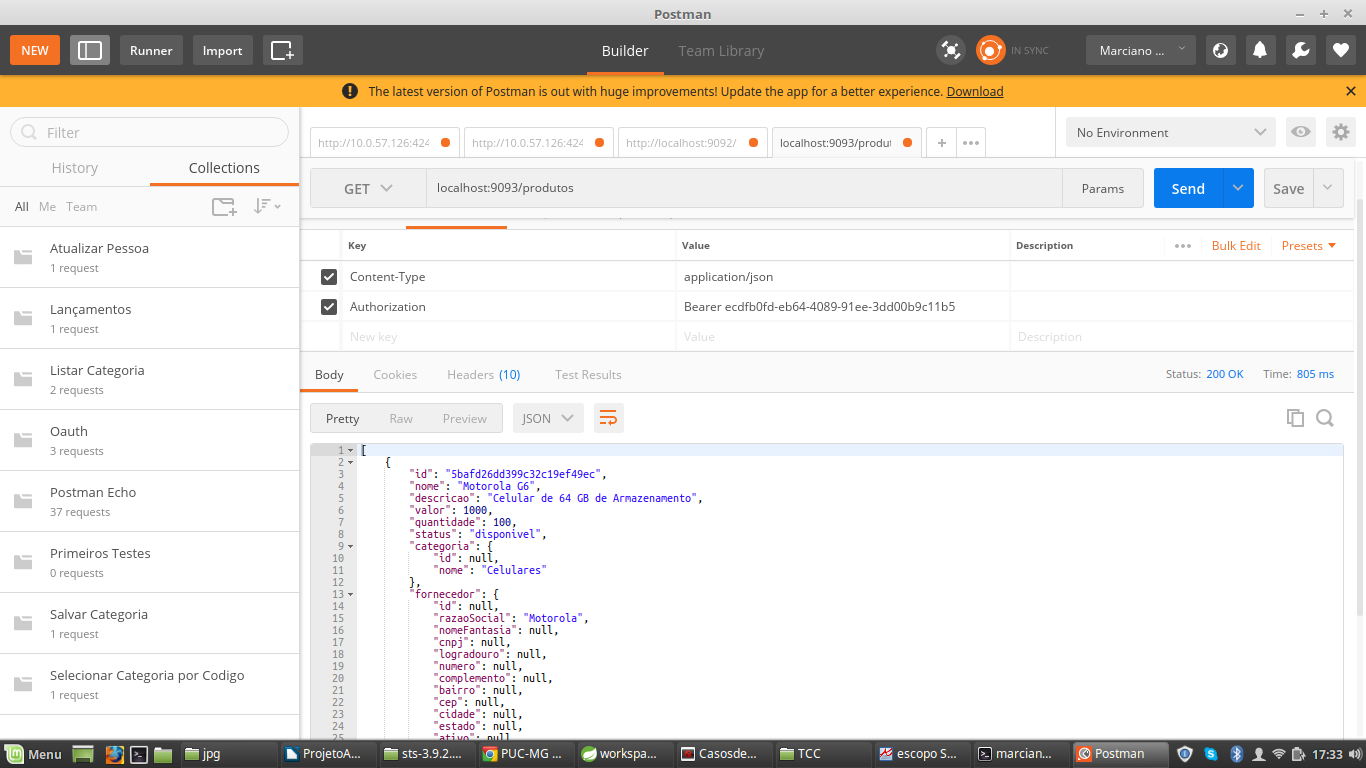


Após a geração do token o acesso as APIs pode ser realizado.

Exemplo:

API de pesquisa de Produtos.





6. Avaliação da Arquitetura

6.1. Análise das abordagens arquiteturais

A arquitetura provou ser uma arquitetura que facilita o desenvolvimento por times multidisciplinares dando aos desenvolvedores facilidade de manutenção e velocidade no desenvolvimento de novas APIs através do Padrão de MicroServiços.

A arquitetura também prove altos padrões de segurança através do uso de Autenticação e Autorização e Tokenirização atraves do Oauth 2 e JWT, a segurança por ter sido implementada em um componente separado facilita a adoção por novas APIs com a necessidade apenas de uma classe de configuração para que a nova funcionalidade utilize essa API de segurança.

A utilização das bibliotecas da Netflix OSS trouxe segurança no desenvolvimento com Spring Cloud através de uma vasta documentação de bibliotecas que são extremamente utilizadas no mercado, possibilitando através de anotação e pouca configuração prover funcionalidades avançadas as APIs como Gateway, Circuit Breaker, Balanceamento e Service Discovery.

6.2. Identificação dos atributos de qualidade

Os atributos identificados estão relacionados aos requisitos listados na seção anterior: Segurança, Desempenho e Performance e Usabilidade.

6.2. Cenários

Cenário 1: Ao realizar o acesso a uma API, o sistema deve apresentar padrões de segurança necessários, garantindo que cadas funcionalidade da API só seja liberada mediante a prévia autenticação e autorização para a obtenção de token de segurança. Caso o acesso seja feito sem esse token ou com um token inválido o sistema deve retornar o código HTTP correspondente a tentativa de acesso indevido.

Cenário 2: Ao navegar no sistema através de um dispositivo movel ou desktop com browser a tela deverá se adaptar automaticamente ao dispositivo, redimensionando todos os seus componentes não ferindo de maneira alguma a experiência da utilização do sistema por parte do usuário.

Cenário 3: Ao utilizar o sistema o usuário deve ser levado a cada funcionalidade de maneira rápida e objetiva, as telas devem demonstrar ao usuário de forma clara qual a função realizada por cada tela do sistema.

