**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS  
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA**

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

**Thallys Freed dos Santos Silva**

**SISTEMA INTEGRADO DE GESTÃO E OPERAÇÃO**

Belo Horizonte

2021

**Thallys Freed dos Santos Sliva**

**SISTEMA INTEGRADO DE GESTÃO E OPERAÇÃO**

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

Orientador(a):

Belo Horizonte

2021

**RESUMO**

Este trabalho tem por objetivo abordar a criação de um sistema para auxiliar a transformação digital em uma empresa do ramo *têxtil,* atualmente a empresa passa por dificuldades financeiras e vê a transformação digital como um disruptor no mercado. Por se tratar de uma grande empresa em âmbito nacional, a IndTexBr já possui vários sistemas, porém os mesmos não possuem integração eficaz.

Com o desejo de realizar a transformação digital através da integração e aproveitamento das informações surge o SIGO (Sistema Integrado de Gestão e Operação), que deverá se integrar aos sistemas existentes e centralizar a gestão da informação, fornecendo informações assertivas e relevantes para a empresa.

Este trabalho visa fornecer um projeto arquitetural que possibilite uma solução de integração entre o SIGO e os sistemas existentes. Como conclusão do projeto esperasse que as integrações estejam bem definidas e demonstradas com a prova de conceito realizada a partir das definições deste trabalho.

**Palavras-chave:** arquitetura de software, projeto de software, requisitos arquiteturais, integração, transformação digital.

**SUMÁRIO**

1. Objetivos do trabalho 7

2. Descrição geral da solução 7

2.1. Apresentação do problema 7

2.2. Descrição geral do software (Escopo) 7

3. Definição conceitual da solução 8

3.1. Requisitos Funcionais 8

3.2 Requisitos Não-Funcionais 8

3.3. Restrições Arquiteturais 8

3.4. Mecanismos Arquiteturais 8

4. Modelagem e projeto arquitetural 9

4.1. Modelo de componentes 9

4.2. Modelo de implantação 9

4.3. Modelo de dados (opcional) 9

5. Prova de Conceito (POC) / protótipo arquitetural 9

5.1. Implementação e Implantação 10

5.2 Interfaces/ APIs 10

6. Avaliação da Arquitetura 11

6.1. Análise das abordagens arquiteturais 11

6.2. Cenários 11

6.3. Avaliação 11

6.4. Resultado 11

7. Conclusão 12

REFERÊNCIAS 13

APÊNDICES 14

CHECKLIST PARA VALIDAÇÃO DOS ITENS E ARTEFATOS DO TRABALHO 15

## 1. Objetivos do trabalho

O objetivo deste trabalho é apresentar uma proposta de solução arquitetural para o SIGO (Sistema de Gestão Integrado de Gestão e Operação), esse sistema visa oferecer aos setores da empresa IndTextBr a integração com sistemas adjacentes, fornecendo aos setores da empresa maior assertividade nas informações sendo de grande apoio para a transformação digital com a redução de custos na produção, aumento da lucratividade e aumento na competitividade no mercado. O sistema deverá ser uma aplicação distribuída e implantável por módulos.

Esse trabalho tem os seguintes objetivos específicos:

* Criar módulo para gestão das normas técnicas existentes nas áreas industriais, ambientais e outras. Esse módulo deve possuir uma base de *compliance* que irá auxiliar no uso correto e eficiente de recursos. Deverá ser disponibilizado um relatório de normas através de função Saas/Faas acessível por micro serviço.
* Criar módulo de consultorias e assessorias que deverá permitir a gestão contratual de consultorias do mercado que irão auxiliar à empresa na adequação às normas e padrões de negócios. Esse módulo deverá ser integrado aos outros módulos permitindo acesso às informações geradas.
* Criar módulo para gestão do processo industrial que terá como objetivo realizar integrações entre os demais módulos e também com a solução ERP utilizada pela empresa.
* A aplicação deverá ser distribuída e ser hospedada em nuvem híbrida com componentes *on premise,* deve ser implantável por módulos e ter arquitetura de micro serviços.

# 2. Descrição geral da solução

## 2.1. Apresentação do problema

Atualmente organizações consolidadas no mercado acabam perdendo muita competitividade por não ainda não terem passado pelo processo de transformação onde a tecnologia é peça chave para solucionar problemas tradicionais, o caso da IndTexBr não é diferente, apesar da empresa já possuir sistema que auxiliam seu processo existem grandes perdas financeiras e operacionais ocasionados pela não aproveitamento de informações relevantes para o negócio.

