

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

Murilo Eduardo Bacagini Santos

**PROJETO ARQUITETURAL DO SISTEMA INTEGRADO DE GESTÃO E
OPERAÇÃO (SIGO)**

Belo Horizonte

2021

Murilo Eduardo Bacagini Santos

**PROJETO ARQUITETURAL DO SISTEMA INTEGRADO DE GESTÃO E
OPERAÇÃO (SIGO)**

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

Orientador(a): Prof. Pedro Alves de Oliveira

Belo Horizonte

2021

AGRADECIMENTOS

Agradeço a Deus por me capacitar finalizar esse projeto, me dar saúde, inteligência e sabedoria. Agradeço a toda minha família pelo apoio e compreensão durante a elaboração desse trabalho. Agradeço a todos os colegas de turma que se disponibilizaram no compartilhamento de ideias e que apoiaram a mim e aos demais colegas para execução dos trabalhos. Por último agradeço ao orientador Pedro pelas respostas detalhadas e atenciosas no fórum do TCC.

RESUMO

Esse projeto apresenta a definição da arquitetura de um projeto de software para uma empresa fictícia chamada IndTexBr Indústria Têxtil do Brasil SA. A IndTexBr é uma indústria nacional do ramo têxtil e se encontra em fase de transição para ingresso no mercado de exportação. A empresa planeja aumentar o seu lucro através da redução de custos operacionais e de matéria prima, passando a importar parte dos produtos da China. A construção e implantação do Sistema Integrado de Gestão e Operação (SIGO) é vista como um elemento chave para auxiliar nas tomadas de decisões e otimização de processos. Um dos grandes desafios do SIGO é a integração com os diversos ativos de software existentes na organização e a obtenção de vantagens competitivas, redução de custos e otimização de processos através dessas integrações. O projeto aborda requisitos e aspectos relevantes para a definição da arquitetura e contempla a criação de provas de conceitos para validação da arquitetura proposta.

Palavras-chave: arquitetura de software, projeto de software, requisitos arquiteturais, integração de aplicações

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	11
3.3. Restrições Arquiteturais	16
3.4. Mecanismos Arquiteturais	16
4. Modelagem e projeto arquitetural.....	17
4.1. Modelo de componentes.....	17
4.2. Modelo de implantação	21
5. Prova de Conceito (POC) / protótipo arquitetural	22
5.1. Implementação e Implantação	22
6. Avaliação da Arquitetura.....	24
6.1. Análise das abordagens arquiteturais	24
6.2. Cenários	24
6.3. Avaliação	25
6.4. Resultado.....	36
7. Conclusão.....	38
REFERÊNCIAS.....	39
APÊNDICES.....	40

1. Objetivos do trabalho

O objetivo desse trabalho é descrever o projeto arquitetural do Sistema Integrado de Gestão e Operação (SIGO) da empresa fictícia IndTexBr. O sistema deverá utilizar arquitetura baseada em microsserviços e deve ter boa usabilidade, segurança no tratamento de dados, boa manutenibilidade, bom desempenho, alta disponibilidade e integração com outros sistemas da empresa.

Os objetivos específicos são:

- Descrever os requisitos funcionais relevantes para definição da arquitetura dos três módulos do sistema: Módulo de Gestão de Normas, Módulo de Consultorias e Assessorias e Módulo de Gestão do Processo Industrial
- Descrever os requisitos de qualidade do sistema
- Prover diagramas para entendimento da solução proposta
- Implementar provas de conceitos (PoC) para validação da arquitetura

2. Descrição geral da solução

2.1. Apresentação do problema

Com o advento dos computadores, as empresas começaram a se informatizar: substituíram as pilhas de papel e grandes arquivos de aço por planilhas, sistemas e banco de dados. Mais adiante, com o surgimento da Internet, as empresas começaram a vender on-line e migraram boa parte de seus sistemas desktops para os navegadores. Porém muitas dessas soluções já não atendem as novas necessidades das empresas. Hoje, em 2021, o mercado é muito dinâmico e competitivo, os clientes estão mais exigentes, e as pessoas mais impacientes. As soluções de T.I hoje precisam ir além do famoso CRUD, e proporcionar uma melhor experiência para o usuário, maior integração entre as soluções, e prover dados estratégicos.

Nesse contexto se insere a IndTexBr Indústria Têxtil do Brasil SA, uma empresa fictícia do ramo têxtil que vem passando por dificuldades financeiras devido à retração do mercado interno por conta da concorrência chinesa e a dependência de fornecedores brasileiros que possuem preços menos competitivos e prazos mais longos. Por isso, a

IndTexBr pretende começar importar produtos da China, assim reduzir o custo da produção e aumentar a margem de lucro.

