

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

Paulo César Dias da Silva
Murilo Caldeira de Sena Goes

SIGO – SISTEMA INTEGRADO DE GESTÃO E OPERAÇÕES

Belo Horizonte

2021

Paulo César Dias da Silva
Murilo Caldeira de Sena Goes

SIGO – SISTEMA INTEGRADO DE GESTÃO E OPERAÇÕES

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

Orientador: Prof. Pedro Alves de Oliveira

Belo Horizonte

2021

Dedicamos este trabalho a nossos avós, pais, esposas e irmãos que nos apoiaram e estiveram sempre ao nosso lado nos impulsionando, mesmo durante os momentos mais difíceis. Agradecemos do fundo do nosso coração.

AGRADECIMENTOS

Agradecemos primeiramente a Deus por nos conceder este presente que foi este curso de pós-graduação, pois com ele tivemos uma enorme evolução profissional e pessoal, por não permitir que desistíssemos no meio da jornada. Agradecemos aos nossos avós, pais e irmãos pelo apoio e o incentivo que nos deram no decorrer desta jornada. Agradecer aos amigos de curso pelo companheirismo no decorrer da jornada e na resolução de dúvidas das matérias apresentadas. Agradecer a todos que de alguma forma se envolveram nesta jornada fantástica. Obrigado a todos.

RESUMO

Este projeto contempla a adoção da Transformação digital como uma solução para a indústria têxtil no sentido de integrar toda a sua cadeia produtiva, no âmbito de alcançar melhores lucratividades, com baixo custo e alta performance, atingindo o objetivo proposto que é o de integração não só de processos, mas de todos os setores de uma indústria da área têxtil. O SIGO (Sistema Integrado de Gestão e Operações) trará a integração de todos os processos fabris que vai desde a operação, gestores, qualidade do produto desenvolvido até os diretores e responsáveis de uma empresa. Ele se encarregará de fazer a integração de todos os sistemas por meio de microsserviços, *API's*, serviços de mensagerias, *cloud computing*, *BI (Business Intelligence)*, entre outras tecnologias que se fizerem necessárias para garantir uma melhor integração entre os setores. O SIGO permitirá escalabilidade, performance e disponibilidade, garantindo assim a integração de toda a empresa. Este projeto é composto por 3 módulos principais e alguns componentes extras, que terão sua arquitetura proposta de modo a permitir a integração com os outros sistemas legados já existentes na empresa.

Palavras-chave: Arquitetura de software, *BI (Business Intelligence)*, *API's*, escalabilidade, *Web*, Design Responsivo.

SUMÁRIO

1. Objetivos do trabalho.....	7
2. Descrição geral da solução.....	8
2.1. Apresentação do problema.....	8
2.2. Descrição geral do software (Escopo).....	9
3. Definição conceitual da solução.....	9
3.1. Requisitos Funcionais.....	9
3.2 Requisitos Não-Funcionais.....	11
3.3. Restrições Arquiteturais.....	14
3.4. Mecanismos Arquiteturais.....	14
4. Modelagem e projeto arquitetural.....	16
4.1. Modelo de componentes.....	16
4.2. Modelo de implantação.....	19
5. Prova de Conceito (POC) / protótipo arquitetural.....	22
5.1. Implementação e Implantação.....	22
5.2 Interfaces/ <i>API's</i>	25
6. Avaliação da Arquitetura.....	36
6.1. Análise das abordagens arquiteturais.....	36
6.2. Cenários.....	36
6.3. Avaliação.....	37
6.4. Resultado.....	50
7. Conclusão.....	51
REFERÊNCIAS.....	52
APÊNDICES.....	54

1. Objetivos do trabalho

O objetivo deste trabalho é apresentar a descrição do projeto arquitetural de uma aplicação para gestão da indústria Têxtil que vai auxiliar a equipe de operadores, gestores, diretores e em geral toda a empresa, nas atividades do seu dia a dia relacionado à fabricação e na qualidade dos produtos desenvolvidos. O SIGO vai apoiar em atividades como cadastro de normas, atualização das mesmas, na gestão de processos de fabricação dos produtos, na tomada de decisões e estratégias, auxiliar na divulgação e transparência com seus consumidores, gerar informações através de relatórios e integrações com sistemas legados, a fim de garantir o *compliance* da empresa.

Os objetivos específicos são:

1. Criar um modulo de Gestão de Normas visando manter as normas técnicas existentes nas áreas industrial, ambiental e outras aplicáveis, atualizadas e fornecendo toda a base para *compliance*, incluindo o planejamento de ações que envolvam o uso correto e eficiente e recursos, descarte de subprodutos do processo industrial no meio ambiente, monitoramento de riscos de acidentes e suas consequências. Isto será feito através de cadastramento, atualização e utilização das normas existentes, permitindo também a visualização das mesmas por toda a indústria têxtil, cada uma com o seu devido perfil de acesso. Este módulo ainda fará uso de um *web service* responsável por notificar eventuais atualizações de normas externas.
2. Criar um modulo de Consultorias e Assessorias para permitir a contratação e gestão contratual de empresas de assessoria e consultoria existentes no mercado, de modo a adequar a empresa às normas e padrões da sua área de negócio, incluindo os padrões internacionais de governança corporativa. O serviço prestado pelos órgãos reguladores e normalizadores deve gerar insumos para o planejamento e controle das atividades industriais. Além disso, deve ser integrado aos demais módulos de modo a permitir acesso às informações neles geradas.
3. Criar um modulo de Gestão do Processo Industrial que será baseado em tecnologias robustas, livres e gratuitas, tais como Linux e Apache e utilizar componentes reutilizáveis. Este módulo deve possuir diversas integrações, envolvendo trocas de mensagens com os demais módulos e integração completa com a solução ERP implantada, que utiliza as tecnologias da empresa SAP ([HTTPS://sap.com/brazil](https://sap.com/brazil)).
4. Criar um componente de integração entre os sistemas legados e os novos módulos, garantindo a disponibilidade de informações em tempo real por todos usuários, e também resolvendo problemas atuais na integração entre os sistemas legados da empresa.

5. Criar um módulo de autenticação que será responsável pela segurança no acesso e manipulação dos dados.

2. Descrição geral da solução

2.1. Apresentação do problema

O início da quarta revolução industrial, que tem como protagonista a internet para o desenvolvimento tecnológico, em conjunto com o avanço da ciência permitiu unir os conhecimentos analógicos, biológicos e digitais para criar tecnologias mais eficientes, de rápido acesso à informação e ampla comunicação. No entanto, com a rápida degradação ambiental, essa nova era também é marcada pela necessidade de meios de produção sustentáveis, junto da redução de desperdícios.

A partir do momento em que a indústria 4.0 aumenta a autonomia das máquinas e dos sistemas digitais, um setor como o têxtil — que é tão caracterizado pela efemeridade das tendências — ganha com a possibilidade de usufruir de uma maior customização sem perder na qualidade. Além disso, a produção escalável atende a uma real necessidade das confecções. Afinal, é preciso se adequar a todo o momento àquilo solicitado pelo público consumidor.

Entre os princípios para o desenvolvimento e implantação da indústria 4.0, que definem os sistemas de produção inteligentes que tendem a surgir nos próximos anos, está a modularidade, que é a produção de acordo com a demanda, acoplamento e desacoplamento de módulos na produção. O que oferece flexibilidade para alterar as tarefas das máquinas facilmente.

Sejam consumidores ou empresas, os clientes estão, de forma crescente, no epicentro da economia, o que significa que “tudo” está relacionado com a melhoria das formas mediante as quais serão servidos. Hoje, a maioria das empresas fabrica para depois vender, agora você vai vender para depois fabricar. É uma inversão da forma de venda e produção. Isso terá impacto na cadeia logística, usá-la menos é uma tendência. Ser muito mais para o cliente final, do que levar para uma loja ou um centro de distribuição, para então o cliente comprar e levar pra casa.

O conceito de transformação digital envolve não somente um modelo de negócio ou meramente tecnologia, mas uma junção dessas duas visões e análises, de forma complementar e integrada. Os métodos de automação e robótica também serão essenciais para uma produção sustentável e econômica.

A entrada de novos *players* no mercado, a necessidade de melhorar os custos de produção, a automatização de processos fabris, a produção estratégica de produtos, a necessidade de maior variedade de fornecedores, a indispensabilidade de agregar valor no produto final, a busca constante por inovações, a

gestão e governança modernas, além do aumento de lucratividade, são todas questões cruciais nessa nova era tecnológica.

Se adequar é um desafio mais do que necessário para se manter competitivo no mercado. Os sistemas legados utilizados precisam permanecer em uso pelas empresas, porque a substituição do sistema legado por um novo sistema é um processo extenso e caro. Nesse caso, a opção mais viável para o crescimento da operação, e necessária para não perder nenhum tipo de dado, é fazer a integração de sistemas legados.

2.2. Descrição geral do software (Escopo)

A solução arquitetural definida para a aplicação SIGO (Sistema Integrado de Gestão e Operação) no ramo da indústria Têxtil, será uma solução que terá integrações com sistemas legados, capaz de prover informações estratégicas e terá uma arquitetura distribuída sendo capaz de ser utilizada tanto por mobile, aplicações Web, IoT, *tablets*, além de gerar insumos para Big Data. Trata-se de um conjunto de módulos onde cada um possui sua função específica, com o objetivo de auxiliar na gestão e operação das atividades da indústria IndTexBr.

O módulo de Gestão de Normais fornecerá toda a base para *compliance* e planejamento de ações para aderência aos padrões de conformidade, além de um serviço de notificação para qualquer alteração que uma norma venha sofrer, garantindo que os processos da empresa não estejam de maneira irregular. Seguir os padrões de acordo com as normas de controle proporciona uma série de benefícios como eficiência operacional, aumento de produtividade, vantagem competitiva, segurança jurídica, redução de custos e boa reputação no mercado.

O módulo de Consultorias e Assessorias permitirá a contratação e gestão contratual de empresas de assessoria e consultoria existentes no mercado, é integrado com os demais módulos para permitir acesso às informações por parte dos consultores. Permite também que os consultores cadastrem insumos gerados pelos seus serviços de modo a auxiliar no planejamento e controle das atividades industriais.

O módulo de Gestão do Processo Industrial é o módulo principal, com diversas integrações, apresentará dados gerenciais com uso de *dashboards*, imagens iterativas e diversos relatórios e informações geradas pelos demais módulos do sistema (novos e legados), que estarão disponíveis aos gestores auxiliando na tomada rápida de decisões.

3. Definição conceitual da solução

3.1. Requisitos Funcionais

Módulo de Autenticação

- O sistema deve permitir o cadastro de novos usuários (Administradores, Consultores e Gestores de Normas)

- O sistema deve permitir que um usuário efetue *login*.

- O sistema deve permitir que um usuário efetue o *logout*.

Módulo de Gestão de Normas

- O sistema deve permitir a busca e acesso a sistemas externos de normas reguladoras nacionais e internacionais, de acordo com as necessidades técnicas e de negócio.

- O sistema deve permitir o gerenciamento (incluir, modificar, excluir e consultar) das normas internas da empresa.

- O sistema deve permitir o gerenciamento (incluir, modificar, excluir e consultar) de atividades que visam o cumprimento das normas.

- O sistema deve alertar os gestores de normas sobre qualquer alteração que uma norma venha a sofrer.

Módulo de Consultorias e Assessorias

- O sistema deve permitir o gerenciamento (incluir, modificar, excluir e consultar) de empresas de consultorias e assessorias existentes no mercado.

- O sistema deve permitir o gerenciamento (incluir, modificar, excluir e consultar) de contratos com empresas de consultorias e assessorias.

- O sistema deve permitir o gerenciamento (incluir, modificar, excluir e consultar) de ações para planejamento e controle das atividades industriais, para os usuários com perfil de Consultores.

- O sistema deve permitir o acesso à informações dos demais sistemas da IndTexBr para que o trabalho das empresas de assessorias possa ser realizado, tendo como exemplo: Gestão de Normas, Sistema de Logística, Sistema de Gestão dos Processos Industriais, Monitoramento de Vendas, Segurança e Qualidade, Inteligência do Negócio, Relatórios de Acompanhamento, e demais soluções SAP.

Módulo de Gestão do Processo Industrial

- O sistema deverá permitir que os usuários autenticados como administradores possam gerar relatórios e visualizar *dashboards* para auxiliar na tomada de decisão.

- O sistema deverá disponibilizar imagens iterativas com as etapas do processo industrial de maneira sintética, com as informações provenientes dos sistemas novos e legados, e permitirá uma visualização analítica de cada etapa do processo em tempo real.

- O sistema deve permitir o acesso à informações dos demais sistemas da IndTexBr para prover informações aos *dashboards* e relatórios.

3.2 Requisitos Não-Funcionais

- Usabilidade – O sistema deve promover uma usabilidade fácil, objetiva e intuitiva.

Estímulo	Usuário buscando as normas.
Fonte de estímulo	Usuário acessando a funcionalidade de pesquisa de normas e efetuando a busca de uma norma externa.
Ambiente	Produção, carga normal.
Artefato	Módulo Gestão de normas.
Resposta	O <i>front-end</i> apresenta facilidade de navegação, simplicidade, objetividade e intuitivo.
Medida da resposta	O usuário conseguiu efetuar a busca da norma em poucos cliques e em poucos segundos.

- Acessibilidade – O sistema deve suportar ambientes web responsivos e ambientes móveis.

Estímulo	Usuário buscando as normas.
Fonte de estímulo	Usuário acessando a funcionalidade de consulta de normas através de um dispositivo móvel (<i>Smartphone</i>) ou através do computador (Web).
Ambiente	Produção, carga normal.
Artefato	Módulo Gestão de normas.
Resposta	O <i>front-end</i> pode ser adaptar conforme o dispositivo que é usado para acesso à aplicação (Computador) modificado a sua resolução para facilitar a navegação ou pode ter acesso com a resolução desenvolvida para aplicações moveis através de um dispositivo móvel.
Medida da resposta	Nos dois casos a identidade visual se mantém a mesma em todas as telas e resoluções e não há perda de funcionalidades.

- Desempenho – O sistema deve fornecer um bom desempenho com um tempo de resposta adequando a cada funcionalidade.

Estímulo	Usuário Buscar, cadastrar, excluir e atualizar uma norma.
-----------------	---

Fonte de estímulo	Usuário acessa o <i>front-end</i> e efetua o cadastro, atualização, exclusão ou a busca das informações das normas internas.
Ambiente	Produção, carga normal.
Artefato	Módulo Gestão de normas.
Resposta	O cadastro, atualização, exclusão ou a busca de uma norma interna foi realizado com sucesso e com um tempo de resposta muito abaixo do esperado.
Medida da resposta	O Tempo de resposta nas funcionalidades de cadastro, atualização, exclusão ou busca da norma interna ocorreu em menos de 5 segundos.

- Manutenibilidade – O sistema deve apresentar manutenção facilitada.

Estímulo	Alteração de um módulo do sistema.
Fonte de estímulo	Desenvolvedor realiza uma alteração em um módulo do SIGO.
Ambiente	Desenvolvimento.
Artefato	Qualquer módulo do sistema.
Resposta	Atrações confirmadas no <i>branch master</i> .
Medida da resposta	Após o <i>commit</i> , os testes são executados e se confirmados, as alterações são salvas no <i>branch master</i> e a entrega contínua é disparada.

- Testabilidade – O sistema deve ser simples de testar, fazendo uso de testes automatizados.

Estímulo	Alteração de um módulo do sistema.
Fonte de estímulo	Pipeline de CI/CD.
Ambiente	Desenvolvimento.
Artefato	Qualquer módulo do sistema.
Resposta	O sistema testou todas as funcionalidades disponíveis no módulo alterado.
Medida da resposta	Uma nova imagem do módulo é gerada.

- Disponibilidade – O sistema deve ser confiável e robusto, se recuperando no caso da ocorrência de erro.

Estímulo	Desativar um dos nós do <i>cluster</i> em tempo de execução do sistema.
Fonte de estímulo	Administrador do <i>cluster</i> de servidores de aplicação.
Ambiente	Produção, carga normal.
Artefato	Qualquer módulo do sistema.
Resposta	Todos os usuários continuaram usando o sistema sem notar que houve uma queda de um dos nós.
Medida da resposta	Todas as solicitações dos usuários devem ser

	atendidas, com atraso máximo de 3 segundos devido à queda de um dos nós.
--	--

- Disponibilidade – O sistema deve estar disponível 24 horas por dia, nos sete dias da semana.

Estímulo	Sistema apresentando instabilidade no tempo de resposta das requisições.
Fonte de estímulo	Muitos usuários acessando um mesmo recurso.
Ambiente	Produção, carga alta.
Artefato	Qualquer módulo do sistema, base de dados ou componentes de negócio.
Resposta	O sistema deverá ser capaz de fazer escalonamento vertical e horizontal no <i>cluster</i> de microserviços por meio do orquestrador de microserviços e do <i>cluster</i> do banco de dados por meio do sistema gerenciado de banco de dados distribuído.
Medida da resposta	O sistema tem seu tempo de resposta normalizado.