A falta de comunicação entre os módulos existentes ocasiona atrasos operacionais e falhas nos processos de decisões que causam atraso em relação ao mercado e defasagem na empresa como um todo.

A forte pressão sofrida principalmente pelo mercado exterior faz com que a empresa que tem abrangência nacional busque melhoria em seus processos a fim de reduzir seus custos e consequentemente aumentar sua competividade frente aos concorrentes do mercado nacional e internacional.

Atualmente todas os dados gerados pelos diversos sistemas utilizados pela empresa não auxiliam no crescimento da mesma não sendo em momento algum utilizados como vantagem nas tomadas de decisões organizacionais. Tendo em vista a atual situação da empresa é necessário fazer com que as informações da empresa se integrem e possam ser utilizadas de forma assertiva e como ferramentas de alavancagem no mercado.

## 2.2. Descrição geral do software (Escopo)

O SIGO através de seus módulos terá a missão de torna a integração com os demais sistemas tradicionais do setor têxtil e também com o ERP utilizado na empresa. Além da integração com ERP baseado em SAP, o SIGO irá se integrar com as seguintes soluções já existentes:

* Sistema de Logística
* Sistema de Gestão de Processos Industriais
* Monitoramento de Vendas
* Segurança e qualidade
* Inteligência de negócio
* Relatórios de acompanhamento

As integrações serão realizadas com soluções baseadas em SAP e também com plataformas legadas como protocolo SOAP, e linguagens como Java e ABAP além do banco relacional Oracle e *middlewares* Orientados a Mensagem, sendo que essas integrações podem acontecer tanto de forma síncrona quanto de forma assíncrona.

Através do SIGO os usuários terão uma fonte centralizada de informações que vão ser fundamentais na tomada de decisões e na mudança organizacional. O SIGO será um sistema utilizado por diversos setores da empesa, desde colaboradores do processo fabril até funcionário que tomam de decisões gerenciais.

# 3. Definição conceitual da solução

## 3.1. Requisitos Funcionais

**Módulo Gestão de Normas**

* Cadastro *compliance*

O sistema deve permitir que os usuários realizem o cadastro de *compliance* que contemple uso de recursos, descarte de sub produtos no meio ambiente e também monitoramento de acidentes.

* Cadastro de normas técnicas

O sistema deve permitir que o usuário realize o cadastro de normas técnicas das áreas industriais, ambiental e técnicas.

* Repositório de normas

O sistema deverá possuir um repositório de normas sempre disponível para consultas através da web.

**Módulo Consultoria e Assessorias**

* Cadastro de consultorias e assessorias

O sistema deverá permitir que o usuário realize o cadastro de consultorias de mercado que possam atender a empresa.

* Contratação de consultorias e assessorias

O sistema deve permitir que o usuário realize a gestão dos contratos com as consultorias e assessorias que prestem serviço para a empresa.

* Integração com módulo de Gestão Industrial e módulo de Gestão de Normas

Deverá existir integração com os módulos de Gestão Industrial e Gestão de Normas para que os usuários possam visualizar as informações desses módulos.

* Geração de insumos para planejamento e controle de atividades industriais

O sistema deverá fornecer informações que permitam aos usuários planejar e controlar as atividades industrias de acordo as normas vigentes

**Módulo Gestão do Processo Industrial**

* Integração com módulo de gestão de normas, Integração com módulo de consultoria e assessorias, Integração com sistemas de logística, Integração com sistema de gestão de processos industriais, Integração com sistema de monitoramento de vendas, Integração com sistema de segurança e qualidade

Deverá existir integração com os novos módulos e sistemas existentes para que através do SIGO os usuários possam ter acesso facilitado as informações.