A empresa precisa de uma solução que se integre a todo seu ativo de software existente e que auxilie nessa transição e recuperação financeira. É necessário cortar custos desnecessários, otimizar processos, garantir a qualidade da produção e se preparar para o mercado internacional.

2.2. Descrição geral do software (Escopo)

O sistema SIGO será composto por três módulos distintos que deverão ter um alto nível de integração entre si e com os sistemas existentes para que seja possível obter resultados para otimização de processos, melhoria da qualidade e redução de custos, porém ao mesmo tempo evitando um alto-acoplamento entre os componentes. Os módulos a serem desenvolvidos são:

Módulo de Gestão Normas: esse módulo é responsável pela gestão de normas técnicas industriais e ambientais, e deve sempre estar atualizado com as últimas normas do setor, assim garantido que a produção esteja seguindo os padrões de qualidades nacionais e internacionais, desta forma também evitando custos com multas. Esse módulo deve-se integrar ao sistema legado de Segurança e Qualidade para garantir que as últimas normas sejam aplicadas na produção.

Módulo de Consultorias e Assessorias: esse módulo deve permitir gerenciar empresas que irão prestar consultorias e assessorias a fim de identificar falhas e oportunidades de melhorias na fábrica. Essas entidades devem ter acesso à relatórios para auxiliar nos seus trabalhos. Esse módulo deverá se integrar com o sistema legado de Relatórios de acompanhamentos.

Módulo de Gestão do Processo Industrial: esse módulo deve se integrar com diversos sistemas, tais como a solução da SAP responsável pela gestão dos processos de produção, o sistema de Monitoramento de Vendas e o sistema de logística. Essas integrações irão fornecer informações e alertas para consultores, assessorias, gerentes e encarregados, bem como deve ser um facilitador para a integração e implantação de um sistema de B.I já adquirido da Oracle.

3. Definição conceitual da solução

3.1. Requisitos Funcionais

RF001 – Gestão de Módulos e Funcionalidades

O sistema deve permitir o gerenciamento (inclusão, alteração, exclusão e consulta) de módulos e funcionalidades.

RF002 – Gestão de Usuários

O sistema deve permitir a criação, alteração e inativação de usuários, solicitando dados de acesso (login e senha), tipo (interno ou externo) e dados de contato (e-mail e celular). Também deve ser possível atribuir os módulos e funcionalidades permitidas ao usuário.

RF003 – Autenticação de Usuários

O sistema deve permitir a autenticação de usuários através de login e senha, e exigir validação adicional através do envio de SMS para o celular do usuário.

Módulo de Gestão de Normas

RF004 – Gerenciamento de Normas

O sistema deve permitir o usuário incluir, alterar, excluir e consultar normas técnicas. O sistema deve receber dados como: data de publicação, data de início, título, órgão responsável, objetivo, status e a inclusão de anexos.

RF005 – Obtenção de novas normas

O sistema deve se integrar periodicamente com sistemas de empresas parceiras especializadas em gestão de normas regulatórias a fim de obter novas normas ou atualizações em normas utilizadas.

RF006 – Disponibilização de atualizações de normas para sistemas internos

O sistema deve disponibilizar uma interface de integração para que o Sistema de Segurança e Qualidade tenha acesso as atualizações e inclusões de normas, assim garantindo que as últimas normas estejam sendo aplicadas na produção.

Módulo de Consultorias e Assessorias

RF007 – Gestão de Consultorias e Assessorias

O sistema deve permitir incluir, alterar, consultar ou inativar empresas de consultoria e assessoria. Deve permitir informar dados cadastrais das empresas, como razão social, CNPJ, endereços e contatos.

RF008 – Vínculo de Usuários à Consultorias e Assessorias

O sistema deve permitir vincular um usuário externo já cadastrado à uma consultoria ou assessoria.

RF009 – Gestão de Acesso ao Sistema de Relatórios e Acompanhamentos

O Módulo de Consultorias e Assessorias, por meio de interfaces de integração com o Sistema de Relatórios e Acompanhamentos, deve permitir o usuário administrador visualizar os perfis disponíveis no sistema de relatórios então atribuir um destes perfis à empresa, e automaticamente criar usuários ou acessos no sistema de relatórios para que os assessores e consultores possam consultar os dados necessários para realizar suas análises.

RF010 – Publicação de Análises

O sistema deve permitir que os assessores e consultores publiquem os resultados de suas análises. O sistema deve solicitar que o usuário informe um resumo da análise e inclua anexos. A análise poderá ser salva como rascunho antes de ser publicada. Uma vez publicada, os dados ou anexos não podem ser alterados. Análises criadas por uma empresa não podem ser acessadas por outra, e usuários internos podem ter acesso a todas as análises publicadas.

