

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

Hitalo Filipe Silva de Araujo

SISTEMA INTEGRADO DE GESTÃO E OPERAÇÃO – SIGO - INDTEXTBR

Belo Horizonte

2020

Hitalo Filipe Silva de Araujo

SISTEMA INTEGRADO DE GESTÃO E OPERAÇÃO – SIGO - INDTEXTBR

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

Orientador: Prof. Dr. Pedro A. Oliveira

Belo Horizonte

2020

AGRADECIMENTOS

Agradeço a Deus por sempre me dar oportunidades e força para alcança-las. Agradeço a minha família e a Liliane Bernardes de Lima pelo apoio constante. Agradeço aos arquitetos Marcus Couto e Horrander Maikel pela inspiração técnica. E por fim, agradeço a Paulo Ottoni e Camillo Jorge Santos pela inspiração técnica e acadêmica.

RESUMO

O objetivo deste trabalho é projetar uma arquitetura completa e desenvolver uma prova de conceito a fim de dar base tecnológica a mudança organizacional que a IndTextBr está inserida. A IndTextBr é uma organização brasileira que fabrica e comercializa produtos têxteis, setor de alta competitividade. O SIGO, Sistema Integrado de Gestão e Operação, foi projetado para ter três módulos e integração com sistemas legados heterogêneos. Durante o projeto, foi dada prioridade a tecnologias livres, modernas e com boa aceitação da comunidade de desenvolvimento de *software*. Durante o trabalho avaliou-se o projeto em termos de segurança, interoperabilidade, usabilidade e acessibilidade. Todos os módulos, a API Gateway e o projeto SPA tem esteira de entrega e integração contínua.

Palavras-chave: arquitetura de software, projeto de software, requisitos arquiteturais, SIGO, microserviços, .NET, Angular, Transformação Digital.

SUMÁRIO

1. Objetivos do trabalho.....	6
2. Descrição geral da solução	6
2.1. Apresentação do problema.....	6
2.2. Descrição geral do software (Escopo)	7
3. Definição conceitual da solução	8
3.1. Requisitos Funcionais	8
3.2. Requisitos Não Funcionais.....	10
3.3. Restrições Arquiteturais.	14
3.4. Mecanismos Arquiteturais	15
4. Modelagem e projeto arquitetural.....	16
4.1. Modelo de componentes.....	16
4.2. Modelo de implantação	21
4.3. Diagrama de Classes	23
5. Prova de Conceito (POC) / protótipo arquitetural	25
5.1. Implementação e Implantação	25
5.1.1 Requisitos não funcionais	25
5.1.1 Casos de Uso.....	27
5.2 Interfaces/APIs.....	29
5.2.1 Gestão de Normas	30
5.2.2 Assessorias e Consultorias	31
5.2.3 Gestão do Processo Industrial.....	32
6. Avaliação da Arquitetura.....	33
6.1. Análise das abordagens arquiteturais	33
6.2. Cenários	34
6.3. Avaliação	35
6.4. Resultado.....	49
7. Conclusão.....	50
REFERÊNCIAS.....	51
APÊNDICES.....	52
CHECKLIST PARA VALIDAÇÃO DOS ITENS E ARTEFATOS DO TRABALHO	Erro! Indicador não definido.

1. Objetivos do trabalho

O objetivo geral desse trabalho é apresentar uma proposta de arquitetura de *software* para o projeto de transformação digital da IndTexBr - Indústria Têxtil do Brasil. O Sistema Integrado de Gestão e Operação (SIGO) visa dispor, principalmente por tecnologias livres e modernas, recursos tecnológicos e dados massivos para proporcionar vantagem competitiva a IndTextBr com alinhamento a nova forma de ação da empresa no ramo têxtil.

Os objetivos específicos são:

- Criar o módulo de Gestão de Normas que servirá de repositório para as normas técnicas relacionadas ao processo industrial. Esse módulo deverá acessar bases de dados externas a fim de identificar atualizações de normas e ser acessível por meio de microsserviços;
- Criar o módulo de Assessorias e Consultorias que permitirá contratação e gestão contratual de assessorias e consultorias. Deve prover integração para os outros módulos;
- Criar o módulo de Gestão do Processo Industrial. Será o módulo principal e proverá meios de integração com os outros módulos e sistemas externos. Será um centro de operações, um *cockpit*;
- Fazer uso de tecnologias livres, robustas e modernas.

2. Descrição geral da solução

Esta seção se destina a descrever a solução arquitetural definida para a aplicação SI-GO. A seção 2.1. Apresentação do problema faz a descrição do problema e a seção 2.2. Descrição geral do software (Escopo) descreve o escopo do projeto.

2.1. Apresentação do problema

Com a globalização as empresas passaram de disputar com concorrentes locais a disputar com pares de todo o planeta. Em um cenário extremamente dinâmico não considerar mudanças disruptivas pode significar a perda de mercados e consequentemente do sucesso da organização.

A IndTexBr é uma atacadista brasileira que fabrica e comercializa produtos têxteis. Neste momento ela está passando por um processo de transformação na sua atuação. Insuficiências geradas pelos mercados local e global precisam ser superadas com ações fortemente baseadas em recursos tecnológicos. Através da redução da dependência de fornecedores internos, da implantação de tecnologias mais avançadas de produção, de automatização, de reformulação no modelo de gestão, da capacitação de colaboradores e de investimentos em inovação, a empresa busca aumentar seu lucro em 5% e economizar de 10 a 20 % o seu custo de produção.

A empresa pretende exportar produtos com valor agregado e importar matérias primas de países que oferecem custo mais baixo e entrega mais rápida. Por meio de uma visão holística acerca de si mesma a IndTexBr vai se reinventar. Nesse contexto, a integração entre sistemas, pessoas e processo é essencial.

2.2. Descrição geral do software (Escopo)

O objetivo deste *software* é oferecer uma plataforma fortemente baseada na integração entre módulos existentes e os que serão construídos. Os módulos e as integrações servirão como ferramenta para todo o processo de renovação que a IndTexBr está passando.

O sistema será composto de três módulos principais: Gestão de Normas, Consultorias e Assessorias e Gestão do Processo Industrial. Com estes módulos será possível gerenciar normas do ramo de atuação, gerenciar contratos com consultorias e assessorias e proporcionar alto nível de integração com todos ativos de *software*.

Para alcançar os objetivos tecnológicos, deve se priorizar tecnologias livres e com boa aceitação da comunidade de desenvolvimento de *software*.

Por motivos econômicos e de agilidade, recursos de sistemas já existentes devem ser reaproveitados por meio de integrações e não desenvolvidos.

O SIGO deverá suportar ambientes *Web* e *mobile*, ser de fácil utilização, seguro, ter alta disponibilidade e disponibilizar meios de integração para os diferentes sistemas de terceiros e desenvolvidos internamente.

3. Definição conceitual da solução

Esta seção apresenta uma definição conceitual da solução a ser desenvolvida, são apresentados os requisitos funcionais e não funcionais, as restrições e os mecanismos arquiteturais considerados.

3.1. Requisitos Funcionais

Nesta seção são apresentados os requisitos funcionais do projeto. Os requisitos funcionais organizados por módulo são:

Módulo Gestão de Normas:

Neste módulo serão armazenadas normas que servem como base para *compliance* e planejamento de ações para uso correto e eficiente de recursos, descarte de sub-produtos, monitoramento de riscos de acidentes e outros.

As normas não são cadastradas ou atualizadas manualmente. Ao se cadastrar um repositório, o sistema busca e salva as referências de todas as normas automaticamente. A referência salva é o endereço *web* da norma. Periodicamente, o sistema verifica alterações das normas nos repositórios. Como pressuposto, o SIGO identificará que uma norma foi alterada com base no trecho final da URL, como de “repo.com/norma/9001/V1” para “repo.com/norma/9001/V2”. As notificações também são enviadas, via integração, para o módulo de Gestão do Processo Industrial.

Os requisitos funcionais são:

- Cadastro de repositórios: O sistema deve permitir que usuários autenticados cadastrem serviços externos de repositório de normas;
- Consulta de repositórios: O sistema deve permitir que usuários autenticados consultem serviços externos cadastrados;
- Exclusão de repositórios: O sistema deve permitir que usuários autenticados removam serviços externos cadastrados;
- Alteração de repositórios: O sistema deve permitir que usuários autenticados alterem serviços externos cadastrados;

- Consulta de normas: O sistema deve permitir que usuários autenticados consultem as normas cadastradas;
- Atualização de normas: O sistema deve acessar repositórios externos cadastrados para identificar e notificar sobre possíveis mudanças e evoluções.

Módulo Consultorias e Assessorias:

Este módulo oferece toda a estrutura necessária para contratação e gestão de empresas de consultoria e assessoria, parceiras da IndTexBr. As situações dos contratos também são enviadas, via integração, para o módulo de Gestão do Processo Industrial.

Os requisitos funcionais são:

- Cadastro de Consultorias e Assessorias: O sistema deve permitir que usuários cadastrados façam o cadastro de empresas de consultoria e assessoria;
- Consulta de Consultorias e Assessorias: O sistema deve permitir que usuários cadastrados consultem a relação de empresas de consultoria e assessoria registradas;
- Exclusão de Consultorias e Assessorias: O sistema deve permitir que usuários cadastrados façam a exclusão de empresas de consultoria e assessoria registradas;
- Alteração de Consultorias e Assessorias: O sistema deve permitir que usuários cadastrados façam a alteração no cadastro de empresas de consultoria e assessoria registradas;
- Cadastro de contratos: O sistema deve permitir que usuários cadastrados façam o cadastro de contratos de consultorias e assessorias;
- Consulta de contratos: O sistema deve permitir que usuários cadastrados consultem a relação de contratos de consultorias e assessorias registradas;
- Exclusão de contratos: O sistema deve permitir que usuários cadastrados façam a exclusão de contratos de consultorias e assessorias;
- Alteração de contratos: O sistema deve permitir que usuários cadastrados façam a alteração de contratos de consultorias e assessorias;

Módulo Gestão do Processo Industrial:

Este é o módulo principal do sistema. Possui integração com todos os outros módulos e sistemas legados (possui pelo menos um evento dedicado a processar mensagens de cada módulo/sistema). É o centro de operações do SIGO, um *cockpit*.

Os requisitos funcionais são:

- Autenticação de usuário: O sistema deve permitir que usuários registrados no AD da empresa possam se autenticar com usuário e senha;
- Autorização de usuários: O sistema deve permitir a definição das permissões de acesso com as opções de acesso de leitura, cadastro, exclusão e alteração. Os usuários devem ser cadastrados primariamente no AD da empresa;
- Cadastro de Eventos: O sistema deve permitir que usuários cadastrados façam o cadastro de eventos relacionados ao processo industrial;
- Consulta de Eventos: O sistema deve permitir que usuários cadastrados consultem a relação eventos registrados;
- Exclusão de Eventos: O sistema deve permitir que usuários cadastrados façam a exclusão de eventos registrados;
- Alteração de Eventos: O sistema deve permitir que usuários cadastrados façam a alteração em eventos registrados;
- Exibição de Eventos: O sistema deve exibir os eventos registrados.

Além dos módulos descritos acima, fazem parte do ecossistema do SIGO o gerador de relatórios *Crystal Reports* e as aplicações legadas: Sistema de Logística, Sistema de Gestão de Processos Industriais, Monitoramento de Vendas, Sistema de Segurança e Qualidade e o Subsistema de Inteligência do Negócio.

3.2. Requisitos Não Funcionais

A seguir são apresentados os requisitos não-funcionais do sistema:

- Usabilidade – O sistema de suportar ambientes *web* e *mobile*

Estímulo	Usuário percorrendo a listagem de normas registradas em um celular.
Fonte do Estímulo	Usuário acessando o sistema com um

	dispositivo móvel.
Ambiente	Produção, carga normal.
Artefato	Módulo de Gestão de Normas.
Resposta	A interface se adapta a tela do dispositivo.
Medida da resposta	O usuário consegue usar as funcionalidades e verificar as informações com experiência completa.

- Acessibilidade – O sistema deve ser de fácil utilização.

Estímulo	Usuário cadastrando um contrato de assessoria.
Fonte do Estímulo	Usuário acessando o módulo de Consultorias e Assessorias.
Ambiente	Produção, carga normal.
Artefato	Módulo de Consultorias e Assessorias.
Resposta	A interface apresenta simplicidade e facilidade de uso.
Medida da resposta	O usuário consegue fazer o cadastro.

- Desempenho – O sistema deve ter bom desempenho.

Estímulo	Usuário cadastrando novo repositório de normas.
Fonte do Estímulo	Usuário cadastrando novo repositório de normas no módulo de Gestão de Normas.
Ambiente	Produção, carga normal.
Artefato	Módulo de Gestão de Normas.
Resposta	O sistema exibe a coleção de normas em poucos segundos.
Medida da resposta	O sistema exibe a listagem de normas em 3 segundos.

- Manutenibilidade – O sistema deve ser de fácil manutenção.

Estímulo	Usuário encontra inconsistência na interface.
Fonte do Estímulo	Usuário encontra botão desalinhado no módulo de Gestão de Normas.
Ambiente	Produção, carga normal.
Artefato	Módulo de Gestão de Normas.
Resposta	A interface é atualizada isoladamente.
Medida da resposta	A correção é aplicada sem interromper o funcionamento módulo.

- Testabilidade – O sistema deve ser passível de ser testado em todas as suas funcionalidades.