Cenário 4: Ao realizar o acesso a qualquer tela do sistema, a tela deve ser renderizada no máximo em 5 segundos mostrando o componente que o usuário deseja acessar.

6.3. Avaliação

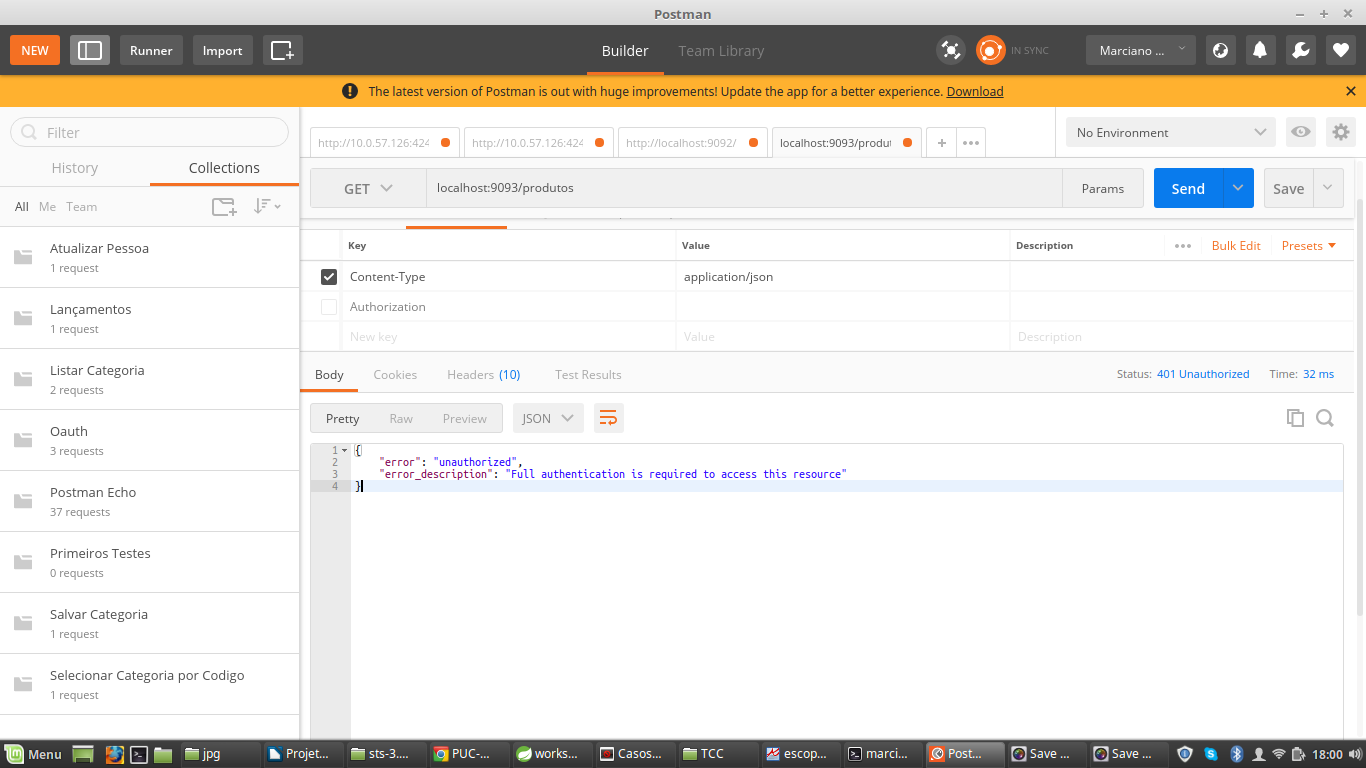
Processo de avaliação dos cenários identificados no item 6.1 são analisados.

O objetivo é determinar os riscos, não riscos, pontos de sensibilidade e tradeoffs e as evidências mostrando o requisito de qualidade sendo atendido.

|  |  |
| --- | --- |
| Atributo de Qualidade: | Segurança |
| Requisito de Qualidade: | O sistema deve apresentar altos padrões de segurança |
| Preocupação: | |
| Impossibilitar o acesso a APIs privadas do sistema sem autenticação no sistema. | |
| Cenários(s): | |
| Cenário1 | |
| Ambiente: | |
| Sistema em operação normal | |
| Estimulo: | |
| Usuário tenta acessar a API de Produto sem realiza a devida autenticação e obtenção do Token | |
| Mecanismo: | |
| Através do mecanismo de autenticação, autorização e geração de token da API de segurança todas as requisições devem ser feitas com um Token válido. | |
| Medida de Resposta: | |
| A API deve retornar um código de erro HTTP mostrando que o acesso sem o Token ou com um Token inválido é proibida. | |
| Considerações sobre a arquitetura: | |
| Riscos: | A autenticação e autorização na utilização das APIs do sistema é critica para a segurança do sistema. |
| Ponto de Sensibilidade: | A falta de configuração em novas APIs para a utilização do servidor de segurança |
| TradeOff: | Não existe |