- Interoperabilidade – O sistema deve se comunicar com sistemas externos via *API's Restful* de integração.

Estímulo	Consulta a sistema externo de Gestão de Normas.
Fonte de estímulo	Módulo de Gestão de Normas efetuando busca de uma norma externa.
Ambiente	Produção, carga normal.
Artefato	Módulo de Gestão de Normas.
Resposta	O sistema externo consultado retorna os dados solicitados.
Medida da resposta	Comunicação com o sistema externo efetuada com sucesso.

- Interoperabilidade – O sistema deve se comunicar com os sistemas legados já existentes *on premise*.

Estímulo	Consulta de uma etapa do processo industrial.
Fonte de estímulo	Sistema de Gestão de Processos Industriais criando informações de um processo industrial.
Ambiente	Produção, carga normal.
Artefato	Módulo de Gestão do Processo Industrial.
Resposta	O integrador dos sistemas legados retorna os dados do processo.
Medida da resposta	Os dados do processo industrial são exibidos na tela.

- Segurança – O sistema deve apresentar segurança no acesso e manipulação de dados.

Estímulo	Acesso a uma página ou recurso protegido do sistema sem estar autenticado.
Fonte de estímulo	Usuário.
Ambiente	Produção, carga normal.
Artefato	Módulos de Gestão de Normas, Gestão do Processo Industrial, Autenticação e Gestão de Consultorias e Assessorias.
Resposta	O sistema deve redirecionar o usuário para a tela solicitando usuário e senha ou retornar o código 401 (<i>unauthorized</i>).
Medida da resposta	O sistema não deve permitir o acesso às páginas ou recursos privados, obrigando a autenticação.

3.3. Restrições Arquiteturais

- Apresentar características de aplicações distribuídas, tais como abertura, portabilidade e uso extensivo de recursos de rede, sendo capaz de ser acessada de qualquer lugar do mundo, se comunicar e integrar com sistemas de diferentes tecnologias.
- Ser hospedado em nuvem híbrida, com parte dos componentes sendo mantidos *on premise*.
- Ser modular e implantável por módulos, de acordo com a prioridade e necessidade da empresa.
- O sistema vai ter integrações com sistemas de terceiros e deve ter sua arquitetura baseada em micro-serviços e *API's*.
- O sistema deve abrir de forma responsiva em aparelhos menores, como celular e *tablet*.
- O sistema deve ter seu *build* feito através de integração contínua.
- O sistema deve ter *pipelines* de testes em sua integração contínua.

3.4. Mecanismos Arquiteturais

Inclua nesta seção os mecanismos arquiteturais de análise, projeto e implementação da arquitetura inicialmente contemplada para a sua aplicação.

Mecanismo de análise	Mecanismo de design	Mecanismo de implementação
<i>Front-end</i>	Interface de comunicação com o usuário do sistema	React.js
<i>Back-end</i>	Regras de negócio da aplicação	.NET Core
Integrações com outros módulos e sistemas	Interfaces utilizando XML e/ou JSON, Mensageria com conectores KAFKA	<i>WebServices</i> WS-*, <i>Restful API's</i> , AWS MSK e AWS Direct Connect
Persistência	ORM e tecnologias de acesso	EFCore e ADO.Net

	a dados	
Persistência	Banco de dados relacional	AWS RDS (Relational Database Service) e Microsoft SQL Server
Alta disponibilidade	Balanceamento de carga das aplicações	AWS ElasticBeanstalk e Balanceador de cargas AWS Elastic Load Balancer
Autenticação e autorização	Verificação das credenciais para execução de ações	Gerador de <i>token</i> JWT OAuth 2.0
Exposição de <i>API's</i>	Exposição de <i>Restful API's</i>	AWS API Gateway
Notificações de usuários	Envio de notificações de atualizações de normas externas	AWS Lambda
Disponibilização de conteúdo estático	Aplicação para servir conteúdo estático, como HTML, Javascript, CSS, fontes e imagens	AWS Amplify
CI/CD	Ferramenta para pipeline de integração e entrega contínua	GitHub
Build	Geração de artefatos para publicação nos servidores de aplicação	MSBuild e NPM
Automação de testes	Execução de testes automatizados das aplicações	MSTest e Jest
Deploy	<i>Deploy</i> de artefatos para os servidores de aplicação	AWS Code Deploy
Versionamento	Controle de código-fonte	Git e Github

4.1. Modelo de componentes

O diagrama ilustra a arquitetura de uma solução híbrida, dividida em duas partes principais: **AMBIENTE EM NUVEM** e **AMBIENTE ON-PREMISSE**.

AMBIENTE EM NUVEM:

- «API EXTERNA» NORMAS EXTERNAS** e **«WEB BROWSER» SIGO** interagem com a infraestrutura via **«AWS WEB SERVICE» AMBIENTE EM NUVEM**.
- O **«AWS API GATEWAY» API GATEWAY** atua como ponto central de entrada, recebendo requisições REST e distribuindo-as para:
- «AWS ELASTIC LOAD BALANCER» BALANCEADOR DE CARGA**, que direciona o tráfego para o **«AWS ELASTIC BEANSTALK» API WEB**.
- O **«AWS ELASTIC BEANSTALK» API WEB** hospeda quatro aplicações baseadas em **«NET Core»**:
 - GESTÃO DE NORMAS**
 - CONSULTORIAS E AESSORIAS**
 - GESTÃO DO PROCESSO INDUSTRIAL**
 - AUTENTICAÇÃO**
- As aplicações de **«NET Core»** utilizam o **«AMAZON MSK» APACHE KAFKA** para comunicação assíncrona (**«Publish/Subscribe»**).
- O **«AWS LAMBDA» NOTIFICAÇÃO DE ALTERAÇÕES DE NORMAS EXTERNAS** também se integra ao sistema via REST.
- Para a infraestrutura de backend, são utilizados:
 - «AWS CODE DEPLOY» DEVOPS CI/CD** para automação de deploys.
 - «GITHUB» CONSTRUÇÃO TESTES UNITÁRIOS INTEGRAÇÃO CONTÍNUA ENTREGA CONTÍNUA** para controle de versão e CI/CD.
 - «AWS RDS» BANCO DE DADOS SQL SERVER** para armazenamento de dados, com instâncias específicas para:
 - AUTENTICAÇÃO**
 - GESTÃO DE NORMAS**
 - CONSULTORIAS E AESSORIAS**
 - GESTÃO DO PROCESSO INDUSTRIAL**

AMBIENTE ON-PREMISSE:

- O **«AWS DIRECT CONNECT» CONEXÃO DEDICADA** estabelece a ligação entre a nuvem e o ambiente local.
- O **«KAFKA CONNECTOR» MIDDLEWARE SAP** atua como ponte entre o ambiente on-premise e o **«AMAZON MSK» APACHE KAFKA**.
- Existem cinco instâncias de **«KAFKA CONNECTOR» MIDDLEWARE** especializadas:
 - MIDDLEWARE SAP** (conectado ao **«LEGADO SAP» SISTEMA DE LOGÍSTICA** via **«JDBC»**).
 - MIDDLEWARE MONITORAMENTO DE VENDAS** (conectado ao **«LEGADO SAP» GESTÃO DO PROCESSO INDUSTRIAL** via **«JDBC»**).
 - MIDDLEWARE SEGURANÇA E QUALIDADE** (conectado ao **«LEGADO REACT» MONITORAMENTO DE VENDAS** via **«JDBC»**).
 - MIDDLEWARE INTELIGÊNCIA DO NEGÓCIO** (conectado ao **«LEGADO .NET FRAMEWORK 4.7.2» SEGURANÇA E QUALIDADE** via **«JDBC»**).
 - MIDDLEWARE RELATÓRIOS DE ACOMPANHAMENTO** (conectado ao **«LEGADO ORACLE» INTELIGÊNCIA DO NEGÓCIO** via **«JDBC»**).
- Adicionalmente, há uma conexão **«SOAP»** entre o ambiente on-premise e o **«AWS RDS»** na nuvem.

Figura 1 – Diagrama de componentes.

A arquitetura prevista tem como característica a separação do sistema em módulos implantáveis individualmente. Todo o sistema foi projetado para ser executado em nuvem híbrida, estando parte em ambiente *on premise* e parte na nuvem AWS.

A separação entre as aplicações de *front-end* e suas *API's* de *back-end* provê a independência entre componentes e auxilia a manutenibilidade do sistema. As aplicações de *front-end* são responsáveis pelo *layout* visual, interface e interação com os usuários, nelas não existem regras de negócio. A utilização da tecnologia React.js proporciona uma experiência intuitiva aos usuários, possuindo também a capacidade de responsividade para exibição de recursos em dispositivos com diversos tamanhos de tela. Isso garante o atendimento aos requisitos não funcionais de usabilidade e acessibilidade.

As *API's* de *back-end* expõem seus *end-points* para um *API Gateway*, de forma que apenas o Gateway esteja exposto à internet pública, para que as *API's* não estejam acessíveis externamente de forma direta, apenas pelo Gateway. Para isso, foi utilizada a tecnologia Amazon VPC, uma rede virtual privada, com sub-redes públicas e privadas, estando todas as *API's* implantadas em sub-redes privadas. A comunicação entre *front-end* e *back-end* é feita através de chamadas ao *API Gateway*, trafegando os seus dados no formato json. As rotas são protegidas pela *API* de autenticação, que recebe as informações de usuário e senha do *front-end* e disponibiliza um *token* JWT para ser utilizado no cabeçalho das próximas requisições. As autorizações de acessos conforme os perfis de usuários são tratadas também pelo *token* JWT, permitindo que cada usuário tenha acesso apenas às informações pertinentes ao seu perfil (Administradores, Consultores e Gestores de Normas). Não é possível acessar informações ou recursos sem um *token*, ou com um *token* inválido, garantindo o requisito não funcional de segurança.

A utilização do AWS ElasticBeanstalk, em conjunto com o AWS Elastic Load Balancer, que são serviços para implantação e escalabilidade de aplicações, provê recursos como o provisionamento de capacidade, o balanceamento de carga, a escalabilidade da aplicação e até o monitoramento da saúde do aplicativo, de maneira dinâmica, conforme o número de requisições, diminuindo a latência. Ao mesmo tempo, mantém total controle sobre os recursos da AWS que possibilitam a operação do aplicativo e acesso aos recursos a qualquer momento. A aplicação utiliza um servidor de banco de dados relacional, SQL Server, através do serviço de banco de dados relacionais da AWS, o AWS RDS (*Relational Database Service*). Por se tratar de microsserviços cada aplicação terá sua própria instância de banco de dados, garantindo a independência e resiliência entre os componentes. A infraestrutura proporcionada pelo serviço AWS RDS facilita a configuração, a operação e a escalabilidade de bancos de dados relacionais na nuvem com performance rápida, alta disponibilidade e segurança. Tais características garantem a conformidade com os requisitos não funcionais de desempenho e disponibilidade.

Um *pipeline DevOps* utilizando das ferramentas GitHub e GitHub Actions será responsável pela integração contínua, construção da aplicação, testes automatizados e entrega contínua, aplicando qualquer alteração na infraestrutura do AWS Code Deploy em questão de minutos, garantindo os requisitos funcionais de

testabilidade e manutenibilidade.

Uma das integrações previstas é a de consultas às normas externas, pelo módulo de Gestão de Normas. Para isso é consumida uma *API Restful* de um fornecedor terceirizado com informações das normas aplicadas à empresa.

Outra integração prevista foi o serviço de notificação de alterações de normas. Aqui não é necessária uma estrutura alocada de maneira definitiva, então optou-se por uma solução *serverless* (computação sem servidor) utilizando o serviço AWS Lambda, pagando-se apenas pelo processamento executado. O fornecedor da *API* de normas externas irá disparar uma notificação de alteração de norma, que será consumida pelo serviço Lambda. A estrutura somente será alocada para realizar a comunicação com a *API* externa e notificar a *API* de Gestão de Normas, sendo encerrada em seguida.

Toda a integração entre os novos módulos, os sistemas legados já implantados e os sistemas legados ainda não implantados ou com problemas de integração – como os legados de Segurança e Qualidade e Inteligência de Negócio – foi prevista utilizando a tecnologia Apache Kafka e o Amazon MSK que é um serviço gerenciado que facilita a criação e execução de aplicativos que usam o Apache Kafka para processar dados de *streaming*. O Apache Kafka foi desenvolvido para processar fluxos de dados provenientes de diversas fontes, bem como para entregá-los a vários clientes e é capaz de lidar com milhões de pontos de dados por segundo, que é uma vantagem para os casos de uso que exigem disponibilidade de dados em tempo real.

O Amazon MSK monitora continuamente a integridade do *cluster* e substitui automaticamente nós não íntegros sem tempo de inatividade. Além disso, o Amazon MSK protege o *cluster* do Apache Kafka criptografando dados ociosos.

Os novos módulos foram projetados para funcionar diretamente com o Kafka relatando e consumindo os fluxos de eventos do Kafka.

O Kafka Connect, foi utilizado como uma estrutura para *streaming* de dados entre o Kafka e os outros sistemas de dados, com conectores JDBC para integração com os bancos de dados legados existentes.

Para acesso aos bancos de dados internos, foi utilizada uma conexão dedicada entre o *datacenter* local e a nuvem, utilizando a tecnologia AWS Direct Connect.

O Kafka é o barramento central que conecta todos os aplicativos. Ter um tubo central significa que você tem apenas um link entre *datacenters* para monitorar e gerenciar. Isso permite acompanhar os eventos produzidos em cada *datacenter*, quem os está consumindo e qual é a latência. Garantir que você tenha um único pipeline e que todos os dados de origem sejam transmitidos para a nuvem apenas uma vez (e não várias vezes para vários aplicativos), reduz o tráfego cruzado do *datacenter*.

A comunicação entre os bancos de dados legados e o Amazon MSK é feita através de *middlewares* implementados como conectores JDBC que efetuam a publicação de mensagens com as alterações nos dados, nos tópicos do Kafka (*publish*) e também consomem mensagens no barramento Kafka dos tópicos inerentes a cada aplicação (*subscribe*). A configuração dos tópicos, seus produtores e consumidores garante que cada sistema, novo ou legado, tenha acesso às informações que precisarem, e também disponibilizem infor-

mações aos demais módulos. O Sistema de Segurança e qualidade tem sua integração com o sistema SAP resolvida dessa forma, assim como todos os demais sistemas legados. O sistema de Inteligência de Negócio receberá os dados consumidos pelo seu conector Kafka, que então serão extraídos e descarregados na base do Hadoop do sistema de *B.I.* adquirido da empresa Oracle.

Também foi prevista uma integração entre o novo módulo de Consultorias e Assessorias e o sistema legado de Relatórios de Acompanhamento, para que seja possível o acesso pelos consultores aos relatórios da empresa. O módulo de consultorias e assessorias terá acesso a todos os demais módulos da empresa pela mensageria Kafka, porém o módulo de relatórios faz uso do banco de dados já existentes dos outros sistemas, não enviando informações à mensageria, apenas recebendo para consultas em tempo real. Portanto, foi previsto um *end-point* SOAP que será consumido no módulo de Consultorias e Assessorias através de protocolo WS-*.

Essas integrações garantem o requisito não funcional de interoperabilidade.

4.2. Modelo de implantação

O diagrama de implantação ajuda a modelar a topologia de hardware do sistema, descrevendo a implementação física dos componentes de software e como a comunicação entre eles irá ocorrer, caracterizando também o que estará na modalidade *on premise* e o que estará em nuvem. O modelo de implantação da arquitetura está documentado e detalhado abaixo.