Módulo de Gestão do Processo Industrial

RF011 – Gerenciamento e monitoramento de atrasos na produção

O sistema deve se integrar à ferramenta de modelagem de processos e permitir que o usuário defina uma meta de tempo que uma ordem de produção deve levar em cada estágio. Após definido esses intervalos, o sistema deve monitorar em tempo real cada ordem de produção e notificar imediatamente os encarregados quando houver atrasos em um estágio.

RF011 – Envio de avisos de parada de produção

O sistema deve se integrar com a solução da SAP e enviar notificações para os responsáveis quando houver uma parada na linha de produção.

RF012 – Relatório de incidentes

O sistema deve enviar diariamente um relatório com o resumo dos problemas e incidentes ocorridos na produção do dia anterior para que os gestores possam discutir soluções para os problemas ocorridos.

RF013 – Solicitação de compra automatizada

O sistema deve cruzar dados do sistema de logística (estoque atual, histórico de prazos de entrega) com os dados do monitor de vendas a fim de identificar quando será necessário iniciar a compra de matéria prima, então gerar automaticamente um rascunho do pedido, o qual será revisado pela área de compras, para então ser enviado aos fornecedores. Esse relatório deverá ser gerado em português e inglês (para fornecedores chineses).

RF014 – Dicionário multi-idioma de materiais

O sistema deve disponibilizar uma função para traduzir em vários os idiomas os nomes dos materiais cadastrados no sistema de logística para auxiliar nas automações envolvendo fornecedores estrangeiros.

RF015 – Extração de dados para B.I

O sistema deverá extrair dados de vários sistemas e consolidá-los para que possam ser utilizados pelo sistema Oracle Business Intelligence que já foi adquirido, mas não está funcionando devido as dificuldades encontradas para integrá-lo com os sistemas existentes na empresa.

3.2 Requisitos Não-Funcionais

- Segurança – o sistema deve impedir acesso de usuários não autenticados

Estímulo	Tentativa de acesso a qualquer página do sistema
Fonte de estímulo	Usuário não autenticado
Ambiente	Produção, carga normal
Artefato	Qualquer módulo do sistema
Resposta	O usuário é encaminhado para a tela de login
Medida da Resposta	O sistema não deve permitir que um usuário não autenticado acesse qualquer recurso do sistema

- Segurança – o sistema deve impedir acesso de usuários não autorizados

Estímulo	Tentativa de acesso a página não permitida ao perfil do usuário
Fonte de estímulo	Usuário autenticado sem permissão ao módulo ou função
Ambiente	Produção, carga normal
Artefato	Qualquer módulo do sistema
Resposta	A página não é exibida
Medida da Resposta	O sistema não deve permitir que um usuário autenticado, porém sem permissão ao módulo ou função acesse o recurso

- Segurança – o sistema deve armazenar senhas e informação sensível de forma segura

Estímulo	Criação de usuário
Fonte de estímulo	Usuário administrador
Ambiente	Produção, carga normal
Artefato	Tela de gestão de usuários
Resposta	O usuário é criado com sucesso
Medida da Resposta	A senha é armazenada utilizando um algoritmo de hash e o e-mail e celular são armazenados utilizando criptografia

- Segurança – conexões por meio da Internet deve utilizar conexões seguras

Estímulo	Requisição HTTP ou AMQP para a Internet
Fonte de estímulo	Qualquer módulo do sistema
Ambiente	Produção, carga normal
Artefato	Qualquer módulo do sistema
Resposta	A requisição é realizada com sucesso
Medida da Resposta	A requisição é feita utilizando HTTPS ou AMQPS

- Testabilidade – o desenvolvimento do sistema deve utilizar testes automatizados

Estímulo	Codificação de novas funcionalidades no sistema
Fonte de estímulo	Programador
Ambiente	Desenvolvimento
Artefato	Código fonte
Resposta	O novo código é criado utilizando testes unitários
Medida da Resposta	O novo código possui uma cobertura de testes automatizados de pelo menos 75% em componentes de negócio, e desejavelmente deve possuir testes de integração em componentes de acesso a dados e/ou APIs

- Manutenibilidade – a substituição de serviços de infraestrutura não deve gerar impactos na arquitetura

Estímulo	Substituição do serviço de SMS
Fonte de estímulo	Área financeira ou de gestão exige substituição do serviço SMS para diminuir custos
Ambiente	Desenvolvimento
Artefato	Código fonte – serviço de envio de SMS
Resposta	É realizado a troca do serviço de envio de SMS sem afetar outras camadas do sistema
Medida da Resposta	É substituído apenas o componente que realiza a integração com o provedor de SMS, sem alteração nos contratos da camada de lógica de negócio