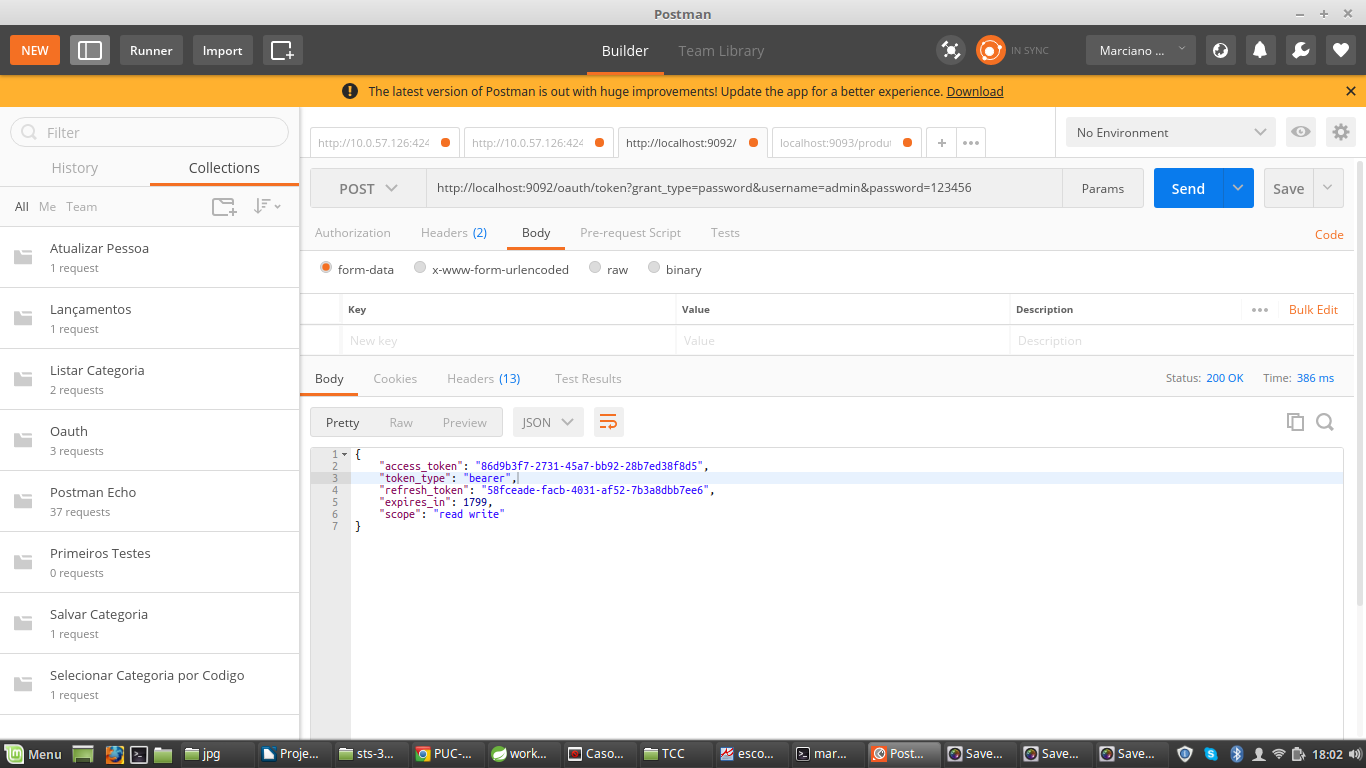
• Evidências do Cenário1

Tentativa de acesso a API de produto sem o Token de autorização de utilização.

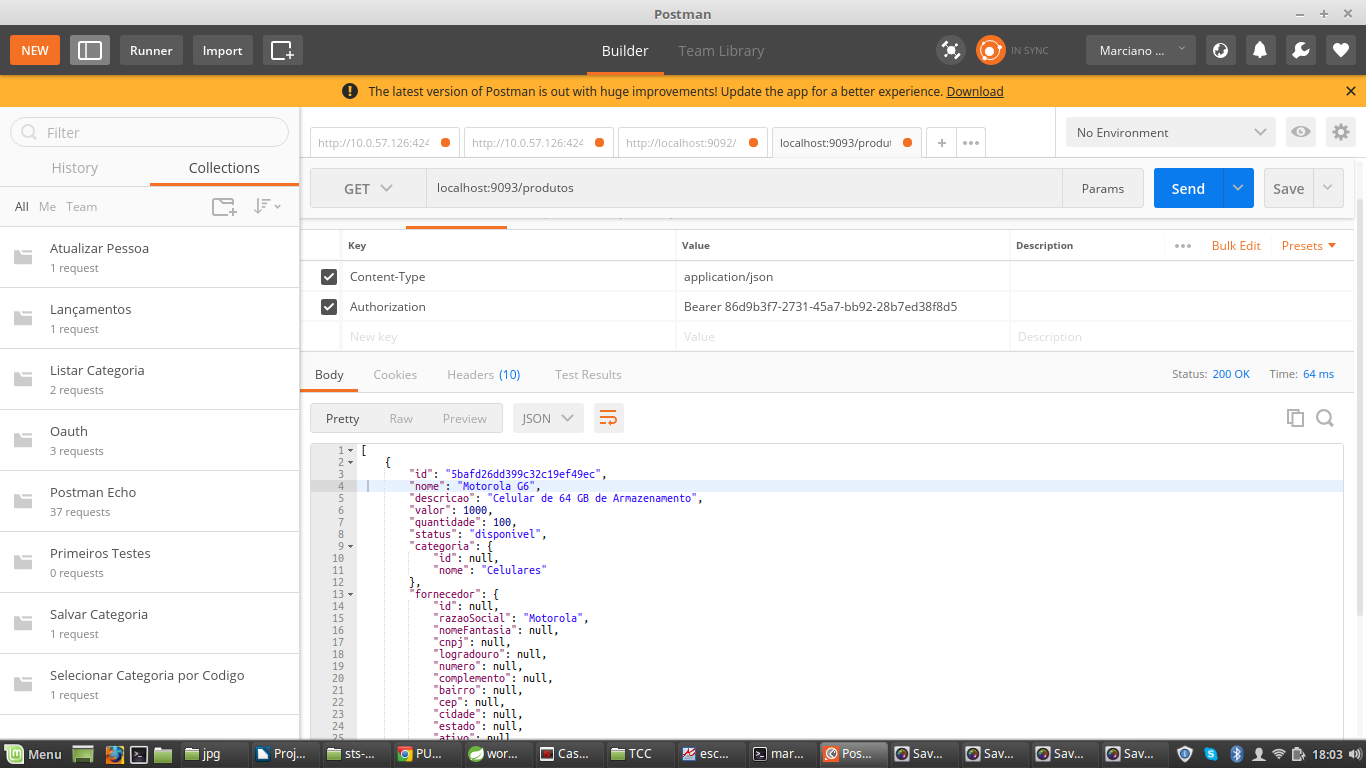


API retorna um erro 401 para o consumidor da API.