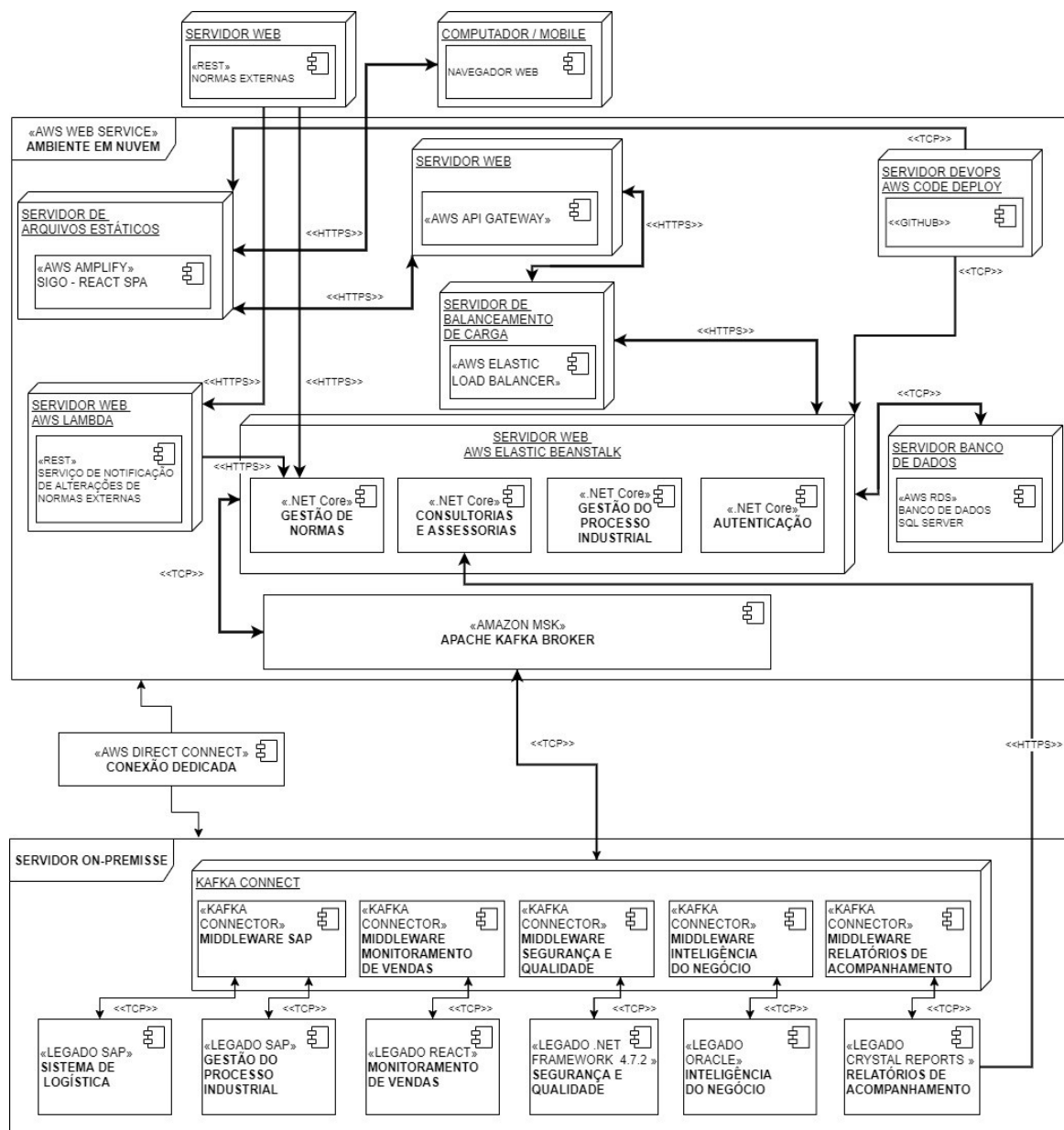


Figura 2 – Diagrama de implantação.

Componente	Descrição
Navegador Web	Representa os <i>browsers</i> utilizados para acesso das aplicações de <i>front-end</i> . Realizam a exibição do HTML para os usuários.
Servidor de arquivos estáticos	Representa servidores web que disponibilizam arquivos estáticos, como imagens, HTML, CSS e Javascript. Foi utilizado o serviço AWS Amplify, que provê grande disponibilidade e cache para recursos estáticos, reduzindo também a latência.
Servidor Web	Representa os servidores que conseguem prover conteúdo estático e dinâmico através do

	<p>protocolo HTTP/HTTPS. No caso das <i>API's</i> o servidor escolhido deverá ter suporte para o pipeline do .NET Core. A infraestrutura utilizada na POC é composta de servidores IIS, rodando em máquinas com S.O. Windows em um ambiente clusterizado do AWS ElasticBeanstalk. Existe a possibilidade de realizar a containerização desses artefatos. O servidor de normas externas não é de nosso controle. A premissa aqui é que esses servidores consigam comunicar-se via HTTPS. No caso do servidor Lambda, a estrutura é provisionada apenas quando uma alteração de norma for notificada, sendo encerrada em seguida.</p>
Servidor de balanceamento de carga	<p>Um servidor para balancear as requisições que chegam entre nós do <i>cluster</i> de servidores Web, diminuindo a latência. Na POC foi utilizado um recurso da AWS chamado AWS Elastic Load Balancer.</p>
Servidor de banco de dados	<p>Servidor Windows para fornecer instância de banco de dados SQL Server. Na POC foi considerada a solução da AWS, o AWS RDS. Essa solução já provê replicação e backup automáticos.</p>
Servidor <i>DevOps</i>	<p>Servidor para pipeline de integração contínua, construção, testes e entrega contínua fazendo uso da infraestrutura AWS Code Deploy e das ferramentas GitHub e GitHub Actions.</p>
Amazon MSK	<p><i>Cluster</i> do Apache Kafka utilizando do padrão <i>publish/subscribe</i> e de conectores para integração com os sistemas legados e os novos módulos.</p>

5. Prova de Conceito (POC) / protótipo arquitetural

5.1. Implementação e Implantação

5.1.1. Requisitos não funcionais

A prova de conceito teve como objetivo evidenciar e validar os requisitos não funcionais considerados mais importantes para esta arquitetura. O *link* do repositório de código da aplicação, assim como o *link* para acesso à aplicação em execução e o *link* para o vídeo de apresentação da POC, estão disponíveis no apêndice deste documento. Foram selecionados os seguintes requisitos não funcionais:

- **Segurança:** O sistema deve apresentar segurança no acesso e manipulação de dados, não permitindo o acesso aos módulos sem o usuário estar devidamente autenticado e autorizado.

Esse requisito não funcional foi escolhido devido à alta criticidade e preocupação com a segurança dos dados sensíveis do sistema.

Os critérios de aceite são:

- No navegador web, ao tentar acessar um recurso protegido sem a devida autenticação e autorização (*token* JWT), redirecionar o usuário para a tela de *login*.

- Nos *end-points* das *API's*, caso seja feita alguma requisição aos recursos privados, sem informar o *token* de acesso, retornar o código 401 (*unauthorized*).

- Uma requisição para o *end-point* da *API* de autenticação, deve retornar um *token* JWT apenas se forem informados usuário e senhas válidos.

- **Usabilidade:** O sistema deve promover uma usabilidade fácil, objetiva e intuitiva.

Esse requisito não funcional foi escolhido devido à necessidade de se navegar entre as funcionalidades do sistema de maneira fácil, objetiva e intuitiva. A usabilidade deve ser levada em consideração para que os usuários tenham uma boa experiência ao operar o sistema e possam focar em suas devidas tarefas, facilitando a tomada de decisões.

Os critérios de aceite são:

- A interface visual do sistema deve apresentar facilidade de navegação.

- O acesso às funcionalidades devem apresentar objetividade e não serem confusos.

- O usuário deve ser capaz de encontrar uma norma e acessar seu documento, com poucos cliques e em poucos segundos, de maneira simples e intuitiva.

- O usuário pode acessar qualquer módulo disponível em seu perfil a partir de qualquer outra tela do sistema.

- **Acessibilidade:** O sistema deve suportar ambientes web responsivos e ambientes móveis.

Esse requisito não funcional foi escolhido pois o sistema deve possuir uma interface responsiva, se adaptando a diversos tipos de tela e dispositivos, sejam computadores, *smartphones* ou *tablets*, proporcionando uma melhor experiência aos usuários, sendo possível efetuar qualquer operação, sem maiores dificuldades independente do dispositivo utilizado.

Os critérios de aceite são:

- Os componentes do *layout* do sistema devem apresentar responsividade, se adaptando aos diferentes tipos de telas, sem perder qualquer funcionalidade.

- A navegação entre as funcionalidades do sistema deve continuar fácil, objetiva e intuitiva.

- A identidade visual da aplicação deve ser mantida, independente do tamanho de tela do dispositivo utilizado.

- **Interoperabilidade:** O sistema deve se comunicar com os sistemas legados já existentes *on premise*. O sistema deve se comunicar com sistemas externos via *API's Restful* de integração.

Esse requisito não funcional foi escolhido pois é parte crucial na arquitetura desenvolvida. O ambiente em nuvem deve comunicar-se com os sistemas *on premise*, e estes, devem comunicar também entre si, provendo toda integração necessária. O sistema também deve comunicar-se com uma *API* externa de normas, buscando dados referentes às normas externas da empresa.

Os critérios de aceite são:

- O sistema deve conseguir produzir uma mensagem no canal de mensageria que será consumida pelos demais sistemas da empresa que se interessarem nesta informação.

- O sistema deve consultar dados referentes às normas externas via *API Restful* fornecida por um terceiro e exibir os dados em tela.

- **Desempenho:** O sistema deve fornecer um bom desempenho com um tempo de resposta adequado a cada funcionalidade.

Esse requisito não funcional foi escolhido pois a aplicação necessita de informações em tempo real. O requisito de desempenho tem impacto mais global sobre o sistema e, por essa razão, está entre os requisitos não funcionais mais importantes. Tarefas como busca, inclusão, atualização ou exclusão de registros devem apresentar um tempo de resposta rápido, independente da localização em que se está acessando o sistema. Adicionalmente, desempenho é importante porque afeta a usabilidade de um sistema.

Os critérios de aceite são:

- O cadastro, atualização, exclusão ou a busca de uma norma interna foi realizado com sucesso e com um tempo de resposta abaixo de 5 segundos.
- O cadastro, atualização ou exclusão de uma linha de produção do processo industrial foi enviado para canal de mensageria em menos de 2 segundos.

5.1.2. Casos de Uso

Os casos de uso implementados nessa prova de conceito, visando a validação dos requisitos não funcionais de maior prioridade, estão discriminados abaixo:

Módulo	Caso de uso	Requisito não funcional
Autenticação	Efetuar <i>login</i> no sistema	Segurança
Gestão de Normas	Buscar uma norma interna	Usabilidade / Acessibilidade / Desempenho
Gestão de Normas	Cadastrar uma norma interna	Usabilidade / Acessibilidade / Desempenho
Gestão de Normas	Alterar uma norma interna	Usabilidade / Acessibilidade / Desempenho
Gestão de Normas	Excluir uma norma interna	Usabilidade / Acessibilidade / Desempenho
Gestão de Normas	Buscar uma norma externa	Usabilidade / Acessibilidade / Desempenho / Interoperabilidade
Legado Gestão de Processos Industriais	Incluir nova linha de produção	Interoperabilidade / Desempenho

5.1.3. Tecnologias Utilizadas

Foram utilizadas as seguintes tecnologias para implementação da prova de conceito:

Caso de uso	Tecnologias
Efetuar <i>login</i> no sistema	React.js / .Net Core / AWS API Gateway / OAuth 2.0 / ADO.NET / AWS RDS (SQL Server)
Buscar uma norma interna	React.js / .Net Core / AWS API Gateway / ADO.NET / AWS RDS (SQL Server)
Cadastrar uma norma interna	React.js / .Net Core / AWS API Gateway / ADO.NET / AWS RDS (SQL Server)
Alterar uma norma interna	React.js / .Net Core / AWS API Gateway / ADO.NET / AWS RDS (SQL Server)
Excluir uma norma interna	React.js / .Net Core / AWS API Gateway / ADO.NET / AWS RDS (SQL Server)
Buscar uma norma externa	React.js / .Net Core
Incluir nova linha de produção	Apache Kafka / Kafka Connect

5.1.2. Implantação

A prova de conceito foi implantada parte na nuvem, utilizando da infraestrutura da Amazon, e parte local para simular uma linha de produção do processo industrial sendo criada. A mensageria estará toda configurada no serviço AWS MSK, porém para a POC, não foi utilizada a tecnologia AWS Direct Connect, portanto a mensageria Kafka foi simulada em um contêiner *docker* local, para mostrar a publicação de mensagens. Abaixo estão listadas todas as implementações e recursos de implantação:

Implementação	Recurso de Implantação
SIGO (<i>front-end</i>)	AWS Amplify
API Gestão de Normas	AWS ElasticBeansTalk / AWS API Gateway
API Gestão do Processo Industrial	AWS ElasticBeansTalk / AWS API Gateway
API Consultorias e Assessorias	AWS ElasticBeansTalk / AWS API Gateway
API Autenticação	AWS ElasticBeansTalk / AWS API Gateway
Mock serviço de normas externas	AWS ElasticBeansTalk
Mock inclusão de mensagens do processo industrial	Apache Kafka / Kafka Connect
Banco de dados SQL Server	AWS RDS

5.2 Interfaces/ API's

5.2.1. Consumidas

Para a implementação da prova de conceito foi feita a simulação de consumo de uma API dos fornecedores de normas externas. A API tem por objetivo simular uma consulta às normas externas da empresa. Essa API foi desenvolvida em .NET Core utilizando o estilo arquitetural Rest.

API	API REST para consulta de Normas Externas	
URL	https://wu63epzr79.execute-api.us-east-1.amazonaws.com/GestaoNormas/api/NormasExternas/	
Método	GET	
Headers	Authorization	token

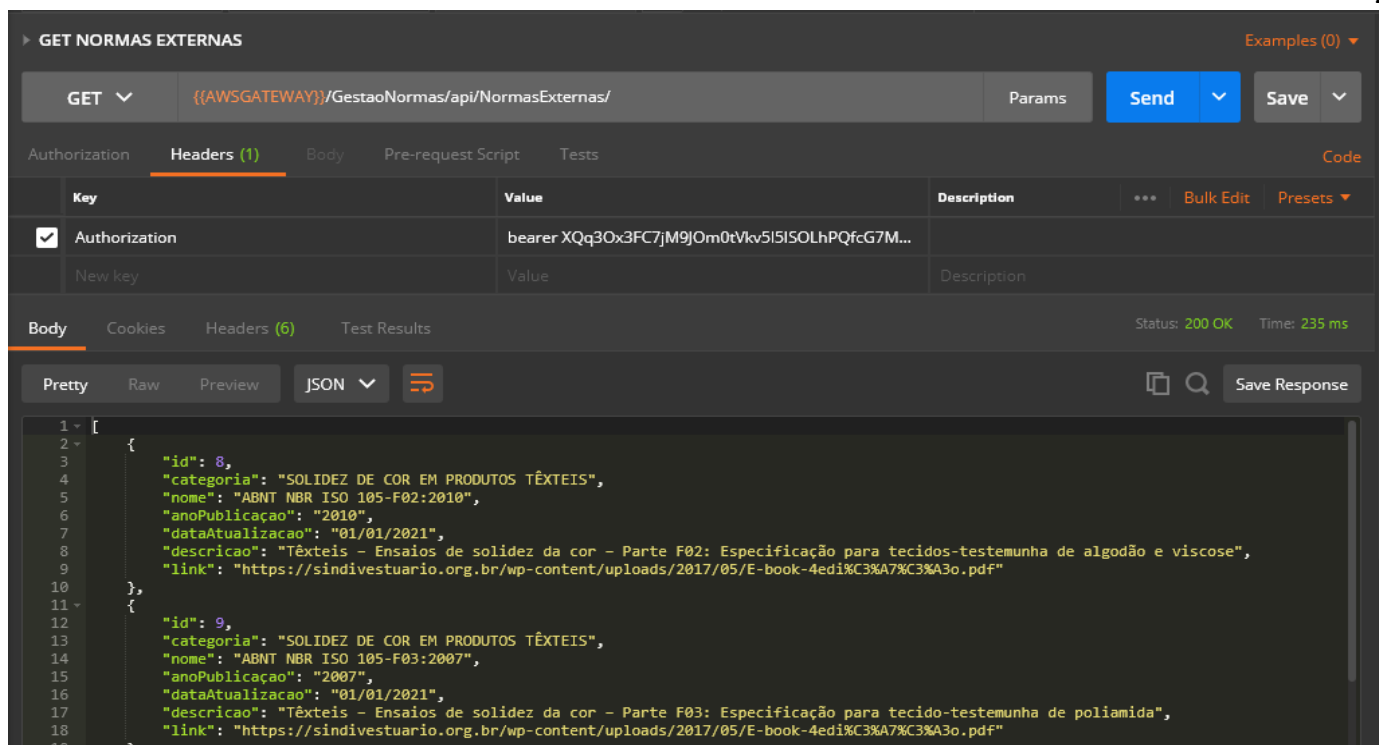


Figura 3 – Exemplo de *request* e *response* da *API* de consulta de normas externas.

5.2.2. Fornecidas

Na implementação da prova de conceito, foram criadas algumas *API's Restful*, sendo elas: *API* de Autenticação, *API* de Normas, *API* de Consultoria e Assessorias e *API* de Gestão do Processo Industrial, para operação de todo processo da indústria têxtil, conforme escopo deste documento.