- Desempenho – o sistema deve possuir bom desempenho

Estímulo	Usuário realizando operações de CRUD
Fonte de estímulo	Usuário acessando o sistema pelo navegador do computador, conectado ao WI-FI
Ambiente	Produção, carga normal
Artefato	Qualquer módulo do sistema
Resposta	O sistema atende a requisição com sucesso
Medida da Resposta	O sistema deve responder em até 3 segundos

- Disponibilidade – o sistema deve estar disponível 24 horas por dia

Estímulo	Falha em um dos nós do cluster
Fonte de estímulo	Falha generalizada no container ou no host
Ambiente	Produção, carga normal
Artefato	Qualquer módulo do sistema
Resposta	Novas instâncias do serviço afetado começam a ser provisionadas automaticamente
Medida da Resposta	O sistema continua operando, porém com um possível um aumento no tempo de resposta das requisições até a normalização do ambiente

- Usabilidade – o sistema deve ser de fácil utilização e aprendizagem

Estímulo	Consultoria publicando uma análise
Fonte de estímulo	Usuário com nenhuma ou pouca familiaridade com o sistema
Ambiente	Produção, carga normal
Artefato	Qualquer módulo do sistema
Resposta	O usuário consegue utilizar o sistema com facilidade
Medida da Resposta	O usuário consegue utilizar o sistema sem a necessidade de treinamentos pois o sistema oferece uma interface intuitiva e utiliza termos e jargões da área de negócio do usuário.

- Usabilidade – o sistema deve possuir interface responsiva

Estímulo	Acesso do sistema a partir de um dispositivo móvel
Fonte de estímulo	Usuário acessa sistema pelo celular
Ambiente	Produção, carga normal
Artefato	Qualquer módulo do sistema
Resposta	O sistema é apresentado com um layout adequado ao dispositivo do usuário
Medida da Resposta	A exibição do sistema é reajustada de forma que se torne viável e agradável a utilização do sistema sem necessidade de utilização de zoom e com espaçamento suficiente entre os botões/links para evitar que o usuário clique em opções indesejadas por engano.

- Interoperabilidade – sistema deve se comunicar com sistemas externos via APIs REST de integração

Estímulo	Obtenção de novas normas técnicas da API contratada
Fonte de estímulo	Serviço agendado
Ambiente	Produção, carga normal
Artefato	Módulo de Gestão de Normas
Resposta	Novas normas são inseridas na base de normas
Medida da Resposta	É realizado a consulta ao serviço externo de normas técnicas através de uma requisição HTTPS consumindo a API REST disponibilizada pela empresa parceira

- Interoperabilidade – comunicação entre os módulos e sistemas legados deve utilizar um broker de comunicação preferencialmente

Estímulo	Autenticação de usuário
Fonte de estímulo	Microserviço de usuários
Ambiente	Produção, carga normal
Artefato	Broker de comunicação
Resposta	Usuário é informado que um SMS será enviado
Medida da Resposta	A mensagem de envio de SMS é adicionada à fila do broker

- Interoperabilidade – a comunicação com sistemas legados deve utilizar o protocolo SOAP quando não for possível a utilização o broker de mensagens ou REST

Estímulo	Vincular perfil do sistema legado de Relatórios à uma consultoria ou assessoria
Fonte de estímulo	Usuário tentando dar acesso a uma consultoria ao sistema legado de Relatórios
Ambiente	Produção, carga normal
Artefato	Módulo de Consultorias e Assessorias
Resposta	É obtido a listagem de perfis do sistema legado de Relatórios e realizado o vínculo
Medida da Resposta	É realizado uma conexão utilizando SOAP para obter e gravar dados no sistema legado de Relatórios

- Portabilidade – o sistema deve armazenar datas no formato UTC

Estímulo	Inclusão de qualquer registro que tenha data
Fonte de estímulo	Qualquer solicitação de inclusão/edição de informação
Ambiente	Produção, carga normal
Artefato	Qualquer módulo do sistema
Resposta	O usuário visualiza as datas no horário correspondente ao seu fuso-horário
Medida da Resposta	A data é inserida no banco de dados no formato UTC

- Portabilidade – o sistema deve ser desenvolvido com tecnologias que possam rodar em containers Linux

Estímulo	Desenvolvimento do sistema
Fonte de estímulo	Desenvolvimento de novo módulo ou serviço
Ambiente	Desenvolvimento
Artefato	Qualquer módulo do sistema
Resposta	É feito deploy ou build do módulo usando Docker
Medida da Resposta	O serviço desenvolvido pode ser executado em um container Docker

3.3. Restrições Arquiteturais

- Apresentar características de aplicações distribuídas
- Ser hospedado em nuvem híbrida
- Utilizar arquitetura baseada em microsserviços
- Ser modular e implantável por módulos