Estímulo	Execução de testes manuais.
Fonte do Estímulo	Analista de Qualidade testando o módulo de Assessorias e Consultorias.
Ambiente	Homologação, carga normal.
Artefato	Módulo de Assessorias e Consultorias.
Resposta	O sistema não apresenta erros.
Medida da resposta	A versão é homologada.

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

Estímulo	Falha em <i>container</i> de microsserviço.
Fonte do Estímulo	<i>Container</i> do Módulo de Normas apresenta falha e é desativado.
Ambiente	Produção, carga normal.
Artefato	Módulo de Gestão de Normas.
Resposta	Outro <i>container</i> de Módulo de Normas é inicializado.
Medida da resposta	O módulo de Gestão de Normas continua disponível.

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

Estímulo	Consulta a um dos repositórios externos de normas.
Fonte do Estímulo	Módulo de Gestão de Normas fazendo requisição no repositório de normas externo.
Ambiente	Produção, carga normal.
Artefato	Módulo de Gestão de Normas.
Resposta	O sistema externo responde a requisição com as normas requisitadas.
Medida da resposta	A comunicação é feita com sucesso.

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

Estímulo	Tentativa de acesso sem permissão.
Fonte do Estímulo	Usuário não autenticado tenta fazer acesso aos módulos.
Ambiente	Produção, carga normal.
Artefato	Todos os módulos.
Resposta	O sistema redireciona o usuário para a página de autenticação.
Medida da resposta	O sistema não permite o acesso.

- Disponibilidade - Estar disponível 24 horas por dia, nos sete dias da semana, nas funcionalidades ligadas à produção.

Estímulo	Atualização de módulo.
Fonte do Estímulo	Atualização de módulo após homologação de versão.
Ambiente	Produção, carga normal.
Artefato	Módulo de Assessorias e Consultorias.
Resposta	Todos os usuários continuam usando o sistema.
Medida da resposta	Os usuários não são afetados durante a atualização.

- CI/CD - Ser desenvolvido utilizando recursos de integração contínua e entrega contínua.

Estímulo	Geração de versão.
Fonte do Estímulo	Uma alteração de código é aprovada e concluída.
Ambiente	Produção, carga normal.
Artefato	Todos os módulos.
Resposta	O código é atualizado.
Medida da resposta	O código é atualizado com as alterações aprovadas.

3.3. Restrições Arquiteturais.

Seguem as restrições arquiteturais do projeto:

- O sistema deve ser desenvolvido com a tecnologia multiplataforma .NET Core no *backend*;
- O sistema deve ser desenvolvido com a tecnologia Angular no *frontend*;
- As interfaces devem ser responsivas com boa adaptação em telas de diferentes tamanhos;
- O sistema deve ser desenvolvido em módulos.
- Deve apresentar características de aplicações distribuídas, tais como abertura, portabilidade e uso extensivo de recursos de rede;
- Deve ser modular e implantável por módulos;
- Deve adotar arquitetura baseada em microsserviços;
- Deve ser hospedado em nuvem híbrida, com parte dos componentes sendo mantidos *on Premise*;
- O *frontend* de cada módulo deve ser em página única (*Single Page Application* – SPA).

3.4. Mecanismos Arquiteturais

Mecanismo de Análise	Mecanismo de Design	Mecanismo de Implementação
Linguagem e plataforma	Linguagem de programação e plataforma de desenvolvimento	C#, .NET Core 3.1 e .NET Standard 2.1
Persistência	Banco de dados relacional	MySQL
Persistência	ORM	Entity Framework Core
Persistência	Micro ORM	Dapper
API	Microserviços	REST/RESTful
Frontend	Interface homem máquina	Angular 10 e Bootstrap 4
Segurança	Autenticação e autorização	JWT e Microsoft.AspNetCore.Identity
Versionamento	Versionamento de código fonte	Git (no Azure)
Repositório	Repositório do código fonte	GitHub
Repositório	Repositório de imagens de <i>containers</i>	Registros de <i>container</i> do Azure
Devops	Plataforma de Devops	Azure Devops
Integração entre módulos desenvolvidos	JSON	API's REST
Implantação	<i>Container</i>	Docker
CI/CD	<i>Pipeline</i> de integração e entrega contínua	Azure Pipelines
Testes	Testes automatizados	xUnit
Compilação	Geração de artefatos de <i>backend</i>	MSBuild
Transpilação	Geração de artefatos de <i>frontend</i>	Webpack
ESB	Integração	Mule
Mensageria	Integração	RabbitMQ
API Gateway	Integração	Ocelot para .NET Core

Descrição de API	Descrição, consumo e visualização de API's RESTful	Swagger
Tarefas em segundo plano	Gerenciados de tarefas recorrentes em segundo plano	HangFire
Comunicação	Comunicação entre clientes web e servidores em tempo real	SignalR

4. Modelagem e projeto arquitetural

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

4.1. Modelo de componentes

O modelo de componentes na Figura 1 detalha os componentes, a comunicação entre eles e as tecnologias utilizadas. Os módulos foram organizados para serem reutilizáveis e possíveis de serem implantáveis em módulos e para prover integração entre ativos existentes e que serão criados.

O SIGO foi planejado para ter módulos com arquitetura baseada em microsserviços e implantáveis independentemente. Cada módulo é formado por um *frontend* Angular, um *backend* .NET Core com C# e um banco de dados relacional MySQL, todas tecnologias livres, robustas e gratuitas. A utilização de SPA's Angular permite maior velocidade na transmissão de dados e na renderização de interfaces. O .NET Core é um arcabouço robusto, moderno e está entre as tecnologias mais rápidas de contexto geral. O MySQL está entre os mais populares bancos de dados livres.

A comunicação entre os módulos será feita por através de mensagens por meio do sistema RabbitMQ. Este sistema livre permite vários protocolos de mensagens, enfileiramento, confirmação de entrega e monitoramento de recursos.

A autenticação dos usuários é feita por meio do AD da empresa, onde os usuários devem estar pré cadastrados.

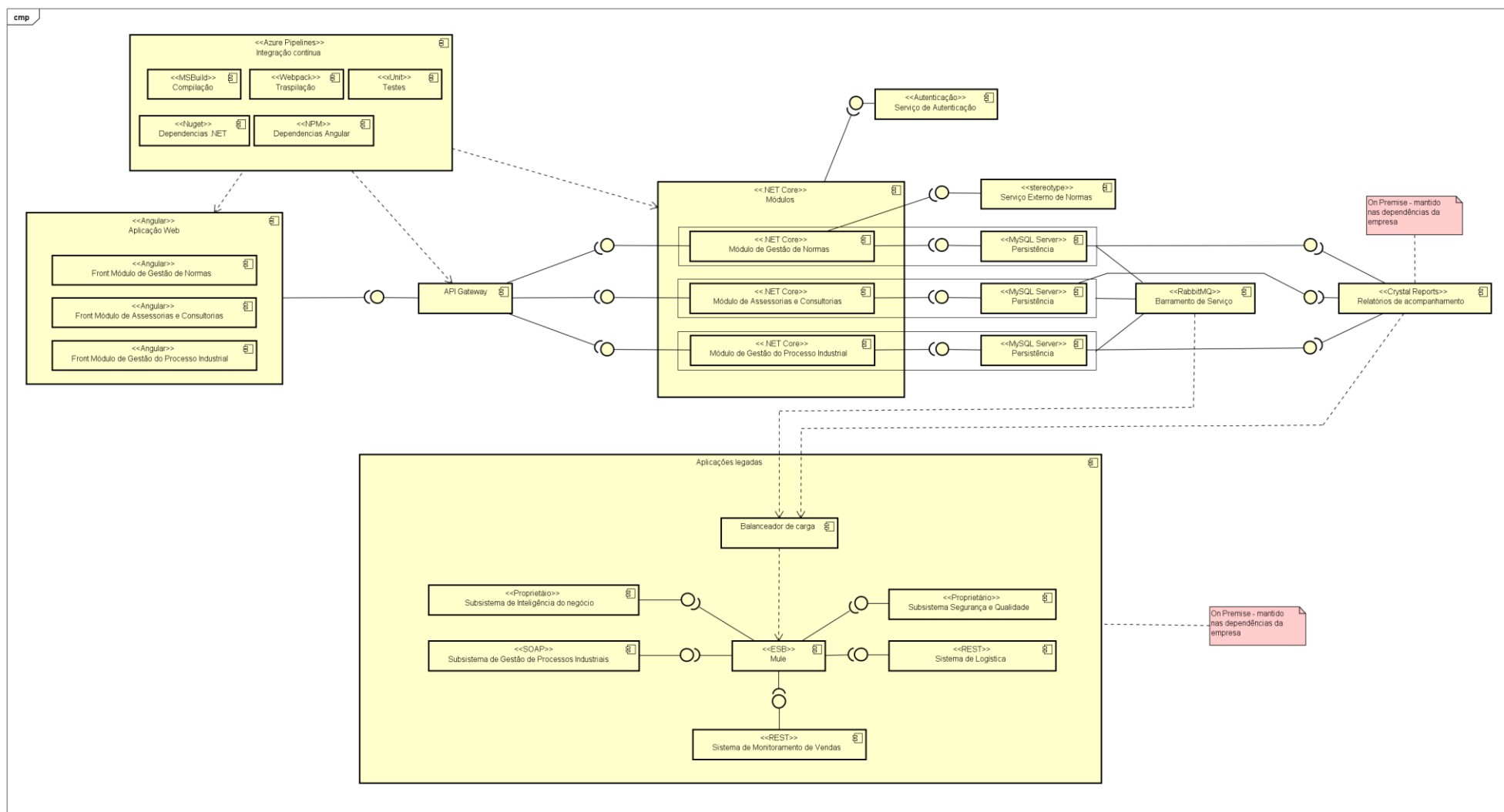


Figura 1 - Diagrama de Componentes

Todas as requisições entre a aplicação cliente e a aplicação no servidor devem usar um *token* JWS obtido na *API gateway* quando o usuário é autenticado no AD. Nenhuma interface de comunicação dos módulos é exposta diretamente, a *API gateway* vai intermediar as comunicações, além de registrar logs e verificar permissões. Os módulos são configurados para aceitar requisições originadas apenas da *API gateway* (CORS) e a *API gateway*, por sua vez, só aceita requisições do *frontend* SIGO.

Cada *token* JWT é gerado em combinação com a tecnologia OAuth. Quando o usuário é autenticado no AD, o *token* é montado com informações desprotegidas (como nome do usuário) e protegidas (dados de acesso gerados com OAuth). Assim é possível recuperar informações no *frontend* sem a necessidade de manter estado no servidor e sem detrimento a segurança.

Escolheu-se fazer uso de duas formas distintas de acesso ao banco de dados: EF Core e Dapper. Um ORM e um micro ORM respectivamente. A escolha de duas tecnologias, que a princípio servem para o mesmo propósito de manipular bancos de dados, se deve a prerrogativas de desempenho e manutenibilidade. O EF Core deve ser usado no SIGO para inclusões, atualizações e listagens simples de entidades. Já o Dapper deve ser usado para consultas complexas, com muitas relações, conversões e que não foram definidas como simples. Algumas características são destacadas a seguir:

EF Core:

- Agilidade na escrita de expressões;
- Facilidade no versionamento de migrações de bancos de dados;
- Expressões *lambda* podem se tornar complexas.

Dapper:

- Uso de scripts SQL padrão;
- Mapeamento consulta-entidade pouco ágil.

É esperado que a agilidade na escrita de expressões com EF Core e a legibilidade do Dapper se complementem. Embora essa escolha tenha sido originada por experiências em outros projetos em .NET e .NET Core, planeja-se avaliar continuamente a viabilidade do uso dos

dois recursos. O desenho da arquitetura foi feito para permitir a mudança na forma de acesso de dados com o devido isolamento, como visto na Figura 2.

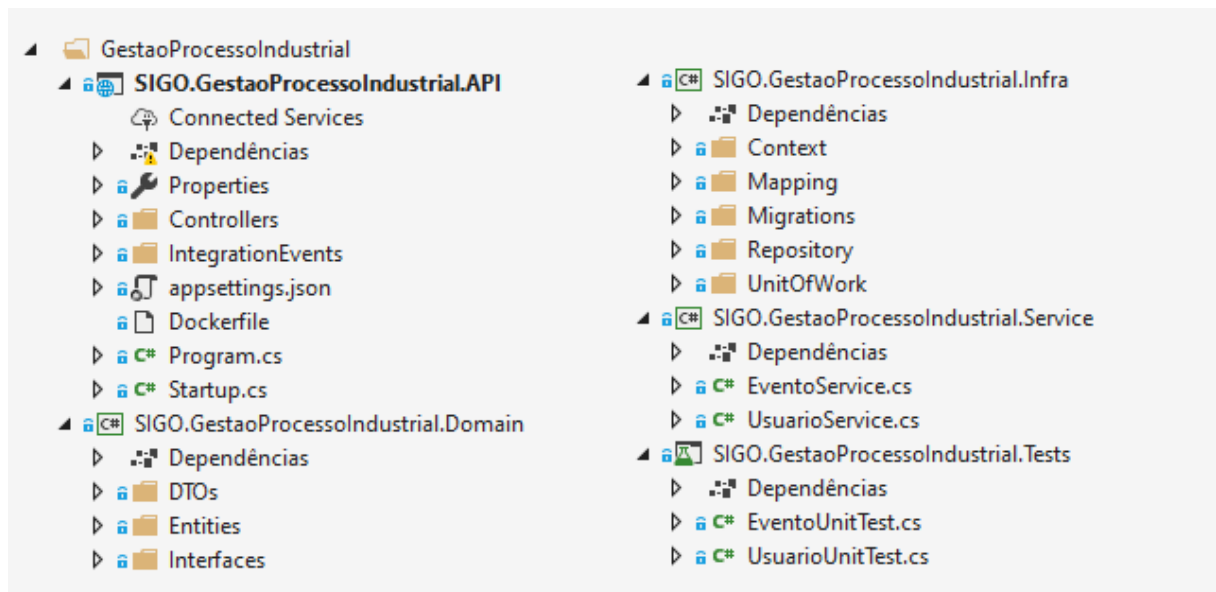


Figura 2 - Estrutura do projeto

É necessário ressaltar que nenhum dado deve ser apagado. Cada entidade deve possuir o atributo “DataExclusao”, para registro virtual de exclusão.