API	API REST para consulta de Normas Internas	
URL	https://wu63epzr79.execute-api.us-east-1.amazonaws.com/GestaoNormas/api/Normas/	
Método	GET	
Headers	Authorization	token

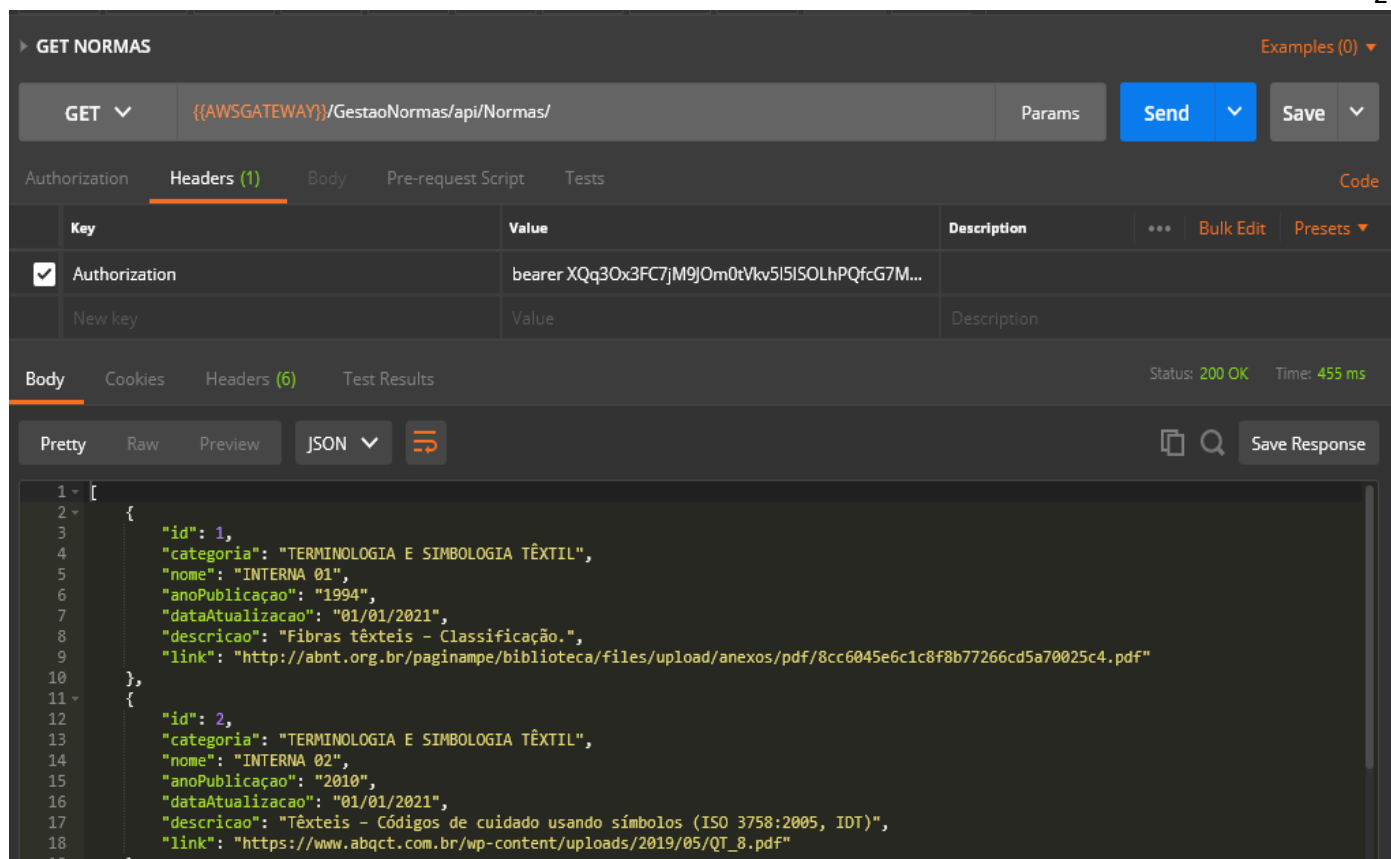


Figura 4 – Exemplo de *request* e *response* da API para consulta de normas internas.

API	API REST para cadastro de Norma Interna	
URL	https://wu63epzr79.execute-api.us-east-1.amazonaws.com/GestaoNormas/api/Normas/	
Método	POST	
Headers	Authorization	token
	Content-Type:	application/json
Body	categoria	Categoria da Norma
	nome	Nome da Norma
	anoPublicação	Data da publicação da norma
	descricao	Descrição da norma
	link	Link para acesso ao PDF da norma

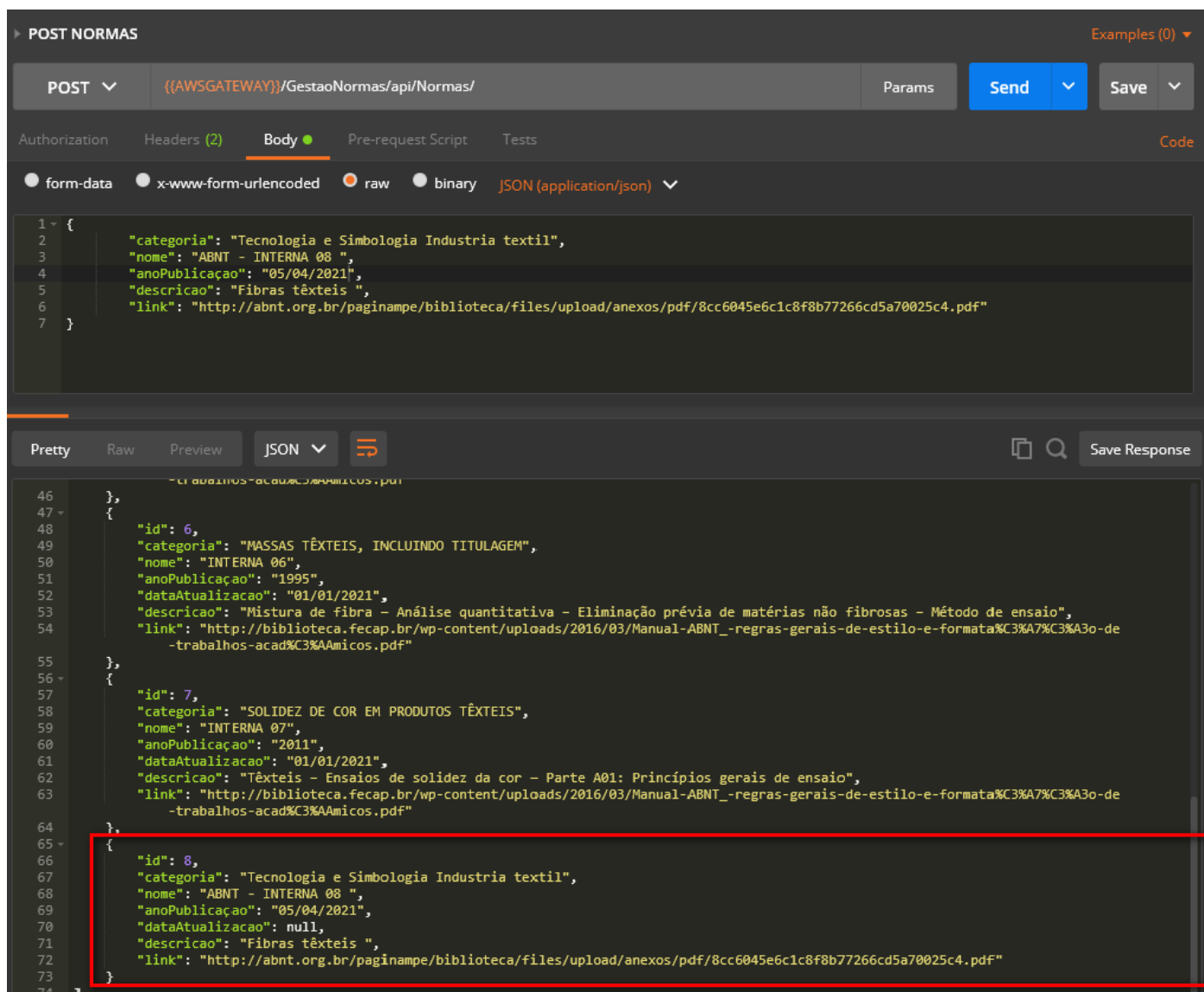


Figura 5 – Exemplo de *request* e *response* da API para cadastro de norma interna.

API	API REST para atualização de Norma Interna	
URL	https://wu63epzr79.execute-api.us-east-1.amazonaws.com/ GestaoNormas/api/Normas/1	
Método	PUT	
Headers	Authorization	token
	Content-Type:	application/json
Body	categoria	Categoria da Norma
	nome	Nome da Norma
	anoPublicação	Data da publicação da norma
	descricao	Descrição da norma

	link	Link para acesso ao PDF da norma
--	------	----------------------------------

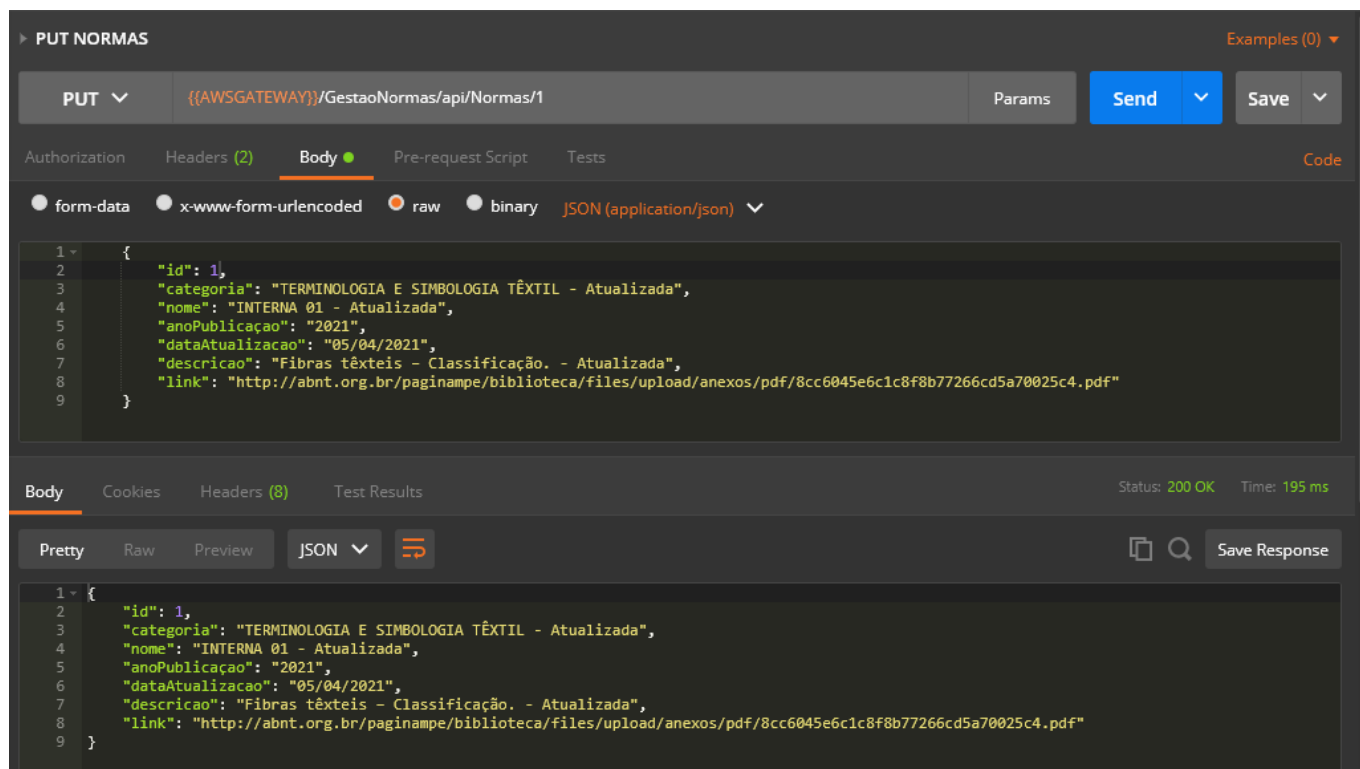


Figura 6 – Exemplo de *request* e *response* da API para atualização de norma interna.

API	API REST para exclusão de Norma Interna	
URL	https://wu63epzr79.execute-api.us-east-1.amazonaws.com/GestaoNormas/api/Normas/8	
Método	DELETE	
Headers	Authorization	token
	Content-Type:	application/json

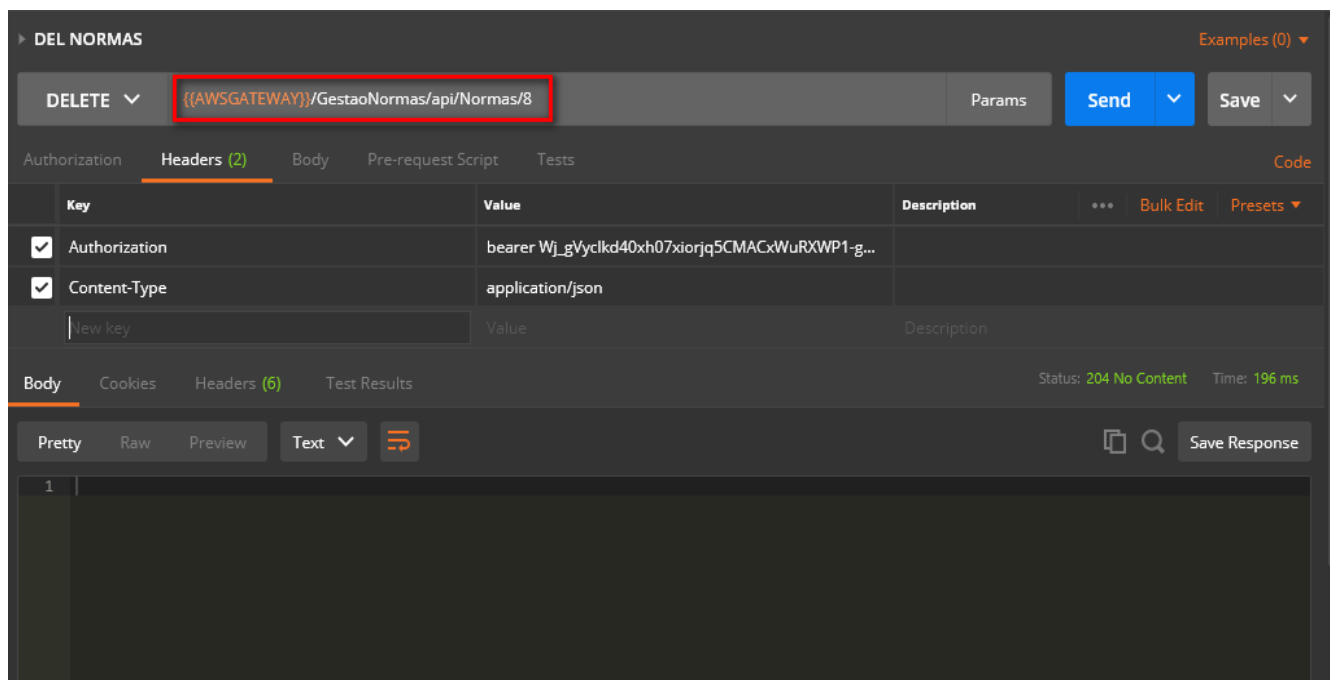


Figura 7 – Exemplo de *request* e *response* da API para exclusão de norma interna.

API	API REST para busca de Consultorias	
URL	https://wu63epzr79.execute-api.us-east-1.amazonaws.com/Consultoria-Assessoria/api/Consultorias/	
Método	GET	
Headers	Authorization	token

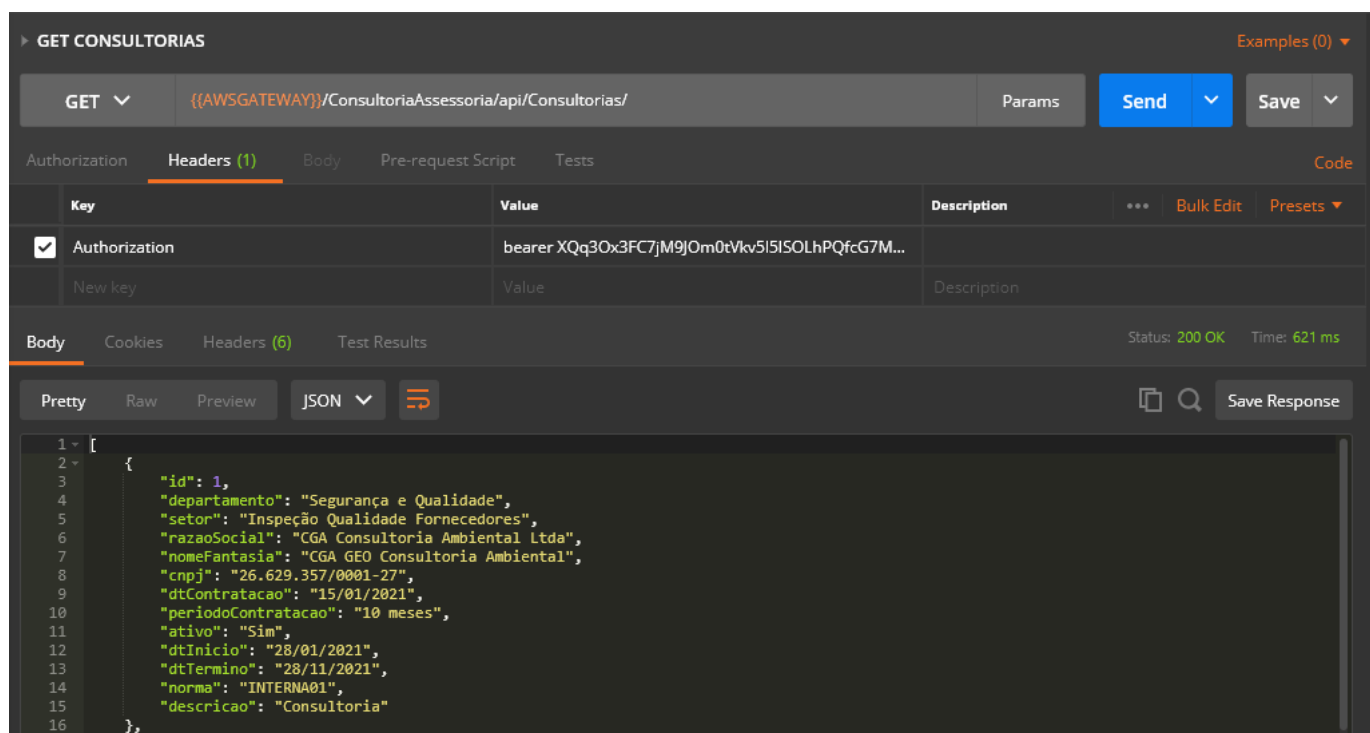


Figura 8 – Exemplo de *request* e *response* da API para busca de consultorias.