Usuário realizando a autenticação, obtendo um Token de acesso e chamando a API novamente.



Utilizando o Token na API de Produto:



Cenário 2.

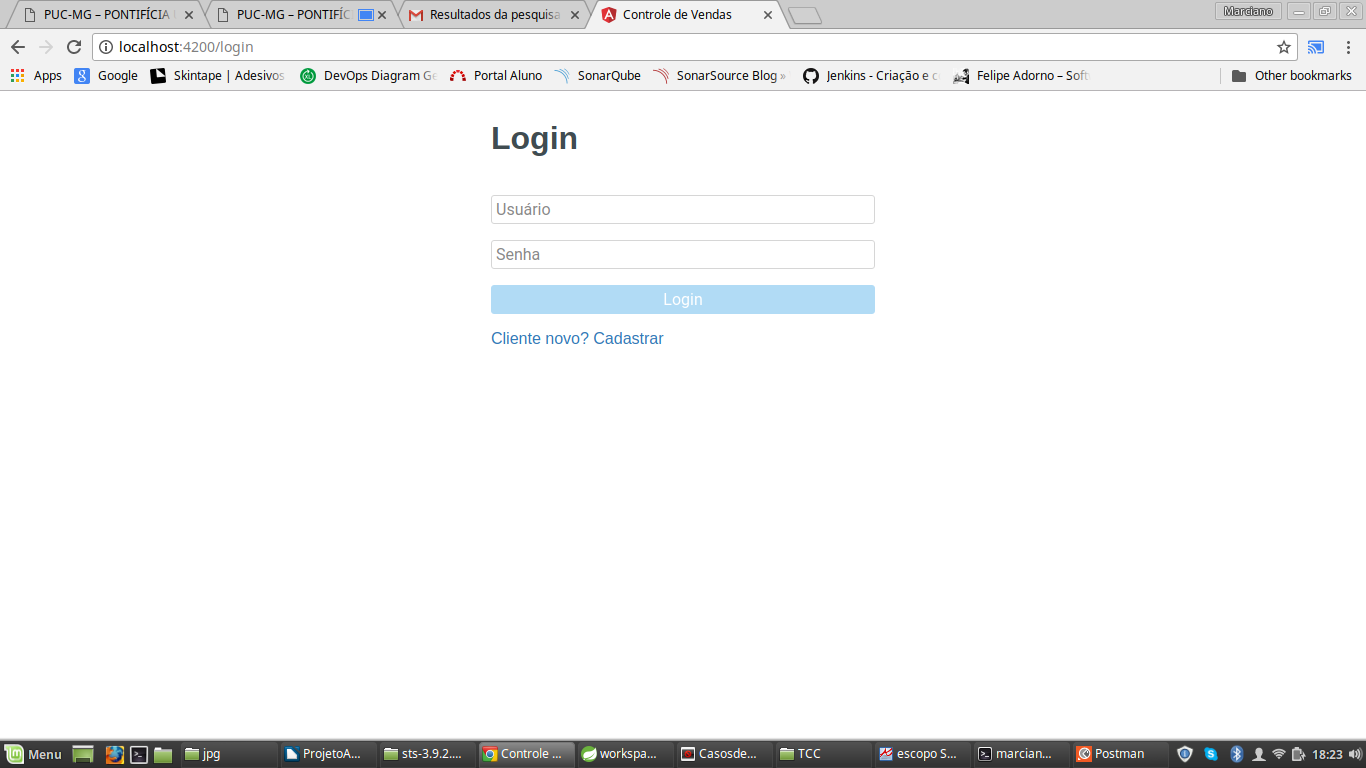
|  |  |
| --- | --- |
| Atributo de Qualidade: | Usabilidade |
| Requisito de Qualidade: | O sistema deve apresentar padrões de usabilidade |
| Preocupação: | |
| Sistema deve apresentar desempenho satisfatório. | |
| Cenários(s): | |
| Cenário2 | |
| Ambiente: | |
| Sistema em operação normal | |
| Estimulo: | |
| Usuário realizando login no site e sendo direcionado para a tela de pesquisa de produtos. | |
| Mecanismo: | |
| Telas com design clean sem poluição e muitos componentes possibilitando a rápida renderização dos componentes do lado cleinte.  Os processos AJAX através da tecnologia SPA do Angular e Bootstrap permitem o carregamento das páginas de forma rápida devido ao carregamento único do sistema pelo SPA e após esse carregamento apenas as partes necessárias são carregadas pelo cliente. | |
| Medida de Resposta: | |
| O usuário não deve demorar mais que 5 segundos para realizar o Login e ser direcionado a tela de pesquisa de produtos. | |
| Considerações sobre a arquitetura: | |