3.4. Mecanismos Arquiteturais

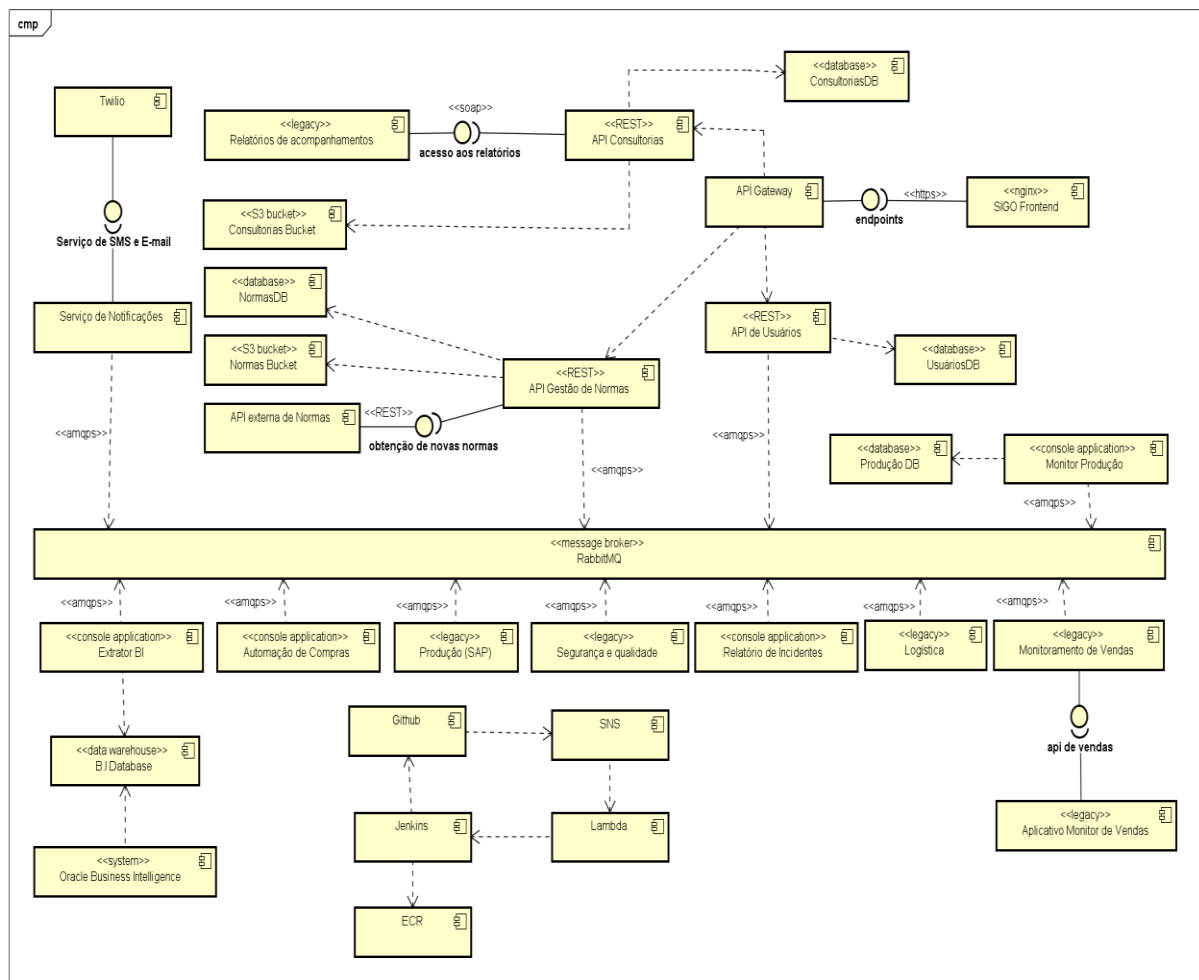
Mecanismo de análise	Mecanismo de design	Mecanismo de implementação
Frontend	Framework	Angular 8
Backend	APIs REST	ASP.NET Core 3.1
Backend	Serviços/Jobs	.NET Core 3.1
Interoperabilidade	Broker de mensagens	Amazon MQ para RabbitMQ
Interoperabilidade	Protocolos de comunicação	HTTP e SOAP
Disponibilidade	Balanceamento de carga	AWS Application Load Balancer
Gerenciamento de API's	API Gateway	ASP.NET Core e Ocelot
Segurança	Algoritmo de Hash	SHA-512
Segurança	Criptografia Simétrica	AES/Rijndael
Segurança	Autenticação e Autorização	JWT
Persistência	Banco de dados	Postgres
Persistência	Mapeamento objeto relacional (ORM)	Entity Framework Core
Armazenamento	Armazenamento de documentos	AWS S3
Inteligência de negócio	Data Warehouse	Redshit
Inteligência de negócio	Plataforma de BI	Oracle Business Intelligence
Versionamento	SCM	Github
CI/CD	Ferramenta para integração e implantação continua	Jenkins
CI/CD	Análise estática de código	SonarQube
Deploy	Mecanismo de virtualização	Docker