API	API REST para cadastro de Consultoria	
URL	https://wu63epzr79.execute-api.us-east-1.amazonaws.com/Consultoria-Assessoria/api/Consultorias/	
Método	POST	
Headers	Authorization	token
	Content-Type:	application/json
Body	departamento	Departamento que a empresa presta a Consultoria
	setor	Setor que a empresa presta a Consultoria
	razaoSocial	Nome razão social da empresa que presta a consultoria
	nomeFantasia	Nome fantasia da empresa que presta a Consultoria
	cnpj	Cnpj da empresa da consultoria
	dtContratacao	Data da contratação da consultoria
	periodoContratacao	Período da contratação da consultoria
	ativo	Consultoria está ativa ou não
	dtInicio	Data do início da consultoria
	dtTermino	Data do término da consultoria
	norma	Norma que a consultoria segue
	descricao	Descrição da consultoria

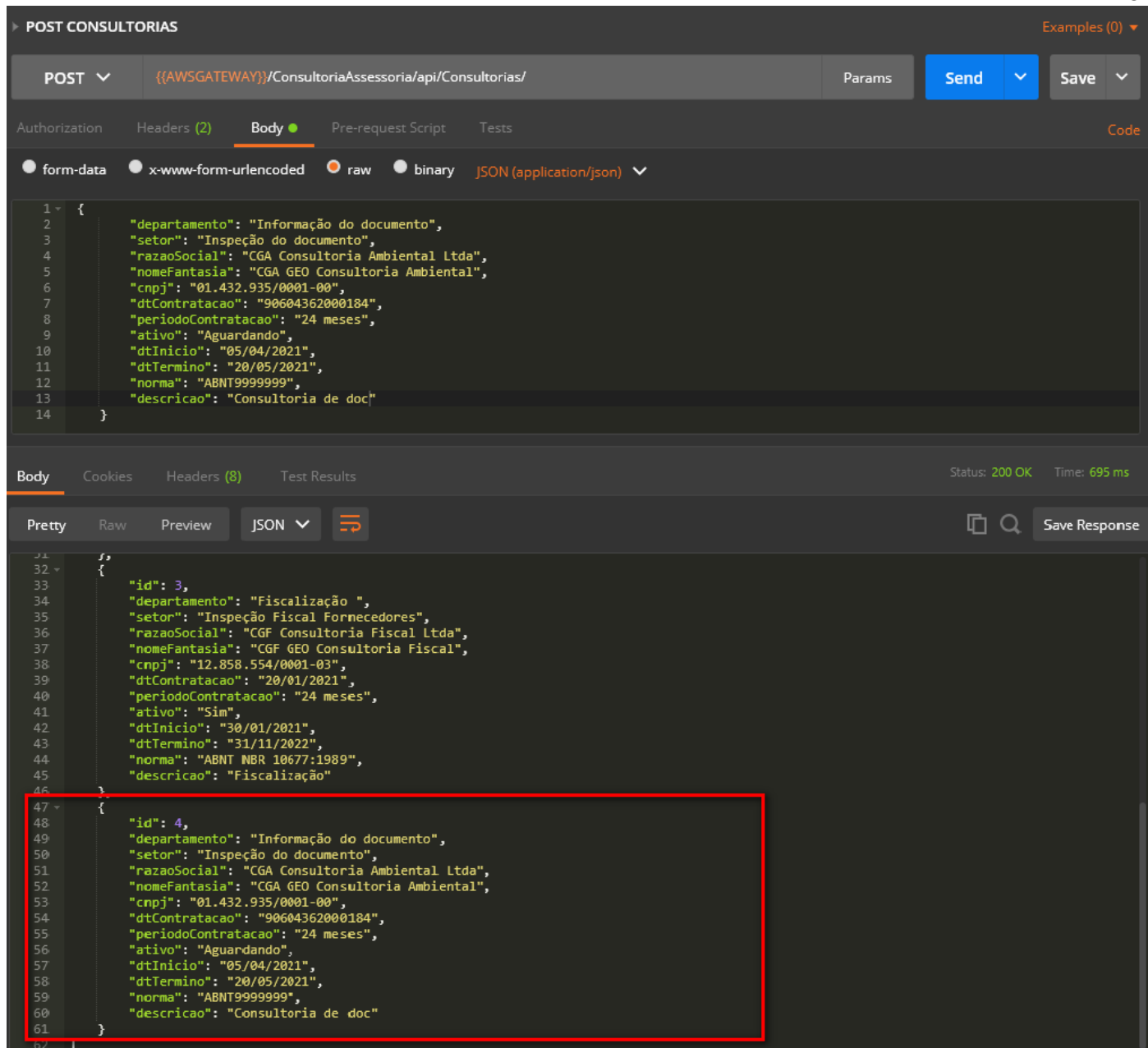


Figura 9 – Exemplo de *request* e *response* da API para cadastro de consultorias.

API	API REST para atualização de Consultoria	
URL	https://wu63epzr79.execute-api.us-east-1.amazonaws.com/Consultoria-Assessoria/api/Consultorias/1	
Método	PUT	
Headers	Authorization	token
	Content-Type:	application/json
Body	departamento	Departamento que a empresa presta a Consultoria
	setor	Setor que a empresa presta a Consultoria

	razaoSocial	Nome razão social da empresa que presta a consultoria
	nomeFantasia	Nome fantasia da empresa que presta a Consultoria
	cnpj	Cnpj da empresa da consultoria
	dtContratacao	Data da contratação da consultoria
	periodoContratacao	Período da contratação da consultoria
	ativo	Consultoria está ativa ou não
	dtInicio	Data do início da consultoria
	dtTermino	Data do término da consultoria
	norma	Norma que a consultoria segue
	descricao	Descrição da consultoria

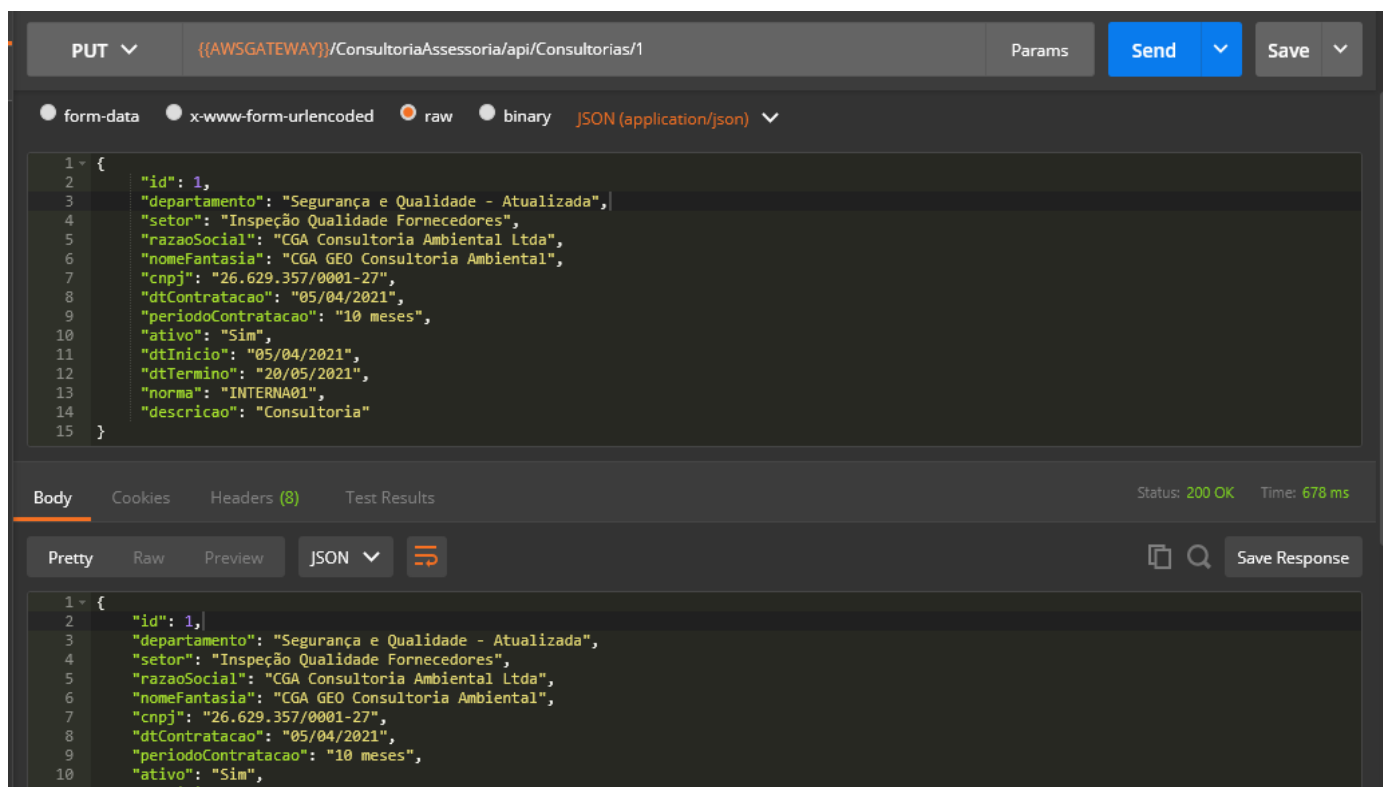


Figura 10 – Exemplo de *request* e *response* da API para atualização de consultorias.

API	API REST para exclusão de Consultoria
URL	https://wu63epzr79.execute-api.us-east-1.amazonaws.com/Consultoria-Assessoria/api/Consultorias/5
Método	DELETE

Headers	Authorization	token
	Content-Type:	application/json

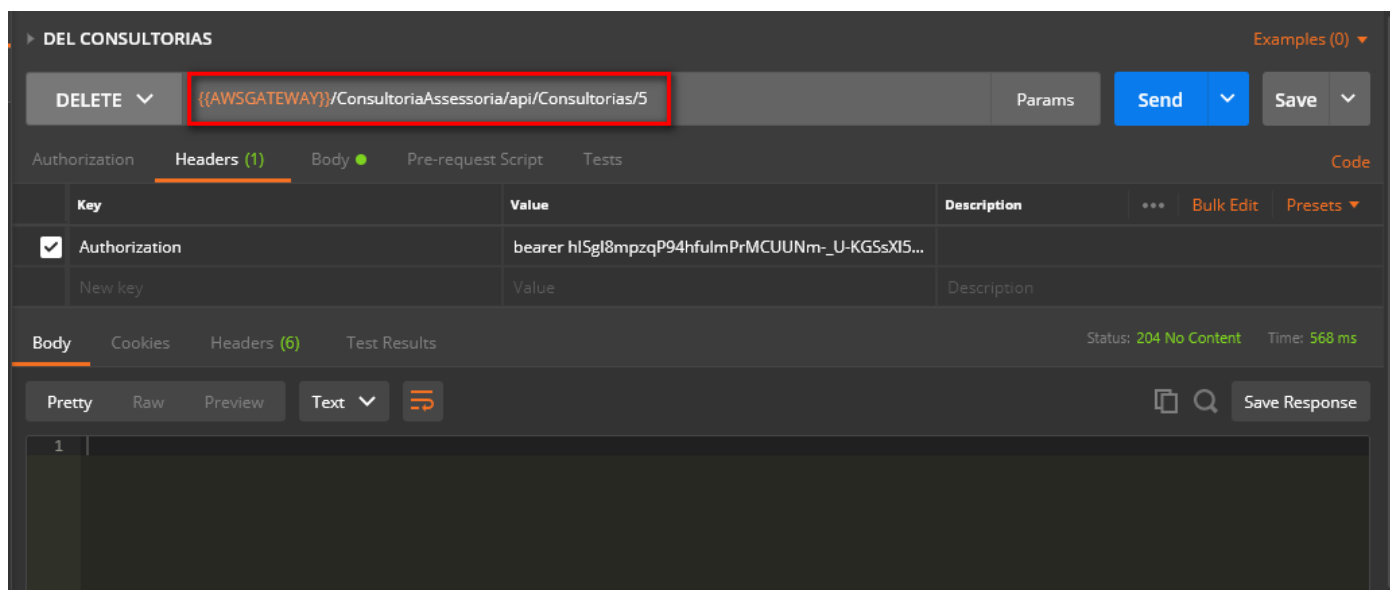


Figura 11 – Exemplo de *request* e *response* da API para exclusão de consultorias.

API	API REST para busca de Processos Industriais	
URL	<code>https://wu63epzr79.execute-api.us-east-1.amazonaws.com/GestaoProcessos/api/Processos/</code>	
Método	GET	
Headers	Authorization	token

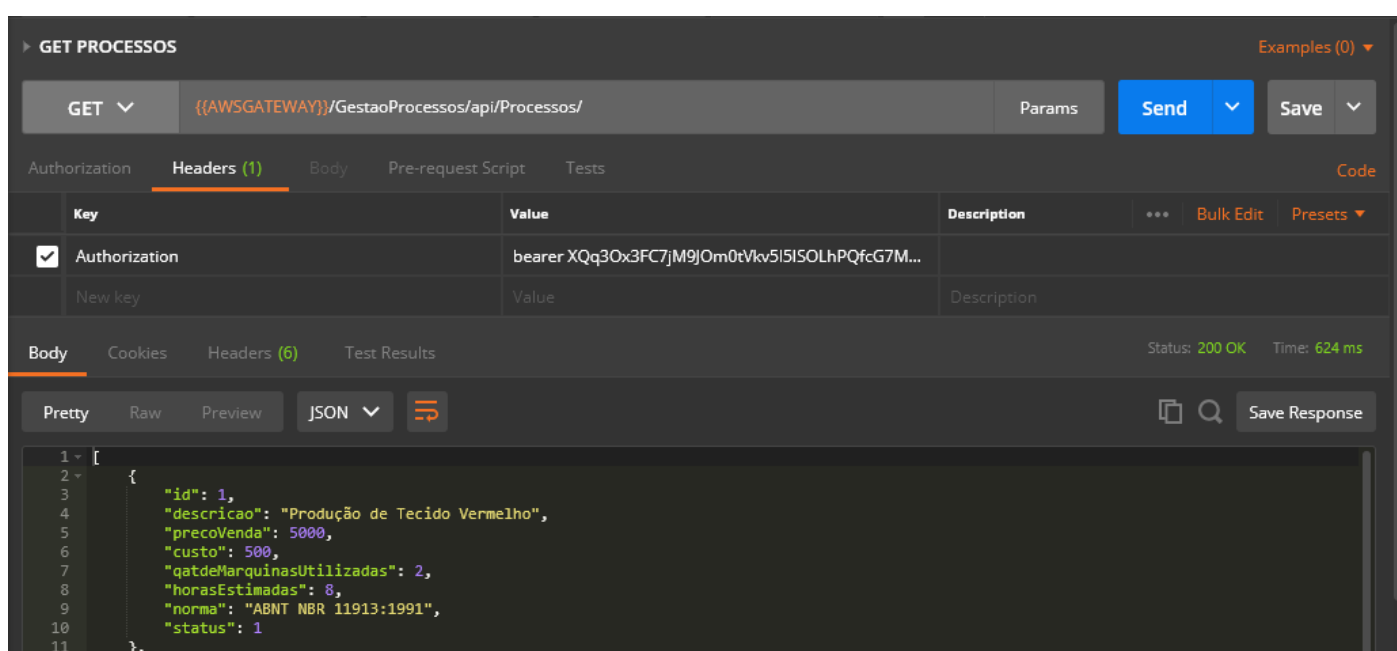


Figura 12 – Exemplo de *request* e *response* da API para consulta de processos industriais.