## 3.2 Requisitos Não-Funcionais

Abaixo estão descritos os requisitos não funcionais do sistema:

* Usabilidade – O sistema deve prover boa usabilidade

|  |  |
| --- | --- |
| **Estímulo** | Usuário inserindo norma técnica |
| **Fonte do Estímulo** | Usuário acessando a funcionalidade de cadastro de normas técnicas |
| **Ambiente** | Funcionamento, carga normal |
| **Artefato** | Módulo Gestão de normas |
| **Resposta** | Sistema apresenta navegação fácil, simples e objetiva |
| **Medida de resposta** | O cadastro da norma realizada em 10 minutos |

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

|  |  |
| --- | --- |
| **Estímulo** | Cadastro de Consultoria |
| **Fonte do Estímulo** | Usuário acessando o sistema através de um tablet |
| **Ambiente** | Funcionamento, carga normal |
| **Artefato** | Módulo Consultoria e Assessorias |
| **Resposta** | As telas do sistema se adaptaram a resolução do dispositivo, fazendo a renderização correta dos componentes na tela deixando os mesmo em posição que facilite a navegação do usuário. |
| **Medida de resposta** | O sistema apresenta layouts semelhantes em todas as resoluções que for utilizado, onde componentes e imagens são exibidos de acordo com resolução e tamanho do dispositivo utilizado. |

* Desempenho – O sistema deve ser rápido

|  |  |
| --- | --- |
| **Estímulo** | Geração de insumos para planejamento e controle de atividades industriais |
| **Fonte do Estímulo** | Usuário coletando informações para o planejamento de atividades industriais |
| **Ambiente** | Funcionamento, carga normal |
| **Artefato** | Módulo Consultoria e Assessorias |
| **Resposta** | O sistema respondeu todos os dados necessários para que o usuário realize suas atividades. |
| **Medida de resposta** | O tempo de resposta do sistema foi menor que 10 segundos. |

* Desempenho – Processamento de dados nas integrações com o relatório de acompanhamento

|  |  |
| --- | --- |
| **Estímulo** | Geração do relatório de acompanhamento |
| **Fonte do Estímulo** | Usuário solicitando a geração de relatório de acompanhamento |
| **Ambiente** | Funcionamento, carga normal |
| **Artefato** | Módulo de Gestão do Processo Industrial |
| **Resposta** | O sistema gerou o relatório com os dados vindos da integração |
| **Medida de resposta** | O sistema não deverá possuir defasagem de dados superior a 30 segundos em relação ao módulo de acompanhamento e o relatório deve ser gerado em no máximo 20 segundos |

* Testabilidade – O sistema deve ser simples para testar

|  |  |
| --- | --- |
| **Estímulo** | Execução de testes no sistema |
| **Fonte do Estímulo** | Desenvolvedor |
| **Ambiente** | Ambiente de desenvolvimento |
| **Artefato** | Módulo de Gestão do Processo Industrial, Módulo de Gestão de Normas, Modulo de Consultoria e Assessoria |
| **Resposta** | Testes realizadas em funcionalidades de cadastro, edição e exclusão de dados e nas integrações entre os módulos |
| **Medida de resposta** | O sistema dede possibilitar que os testes sejam executados com apenas um comando. |

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

|  |  |
| --- | --- |
| **Estímulo** | Teste de integração |
| **Fonte do Estímulo** | Sistema externo |
| **Ambiente** | Funcionamento, carga normal |
| **Artefato** | Módulo de Gestão do Processo Industrial |
| **Resposta** | O serviço que recebe os dados para a integração respondeu as solicitações com sucesso |
| **Medida de resposta** | Integração realizada com tempo de resposta menor que 2 segundos |

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

|  |  |
| --- | --- |
| **Estímulo** | Deploy em ambiente de produção |
| **Fonte do Estímulo** | Ciclo de entrega continua |
| **Ambiente** | Funcionamento, carga elevada com diversos usuários logados |
| **Artefato** | *Load balance* |
| **Resposta** | Os usuários que estão utilizando o sistema continuaram normalmente, sem perceber que o sistema foi atualizado. |
| **Medida de resposta** | Todas as requisições realizadas pelos usuários devem ser atendidas em no máximo 3 segundos. |

* Segurança – O sistema deve apresentar altos padrões de segurança

|  |  |
| --- | --- |
| **Estímulo** | Enviar requisição para integração de dados sem informar um *token* válido |
| **Fonte do Estímulo** | Requisição http externa |
| **Ambiente** | Funcionamento com carga normal |
| **Artefato** | Módulo de gestão do processo industrial |
| **Resposta** | O sistema deve barrar a requisição. |
| **Medida de resposta** | O sistema não deve permitir qualquer tipo de integração que não possua um *token* válido. |