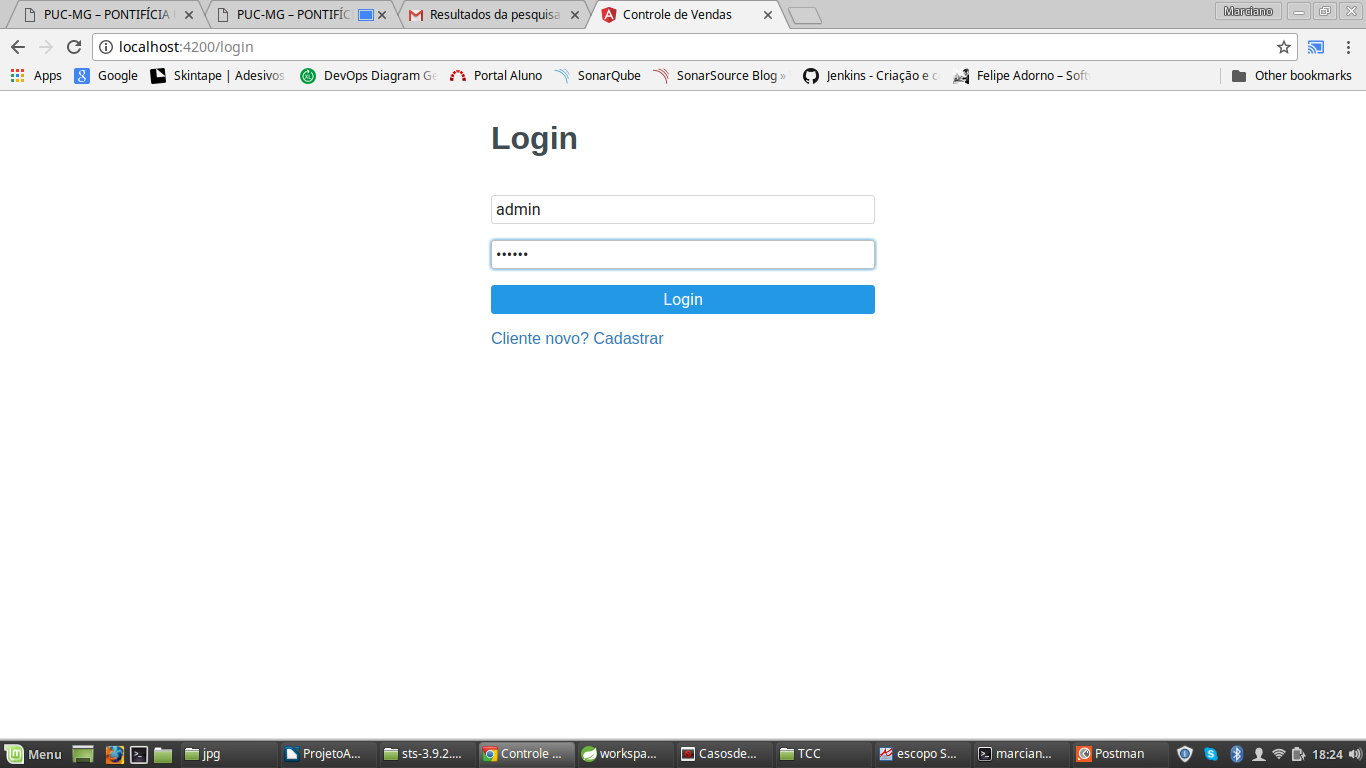
|  |  |
| --- | --- |
| Riscos: | A máquina cliente(Dispositivo Móvel ou Desktop) pode não ser uma máquina com boa memória RAM impossibilitando que o usuário faça bom uso do SPA devido ao carregamento de Javascript na máquina cliente. |
| Ponto de Sensibilidade: | Configurações de Hardware do lado Cliente |
| TradeOff: | Não existe |

Evidências do Cenário2.

Usuário realizando o Login no sistema e sendo direcionado para a tela de pesquisa de produtos.

Evidências do Cenário2.







Cenário 3.

|  |  |
| --- | --- |
| Atributo de Qualidade: | Acessibilidade |
| Requisito de Qualidade: | O sistema deve suportar diversos ambientes e ser responsivo |
| Preocupação: | |
| O Sistema deve redimensionar todos os seus componentes de acordo com o dispositivo utilizado. | |
| Cenários(s): | |
| Cenário3 | |
| Ambiente: | |
| Sistema em operação normal | |
| Estimulo: | |
| Usuário realizando login no site e sendo direcionado para a tela de pesquisa de produtos. | |
| Mecanismo: | |
| Atraves da junção do Angular com o Bootstrap foi possivel criar telas totalmente responsivas e que se adaptam totalmente aos dispositivos de forma automática e transparente ao usuário. | |
| Medida de Resposta: | |
| O usuário deve poder utilizar o sistema em qualquer dispositivo móvel ou desktop de maneira parecida sem que haja nenhuma perda de acessibilidade com a mudança de dispositivos. | |
| Considerações sobre a arquitetura: | |

|  |  |
| --- | --- |
| Riscos: | Usuário podem tentar acessar o sistema em browser que não possibilitem o uso de SPA. |
| Ponto de Sensibilidade: | Diferentes Browsers em diferentes dispositivos móveis |
| TradeOff: | Não existe |