API	API REST para Autenticação	
URL	https://wu63epzr79.execute-api.us-east-1.amazonaws.com/token	
Método	POST	
Headers	Authorization	token
	Content-Type:	application/json
Body	username	Login do Usuário
	password	Senha do Usuário
	grant_type	password

POST `{{AWSGATEWAY}}/token` Params Send Save

form-data x-www-form-urlencoded raw binary

Key	Value	Description
<input checked="" type="checkbox"/> username	murilo@sigo.com.br	
<input checked="" type="checkbox"/> password	123456	
<input checked="" type="checkbox"/> grant_type	password	

Body Cookies Headers (11) Test Results Status: 200 OK Time: 470 ms

Pretty Raw Preview JSON Save Response

```

1 {
2   "access_token": "XQq30x3FC7jM9J0m0tVkv515IS0LhPQfcG7MxEj0Y7-Cpwq4UUZMeucEYq6ghImNrj
3     -sLpUAid7hXoZFKYX501urHRZqMp2eAkVcwMREs7gH0wT61C0oLWiEwo_1y4M_cprcv2in1X_EVfbBuMEHbpy9a1K667ahNY6_kmp7ZQShzKGeNB3eG8Qb5e8aDokPNdc_A7S9
4     cXuaP-5YU611IV1DxN6s5c5wg70n5LxuaSLT2tKNn_HH486ideNr0VEsWPomALkFqDs_YhGGsXgedeik0aF1Eg_T53Yg0CeAPQKbu9ZP6qhkprI1z4uULqXJ",
5   "token_type": "bearer",
6   "expires_in": 3599,
7   "Username": "murilo@sigo.com.br",
8   ".issued": "Mon, 05 Apr 2021 23:23:42 GMT",
9   ".expires": "Tue, 06 Apr 2021 00:23:42 GMT",
  "http://schemas.microsoft.com/ws/2008/06/identity/claims/role": "[\"Administrador\"]"
}

```

Figura 13 – Exemplo de *request* e *response* da API para autenticação.

6. Avaliação da Arquitetura

6.1. Análise das abordagens arquiteturais

A arquitetura proposta abrange vários componentes modulares, cada um com suas tecnologias e em diferentes recursos de implantação. Esta arquitetura foi projetada para rodar parte na nuvem e parte *on premise*, e embora faça uso de uma estrutura específica em nuvem, a implementação foi desenvolvida para ser o mais independente possível de qualquer plataforma. A integração entre os sistemas foi projetada para que todos estes, novos ou já existentes, tenham suas informações disponibilizadas mutuamente. Desta forma favorecendo a simplicidade de possíveis migrações futuras dos sistemas legados para a nuvem, sem prejuízos da operação do sistema como um todo. Além disso foi disposta uma esteira *DevOps* para tarefas de construção, testes automatizados, integração contínua e entrega contínua, automatizando a entrega de novas funcionalidades e implementações, além de proporcionar maior garantia de qualidade.

6.2. Cenários

Os cenários descritos abaixo são os utilizados para testar os requisitos não funcionais listados na seção 5.1.1: Segurança, acessibilidade, usabilidade, interoperabilidade e desempenho.

Cenário 1: No navegador web, ao tentar acessar um recurso protegido sem a devida autenticação, o usuário será redirecionado para a tela de *login*. Um recurso protegido não deve ser acessível sem o usuário estar autenticado.

Cenário 2: Ao tentar acessar algum recurso protegido diretamente pelo seu *end-point*, o usuário é obrigado a passar um *token* válido no cabeçalho, caso o *token* não seja válido ou não exista, então deverá ser retornado o código 401 (unauthorized). O *token* é obtido pelo *end-point* de autenticação, apenas se forem informados usuário e senha válidos no corpo da requisição.

Cenário 3: Ao navegar entre as telas, o sistema deve apresentar uma usabilidade fácil, objetiva e intuitiva. O usuário acessa qualquer módulo disponível em seu perfil a partir de qualquer outra tela do sistema, com menos de 3 cliques e em menos de 10 segundos. O usuário acessa o módulo de normas externas, encontra uma norma externa e acessa seu documento com poucos cliques e em menos de 30 segundos.

Cenário 4: Ao realizar o acesso na aplicação através de um dispositivo móvel ou qualquer outro com resoluções diferentes de telas, os componentes visuais são redimensionados adequando-se à resolução do dispositivo, sem perder qualquer funcionalidade ou identidade visual.

Cenário 5: Ao realizar operações de inclusão, atualização, exclusão ou busca de uma norma interna, o usuário é capaz de realizar cada operação em menos de 1 minuto e com tempo de resposta abaixo de 5 segundos.

Cenário 6: O sistema deve produzir uma mensagem no canal de mensageria que será consumida pelos demais sistemas da empresa que se interessarem nesta informação. A inclusão de uma linha de produção do processo industrial é enviada para canal de mensageria em menos de 2 segundos. A mensageria foi documentada para estar toda configurada no serviço AWS MSK, porém para a POC, não foi utilizada a tecnologia AWS Direct Connect, portanto a mensageria Kafka foi simulada em um contêiner *docker* local, para mostrar a publicação de mensagens.

Na priorização foi utilizado o método de árvore de utilidades reduzida e com prioridades. Os cenários foram categorizados de acordo os atributos de qualidade a que estão relacionados, e então foram classificados em função de sua importância e complexidade, considerando a percepção de negócio e arquitetura. As duas variáveis de priorização “Importância” e “Complexidade”, apresentadas nas colunas IMP. e COM. respectivamente foram classificadas em alta (A), média (M) e baixa (B).

Categoria	Cenários	IMP.	COM.
Segurança	Cenário 1: Acesso às páginas privadas via login	A	A
Segurança	Cenário 2: Acesso a recursos privados via <i>token</i>	A	A
Usabilidade e Interoperabilidade	Cenário 3: O sistema deve apresentar uma usabilidade fácil, objetiva e intuitiva. O sistema deve se comunicar com sistemas externos via <i>API's Restful</i> de integração	A	M
Acessibilidade	Cenário 4: Responsividade das telas da aplicação	M	A
Desempenho	Cenário 5: As requisições devem ser atendidas em até 5 segundos	A	M
Interoperabilidade e Desempenho	Cenário 6: O sistema deve se comunicar com os sistemas legados já existentes <i>on premise</i> . A inclusão de informações no canal de mensageria deve apresentar um bom desempenho com um tempo de resposta adequado.	A	A

6.3. Avaliação

O processo de avaliação para validar os cenários do item 6.2 será descrito nos quadros de cada cenário abaixo com o objetivo de determinar os riscos, não riscos, pontos de sensibilidade, *trade-off's* e evidenciar o atendimento aos requisitos de qualidade.

- Cenário 1**

Atributo de qualidade:	Segurança
Requisito de qualidade:	Acesso às páginas privadas via login.
Preocupação	
Impedir o acesso às páginas privadas sem autenticação no sistema.	

Cenário 1	
Ambiente:	
Produção, carga normal.	
Estímulo:	
Usuário tentando acessar uma página privada sem estar devidamente autenticado no sistema.	
Mecanismo:	
Criar um mecanismo de validação de credenciais (<i>token</i> de acesso) e suas permissões associadas a cada perfil permitindo ou restringindo o acesso a páginas do sistema.	
Medida de resposta:	
O usuário deve ser redirecionado para tela de login.	
Considerações sobre a arquitetura:	
Riscos:	O gerenciamento de autenticação e autorização são pontos críticos para a segurança de sistemas. Falhas neste requisito podem causar vazamento de credenciais e informações sensíveis. Uma maneira de proteger <i>API's</i> de acesso indevidos é a utilização de JWT, que combinada com o bom gerenciamento dos algoritmos e chaves para assinalar o <i>token</i> , se mostra simples, rápida e segura.
Pontos de sensibilidade:	O Servidor da aplicação opera no protocolo HTTPS para garantir maior segurança.
Trade-off:	Não há.

• Evidências do cenário 1:

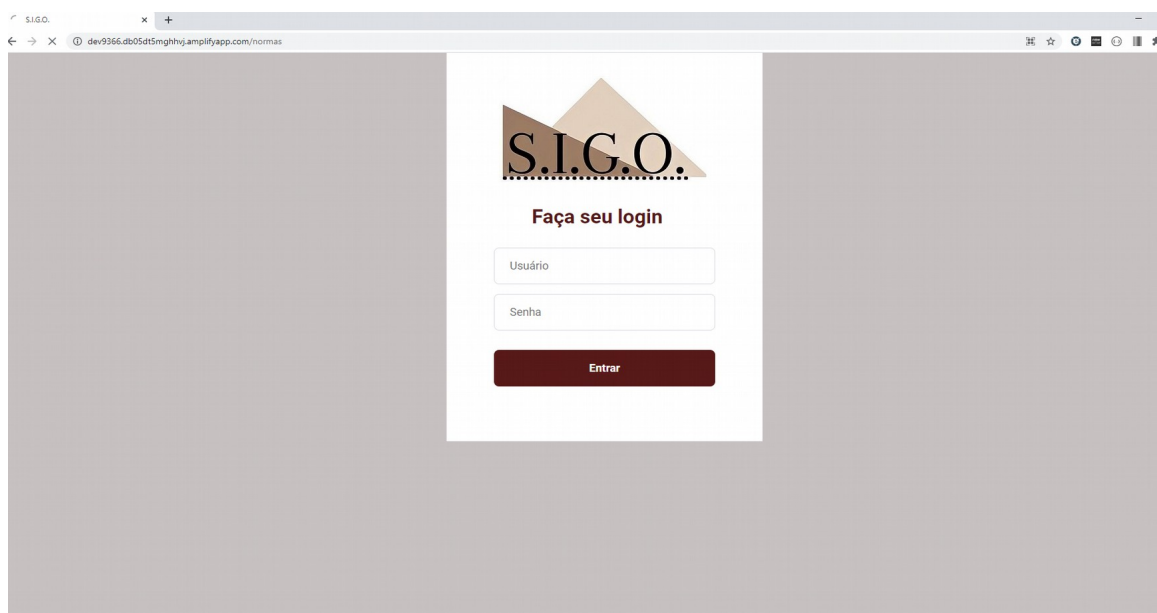


Figura 14 – Usuário tentando acessar uma tela protegida do sistema sem estar autenticado e sendo redirecionado para a tela de *login*.



Figura 15 – Usuário com acesso ao sistema, após ter feito o *login*.

- **Cenário 2**

Atributo de qualidade:	Segurança
Requisito de qualidade:	Acesso a recursos privados via token.
Preocupação	
Impedir o acesso aos recursos do sistema sem um <i>token</i> válido.	
Cenário 2	
Ambiente:	
Produção, carga normal.	
Estímulo:	
Usuário tentando acessar um recurso privado sem passar um <i>token</i> de acesso válido no cabeçalho da requisição.	
Mecanismo:	
Criar um mecanismo de validação de credencias (<i>token</i> de acesso) e suas permissões associadas a cada perfil permitindo ou restringindo o acesso a recursos do sistema.	
Medida de resposta:	
O sistema deve retornar o código 401 (<i>unauthorized</i>) como resposta. O sistema deve retornar um <i>token</i> JWT apenas se forem informados usuário e senhas válidos.	
Considerações sobre a arquitetura:	
Riscos:	O gerenciamento de autenticação e autorização são pontos críticos para a segurança de sistemas. Falhas neste requisito podem causar vazamento de credenciais e informações sensíveis. Uma maneira de proteger <i>API's</i> de acesso indevidos é a utilização de JWT, que combinada com o bom gerenciamento dos algoritmos e chaves para assinalar o <i>token</i> , se mostra simples, rápida e segura.
Pontos de sensibilidade:	O Servidor da aplicação opera no protocolo HTTPS para garantir maior segurança.

- Cenário 3

Atributo de qualidade:	Usabilidade e Interoperabilidade
Requisito de qualidade:	O sistema deve apresentar uma usabilidade fácil, objetiva e intuitiva. O sistema deve se comunicar com sistemas externos via <i>API's Restful</i> de integração.
Preocupação	
O acesso às funcionalidades devem apresentar objetividade e não serem confusos. O usuário deve ser capaz de encontrar uma norma externa e acessar seu documento, além de acessar qualquer outro módulo disponível em seu perfil a partir de qualquer outra tela do sistema, com poucos cliques e em poucos segundos.	
Cenário 3	
Ambiente:	
Produção, carga normal.	
Estímulo:	
Usuário tentando acessar uma norma externa.	
Mecanismo:	
Criar telas objetivas e intuitivas, todo conteúdo estático (HTML, JavaScript e CSS) é carregado no início da navegação, se comunicando com o <i>back-end</i> com dados em formato JSON. Um menu lateral de navegação está disponível em todas as telas. A <i>API</i> de Gestão de Normas faz uma chamada à <i>API</i> externa de normas.	
Medida de resposta:	
O usuário acessa o módulo de normas, encontra uma norma externa e acessa seu documento com poucos cliques e em menos de 30 segundos. O usuário acessa qualquer módulo disponível em seu perfil a partir de qualquer outra tela do sistema, com menos de 3 cliques e em menos de 10 segundos.	
Considerações sobre a arquitetura:	
Riscos:	Alto número de requisições pode levar a uma instabilidade no sistema até que o balanceador de cargas o normalize. A <i>API</i> de normas externas pode estar indisponível por problemas no lado do fornecedor da mesma.
Pontos de sensibilidade:	Carga excessiva de <i>requests</i> , balanceador de carga ativo.
Trade-off:	Não há.

- Evidências do cenário 3:



Figura 19 – Usuário acessando a página inicial do sistema, com barra lateral disponível para navegação.



Figura 20 – Usuário acessando a página de gestão de normas, com barra lateral disponível para navegação.

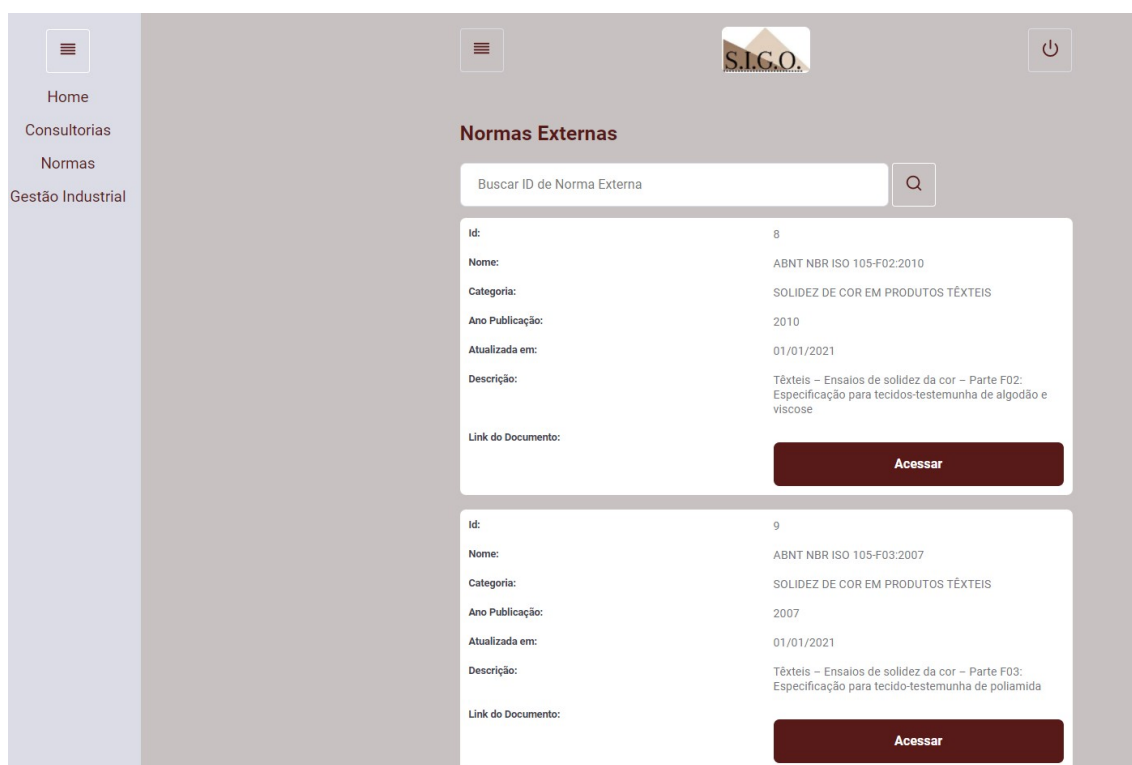


Figura 21 – Usuário acessando a página de gestão de normas externas, com informações da API externa, com barra lateral disponível para navegação e botão de acesso ao link da norma.

- **Cenário 4**

Atributo de qualidade:	Acessibilidade
Requisito de qualidade:	Responsividade das telas da aplicação.
Preocupação	
Os componentes do <i>layout</i> do sistema devem apresentar responsividade, se adaptando aos diferentes tipos de telas, sem perder qualquer funcionalidade, mantendo a objetividade e a identidade visual do sistema.	
Cenário 4	
Ambiente:	
Produção, carga normal.	
Estímulo:	
Usuário acessando o sistema em diferentes formatos de telas e em diferentes dispositivos.	
Mecanismo:	
Criar um <i>layout</i> responsivo, capaz de se adequar automaticamente conforme resolução da tela do	

dispositivo utilizado.	
Medida de resposta:	
Ao realizar o acesso na aplicação através de um dispositivo móvel ou qualquer outro com resoluções diferentes de telas, os componentes visuais são redimensionados adequando-se à resolução do dispositivo, sem perder qualquer funcionalidade ou identidade visual.	
Considerações sobre a arquitetura:	
Riscos:	Experiencia pode ser prejudicada ao utilizar dispositivos ou telas mais antigas, com baixa resolução ou qualidade de imagem. A experiência do usuário pode ser prejudicada se a qualidade de conexão da rede que estiver utilizando for muito baixa.
Pontos de sensibilidade:	Não há.
Trade-off:	Não há.