A esteira de CI/CD é implementada na estrutura da Microsoft Azure. Para versionamento, espera-se adotar a tecnologia Git com as diretrizes de fluxo de trabalho *GitFlow*.

A integração entre sistemas neste projeto é de suma importância. Os sistemas legados da IndTexBr são bastante heterogêneos, e da mesma maneira as formas de comunicação dos mesmos. O ESB Mule permite a comunicação entre os microsserviços desenvolvidos e as aplicações Inteligência do negócio, Gestão de Processos Industriais, Monitoramento de Vendas, Segurança e Qualidade e de Logística encaminhando e traduzindo as mensagens para o formato necessário. Todos os sistemas legados, o ESB Mule e um balanceador de carga estão implantados na infraestrutura local da empresa.

Os sistemas legados são heterogêneos. Espera-se que o uso de ESB proporcione uniformidade no consumo dos recursos destes sistemas com diminuição de complexidade.

Alguns funcionários da IndTextBr poderão gerar relatórios com o programa Crystal Reports. Este programa é instalado no computador dos usuários com permissão. O quadro Quadro 1 apresenta e descreve os componentes.

Componente	Descrição
Integração contínua	Recursos de compilação, transpilação, gerenciamento de dependências e testes.
Aplicação Web	Módulos <i>frontend</i> em Angular para disponibilização das páginas para os navegadores.
API <i>gateway</i>	Ponto de entrada para os serviços.
Módulos (microserviços)	Microserviços utilizados para expor operações de cada módulo.
Servidor de autenticação	Recurso utilizado para autenticar o acesso dos usuários.
Serviço externo de normas	Conjunto de repositórios externos que disponibilizam normas relacionadas ao setor têxtil.
Barramento de serviço	Servidor de mensageria para prover integração entre os microserviços.
Relatórios de acompanhamento	Programa de geração de relatórios.
Balanceador de carga	Componente para distribuir carga de trabalho entre os servidores.
ESB Mule	Recurso utilizado para realizar integração entre sistemas heterogêneos com comunicação baseada em XML, JSON e outros meios.
Subsistema de Inteligência do negócio	Sistema legado que será integrado ao SIGO.
Subsistema de Gestão de Processos Industriais	Sistema legado que será integrado ao SIGO.
Sistema de Monitoramento de Vendas	Sistema legado que será integrado ao SIGO.
Sistema de Logística	Sistema legado que será integrado ao SIGO.
Subsistema Segurança e Qualidade	Sistema legado que será integrado ao SIGO.

Quadro 1 - Quadro de componentes

4.2. Modelo de implantação

A Figura 3 apresenta o diagrama de implantação da aplicação, indicando o mapeamento dos elementos de *software* da arquitetura para os elementos de *hardware* onde eles executarão. Toda a aplicação construída está na modalidade em nuvem e os sistemas legados e o ESB Mule na modalidade *on premise*. A nuvem Azure além de hospedar os microsserviços do SIGO também hospeda as esteiras de CI/CD.

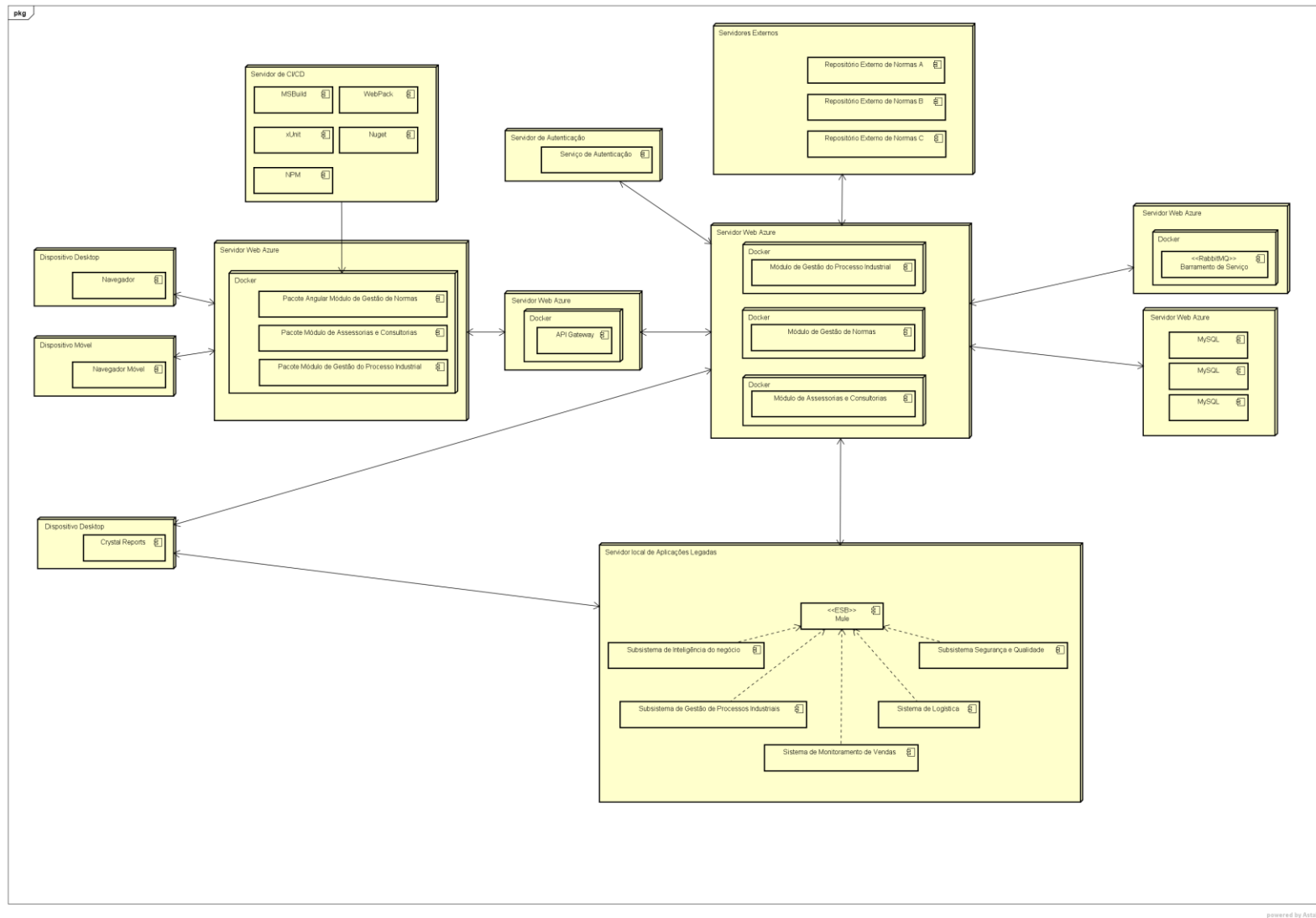


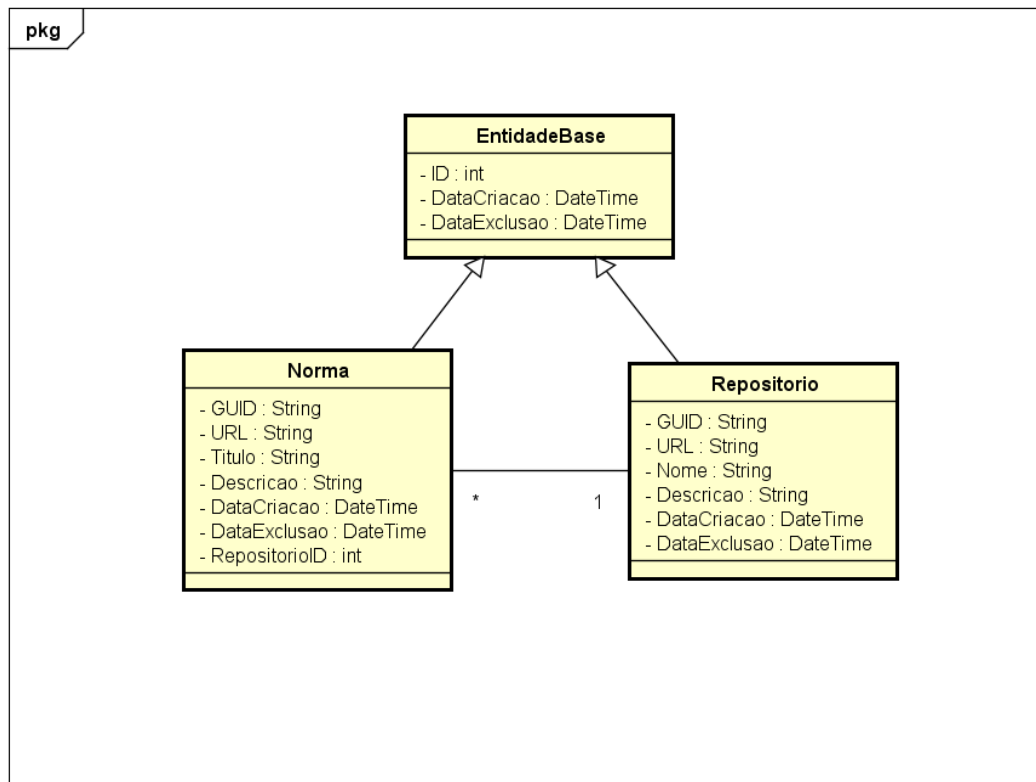
Figura 3 - Diagrama de Implantação

Componente	Descrição
Navegador	Navegador utilizado pelos usuários para interagir com o SIGO.
Navegador Móvel	Navegador utilizado pelos usuários para interagir com o SIGO.
Servidor de Integração Contínua	Recursos de compilação, transpilação, gerenciamento de dependências e testes.
Servidor Web Azure - Frontend	Servidos que armazena os arquivos estáticos das páginas <i>Web</i> em Angular.
Servidor de autenticação	Servidor utilizado para autenticar o acesso dos usuários.
Servidor Web Azure – API Gateway	Servidor responsável para prover o ponto de entrada para os serviços do SIGO.
Servidor Web Azure – Módulos	Servidores com os <i>containers</i> dos módulos SIGO.
Dispositivo Desktop – Crystal Reports	Dispositivo com <i>Crystal Reports</i> instalado e configurado para geração de relatórios.
Servidor Web Azure - RabbitMQ	Responsável pelo serviço de mensageria.
Servidor local de Aplicações Legadas	Responsável por armazenar as aplicações legadas e o ESB Mule.

Quadro 2 - Implantação - Quadro de Componentes

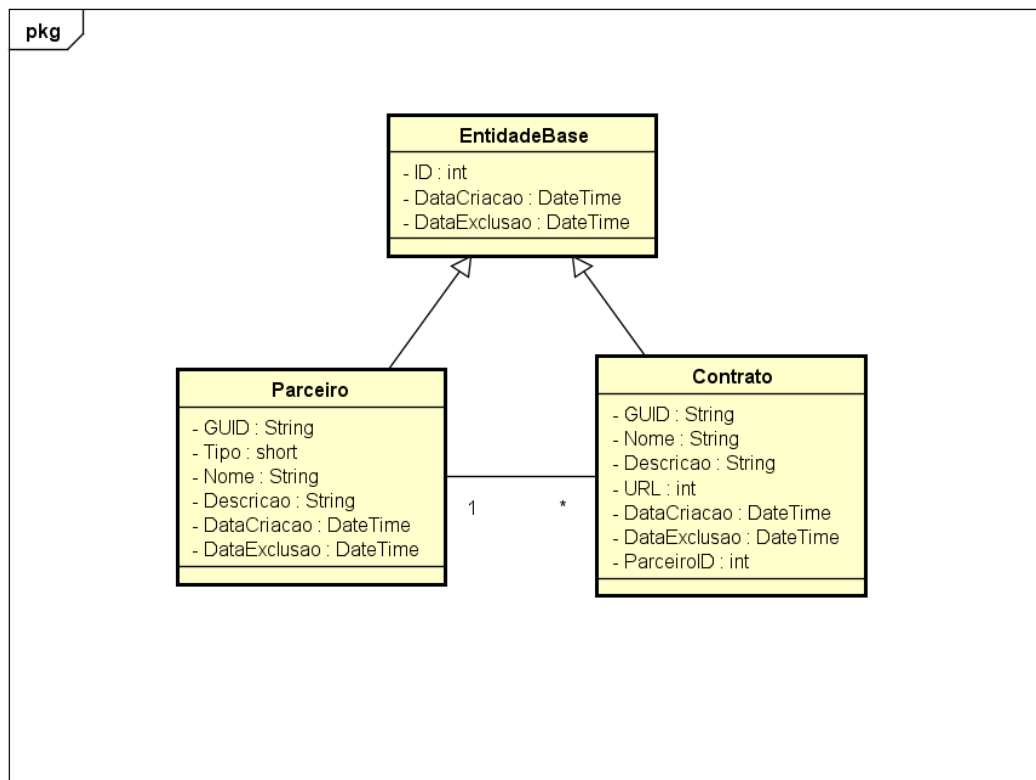
4.3. Diagrama de Classes

Os itens Figura 4, Figura 5 e Figura 6 exibem os diagramas de classes dos módulos Gestão de Normas, Assessorias e Consultorias e Gestão do Processo Industrial respectivamente.