* Manutenibilidade – O sistema deve apresentar manutenção facilitada

|  |  |
| --- | --- |
| **Estímulo** | O componente responsável pela integração com o módulo de vendas precisa ser atualizado |
| **Fonte do Estímulo** | Lentidão e perca de dados na integração |
| **Ambiente** | Funcionamento com carga normal |
| **Artefato** | Módulo de Gestão do Processo Industrial |
| **Resposta** | Após a atualização todos os dados perdidos devem ser integrados. |
| **Medida de resposta** | Todos os dados não processados devem ser importados e estar consistentes com o módulo vendas, após o reprocessamento o sistema deve estar operando com no máximo 30 segundos de defasagem nos dados. |

## 3.3. Restrições Arquiteturais

* O sistema deve ser desenvolvido em JAVA utilizando o framework Spring Boot
* O sistema deve utilizar o serviço dos correios para disponibilizar os dados precisos de cadastro de endereço e CEP.
* O sistema deve abrir de forma responsiva em aparelhos menores, como celular e tablet.
* O sistema deve ser modular para facilitar a implantação.
* As integrações entre os sistemas devem utilizar JWT(Json Web Token).

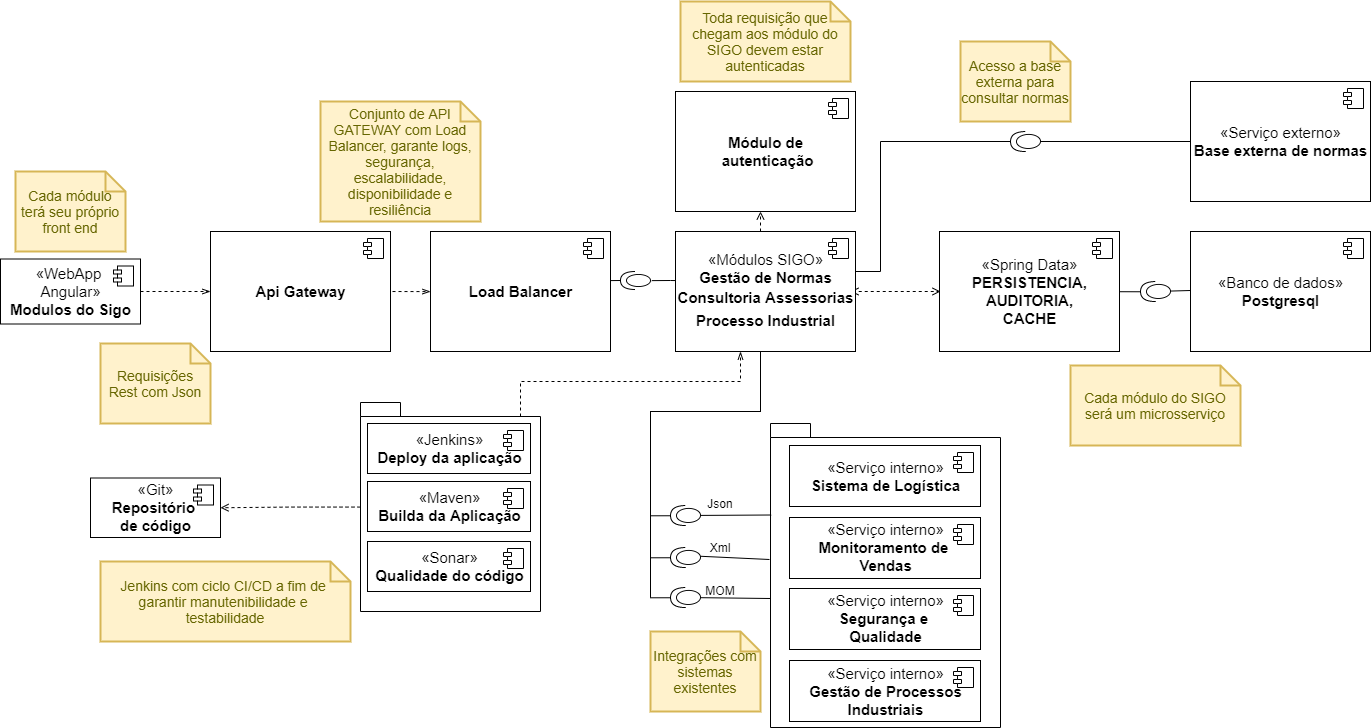
## 3.4. Mecanismos Arquiteturais

|  |  |  |
| --- | --- | --- |