• Evidências do cenário 4:



Figura 22 – Usuário acessando a página inicial do sistema em uma resolução de 1920 x 1080.



Figura 23 – Usuário acessando a página inicial do sistema em uma resolução de 360 x 640 (Moto G4).



Figura 24 – Usuário acessando a página inicial do sistema, exibindo a barra lateral de navegação, em uma resolução de 360 x 640 (Moto G4).

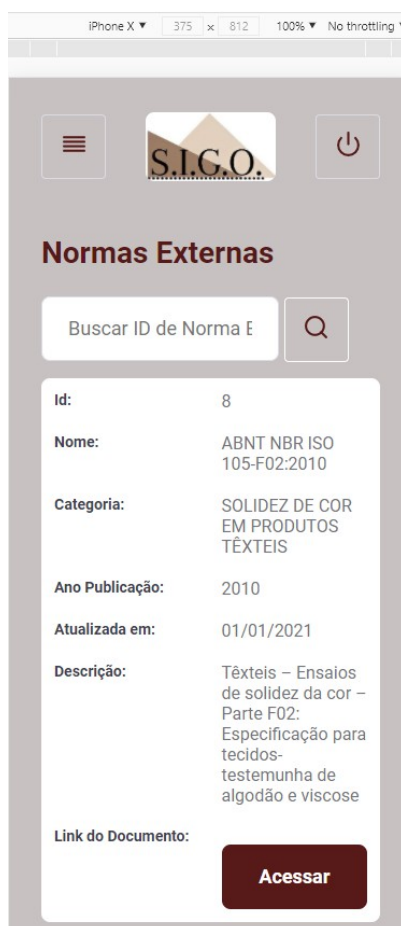


Figura 25 – Usuário acessando a página de normas externas, em uma resolução de 375 x 812 (iPhone X).

- **Cenário 5**

Atributo de qualidade:	Desempenho
Requisito de qualidade:	As requisições devem ser atendidas em até 5 segundos.
Preocupação	
Ao realizar operações de inclusão, atualização, exclusão ou busca de uma norma interna, o usuário deve ser capaz de realizar cada operação em menos de 1 minuto e com tempo de resposta abaixo de 5 segundos.	
Cenário 5	
Ambiente:	
Produção, carga normal.	
Estímulo:	
Usuário realizando operações de inclusão, busca, edição e atualização de normas internas.	
Mecanismo:	
Criar um mecanismo de comunicação com <i>back-end</i> apenas em formato JSON, para que o mesmo fique responsável pelas regras de negócio e pelo envio de informações, agilizando a troca de informações, além de um balanceador de cargas com objetivo de prover alta disponibilidade e escalabilidade automática. Criar uma API de <i>back-end</i> altamente disponível, robusta e segura para processamento dos dados e regras de negócio.	
Medida de resposta:	
O cadastro, atualização, exclusão ou a busca de uma norma interna foi realizado com sucesso e com um tempo de resposta abaixo de 5 segundos.	
Considerações sobre a arquitetura:	
Riscos:	Alto número de requisições pode levar a uma instabilidade no sistema até que o balanceador de cargas o normalize.
Pontos de sensibilidade:	Carga excessiva de <i>requests</i> , balanceador de carga ativo.
Trade-off:	Não há.

- **Evidências do cenário 5:**

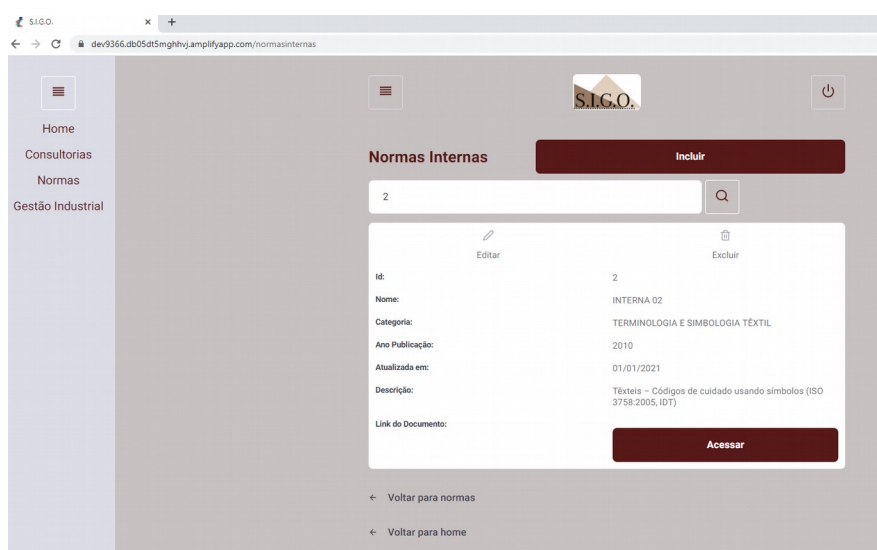


Figura 26 – Usuário efetuando a busca de uma norma interna.

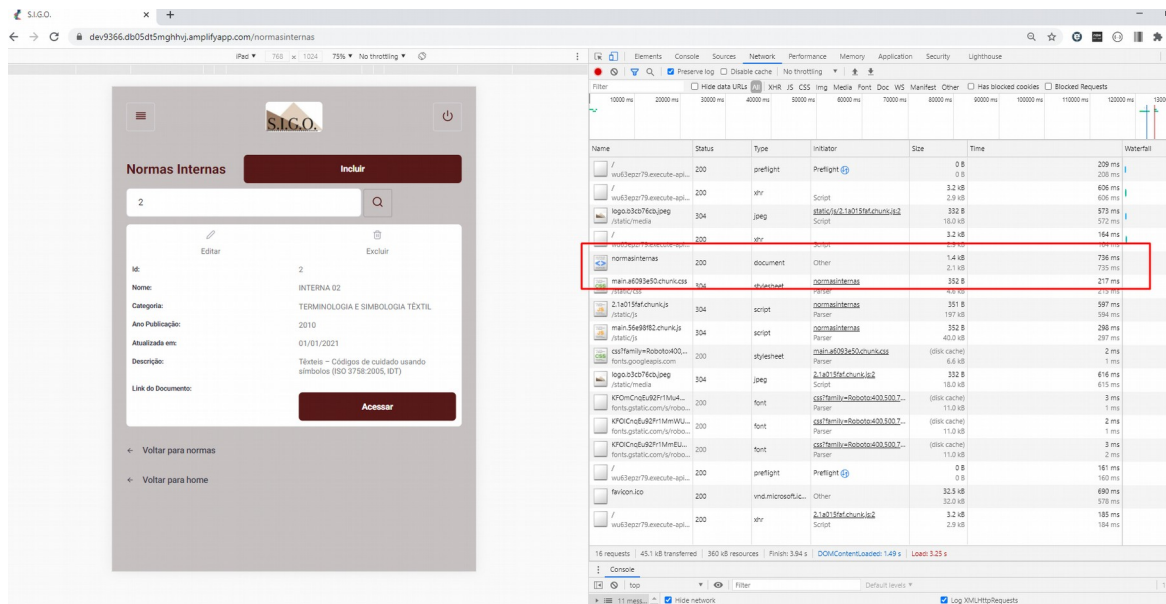


Figura 27 – Tempo de resposta da requisição de busca de uma norma interna.

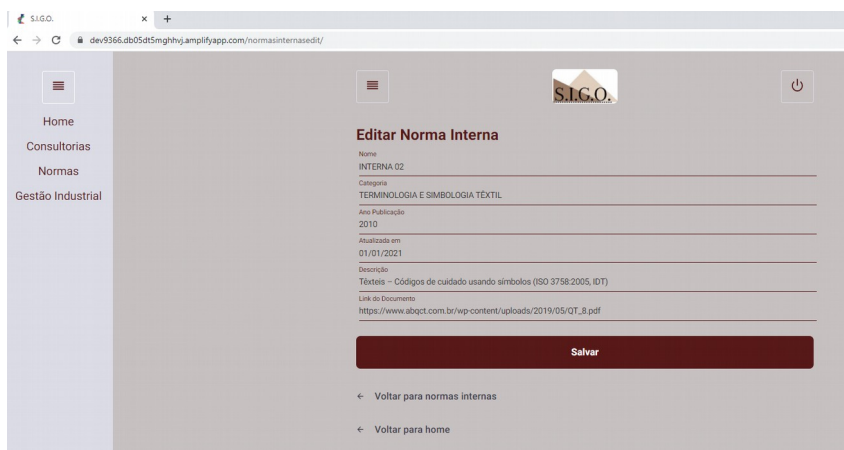


Figura 28 – Usuário efetuando a atualização de uma norma interna.

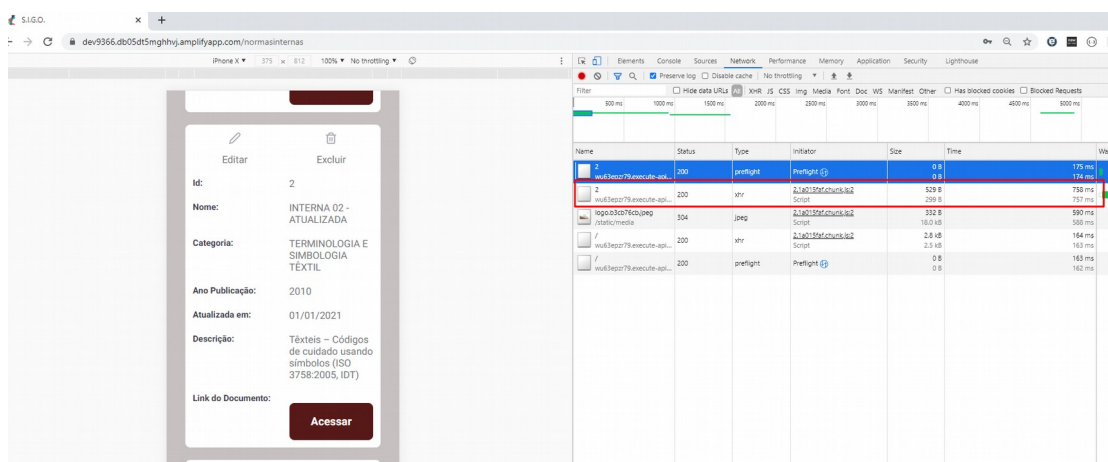


Figura 29 – Tempo de resposta da requisição de atualização de uma norma interna.

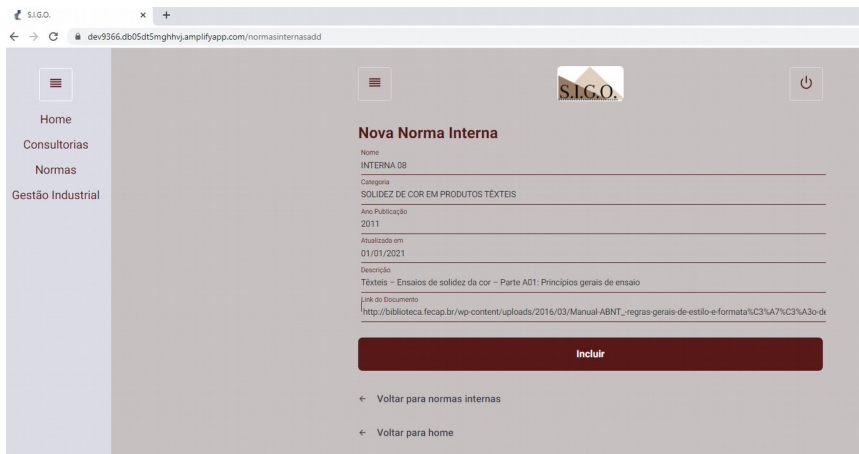


Figura 30 – Usuário efetuando o cadastro de uma norma interna.

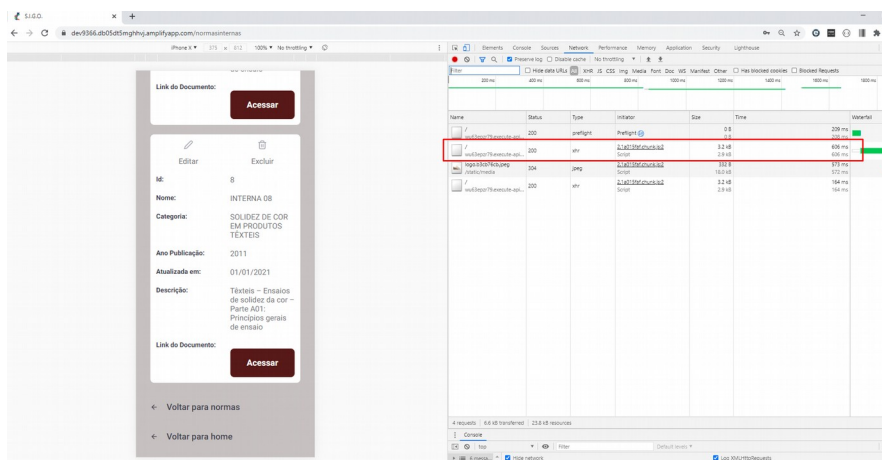


Figura 31 – Tempo de resposta da requisição de inclusão de uma norma interna.

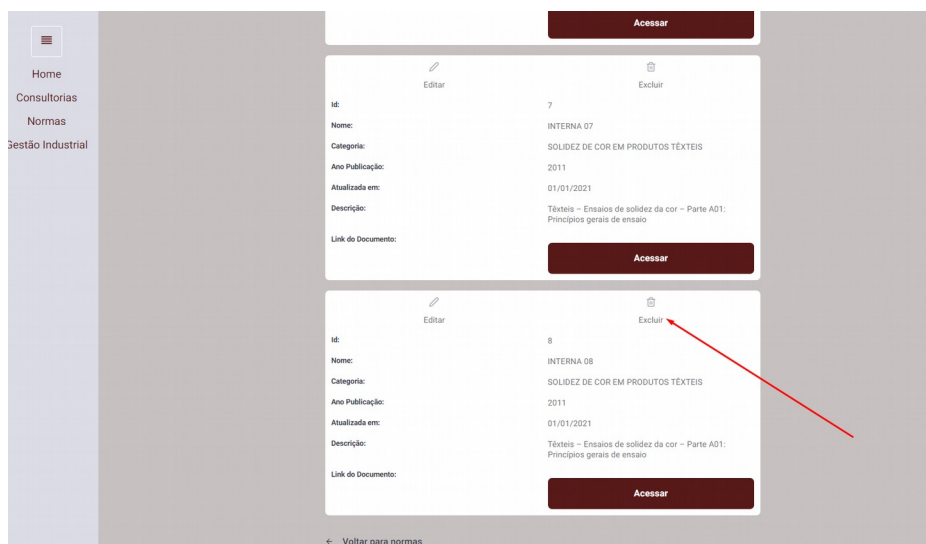


Figura 32 – Usuário efetuando a exclusão de uma norma interna.

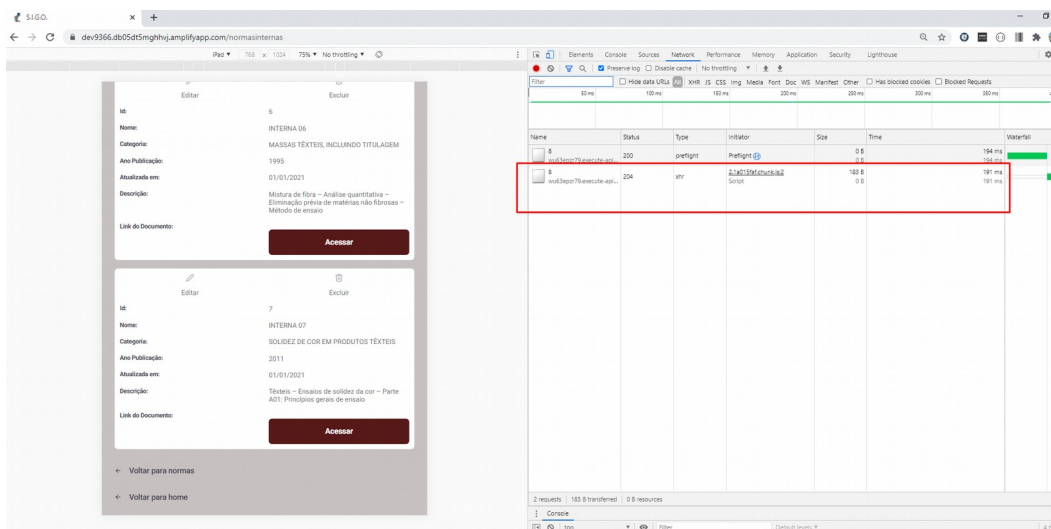


Figura 33 – Tempo de resposta da requisição de exclusão de uma norma interna.