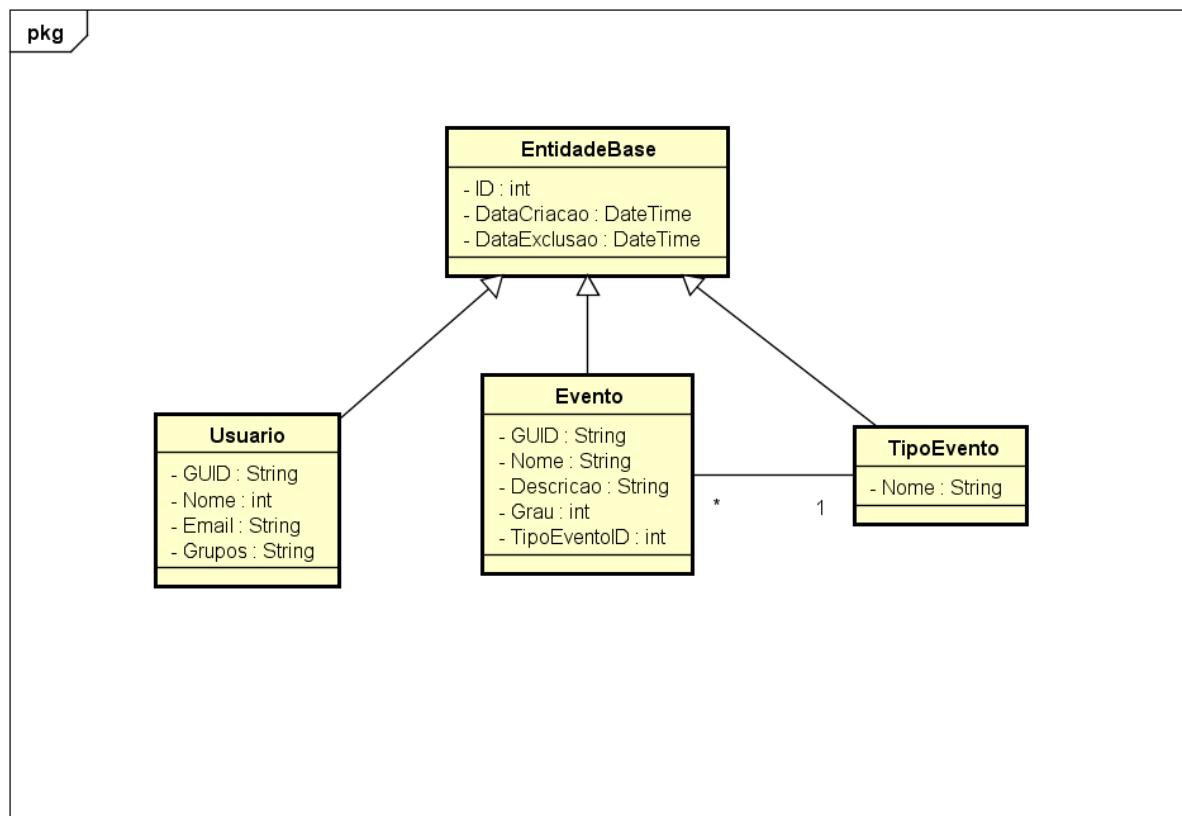
powered by Astah

Figura 4 - Diagrama de Classes - Gestão de Normas



powered by Astah

Figura 5 - Diagrama de Classes - Assessorias e Consultorias



powered by Astah

Figura 6 - Diagrama de Classes - Gestão do Processo Industrial

5. Prova de Conceito (POC) / protótipo arquitetural

A prova de conceito (POC) foi desenvolvida com o intuito de atender aos requisitos especificados neste relatório. A seção 5.1. Implementação e Implantação descreve a implementação da prova de conceito da arquitetura (protótipo arquitetural) da aplicação. A seção 5.1.1 Casos de Uso apresenta os casos de uso. E finalmente, a seção 5.3 Interfaces/APIs descreve as interfaces disponíveis.

5.1. Implementação e Implantação

A prova de conceito desenvolvida neste trabalho visa avaliar de forma prática se este modelo teórico atende ao que foi planejado. A POC contempla os casos de uso: Cadastro de repositórios e normas, autenticação de usuário e consulta de eventos.

5.1.1 Requisitos não funcionais

A POC visa validar os seguintes requisitos não funcionais:

- Segurança

Esse RNF foi escolhido devido a preocupação e esforço para manter dados sensíveis, recursos e usuários protegidos.

Os critérios de aceite são:

- O sistema não deve permitir que usuários sem permissão possam acessar recursos privados;
- O sistema deve permitir que usuários sem permissão possam acessar recursos não privados;
- O sistema deverá redirecionar o usuário para tela de autenticação ao identificar que um acesso a área segura está sendo feito sem autenticação.

- Interoperabilidade

Esse RNF foi escolhido devido ao fato da comunicação entre os ativos de *software* é a palavra de ordem da arquitetura.

Os critérios de aceite são:

- Os módulos desenvolvidos devem se comunicar entre si e com os sistemas legados;
- O sistema deve se comunicar com recursos externos.

- Usabilidade

Esse RNF foi escolhido devido à importância de se manter fluxos simples e objetivos. O que pode ajudar a empresa a aperfeiçoar seus processos.

Os critérios de aceite são:

- As telas do sistema devem apresentar funcionalidades simples e objetivas;
- O sistema deve apresentar interfaces de fácil navegação.

- Acessibilidade

Esse RNF foi escolhido devido à necessidade das interfaces se adaptarem a diferentes tipos de tela.

Os critérios de aceite são:

- O sistema deve oferecer as mesmas funcionalidades em todos os dispositivos;
- O sistema deve manter cores e estilos de elementos consistentes entre as páginas;
- O sistema deve funcionar de forma consistente em dois ou mais navegadores.
- Integração contínua e entrega contínua

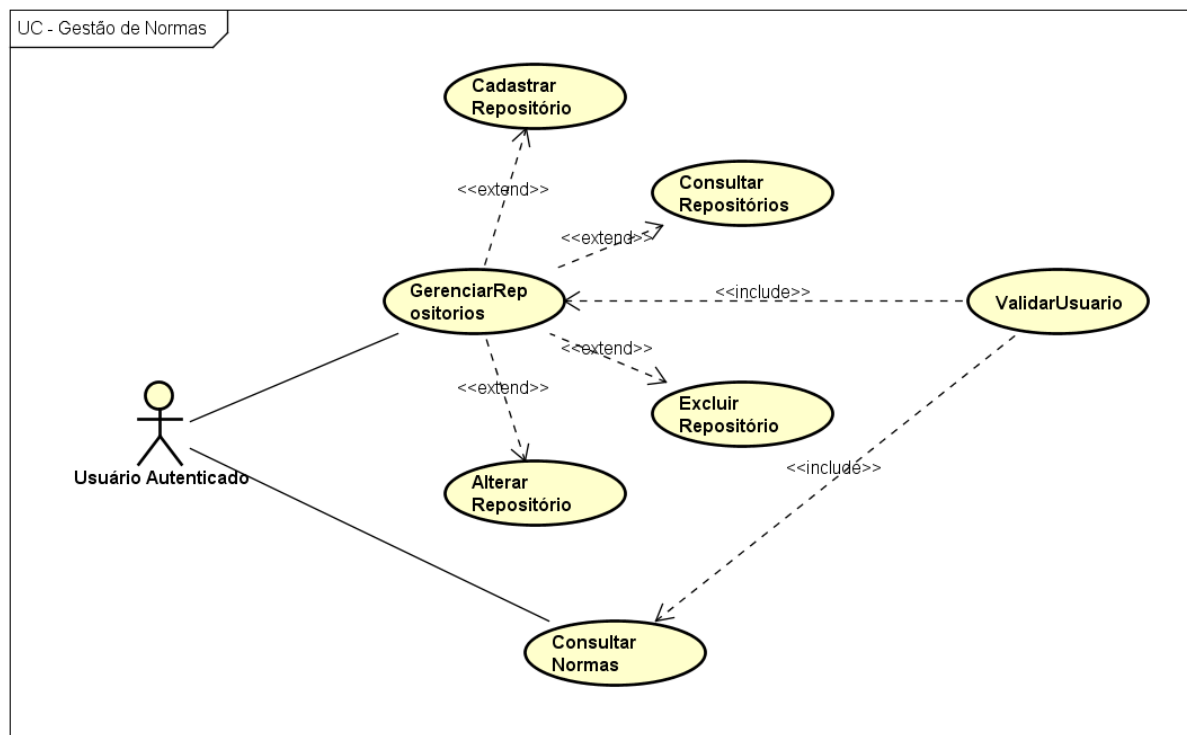
Esse RNF foi escolhido devido a agilidade oferecida por CI/CD durante o processo de desenvolvimento e evolução de sistemas.

Os critérios de aceite são:

- O código fonte deve ser compilado sem intervenção humana quando alterações forem feitas na versão principal do código, assim como também em versões de homologação;
- Uma nova versão da aplicação deve ser publicada sem interferência humana quando o código fonte do SIGO for disponibilizada (posteriormente deve-se avaliar a adoção de confirmação manual de publicação para publicação no ambiente de produção);
- Os responsáveis devem receber por *e-mail*, SMS e/ou por aplicativos de mensagem instantânea o resultado do processo de CI/CD.

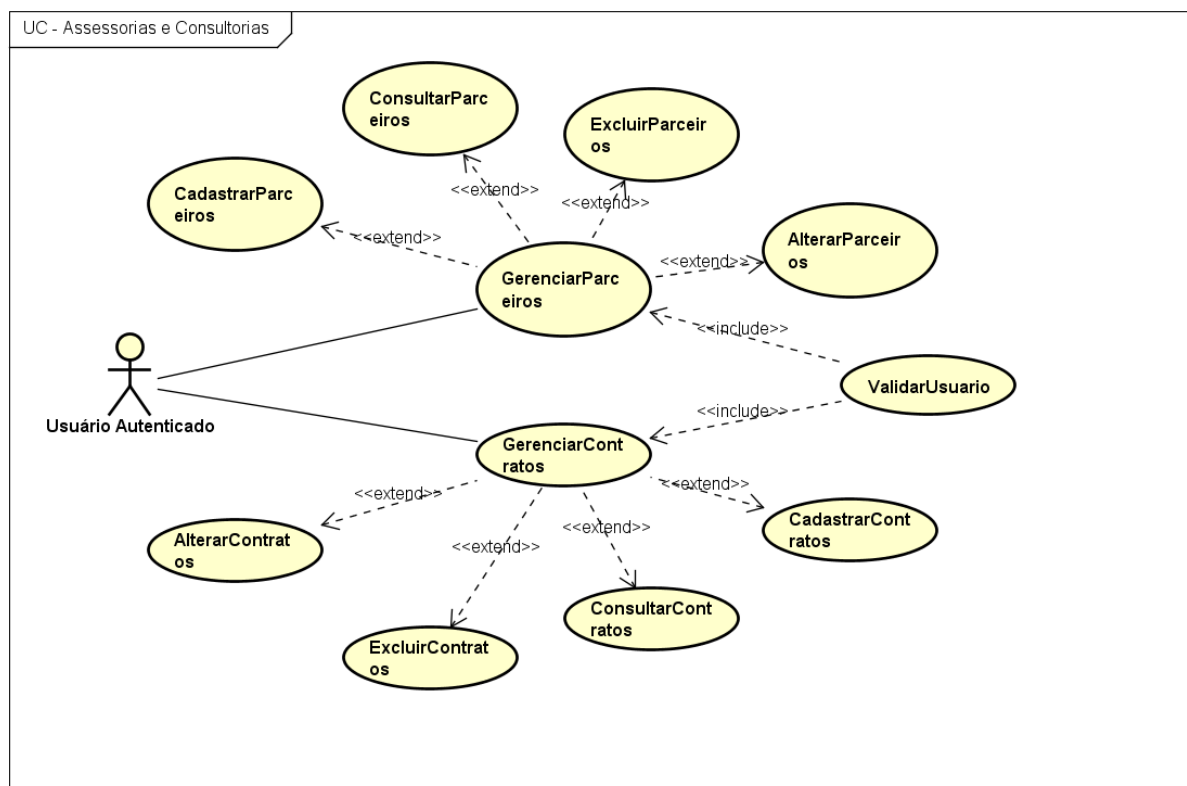
5.1.1 Casos de Uso

Esta seção apresenta os casos de uso (UC) dos três módulos desenvolvidos para o SIGO. A Figura 7 exibe o UC do módulo de Gestão de Normas, a Figura 8 exibe o UC do módulo de Assessorias e Consultorias e por fim a Figura 9 apresenta o UC do módulo de Gestão do Processo Industrial.