| **Mecanismo de Análise** | **Mecanismo de Design** | **Mecanismo de Implementação** |
| Persistência | Banco de dados relacional | PostgreSQL |
| Persistência | Framework ORM | Spring Data JPA |
| Acesso de dados em memória | Framework de Cache | Spring Cache |
| Integração com sistemas legados | Interface utilizando JSON | WebService e WebAPI |
| Logs | Framework | Log4j |
| Auditoria | Framework ORM | Hibernate Envers |
| Build | Geração de artefato para implantação | Maven |
| Deploy | Testes automatizados e deploy da aplicação. Ciclo CI/CD | Jenkins |
| Front-End | Interface utilizada pelo usuário | Angular 9, Bootstrap |
| Versionamento | Versionamento de código fonte da aplicação | Git |
| Alta disponibilidade | *Load balance* | Spring Cloud Loadbalancer |

# 4. Modelagem e projeto arquitetural

## 4.1. Modelo de componentes

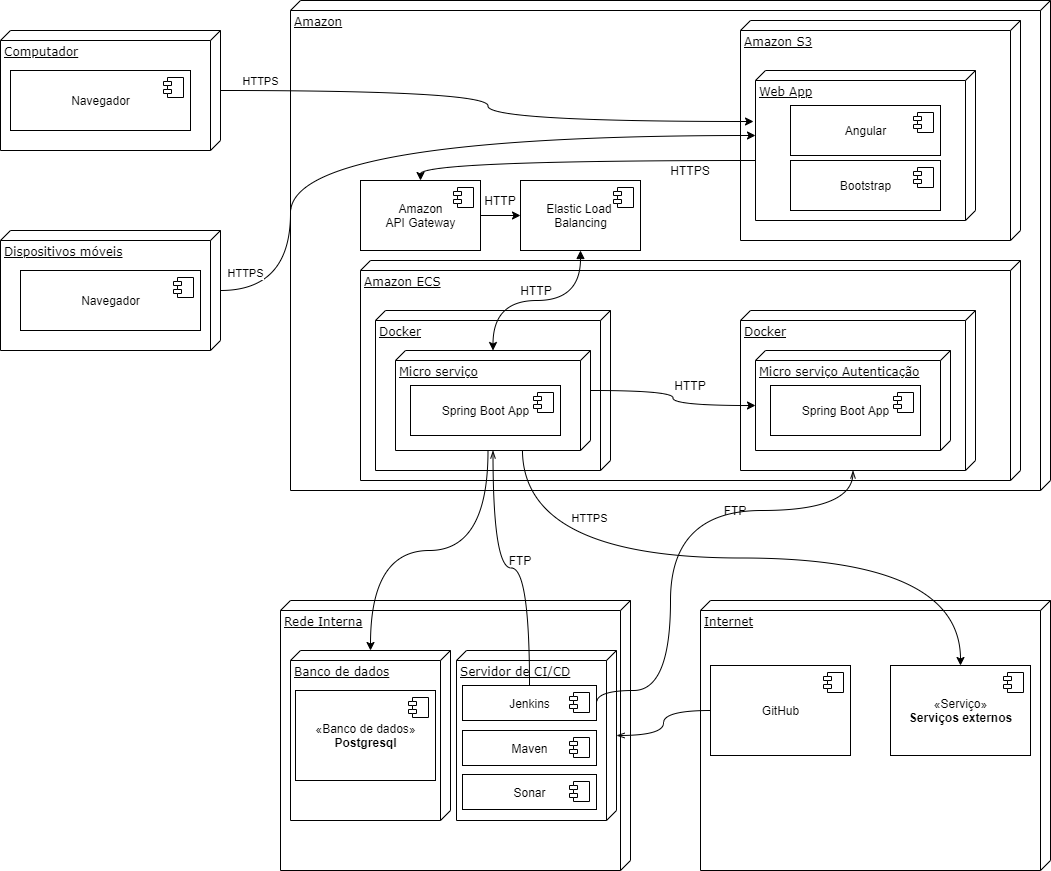
Através do diagrama de componentes do sistema é possível observar o design da arquitetura do sistema e também a seleção das tecnologias. Para o SIGO foram selecionadas tecnologias consolidadas para micro serviços sendo o modelo apresentado abaixo único para todos os módulos do SIGO, pois permite alta escalabilidade, disponibilidade, segurança e manutenibilidade.