- **Cenário 6**

Atributo de qualidade:	Interoperabilidade e Desempenho
Requisito de qualidade:	O sistema deve se comunicar com os sistemas legados já existentes <i>on premise</i> . A inclusão de informações no canal de mensageria deve apresentar um bom desempenho com um tempo de resposta adequado.
Preocupação	
Ao ser criada uma nova linha de produção industrial, a informação deve ser enviada para o canal de mensageria em seu devido tópico, para consumo pelos demais módulos interessados.	
Cenário 6	
Ambiente:	
Produção, carga normal.	
Estímulo:	
Sistema legado de gestão industrial criando uma nova linha de produção.	
Mecanismo:	
Criar um mecanismo de mensageria para comunicação entre os novos módulos e os sistemas legados da empresa. O Apache Kafka foi implementado para processar os fluxos de dados, praticamente em tempo real. Para a POC, não foi utilizada a tecnologia AWS Direct Connect, portanto a mensageria Kafka foi simulada em um contêiner <i>docker</i> local, para mostrar a publicação de mensagens.	
Medida de resposta:	
O sistema deve conseguir produzir uma mensagem no canal de mensageria em menos de 2 segundos.	
Considerações sobre a arquitetura:	
Riscos:	Baixa latência da rede ou indisponibilidade do serviço Kafka podem afetar o tempo processamento das mensagens.
Pontos de sensibilidade:	Amazon MSK e AWS Direct Connect ativos.
Trade-off:	Poderiam ser alterados os sistemas legados para o envio e recebimento de mensagens sem conectores Kafka. Ainda poderiam ser prevista a integração sem mensageria, por meio de API's.

- **Evidências do cenário 6:**

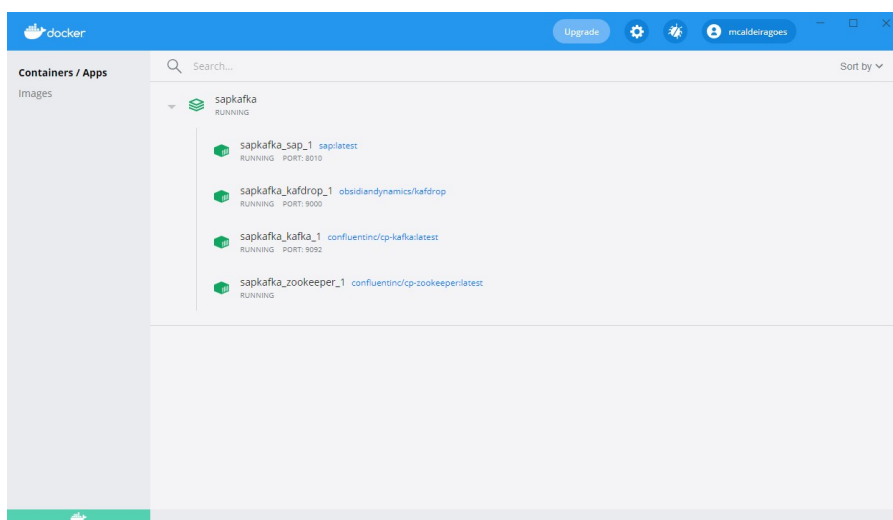


Figura 34 – Contêineres *docker* para simulação do Kafka e do conector com o sistema legado de gestão de processos industriais (SAP).

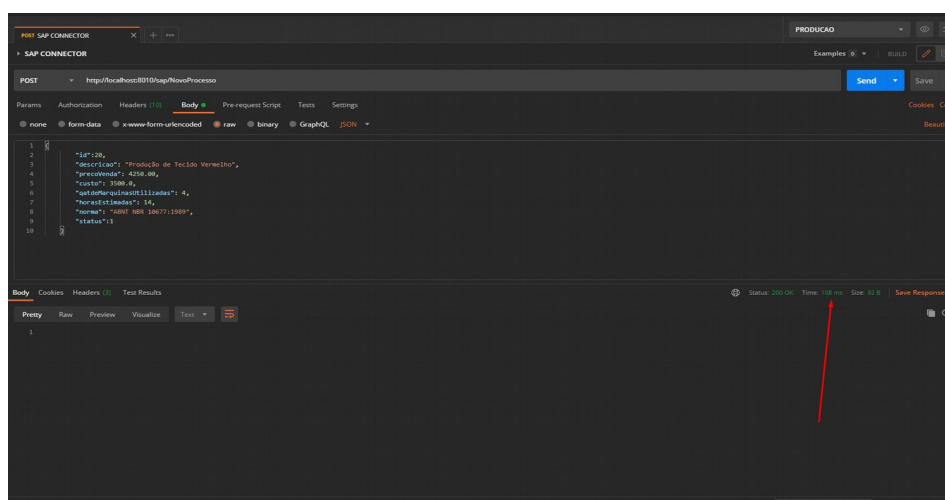


Figura 35 – Simulação de criação de linha de produção pelo sistema legado de gestão de processos industriais.

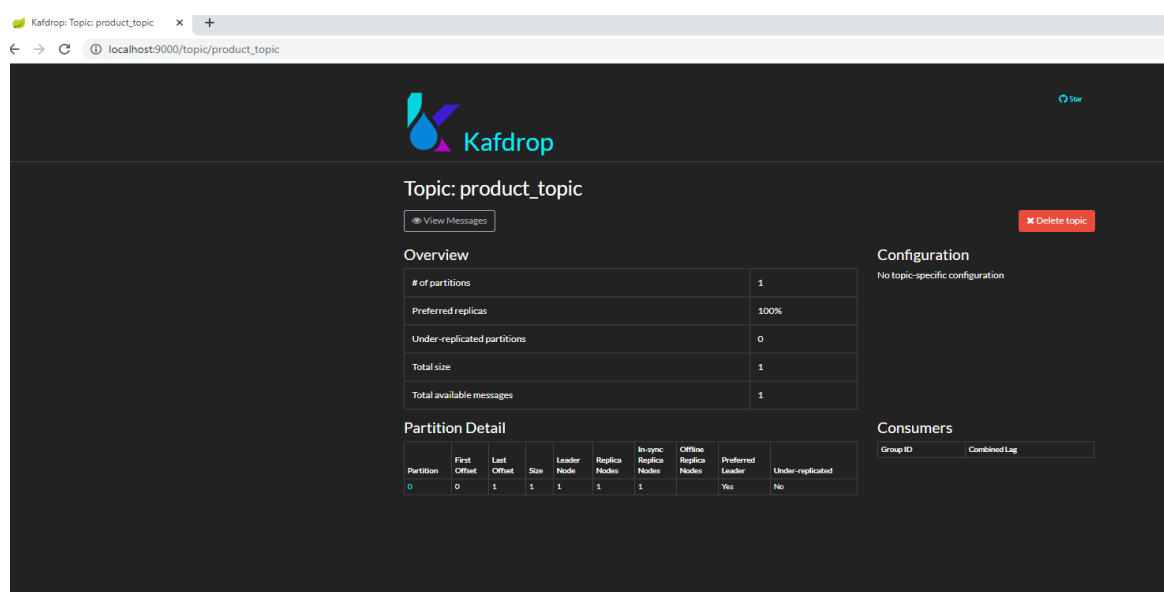


Figura 36 – Tópico da mensageria Kafka com as informações da linha de produção criada.

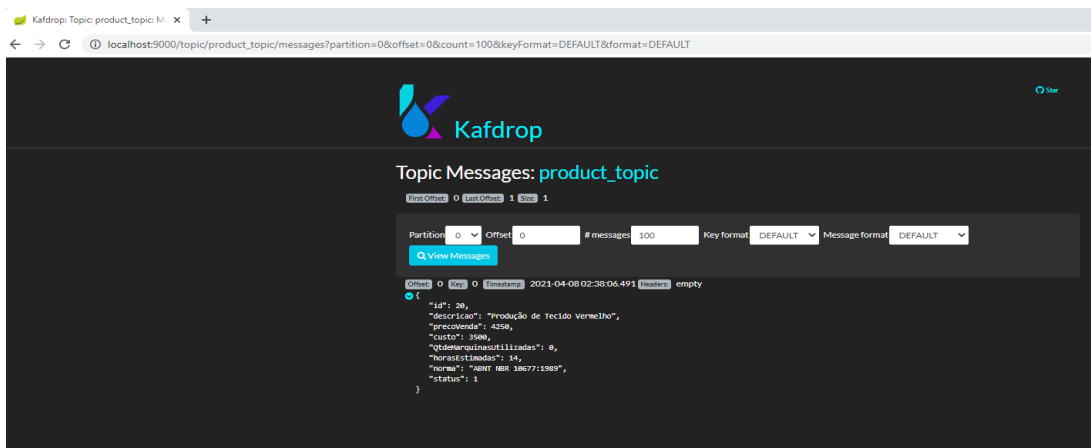


Figura 37 – Mensagem enviada ao Kafka com as informações da linha de produção criada.

6.4. Resultado

Considerando o escopo do projeto proposto, os atributos de qualidade e os requisitos funcionais mencionados no mesmo, foi realizada a validação da arquitetura através da criação da prova de conceito, mostrando que as necessidades propostas foram atendidas e ainda há uma margem para melhorias. A avaliação permitiu concretizar de forma mais objetiva os testes e cenários propostos para definir os pontos fortes e fracos desta arquitetura. Nesta avaliação foram validados os requisitos apresentados no quadro abaixo.

Requisitos não funcionais	Testado	Homologado
Segurança: O sistema deve apresentar segurança no acesso e manipulação de dados, não permitindo o acesso aos módulos sem o usuário estar devidamente autenticado e autorizado.	SIM	SIM
Usabilidade: O sistema deve promover uma usabilidade fácil, objetiva e intuitiva.	SIM	SIM
Acessibilidade: O sistema deve suportar ambientes web responsivos e ambientes móveis.	SIM	SIM
Interoperabilidade: O sistema deve se comunicar com os sistemas legados já existentes <i>on premise</i> . O sistema deve se comunicar com sistemas externos via <i>API's Restful</i> de integração.	SIM	SIM
Desempenho: O sistema deve fornecer um bom desempenho com um tempo de resposta adequado a cada funcionalidade.	SIM	SIM

Avaliando a arquitetura proposta neste projeto, no que diz respeito a interação do usuário com a aplicação, foi possível identificar alguns pontos fortes, nos quais podemos citar a separação do *front-end* e o *back-end*, o que nos permite efetuar melhorias no seu layout sem a necessidade de alterar suas funcionalidades ou suas regras de negócio, além de poder ser aplicado um design modelado através de equipes de UX e desenvolvido por programadores especializados em *front-end* quando em paralelo a equipe de *back-end* trabalha em outras melhorias.

Avaliando a camada de *back-end*, é possível também notar pontos fortes como a facilidade de manutenibilidade caso seja necessário, isto se deve ao fato do sistema ser dividido em módulos separados que expõem apenas suas interfaces para que o módulo de API Gateway consiga se comunicar com seus respectivos *end-points*. A arquitetura também apresenta uma boa integração entre a linguagem utilizada na aplicação com a base de dados utilizada para o armazenamento das informações uma vez que temos vários tipos de conectores de banco de dados já existentes e com grande “*Know-how*” no mercado e que facilitam a integração da aplicação com seu respectivo dispositivo de armazenamento. Avaliando a comunicação e performance necessária para trafegar dados entre todos os sistemas presentes na indústria, e enviando informações pelo barramento de integração entre o SIGO e todos os outros sistemas, temos o Kafka que monitora e permite a integração em tempo real de todo o tráfego de informações que são essenciais para o funcionamento da indústria. A comunicação com *API's* externas também se mostrou bastante eficaz, com os recursos sendo consumidos em um tempo de resposta extremamente baixo.

Em relação à implantação do sistema, a utilização da nuvem da Amazon se mostrou bastante simples em questões de configuração, além de prover recursos de escalabilidade, segurança e integração. Dessa forma não há necessidade de se contratar e manter uma infraestrutura de tal magnitude, com garantias de segurança, disponibilidade e escalabilidade. Utilizar a esteira *DevOps* da própria Amazon, para tarefas de integração contínua, construção, testes automatizados e entrega contínua, proporciona grandes benefícios em questões de manutenibilidade e testabilidade, auxiliando no processo de novas implementações e melhorias. Por outro lado o custo a se pagar por tais serviços se mostrou um pouco elevado, porém há de se considerar que os custos de adquirir, operar e manter tal estrutura por conta própria também não seria de grande economia. Como margem para melhoria futura, existe também a possibilidade de implantar as *API's* em contêineres.

7. Conclusão

Projetar um sistema de alta qualidade, com requisitos funcionais e não funcionais desta magnitude foi algo desafiador, mas o mesmo tempo muito prazeroso. Este trabalho nos possibilitou aprimoramos nosso conhecimento e ampliar a visão de mercado, tecnologias e soluções como um todo.

O protótipo arquitetural apresentado por este trabalho tem uma arquitetura de ponta, com escalabilidade, manutenibilidade, desempenho, segurança, acessibilidade, usabilidade e interoperabilidade. Foram exploradas novas tecnologias, algumas extremamente atuais, como a própria AWS e seus muitos serviços, tecnologias emergentes como o React.js no *front-end* e .Net Core no *back-end*, a plataforma Apache Kafka e toda a sua extensão, permitindo a interoperabilidade com tecnologias legadas.

Em conclusão, todos os objetivos foram alcançados e ainda há margens para melhorias futuras.

REFERÊNCIAS

AWS Documentation. Amazon. Disponível em: https://docs.aws.amazon.com/index.html#lang/pt_br. Acesso em: 12 de dezembro de 2020.

AWS .NET SDK. Amazon. Disponível em: https://docs.aws.amazon.com/pt_br/sdk-fornet/v3/developer-guide/aws-sdk-net-v3-dg.pdf. Acesso em: 12 de dezembro de 2020.

Guia do .NET Core. Microsoft. Disponível em: <https://docs.microsoft.com/ptbr/dotnet/core>. Acesso em: 12 de dezembro de 2020.

O que é Apache Kafka. Red Hat. Disponível em: <https://www.redhat.com/pt-br/topics/integration/what-is-apache-kafka>. Acesso em: 12 de dezembro de 2020.

AWS Elastic Beanstalk. Amazon. Disponível em: <https://aws.amazon.com/pt/elasticbeanstalk/>. Acesso em: 12 de dezembro de 2020.

Amazon Relational Database Service (RDS). Amazon. Disponível em: <https://aws.amazon.com/pt/rds/>. Acesso em: 12 de dezembro de 2020.

O QUE É TRANSFORMAÇÃO DIGITAL. TD – Transformação Digital. Disponível em: <https://transformacaodigital.com/o-que-e-transformacao-digital/>. Acesso em: 03 de novembro de 2020.

Como funciona a indústria têxtil na era da transformação digital. Febratex Group. Disponível em: <https://fcem.com.br/noticias/entenda-como-funciona-a-industria-textil-na-era-da-transformacao-digital/>. Acesso em: 03 de novembro de 2020.

Indústria têxtil, como funciona. Pontual Textil. Disponível em: <http://www.pontualtextil.com.br/destaque/55/industria-textil-como-funciona>. Acesso em: 03 de novembro de 2020.

Indústria 4.0: quais as suas expectativas para o setor têxtil. Febratex Group. Disponível em: <https://fcem.com.br/noticias/industria-4-0-quais-as-suas-expectativas-para-o-setor-textil/>. Acesso em: 03 de novembro de 2020.

Setor têxtil: por que é preciso se adequar à Indústria 4.0. Delta Máquinas Têxteis. Disponível em: <https://www.deltaequipamentos.ind.br/industria-textil/setor-textil/>. Acesso em: 03 de novembro de 2020.

Transformação digital: As 7 principais tendências para 2020. Textília. Disponível em: <<http://www.textilia.net/materias/ler/moda/Tecnologia/transformacao-digital-principais-tendencias-para-2020>>. Acesso em: 03 de novembro de 2020.

A REVOLUÇÃO DIGITAL CHEGA À INDÚSTRIA TÊXTIL. Home Decore. Disponível em: <<http://www.homedecore.com.br/site/a-revolucao-digital-chega-a-industria-textil/>>. Acesso em: 03 de novembro de 2020.

Epson visa a transformação digital na indústria têxtil. Mercado e Consumo. Disponível em: <<https://www.mercadoeconsumo.com.br/2020/02/05/epson-visa-a-transformacao-digital-na-industria-textil/>>. Acesso em: 03 de novembro de 2020.

APÊNDICES

URL do repositório de código fonte:

<https://github.com/murilocaldeira/IndTexBrPUCMG>

URL da apresentação da POC no YouTube:

<https://www.youtube.com/watch?v=jABuVRqOjLA>

URL para acesso à aplicação:

<https://dev9366.db05dt5mgghvj.amplifyapp.com/>

Dados para acesso à aplicação:

Usuário: murilo@sigo.com.br / Senha: 123456