Deploy	Registro de imagens	AWS Elastic Container Registry (ECR)
Deploy	Orquestração de containers	AWS Elastic Container Service (ECS)
Deploy	Servidor de páginas	Nginx
Testes	Testes automatizados	XUnit
Comunicação	Envio de SMS	Twilio

4. Modelagem e projeto arquitetural

4.1. Modelo de componentes

Figura 1 - Diagrama de componentes



powered by Astah

Fonte: autor

No diagrama de componentes acima é possível visualizar os componentes do sistema SIGO, tanto componentes legados quanto os novos módulos. É possível visualizar as

principais interações entre os componentes, sendo notável que boa parte deles se comunicam com o broker de mensagens, assim diminuindo o acoplamento. Segue abaixo detalhamento dos principais componentes no diagrama:

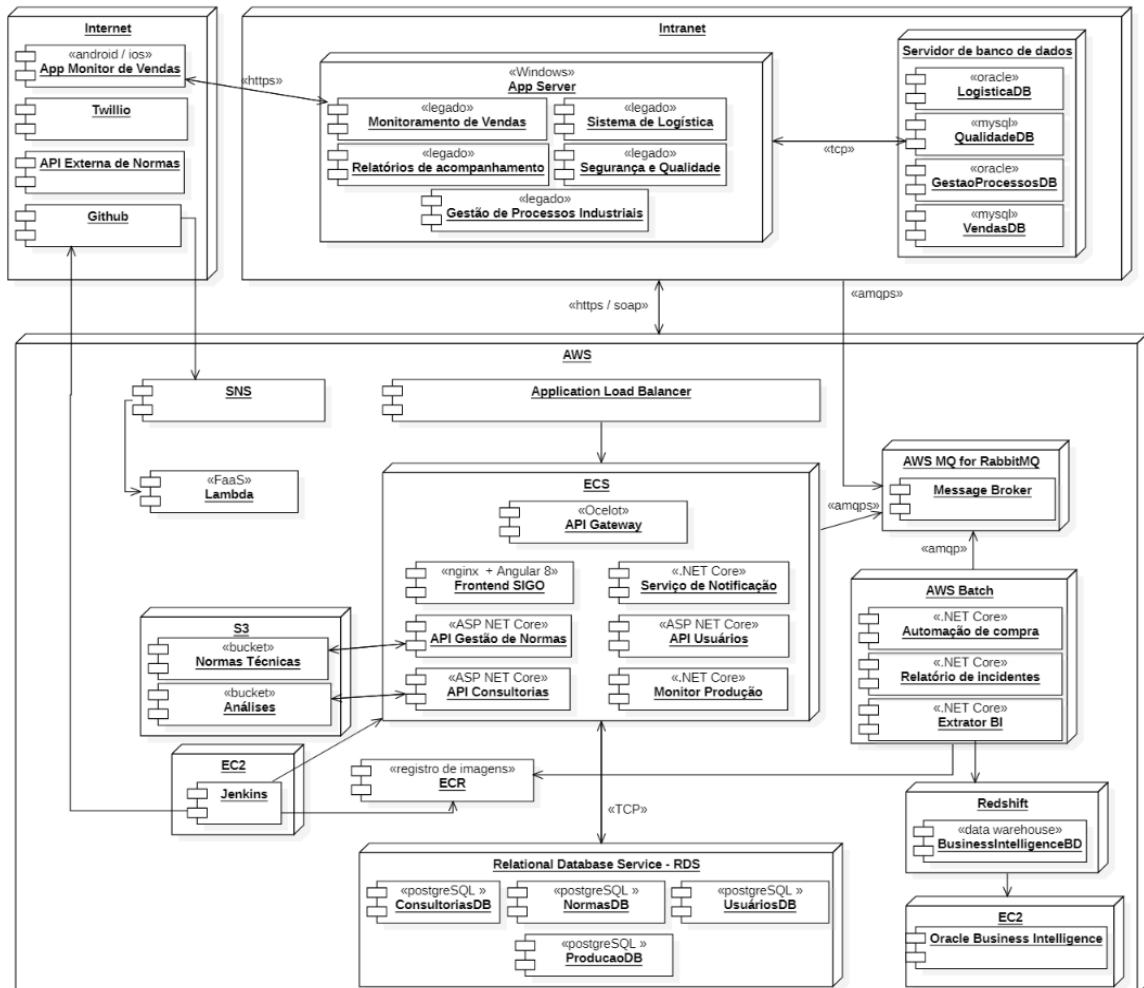
Componente	Descrição
RabbitMQ (broker)	Message broker – componente central que recebe mensagens e distribui para os componentes interessados.
Extrator BI	Aplicação .NET Core que lê mensagens do broker e as armazena no data warehouse e realiza computações e transformações.
BI Database	Banco de dados Redshift a ser utilizado pelo Oracle Business Intelligence.
Oracle Business Intelligence	Solução de B.I da Oracle – utiliza o componente BI Database (Redshift) como fonte de dados.
Automação de compras	Aplicação .NET Core executada em horários programados para consumir do broker mensagens dos sistemas de logística e vendas para gerar pedidos de compras automaticamente.
Produção (SAP)	Sistema legado da SAP responsável pelo controle da produção. Publica mensagens no broker que são de interesse do Monitor Produção e Extrator BI.
Segurança e qualidade	Sistema legado responsável pela qualidade. Se conecta no broker para obter atualizações de normas técnicas.
Relatório de Incidentes	Aplicação .NET Core executada em horários programados para enviar um relatório consolidado de incidentes e paradas na produção através do consumo de mensagens publicadas no broker.