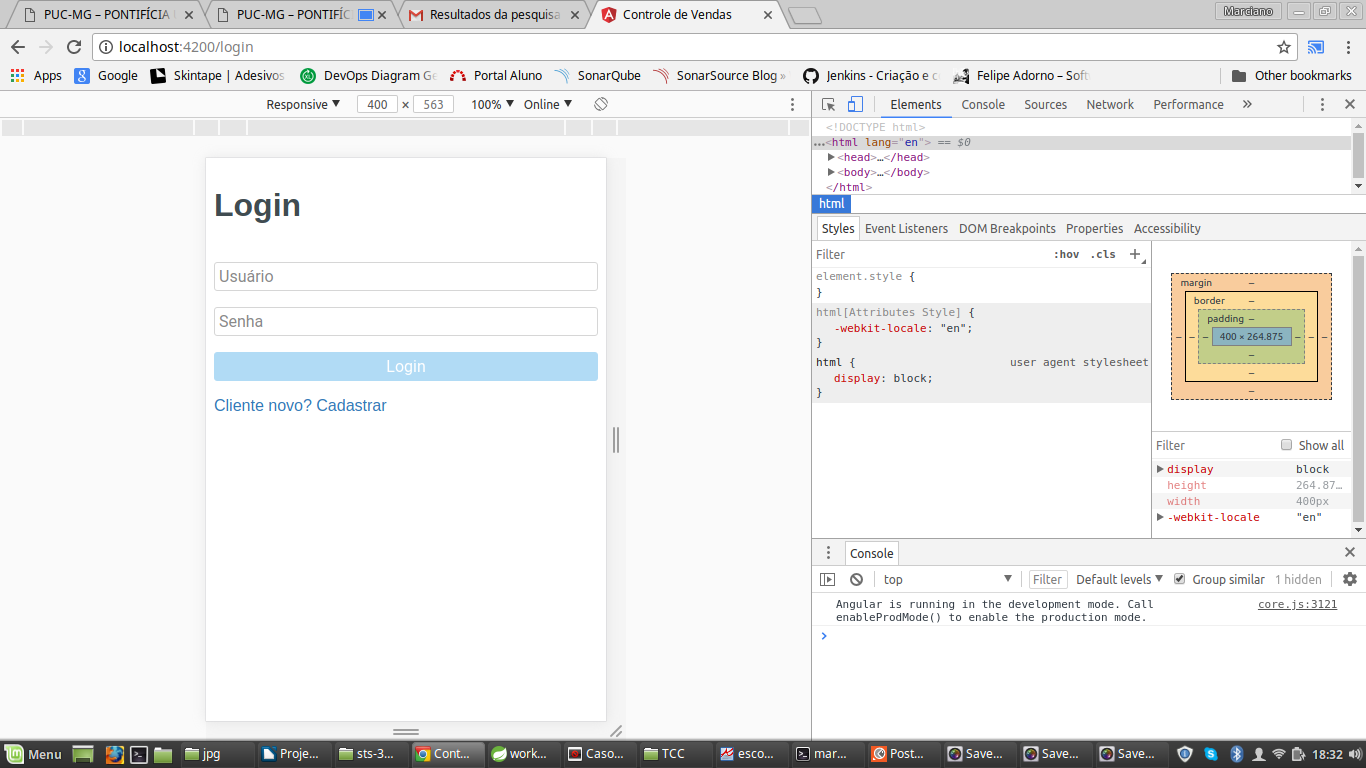
• Evidências do Cenário3

Usuário se autenticando em um desktop e em um dispositivo móvel.