powered by Astah

Figura 7 - UC - Gestão de Normas



powered by Astah

Figura 8 - UC - Assessorias e Consultorias

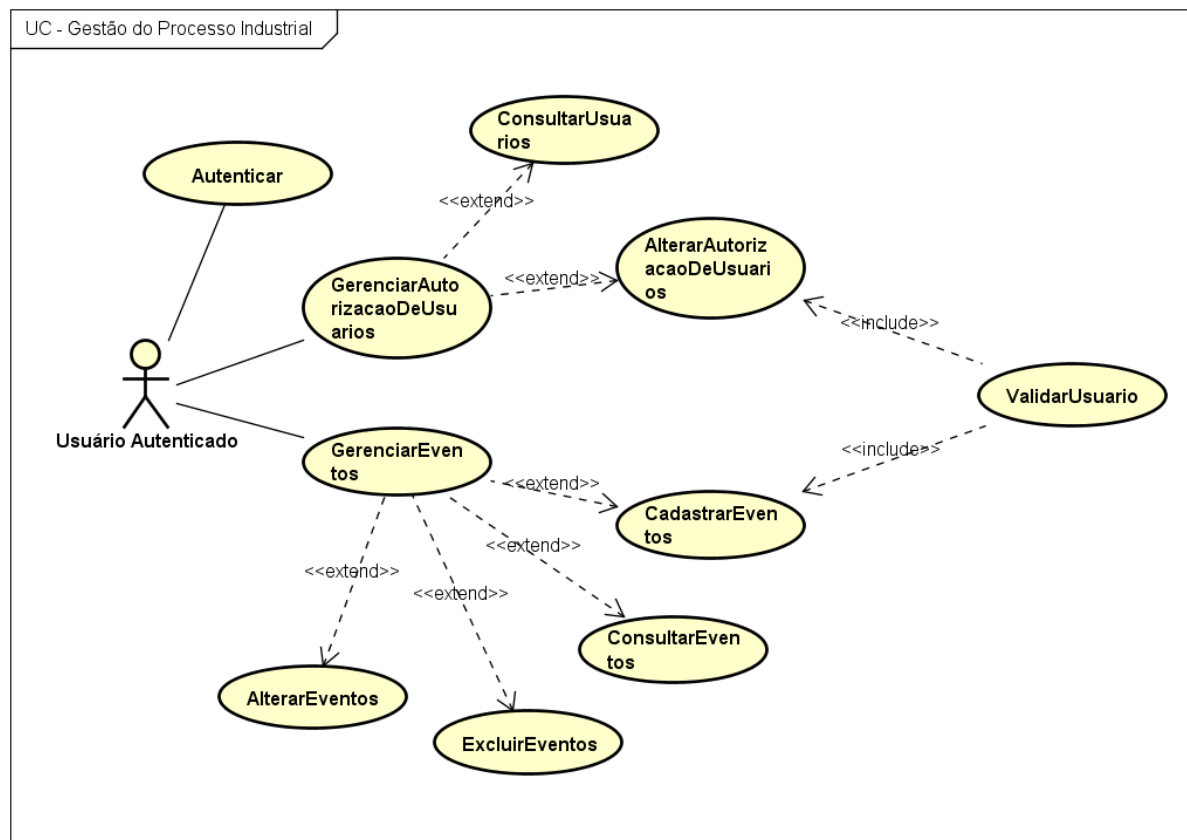


Figura 9 - UC - Gestão do Processo Industrial

Para a criação da POC do SIGO alguns casos de uso foram priorizados. Verificou-se também a adequação aos requisitos não funcionais levantados. Eles são descritos na seção 6. Avaliação da Arquitetura.

5.2 Tecnologias utilizadas

As tecnologias utilizadas na POC foram todas as descritas na seção 3.4. Mecanismos Arquiteturais com exceção de Signalr, Hangfire e ESB Mule. O ESB e as requisições dos sistemas legados foram simulados com uma aplicação .NET.

5.3 Interfaces/APIs

Para visualização, descrição e consumo das API's em tempo de desenvolvimento e para testes foi usado o arcabouço/framework Swagger. Com o Swagger é possível utilizar a especificação OpenAPI que define padrões para descrever API's RESTful. As API's poderão apenas ser acessadas com intermédio do Gateway. Além de melhorias de segurança é possível

centralizar e padronizar os endpoints e possivelmente alterar os endpoints sem afetar as rotas definidas na aplicação cliente.

O compartilhamento de recursos de origens diferentes (CORS - Cross-Origin Resource Sharing) na ApiGateway, está habilitado apenas para o servidor onde a aplicação frontend Angular esta hospedada.

5.3.1 Gestão de Normas

A Figura 10 apresenta os endpoints do módulo de Gestão de Normas. Os recursos são protegidos e o usuário deve informar o token de autorização. Os recursos do módulo só podem ser consumidos por intermediação da ApiGateway.

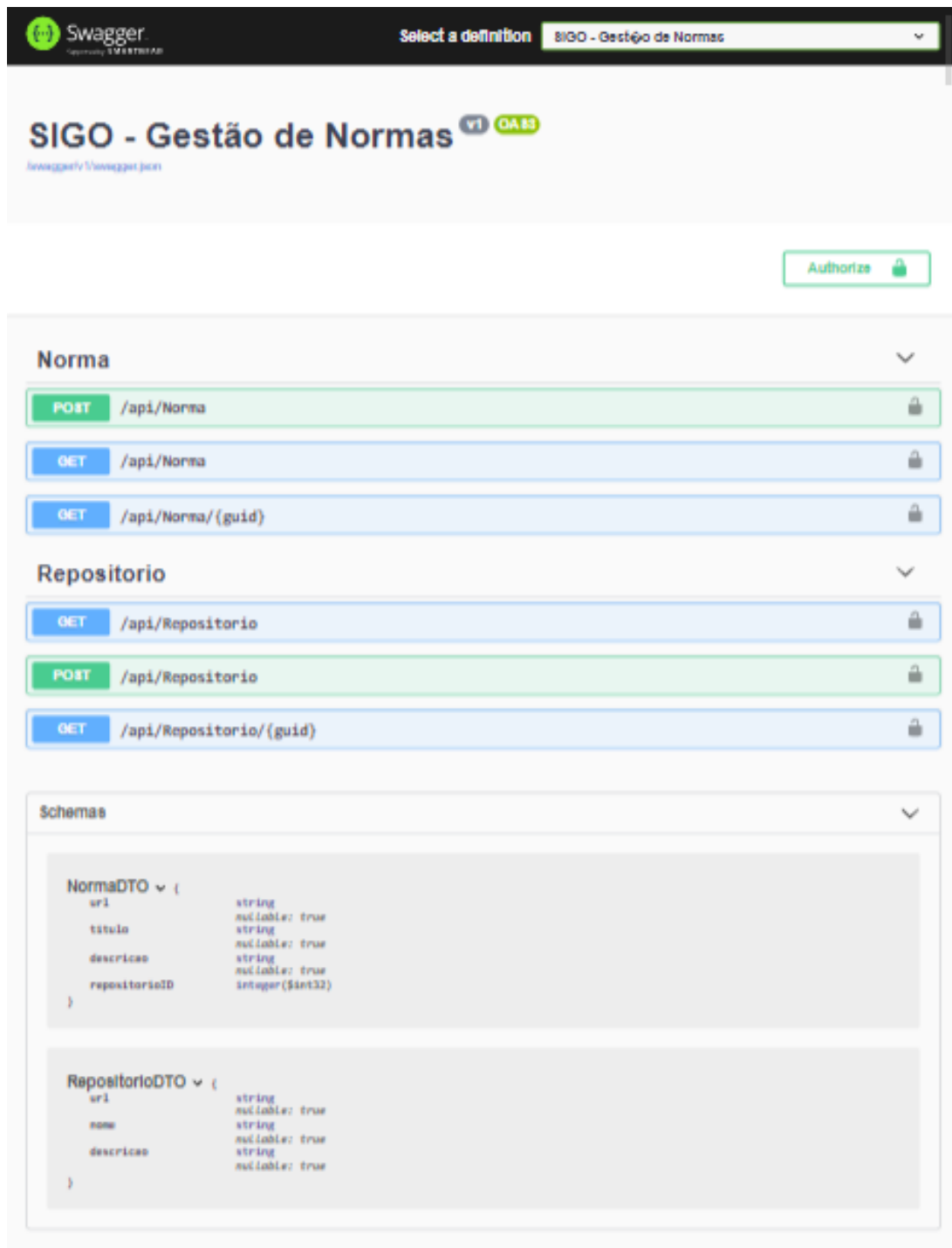


Figura 10 - Interfaces Gestão de Normas

5.3.2 Assessorias e Consultorias

A Figura 11 apresenta os endpoints do módulo de Assessorias e Consultorias. Os recursos são protegidos e o usuário deve informar o token de autorização. Os recursos do módulo só podem ser consumidos por intermediação da ApiGateway.

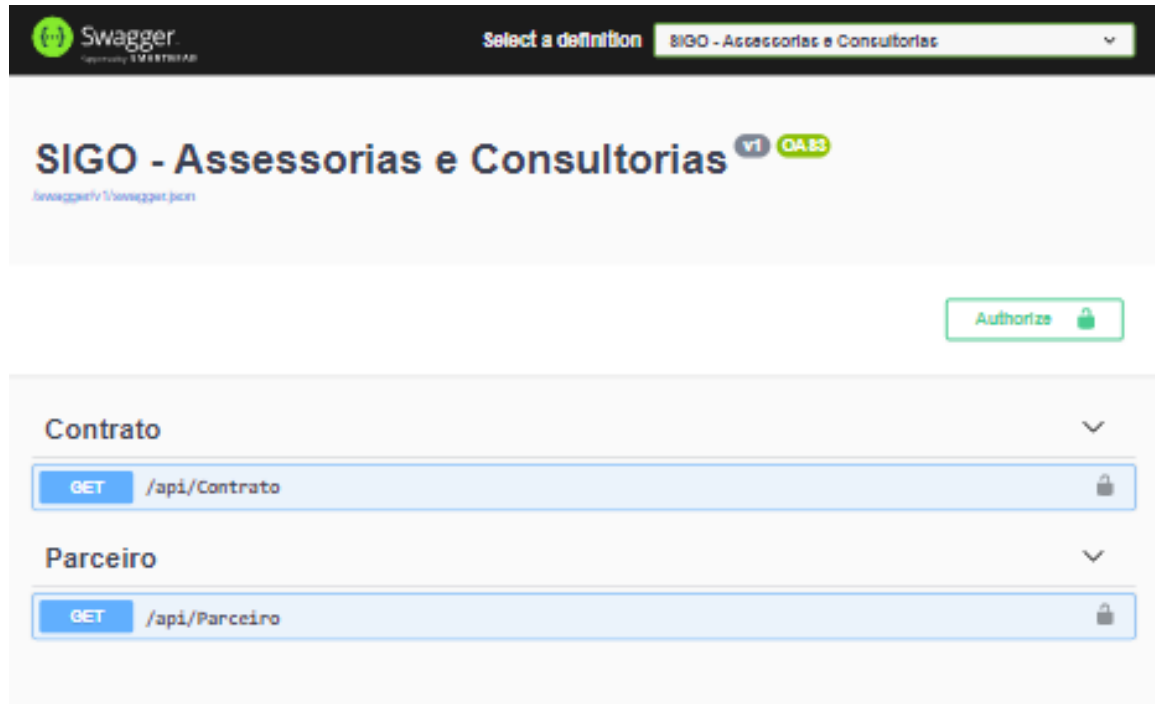


Figura 11 - Interfaces Assessorias e Consultorias

5.3.3 Gestão do Processo Industrial

A Figura 12 apresenta os endpoints do módulo de Gestão do Processo Industrial. Os recursos são protegidos e o usuário deve informar o token de autorização. Os recursos do módulo só podem ser consumidos por intermediação da ApiGateway.



Figura 12 - Interfaces Gestão do Processo Industrial

6. Avaliação da Arquitetura

6.1. Análise das abordagens arquiteturais

A arquitetura desenvolvida neste trabalho contempla um conjunto de módulos desenvolvidos, mecanismos de integração e comunicação e praticas de desenvolvimento de sistemas. Cada tecnologia e produto escolhidos foram pensados para fornecer uma estrutura robusta e leve. Destaca-se também o uso de tecnologias livres e modernas. A plataforma Azure para publicação de artefatos, definição de esteira de CI/CD e para gerenciamento de projetos foi escolhida por sua boa usabilidade, recursos e desempenho. Todas as escolhas levaram em consideração experiências em projetos anteriores e a adoção por parte da comunidade de desenvolvimento de *software*.

6.2. Cenários

Cenário 1: Na tentativa de acesso a algum recurso protegido o usuário não autenticado deve ser impedido e ser redirecionado para a tela de autenticação. O sistema deve permitir que páginas públicas sejam acessíveis a usuários autenticados e não autenticados. Este cenário garante o RNF de segurança.

Cenário 2: Na ocorrência de eventos como uma norma cadastrada no módulo de Gestão de Normas ou no aviso de estoque mínimo do sistema legado de Logística, o módulo de Gestão do Processo Industrial deve emitir uma mensagem de alerta para o usuário. Este cenário em conjunto com o cenário 3 garante o RNF de interoperabilidade.

Cenário 3: No cadastro de um novo repositório, o sistema deve mapear as normas deste repositório automaticamente. Este cenário em conjunto com o cenário 2 garante o RNF de interoperabilidade.

Cenário 4: Ao navegar pelo sistema, as telas devem apresentar funcionalidade simples e objetivas além de controles de navegação bem destacados. Este cenário garante o RNF de usabilidade.

Cenário 5: Ao navegar pelo sistema em diferentes navegadores, sistemas operacionais e resoluções o sistema deve apresentar o conjunto completo de funcionalidades, mesmo que rearranjadas, com cores e estilos consistentes. Este cenário garante o RNF de acessibilidade.