Logística	Sistema legado responsável por processos de logística. Publica no broker mensagens que são de interesse dos componentes de Automação de Compras e Extrator B.I
API Gestão de Normas	API REST em .NET Core para o módulo de gestão de normas. Possui uma tarefa em background para obtenção de novas normas através do consumo de uma API externa. Disponibiliza novas normas no broker para o sistema legado de qualidade.
API Externa de Normas	API REST externa contratada para consulta de novas normas técnicas ou atualizações de normas.
Bucket de Normas	Repositório de arquivos de normas técnicas. Utiliza AWS S3.
API de Consultorias	API REST em .NET Core para módulo de consultorias e assessorias. Demanda interface para obter lista de relatórios do sistema legado para atribuir permissões de relatórios aos consultores.
Bucket de Consultorias	Repositório de arquivos de análises elaboradas pelas consultorias. Utiliza AWS S3.
Relatórios	Sistema legado de relatórios. Disponibiliza interface para listagem de relatórios e atribuição de permissões para os consultores.
API de Usuários	API REST em .NET Core responsável pela autenticação e autorização. Publica mensagens no broker para envio de SMS na autenticação multifator.
Monitor Produção	Aplicação .NET Core responsável por ler mensagens de paradas de produção, avaliação de atrasos na produção entre outras

	verificações. Obtém dados através da leitura de mensagens no broker, e publica mensagens para o serviço de notificação e Extrator BI.
SIGO Frontend	Aplicação frontend (Angular) servida pelo Nginx. Conhece apenas o endereço do API Gateway para não ter um alto-acoplamento com os demais serviços.
API Gateway	Ponto único de entrada das requisições feitas pelo frontend. Responsável em repassar a requisição para o microserviço solicitado.
Serviço de Notificações	Serviço responsável em ler do broker as mensagens de solicitação de envio de SMS ou e-mail, então realizar o envio com o serviço contratado.

4.2. Modelo de implantação

Figura 2 – Diagrama de Implantação



Fonte: autor

No diagrama acima a aplicação está dividida em três nodos principais: Intranet (ambiente on-premise), AWS (ambiente de nuvem) e Internet (aplicações fora da Intranet e da nuvem da empresa). Segue abaixo detalhamento de cada nodo:

No centro do diagrama é visível o cluster de containers gerenciados pelo serviço ECS (Elastic Container Service). Cada API ou componente que roda dentro desse cluster possui políticas de auto escalonamento, sendo garantido a execução de no mínimo dois containers por componente, com o tráfego distribuído pelo Application Load Balancer, assim garantindo a alta-disponibilidade. Cada container se conecta com seu banco de dados de interesse e as bases de dados rodam dentro do serviço RDS (Relational Database Service) que é um serviço gerenciado pela AWS, o que garante alta-disponibilidade no serviço de banco de dados.

O broker RabbitMq, é gerenciado pelo serviço AWS MQ, que é um serviço que proporciona alta-disponibilidade ao serviço de mensageria, uma vez que este deve ser o serviço com maior disponibilidade pois é por meio dele que acontece várias integrações entre os módulos e sistemas.

Alguns containers armazenam documentos em buckets do S3 que é um serviço de armazenamento de objetos de alta-disponibilidade e baixo custo, e realizam conexões SOAP para serviços legados que rodam na Intranet da empresa.

O servidor de deploy, Jenkins, roda dentro de uma instância EC2 Linux, porém essa permanece desligada boa parte do tempo para redução de custos. Ela é iniciada automaticamente por uma função Lambda que é disparada por um tópico SNS que é acionado por um Git Hook.

As imagens Docker resultantes do deploy são armazenadas no ECR (Elastic Container Registry). Essas imagens são utilizadas no cluster de containers do ECS e pelos serviços agendados que é gerenciado pelo serviço AWS Batch.