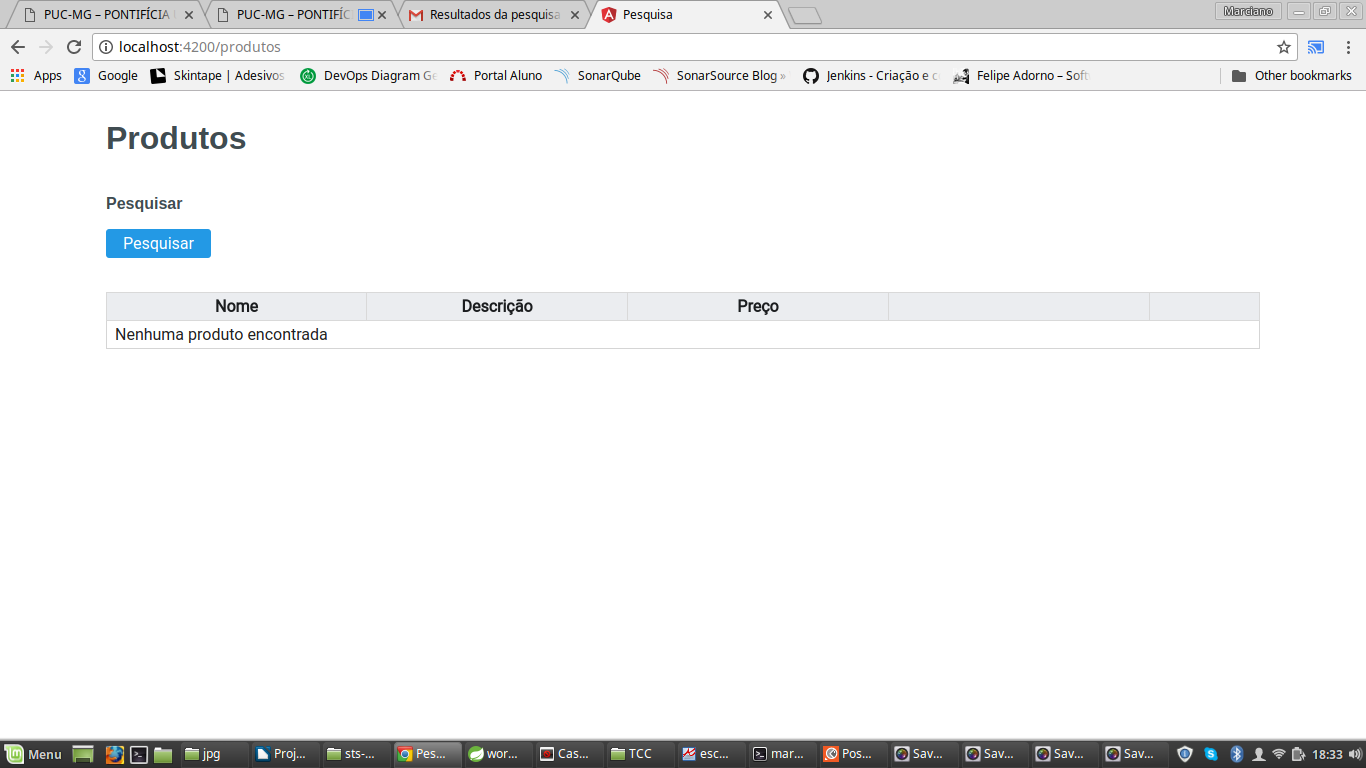
Desktop



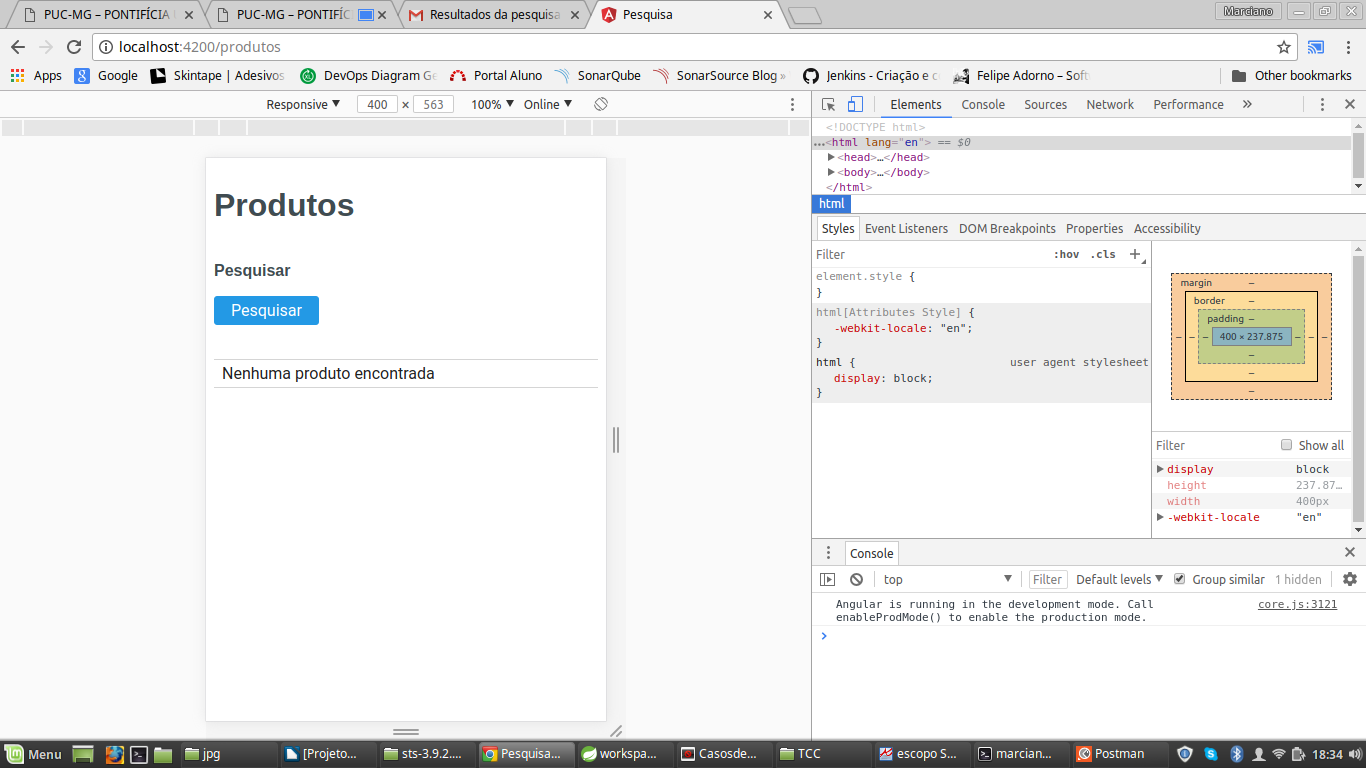
Dispositivo Móvel



Desktop



Dispositivo Móvel

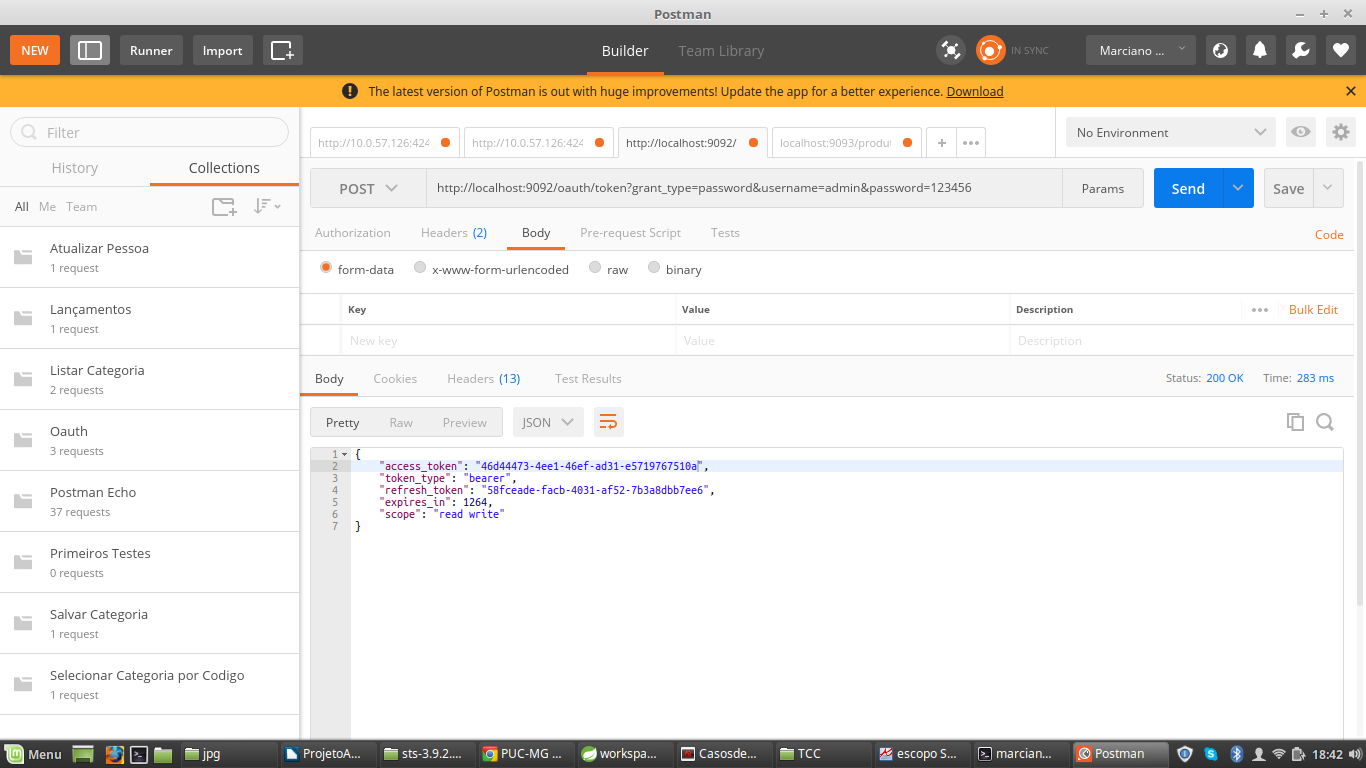


|  |  |
| --- | --- |
| Atributo de Qualidade: | Desempenho e Performance |
| Requisito de Qualidade: | O sistema deve ser rápido |
| Preocupação: | |
| Sistema deve apresentar desempenho satisfatório dentro dos limites aceitáveis. | |
| Cenários(s): | |
| Cenário4 | |
| Ambiente: | |
| Sistema em operação normal | |
| Estimulo: | |
| Pesquisa na API de produtos e Segurança. | |
| Mecanismo: | |
| Cada API foi criada para responder apenas por um motivo possibilitando o facil escalonamento, o cache de serviços e o tunning de consultas. | |
| Medida de Resposta: | |
| As consultas devem responder em menos que 5 segundos. | |
| Considerações sobre a arquitetura: | |
| Riscos: | APIs fora do ar podem sobrecarregar uma única API com consultas. |
| Ponto de Sensibilidade: | Balanceamento de Carga ativo |
| TradeOff: | Não existe |

Evidências do Cenário 4

Utilizando o Postman realizando consulta na API de Segurança e Produtos.

Tempo de Resposta 283ms



Tempo de Resposta 55ms



6.5. Resultados

Considerendo que a arquitetura atende todos os objetivos e atende a proposta de uma arquitetura modular, de facil manutenção com altos padrões de segurança, escalonamento, recuperação a falha e resiliência, a arquitetura atende aos requisitos para a criação de um e-commerce no modelo dropshipping.

A avaliação concluiu através dos testes que existem pontos de melhoria no decorrer da vida útil do software mas nenhuma dessas melhorias afeta a liberação da utilização por parte dos clientes e fornecedores do sistema.

|  |  |  |
| --- | --- | --- |
| Requisitos Não Funcionais | Testado | Homologado |
| O sistema deve apresentar altos padrões de segurança. | Sim | Sim |