Abaixo estão apresentados o detalhamento dos componentes e módulos selecionados para a arquitetura do SIGO. Os módulos do sistema irão possuir uma arquitetura dividida em aplicativo Web desenvolvido utilizando angular 9 e Bootstrap e micros serviço desenvolvido utilizando Spring Boot, sendo que cada módulo possuirá um seu próprio front e um ou mais micro serviços orientados ao assunto do módulo. Para atender ao RNF de disponibilidade e também ter escalabilidade, as requisições enviadas aos micros serviços serão direcionadas para um API Gateway que fará o redirecionamento para o Load Balancer que através da carga atual do sistema irá escolher o melhor container para executar a requisição no momento. O requisito não funcional de segurança será atendido pelo módulo de autenticação, para toda requisição recebida será feita validação nesse módulo para garantir que o usuário está logado e se tem permissão ao recurso solicitado. Será utilizado também o cache e auditoria do Spring Data afim de garantir o RNF de desempenho, uma vez que requisições repetidas serão direcionadas ao cache, com a auditoria haverá rastreabilidade de operações realizadas pelos usuários e também possibilidade de reverter as mesmas. É também um RNF do sistema a testabilidade e manutenibilidade, para isso é de suma importância ciclos de integração e entrega continua, para isso o sistema irá contar com o Jenkins que através de pipelines irá garantir esses requisitos e aliado ao Sonar fazer análise de qualidade do código.

|  |  |
| --- | --- |
| Componente | Descrição |
| WebApp Angular | Componente utilizada para o front end dos apps |
| Api Gateway | Componente utilizada para expor as API´s na web |
| Load Balancer | Componente utilizada para direcionar requisições para containers que não estejam sobrecarregados |
| Módulo de autenticação | Componente utilizada para realizar autenticação no sistema |
| Módulos SIGO | Componente que representa os módulo do SIGO. |
| Spring Data | Framework utilizado para persistência na base de dados, auditoria de transações e cache. |
| Serviço Externo | Serviços expostos na web que o SIGO consome para realizar operações |
| Postgresql | Componente utilizada para armazenar os dados da aplicação |
| Integração Continua | Componentes para a construção dos artefatos a serem implantados nos containers, faz a gestão de dependências, testes e qualidade de código |
| Serviço Interno | Sistemas legados com os quais o SIGO faz integrações |
| Git | Componente utilizada para versionamento de código dos módulos |

## 4.2. Modelo de implantação



O diagrama acima apresenta o diagrama de implantação da aplicação, são apresentados os componentes e onde os mesmos serão executados. A aplicação tem por requisito executar em nuvem e também *on premisse* em infraestrutura própria, por isso a aplicação está dividida, sendo os componentes desenvolvidos serão executados em nuvem da Amazon, a opção pela Amazon se deu pela alta confiabilidade que os serviços da mesma provêm. A aplicação Angular estará hospeda no serviço de armazenamento da Amazon conhecido por S3 (Simple Sotrage Service), todas as requisições feitas pelo Web App serão direcionadas paro Amazon API Gateway que irá direcionar para o Elastic Load Balance também produto Amazon que contribui com a escalabilidade e disponibilidade dos serviços uma vez que analisa o estado dos containers a fim de escolher o menos sobrecarregado para atender a requisição. Os micros serviços serão construídos utilizando o *framework* *Spring Boot* juntamente com a linguagem Java, a escolha se deu por serem tecnologias consagradas no mercado e que atendem todos os requisitos de desempenho e segurança, iram executar no ECS (*Amazon Elastic Container Service*) da Amazon em containers Docker o que irá garantir grande escalabilidade e disponibilidades para os serviços. Todo o ciclo de integração e entrega continua deveram ser configurados em servidos Jenkins interno, o código da aplicação será armazenado no GitHub que possui alta confiabilidade e fácil integração com Jenkins.

|  |  |
| --- | --- |
| Componente | Descrição |
| Navegador | Esse componente representa o meio pelo qual o usuário realiza interações com o sistema, sendo por um dispositivo móvel ou computador |
| Servidor CI/CD | Componentes para a construção dos artefatos a serem implantados nos containers, faz a gestão de dependências, testes e qualidade de código |
| Elastic Load Balance | Componente utilizada para direcionar requisições para containers que não estejam sobrecarregados |
| Amazon S3 | Componente utilizada para armazenar o conteúdo estático do sistema, que nesse caso é aplicação angular |
| GitHub | Componente utilizada para versionamento de código dos módulos |
| Amazon ECS/ Docker | Componente responsável por executar a aplicação Spring Boot que irá atender as requisições do front |
| Banco de dados | Componente onde são armazenados os dados persistidos pelos serviços |
| Amazon | Nuvem utilizada para prover infraestrutura para a aplicação |
| Rede Interna | Rede *on premisse* da empresa |
| Internet | Rede onde estão os serviços que o SIGO consome para realizar operações |