Cenário 6: Ao concluir alguma alteração no código fonte a esteira de CI/CD deve compilar o projeto, gerar uma versão e notificar os responsáveis. Este cenário garante o RNF de integração contínua e entrega contínua.

Na priorização foi utilizado o método de Árvore de Utilidade reduzida e com prioridades. Foram categorizados de acordo os atributos de qualidade a que estão relacionados e então classificados em função de sua importância e complexidade, considerando a percepção de negócio e arquitetura. Importância e Complexidade foram classificadas como alta, média e baixa de acordo com as características do RNF.

Categoria	Cenários	Importância	Complexidade
Segurança	1 – Acesso a recursos privados	Alta	Alta

	com autenticação.		
Interoperabilidade	2 – O sistema deve prover comunicação entre módulos e sistemas legados.	Alta	Alta
Usabilidade	3 – O sistema deve apresentar funcionalidades simples e objetivas e navegação intuitiva.	Média	Alta
Acessibilidade	3 – O sistema deve funcionar de forma consistente em diferentes telas, resoluções, sistemas operacionais e navegadores.	Média	Alta
Esteira CI/CD	4 – O sistema deve possuir esteira de CI/CD configurada.	Média	Média

Quadro 3 - Avaliação da arquitetura

6.3. Avaliação

Nesta seção é apresentado o processo de avaliação dos cenários identificados no item anterior. O objetivo é determinar os riscos, não riscos, pontos de sensibilidade e *trade-offs* e as evidências mostrando o requisito de qualidade sendo atendido.

Cenário 1:

Atributo de Qualidade:	Segurança
Requisito de Qualidade:	Acesso protegido com autenticação e autorização.
Preocupação:	
Acesso a recursos privados apenas com autenticação e autorização.	
Cenários:	
Cenário 1	
Ambiente:	
Produção, carga normal.	
Estimulo:	

Usuário tentando acessar recurso privado sem estar autenticado.	
Mecanismo:	
Mecanismo de geração de <i>tokens</i> mediante email, senha e permissões associadas.	
Medida de Reposta:	
O usuário não autenticado deve ser redirecionado para a tela de autenticação.	
Considerações sobre a arquitetura:	
Riscos:	Falhas nas políticas de autenticação e autorização podem expor dados sigilosos e uso malicioso dos recursos.
Pontos de Sensibilidade:	Servidor respondendo apenas a requisições HTTPS.
Trade-off:	---

Evidências do cenário 1:

A Figura 13 apresenta a página de autenticação do SIGO, uma página pública. Do lado direito da imagem percebe-se a ausência de *token*.

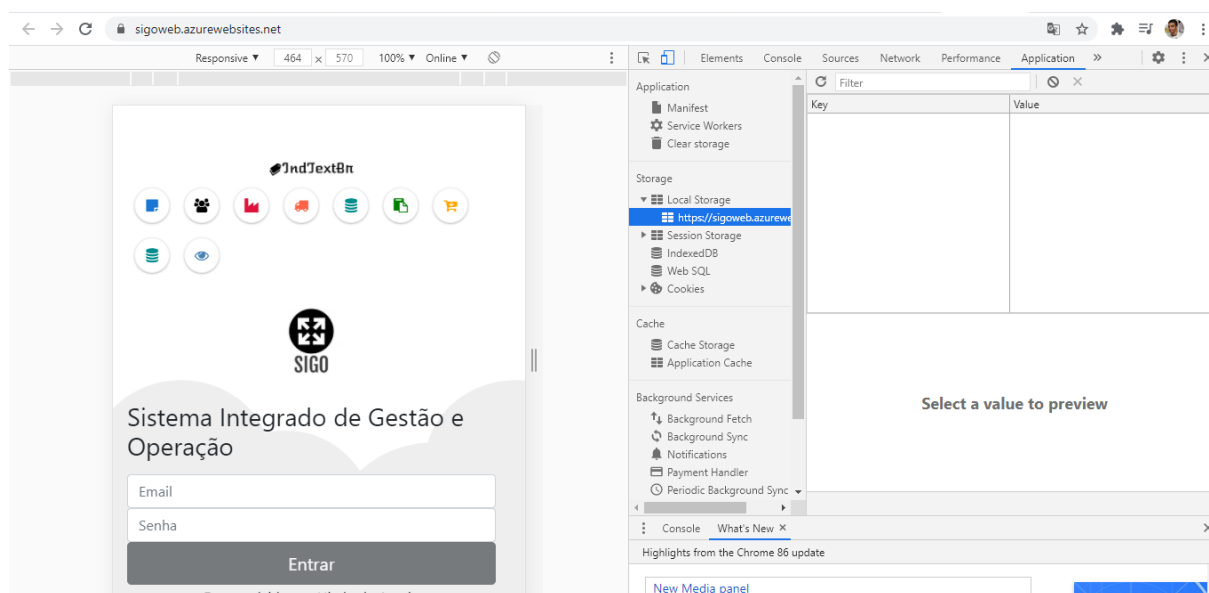


Figura 13 - Cenário 1 - Página pública

A Figura 14 demonstra o acesso a página principal do módulo de Gestão do Processo Industrial, uma página privada. Percebe-se no lado direito da imagem a presença de token.

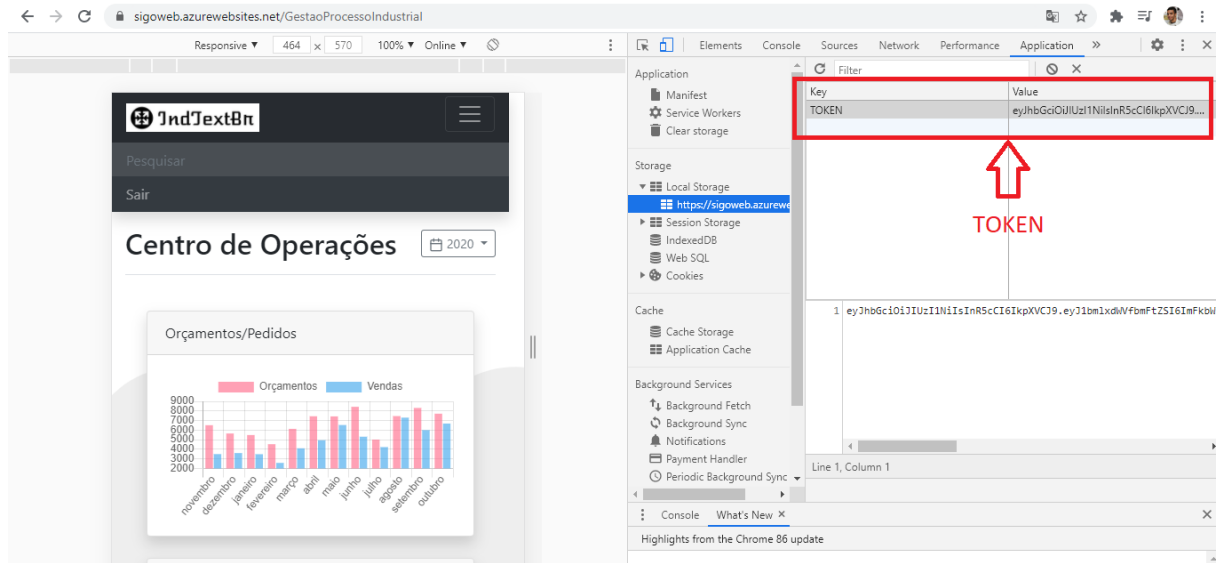


Figura 14 - Cenário 1 - Página privada

A Figura 15 representa o redirecionamento para a tela de autenticação após a exclusão do token. Percebe-se no lado direito da imagem a ausência de token.

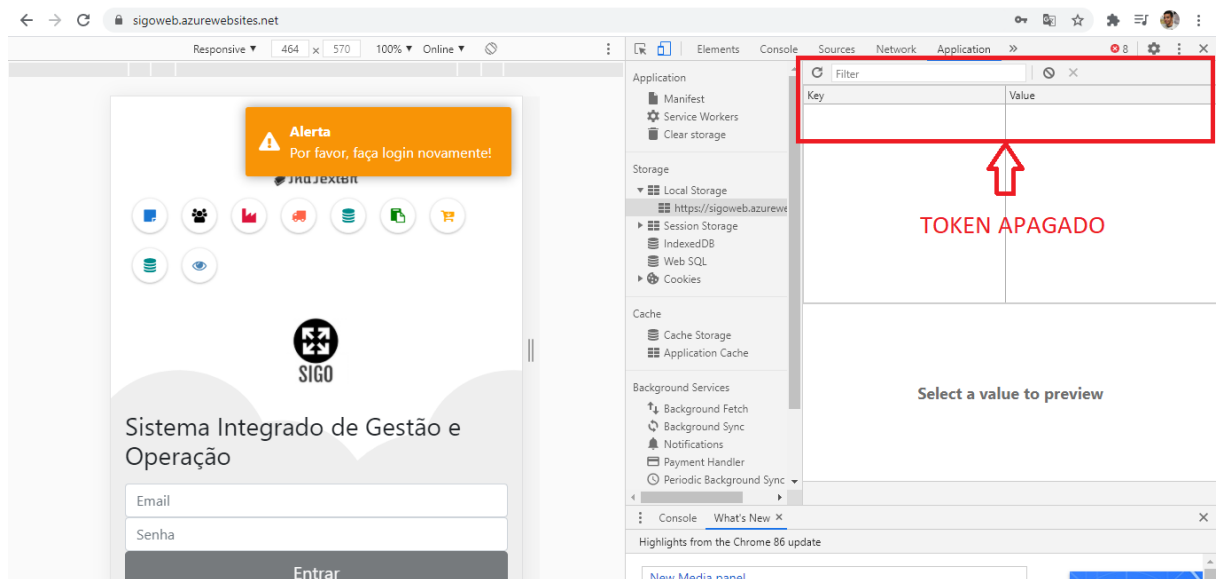


Figura 15 - Cenário 1 - Página com token removido

Cenário 2:

Atributo de Qualidade:	Interoperabilidade
Requisito de Qualidade:	O sistema deve se comunicar com sistemas legados e deve ter integração entre seus módulos.
Preocupação:	

Prover integração entre os módulos e com os sistemas legados heterogêneos	
Cenários:	
Cenário 2	
Ambiente:	
Produção, carga normal.	
Estimulo:	
Módulos e sistemas legados gerando eventos e módulo de Gestão do Processo Industrial exibindo eventos recebidos.	
Mecanismo:	
Configuração de servidor de mensagens para integração de módulos e de ESB para integração com legados.	
Medida de Reposta:	
Os módulos que escutam os eventos recebem as informações.	
Considerações sobre a arquitetura:	
Riscos:	Indisponibilidade do servidor de mensagens e do ESB.
Pontos de Sensibilidade:	Indisponibilidade dos servidores de integração.
Trade-off:	---

Evidências do cenário 2:

A Figura 16 apresenta a publicação de um evento enviado pelo ESB. Essa publicação é feita no servidor RabbitMQ.

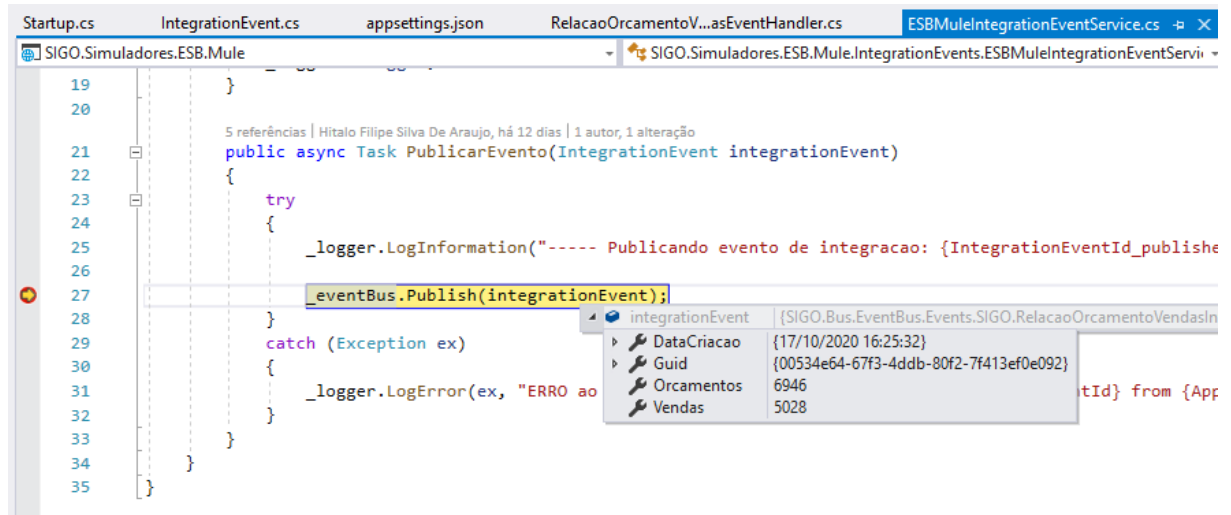


Figura 16 - Cenário 2 - Publicação de Evento de sistema legado

Na Figura 17 é apresentado o recebimento de um evento por parte do módulo de Gestão do Processo Industrial, intermediado pelo RabbitMQ e enviado por um sistema legado.