| O sistema deve prover boa usabilidade | Sim | Sim |
| O sistema deve ser Responsivo | Sim | Sim |
| O sistema deve ser rápido | Sim | Sim |

Com os testes para definir se os requisitos foram atendidos foi possivel identificar pontos importantes como:

As APIs e telas não podem ser acessadas sem a autorização e autenticação, necessitando de um Token para cada acesso.

Todas as telas foram construidas no padrão SPA e são responsivas através do uso de Bootstrap possibilitabdo a padronização de seus componetes.

As APIs tem sua exposição através do padrão REST com JSON possibilitando que elas sejam consumidas por diversos sistemas e assim possibilitam a reutilização dos seus componetes no futuro.

As APIs através do balanceamento de carga e cache e contenirização possibilitam um alto escalonamento e performance em sua utilização.

7. Conclusão

O trabalho em questão apresentou um protótipo de arquitetura para um sistema de vendas na modalidade dropshipping, os objetivos de documentação e prova de conceito provarão que a ideia arquitetural atende totalmente a proposta e que suas limitações não impactam na aceitação da proposta pela facilidade de evolução da arquitetura apresentada.

Todas as melhorias e refactorings podem ser feitas em versões futuras não impedindo o lançamento do sistema de vendas dropshipping.

REFERÊNCIAS

Bibliográficas:

Hohpe, Gregory, Wolf, Bob. Enterprise Application Patterns.

Martin, R C. Clean Architecture.

Documentação Técnica:

<http://projects.spring.io/spring-cloud/>

<https://spring.io/projects/spring-integration>

<https://angular.io/>

Código Fonte

O código fonte da prova de conceito que foi desenvolvido para atender os requisitos não funcionais dessa arquitetura de software.

Configurações do sistema

<https://github.com/MarcianoAnunciacao/ConfigData>

Código fonte e documentação:

<https://github.com/MarcianoAnunciacao/especializacaopuc>