# 5. Prova de Conceito (POC) / protótipo arquitetural

Nesta parte do trabalho é apresentada a POC desenvolvida com o intuito de atender aos requisitos especificados neste relatório.

Além de descrever a POC, faz-se necessário deixar nesta seção o *link* do repositório da aplicação, lembrando que deve ser implementada somente parte da especificação fornecida.

## 5.1. Implementação e Implantação

Descreva a implementação da prova de conceito da arquitetura (protótipo arquitetural) da sua aplicação, apresentando suas características em termos de código. Para isto indique:

* As **tecnologias** utilizadas na sua implementação.
* Os Casos de Uso (UC) ou outro mecanismo, como Histórias de Usuário, que atendem aos requisitos funcionais que serão implementados para validar a arquitetura proposta. **Deve-se pelo menos três casos de uso (ou Histórias de Usuário) descritos(as).** Apresente nos UC um protótipo das interfaces (telas) da aplicação.
* Os **requisitos não funcionais** que serão avaliados. Devem ser definidos pelo menos três requisitos não funcionais. Indique os critérios de aceitação para cada um deles.

Faça a implantação da sua prova de conceito. Isto pode ser feito de diversas formas: em nuvem, utilizando um servidor *web*, aplicativo baixado para *smartphone* etc. Indique no apêndice onde (*link* ou endereço) sua prova de conceito está disponível para ser executada.

Faça um vídeo de apresentação da POC e disponibilize de forma que a banca de avaliação do TCC possa ver. Informe a URL no apêndice deste trabalho.

**Coloque aqui a descrição do código, deixando visível a estrutura de seus componentes (pode utilizar uma figura, se desejar). Acrescente ainda um *link* para o repositório de código utilizado.**

## 5.2 Interfaces/ APIs

Caso exista algum componente na arquitetura da sua aplicação que é genérico e pode ser utilizado em outras aplicações semelhantes, documente a interface desse componente seguindo um modelo de documentação de interfaces.

# 6. Avaliação da Arquitetura

A avaliação da arquitetura desenvolvida neste trabalho é abordada nesta seção, visando avaliar se atende ao que foi proposto. É um item não obrigatório mas desejável, pois permite que você verifique se sua arquitetura está completa e coerente.

## 6.1. Análise das abordagens arquiteturais

Apresente aqui um breve resumo das principais características da proposta arquitetural. Para isto, considerando o método *Architecture. Tradeoff Analysis Method* (ATAM), no qual são utilizados cenários para esta análise.

## 6.2. Cenários

Mostre os cenários utilizados na realização dos testes da sua aplicação. Escolha cenários de testes que demonstrem os requisitos não funcionais (atributos de qualidade) sendo satisfeitos. Priorize os cenários para a avaliação segundo critérios quantitativos ou qualitativos.

## 6.3. Avaliação

Apresente as evidências dos testes de avaliação. Apresente as medidas registradas na coleta de dados. Para aquilo que não for possível quantificar apresente uma justificativa baseada em evidências qualitativas que suportem o atendimento aos requisitos não-funcionais. As evidências das avaliações neste item são fundamentais.

## 6.4. Resultado

Apresente uma avaliação geral da arquitetura produzida, indicando seus pontos fortes e suas limitações. Indique possíveis ajustes/melhorias que podem ser feitos.

## 7. Conclusão

Faça uma avaliação geral do trabalho. Indique se os objetivos foram atendidos, as limitações do resultado alcançado e as dificuldades encontradas, do ponto de vista da área de arquitetura de software.

# REFERÊNCIAS

Como um projeto da arquitetura de uma aplicação não requer revisão bibliográfica a inclusão das referências não é obrigatória, embora seja recomendada. Caso você deseje incluir referências relacionadas às tecnologias, padrões, ou metodologias que utilizadas no seu trabalho relacione-as de acordo com o modelo a seguir.