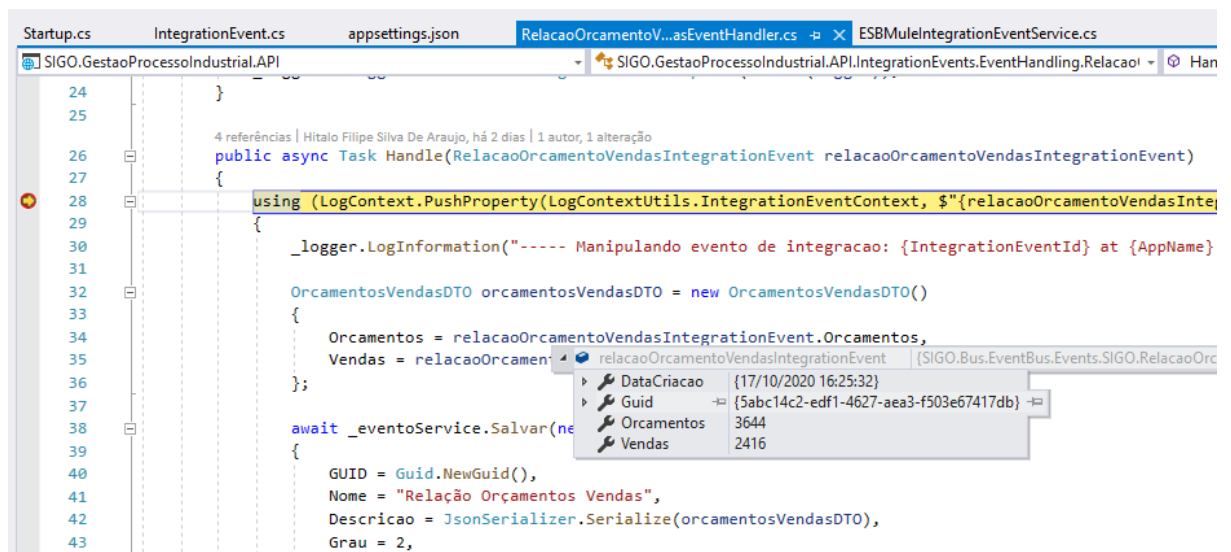


Figura 17 - Cenário 2 - Evento sendo processado em Gestão do Processo Industrial

Cenário 3:

Atributo de Qualidade:	Interoperabilidade
Requisito de Qualidade:	Comunicação com serviços externos.
Preocupação:	
O sistema deve se comunicar com serviços externos.	
Cenários:	
Cenário 3	

Ambiente:	
Produção, carga normal.	
Estimulo:	
O módulo de Gestão de Normas recebe cadastro de novo repositório.	
Mecanismo:	
O módulo de Gestão de Normas faz o mapeamento das normas cadastradas no repositório por meio de requisições HTTP/HTTPS.	
Medida de Reposta:	
O sistema consegue se comunicar e obtém a coleção de normas do repositório.	
Considerações sobre a arquitetura:	
Riscos:	Indisponibilidade ou recusa de conexão com o servidor de repositório de normas.
Pontos de Sensibilidade:	Disponibilidade do servidor do repositório.
Trade-off:	---

Evidências do cenário 3:

A Figura 18 exibe o cadastro de um repositório de normas.

The screenshot shows the JndTextBn web application interface. On the left is a sidebar with navigation links: 'Gestão de Normas', 'Normas', 'Repositórios' (highlighted in blue), and 'MÓDULOS' which includes 'Assessorias e Consultorias' and 'Gestão do Processo Industrial'. The main content area is titled 'Gestão de Normas' and contains a form for registering a repository. The form fields are: 'Nome' (with the value 'Repositório Exemplo RNF'), 'URL' (with the value 'https://github.com/HitaloDeAraujo'), and 'Descrição' (with the value 'Repositório Exemplo RNF - Paginas do TCC de Graduação'). Below the fields are 'Salvar' and 'Cancelar' buttons. The interface also includes a search bar at the top and a 'Sair' button.

Figura 18 – Cenário 3 - Cadastro de Repositório de Normas

Na Figura 19 é apresentado a coleção de normas registradas automaticamente após o cadastro do repositório.



JndTextBn Pesquisar Sair

Gestão de Normas

Normas

Repositórios

MÓDULOS

- Assessorias e Consultorias
- Gestão do Processo Industrial

10 resultados por página Pesquisar RNF

#	Título	Descrição	URL
3ca335a6-2f98-4e7e-bac7-55fd6f2f6174	Norma numero12	Norma numero12 do repositório Repositório Exemplo RNF	URL
7c8b11ad-423a-46e6-9edc-6079ae565d17	Norma numero13	Norma numero13 do repositório Repositório Exemplo RNF	URL
1909d32f-f998-4cf8-abc7-a3a38e8103cc	Norma numero14	Norma numero14 do repositório Repositório Exemplo RNF	URL
ff58c9d-9642-	Norma	Norma numero15 do	URL

Figura 19 - Cenário 3 - Normas obtidas automaticamente do repositório cadastrado

A Figura 20 exemplifica um documento armazenado em um repositório externo.

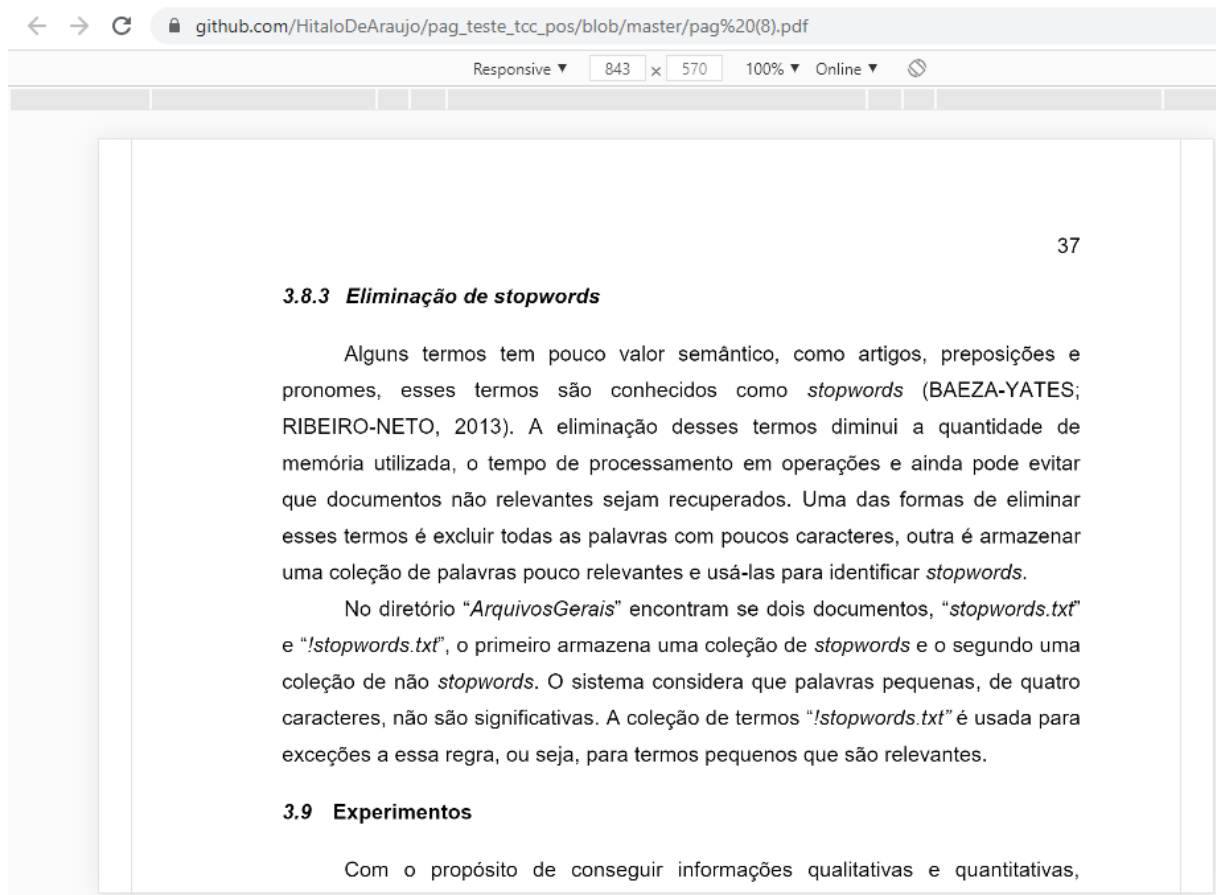


Figura 20 - Cenário 3 - Exemplo de documento cadastrado em Normas

Cenário 4:

Atributo de Qualidade:	Usabilidade
Requisito de Qualidade:	O sistema deve prover boa usabilidade.
Preocupação:	
As telas devem apresentar funcionalidade simples e objetivas além de controles de navegação bem destacados.	
Cenários:	
Cenário 4	
Ambiente:	
Produção, carga normal.	
Estimulo:	
Usuário navegando pelo sistema.	
Mecanismo:	
Criação de telas simples e limpas e uso de SPA (o que agiliza o carregamento de páginas).	

Medida de Reposta:	
O sistema deve conseguir utilizar o sistema de forma intuitiva.	
Considerações sobre a arquitetura:	
Riscos:	O servidor pode apresentar instabilidade, o que pode prejudicar a experiência do usuário.
Pontos de Sensibilidade:	Alta disponibilidade.
Trade-off:	---

Evidências do cenário 4:

A Figura 21 demonstra as transições de *layout* que tem por objetivo facilitar a identificação do fluxo de navegação para o usuário.



Figura 21 - Cenário 4 - Menus e indicativos

A Figura 22 apresenta uma tela com uma padronização de cores que indicam o grau de atenção requerida pelo usuário.



Figura 22 - Cenário 4 - Painéis indicativos de contratos

Cenário 5:

Atributo de Qualidade:	Acessibilidade
Requisito de Qualidade:	O sistema deve apresentar boa acessibilidade.
Preocupação:	
Ao navegar pelo sistema em diferentes navegadores, sistemas operacionais e resoluções o sistema deve apresentar o conjunto completo de funcionalidades, mesmo que rearranjadas, com cores e estilos consistentes.	
Cenários:	
Cenário 5.	
Ambiente:	
Produção, carga normal.	
Estimulo:	
Usuário navegando pelo sistema.	
Mecanismo:	
Criação de interfaces responsivas e com recursos suportados por diferentes navegadores.	

Medida de Reposta:	
O sistema deve se adaptar a diferentes ambientes do cliente sem perder funcionalidades e sem prejudicar a qualidade da interface.	
Considerações sobre a arquitetura:	
Riscos:	Existência de muitos navegadores e de muitas versões de cada um desses navegadores.
Pontos de Sensibilidade:	Compatibilidade com navegadores.
Trade-off:	---

Evidências do cenário 5:

Na Figura 23 é apresentada uma página do SIGO em um dispositivo *desktop* de tela 1366 x 768.

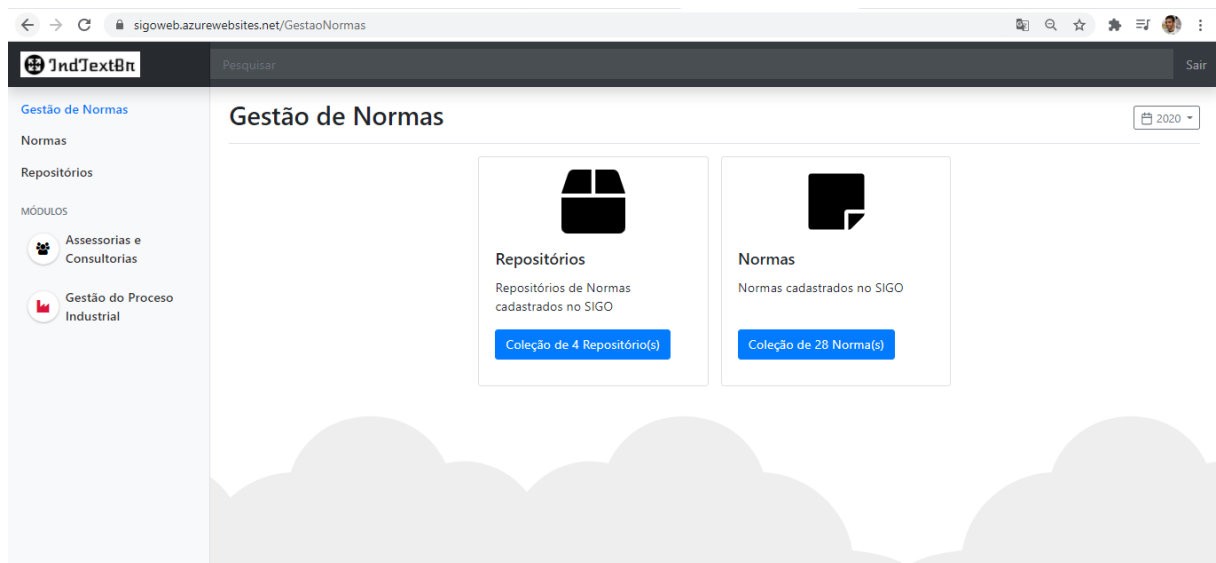


Figura 23 - Cenário 5 - SIGO em computador com resolução 1366 x 768

A Figura 24 é apresentada uma página do SIGO em um dispositivo *smartphone* de tela 375 x 812.



Figura 24 - 22 - Cenário 5 - SIGO em computador com resolução 375 x 812

Cenário 6:

Atributo de Qualidade:	Integração contínua e entrega contínua
Requisito de Qualidade:	O sistema deve possuir esteira de CI/CD.
Preocupação:	
Ao concluir alguma alteração no código fonte a esteira de CI/CD deve compilar o projeto, gerar uma versão e notificar os responsáveis.	
Cenários:	
Cenário 6.	
Ambiente:	
Produção, carga normal.	
Estimulo:	
Conclusão de alteração no código fonte.	
Mecanismo:	
Recursos de compilação e publicação no Azure.	
Medida de Reposta:	

Compilação, publicação e envio de notificação concluídos.	
Considerações sobre a arquitetura:	
Riscos:	Conclusão precipitada de alterações
Pontos de Sensibilidade:	Boas práticas de versionamento e configuração correta de permissões de acesso ao código e as esteiras de CI/CD.
Trade-off:	A liberação de versão de novas funcionalidades poderia ser feita com confirmação manual de responsável.

Evidências do cenário 6:

Uma alteração de código enviada para o servidor é apresentada na Figura 25. Este envio dispara o processo de integração contínua.

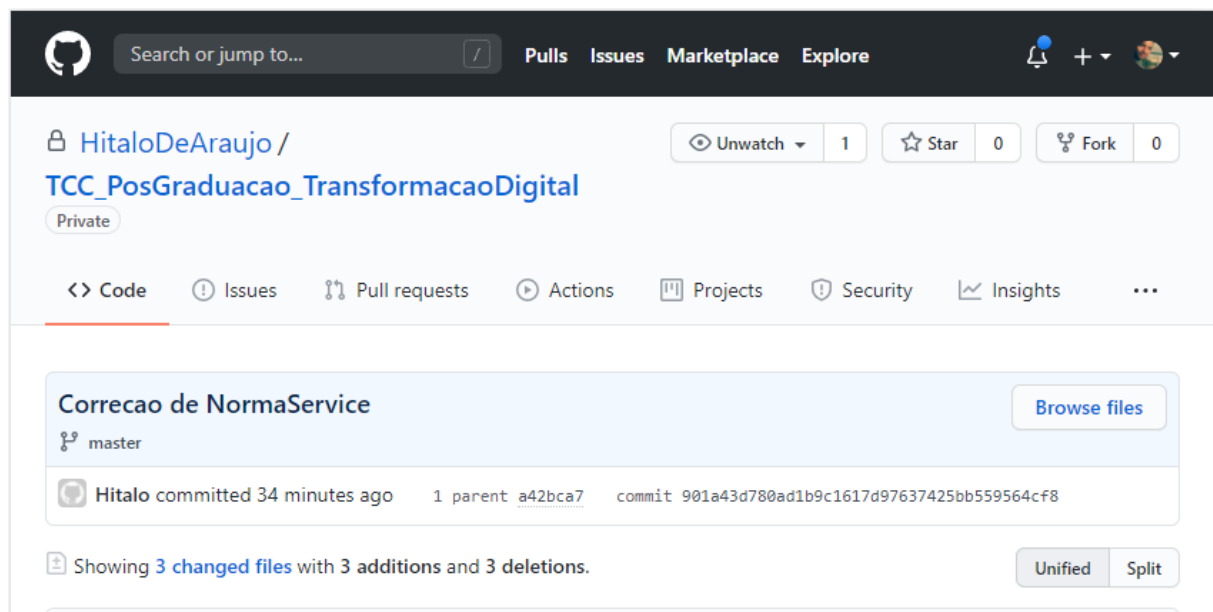


Figura 25 - Cenário 6 - Commit registrado

Na Figura 26 é exibida a conclusão da integração contínua. Após esta etapa a entrega contínua é iniciada.

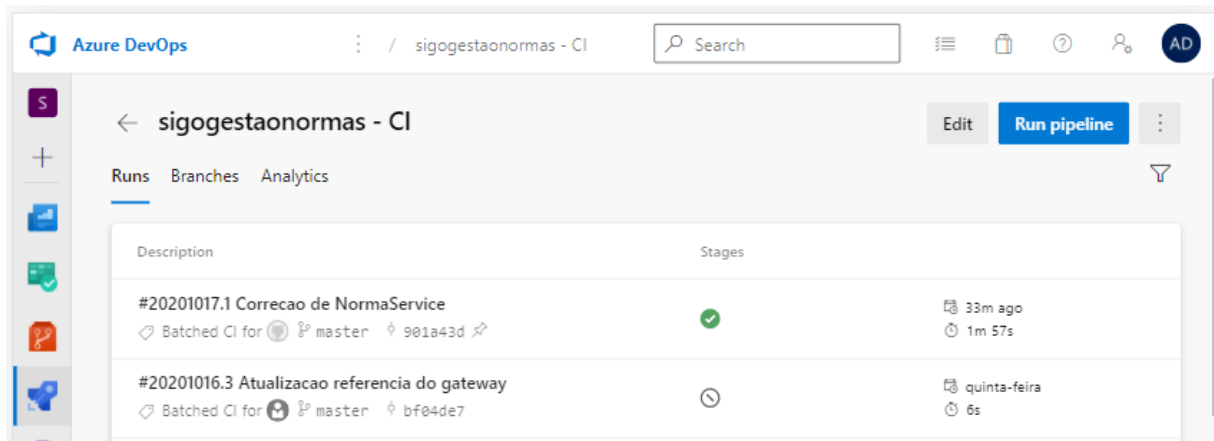


Figura 26 - Cenário 6 - Integração Contínua concluída

A entrega contínua concluída é apresentada na Figura 27. Com a conclusão desta etapa um e-mail é enviado para os responsáveis.

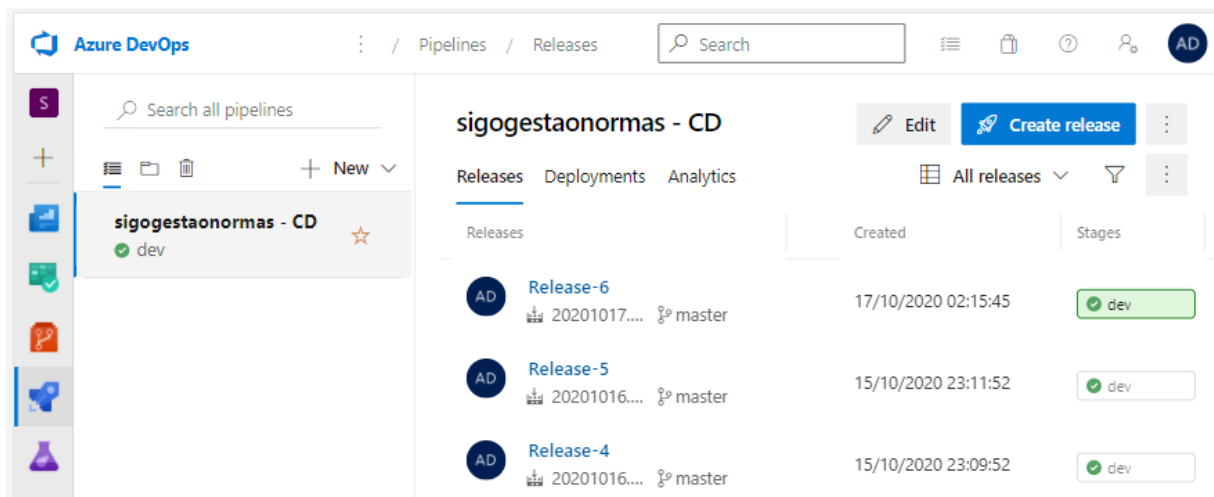


Figura 27 - Cenário 6 - Entrega Contínua concluída

O e-mail enviado pelo Azure com as informações do processo de CI/CD é exibido na Figura 28.

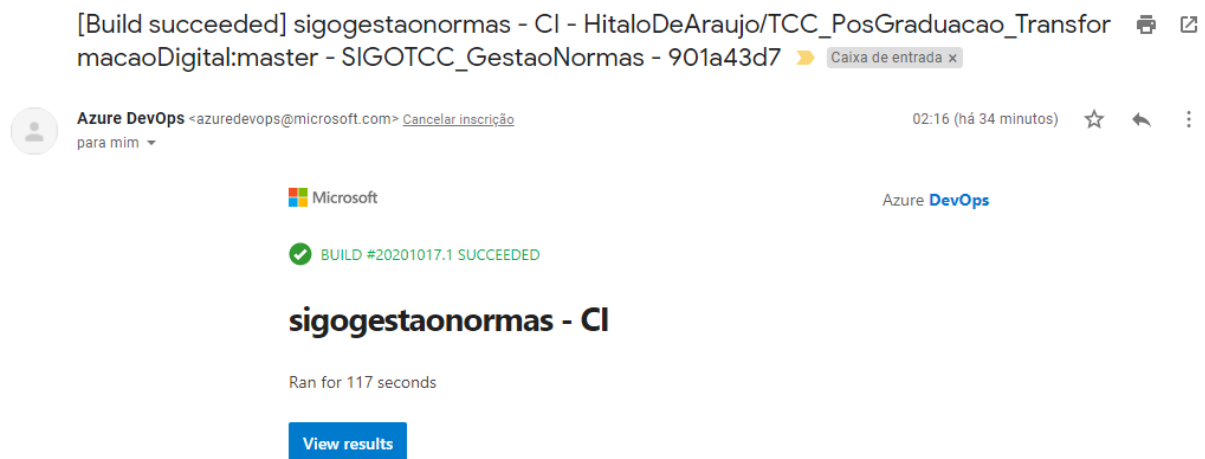


Figura 28 - Cenário 6 - Mensagem de conclusão recebida pelo responsável

6.4. Resultado

Com a realização da validação da arquitetura baseada nos requisitos de qualidade e a criação da POC, percebe-se que os requisitos foram atendidos. O quadro abaixo apresenta os requisitos não funcionais validados:

Requisitos não funcionais	Testado	Homologado
Segurança – Acesso a recursos privados com autenticação.	Sim	Sim
Interoperabilidade – O sistema deve prover comunicação entre módulos e sistemas legados.	Sim	Sim
Usabilidade – O sistema deve apresentar funcionalidades simples e objetivas e navegação intuitiva.	Sim	Sim
Acessibilidade – O sistema deve funcionar de forma consistente em diferentes telas, resoluções, sistemas operacionais e navegadores.	Sim	Sim
Esteira CI/CD – O sistema deve possuir esteira de CI/CD configurada.	Sim	Sim

Quadro 4 - Quadro resultados

7. Conclusão

Este trabalho apresentou o projeto de uma arquitetura completa para o projeto de transformação digital da IndTexBr. O SIGO está inserido na nova forma de atuação da empresa no ramo têxtil e em um contexto desafiador propõe bases tecnológicas consistentes para evolução da organização. O SIGO é formado por três módulos construídos, que são: módulo de Gestão de Normas, módulo de Assessorias e Consultorias e finalmente o módulo de Gestão do Processo Industrial. Além dos módulos planejados e construídos, fazem parte do ecossistema tecnológico do SIGO uma série de sistemas legados devidamente integrados.

Para validar a arquitetura proposta foi desenvolvida uma prova de conceito envolvendo os requisitos críticos do sistema. Neste desenvolvimento foram usadas tecnologias livres, robustas e com boa aceitação no mercado como .NET Core, Angular 10 e MySQL.

Constata-se que o projeto arquitetural do sistema integrado de gestão e operação não é uma tarefa simples. Diferentes abordagens possuem vantagens e desvantagens. As seções de avaliações e a construção do protótipo mensuram a adequação do modelo aos requisitos esperados.

Seguindo a perspectiva de constante evolução, espera-se avaliar alguns pontos durante a utilização do SIGO, como:

- Uso de orquestradores de containers como *Kubernetes*;
- Manutenção ou exclusão do uso combinado do ORM EF Core e do Micro ORM *Dapper*;
- Publicação totalmente automatizada ou publicação com confirmação manual de responsáveis.

REFERÊNCIAS

KRZYWIEC, Wojciech. **Build and run Angular application in a Docker container**. Disponível em: <<https://medium.com/@wkrzywiec/build-and-run-angular-application-in-a-docker-container-b65dbbc50be8>>. Acesso em: 01 outubro de 2020.

DAPPER. **Dapper Tutorial**. Disponível em: <<https://dapper-tutorial.net/dapper>> Acesso em: 15 de setembro de 2020.

MICROSOFT. **Implementar Gateways de API com o Ocelot**. Disponível em: <<https://docs.microsoft.com/pt-br/dotnet/architecture/microservices/multi-container-microservice-net-applications/implement-api-gateways-with-ocelot>> Acesso em: 28 de setembro de 2020.

APÊNDICES

URL do projeto no GitHub:

https://github.com/HitaloDeAraujo/TCC_PosGraduacao_TransformacaoDigital

Gestão do Processo Industrial:

<https://sigogestaoprocessoindustrial.azurewebsites.net/swagger/index.html>

Gestão de Normas:

<https://sigogestaonormas.azurewebsites.net/swagger/index.html>

Assessorias e Consultorias:

<https://sigoassessoriasconsultorias.azurewebsites.net/swagger/index.html>

ApiGateway:

<https://sigoapigateway.azurewebsites.net>

Interface Web:

<https://sigoweb.azurewebsites.net/>

Login no SIGO: admin@sigo.com.br

Senha para o SIGO: (Para fins de teste a validação de senha foi desabilitada, basta digitar algum caractere)

URL do vídeo mostrando uma apresentação da POC: <https://youtu.be/Q36rRY1B5os>

Conta para acesso aos pipelines e serviços do Azure: sigotcc@gmail.com