Siga as normas ABNT, disponíveis em [www.pucminas.br](http://www.pucminas.br), no *link*: <http://portal.pucminas.br/imagedb/documento/DOC_DSC_NOME_ARQUI20160217102425-n.pdf>.

Exemplo:

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

# APÊNDICES

Inclua o URL do repositório (Github, Bitbucket, etc) onde você armazenou o código da sua prova de conceito/protótipo arquitetural da aplicação como anexos. A inclusão da URL desse repositório de código servirá como base para garantir a autenticidade dos trabalhos.

Inclua o URL do vídeo mostrando uma apresentação da POC, contendo de 3 a 5 minutos. Utilize preferencialmente o formato AVI.

# 

# CHECKLIST PARA VALIDAÇÃO DOS ITENS E ARTEFATOS DO TRABALHO

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nº** | **Item a ser cumprido** | **Sim** | **Não** | **Não se aplica** |
| **Completeza do documento** | | | | |
|  | Todos os elementos iniciais do documento (capa, contracapa, resumo, sumário...) foram definidos? |  |  |  |
|  | Os objetivos do trabalho (objetivos gerais e pelo menos três específicos) foram especificados? |  |  |  |
|  | Os requisitos funcionais foram listados e priorizados? |  |  |  |
|  | Os requisitos não funcionais foram listados e identificados usando o estilo estímulo-resposta? |  |  |  |
|  | As restrições arquiteturais foram definidas? |  |  |  |
|  | Os mecanismos arquiteturais foram identificados? |  |  |  |
|  | Um diagrama de caso de uso foi apresentado junto com uma breve descrição de cada caso de uso? |  |  |  |
|  | Um modelo de componentes e uma breve descrição de cada componente foi apresentada? |  |  |  |
|  | Um modelo de implantação e uma breve descrição de cada elemento de hardware foi apresentada? |  |  |  |
|  | Prova de conceito: uma descrição da implementação foi feita? |  |  |  |
|  | Prova de conceito: as tecnologias usadas foram listadas? |  |  |  |
|  | Prova de conceito: os casos de uso e os requisitos não funcionais usados para validar a arquitetura foram listados? |  |  |  |
|  | Prova de conceito: os detalhes da implementação dos casos de uso (telas, características, etc) foram apresentadas? |  |  |  |
|  | Prova de conceito: foi feita a implantação da aplicação e indicado como foi feita e onde está disponível? |  |  |  |
|  | As interfaces e/ou APIs foram descritas de acordo com um modelo padrão? |  |  |  |
|  | Avaliação da arquitetura: foi feita uma breve descrição das características das abordagens da proposta arquitetural? |  |  |  |
|  | Avaliação da arquitetura: Os atributos de qualidade e os cenários onde eles seriam validados foram apresentados? |  |  |  |
|  | Avaliação da arquitetura: uma avaliação com as evidências dos testes foi apresentada? |  |  |  |
|  | Os resultados e a conclusão foram apresentados? |  |  |  |
|  | As referências bibliográficas foram listadas? |  |  |  |
|  | As URLs com os códigos e com o vídeo da apresentação da POC foram listadas? |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nº** | **Item a ser cumprido** | **Sim** | **Não** | **Não se aplica** |
| **Consistência dos itens do documento** | | | | |
|  | Todos os requisitos funcionais foram mapeados para casos de uso? |  |  |  |
|  | Todos os casos de uso estão contemplados na lista de requisitos funcionais? |  |  |  |
|  | Os requisitos não funcionais, mecanismos arquiteturais e restrições c arquiteturais estão coerentes com os modelos de componentes e implantação? |  |  |  |
|  | Os modelos de componentes e implantação estão coerentes com os requisitos não funcionais, mecanismos arquiteturais e restrições arquiteturais? |  |  |  |
|  | As tecnologias listadas na implementação estão coerentes com os requisitos não funcionais, mecanismos arquiteturais e restrições arquiteturais? |  |  |  |
|  | Os casos de uso e os requisitos não funcionais listados na implementação estão coerentes com o que foi listado nas seções anteriores? |  |  |  |
|  | Os atributos de qualidade usados na avaliação estão coerentes com os requisitos não funcionais na sessão 3? |  |  |  |
|  | Os cenários definidos estão no contexto dos casos de uso implementados? |  |  |  |
|  | O apresentado no item resultado está coerente com o que foi mostrado no item avaliação? |  |  |  |