Durante as madrugadas o Extrator de BI é iniciado pelo AWS Batch para disponibilizar dados para a base de BI que fica no Redshift. Outros serviços agendados também são iniciados por esse serviço.

5. Prova de Conceito (POC) / protótipo arquitetural

5.1. Implementação e Implantação

Os casos de usos (requisitos funcionais) escolhidos para implementação da prova de conceito foram:

- Autenticação de usuário
- Gestão de usuários (cadastro de um novo usuário)
- Publicação de análises (módulo de consultorias e assessorias)

Os requisitos não funcionais a serem validados na PoC são:

Segurança

O sistema deve ser seguro e confiável

Critérios de aceite:

- O sistema não deve permitir o acesso de usuários não autenticados; deve redirecionar o usuário não autenticado para a página de login
- O sistema deve solicitar a verificação multifator através do envio de SMS
- O sistema não deve permitir o acesso de um usuário autenticado à um módulo que ele não possui permissão de acesso
- O sistema não deve permitir que uma consultoria visualize análises de outra

Desempenho

O sistema deve ter um bom desempenho

Critérios de aceite:

- O sistema deve responder as solicitações em até 3 segundos

Interoperabilidade

Os módulos do sistema devem se comunicar preferencialmente por troca de mensagens através um broker de comunicação

Critérios de aceite:

- Após o usuário se autenticar com o e-mail e senha, deve ser adicionado uma mensagem de no broker para envio do SMS
- O serviço de notificação deve ler a mensagem do broker e realizar o envio do SMS

Usabilidade

O sistema deve ser de fácil utilização e ter boa usabilidade

Critérios de aceite:

- O sistema deve possuir uma interface simples
- A interface deve ter uma boa usabilidade em dispositivos móveis

Disponibilidade

O sistema deve estar disponível 24 horas por dia

Critérios de aceite:

- O sistema deve continuar atendendo as requisições mesmo com a falha de um nó do cluster

O link da PoC e código fonte estão disponíveis no apêndice desse documento.

6. Avaliação da Arquitetura

6.1. Análise das abordagens arquiteturais

Essa arquitetura foi projetada para atender principalmente os requisitos de desempenho, segurança, manutenibilidade e usabilidade. Foi utilizado vários serviços gerenciados pelo provedor de cloud, tais como o gerenciamento de containers, banco de dados e mensageria, tendo em vista um ganho de disponibilidade e confiabilidade.

6.2. Cenários

Cenário 1: Na criação de um novo usuário, a senha deve ser armazenada utilizando um algoritmo de hash. O e-mail e número de celular devem ser armazenados utilizando criptografia simétrica.

Cenário 2: Ao se autenticar no sistema, após informar o login e senha, o sistema deve enviar um código de verificação por SMS para o celular do usuário para que a autenticação possa ser concluída.

Cenário 3: Na utilização do sistema através de um celular, o sistema deve apresentar uma interface adequada ao dispositivo.

Cenário 4: As operações de leitura e escrita devem ser atendidas rapidamente, levando no máximo 3 segundos.

Cenário 5: O desligamento ou falha de uma das máquinas do cluster não deve tornar o sistema inoperante. O balanceador de carga deve enviar as requisições para as instâncias operantes, e em alguns minutos novas instâncias devem substituir as que falharam.

Cenário 6: O desenvolvimento das principais funcionalidades do sistema possui testes unitários.

Abaixo segue tabela de priorização dos cenários. A coluna “IMP” indica a importância do atributo e cenário avaliado, e a coluna “COM.” indica a complexidade de implementação. Os valores dessas colunas podem ser alta (A), média (M) ou baixa (B).

Atributo de Qualidade	Cenário	IMP.	COM.
Segurança	Cenário 1: criptografia de dados sensíveis	A	A
Interoperabilidade e Segurança	Cenário 2: autenticação multifator	A	A
Usabilidade	Cenário 3: utilização do sistema em dispositivos móveis	M	M
Desempenho	Cenário 4: o sistema deve atender as requisições em até 3 segundos	A	M
Disponibilidade	Cenário 5: o sistema deve continuar atendendo as requisições quando houver falha em um nó do cluster	A	A
Testabilidade	Cenário 6: o desenvolvimento das principais funcionalidades do sistema deve possuir testes unitários	M	A

6.3. Avaliação

- **Cenário 1**

Atributo de Qualidade	Segurança
Requisito de Qualidade	O sistema deve armazenar senhas e informação sensível de forma segura
Preocupação	
Armazenamento seguro dados sensíveis	
Cenário	
Cenário 1	
Ambiente	
Produção, carga normal	
Estímulo	
Criação de um novo usuário	

Mecanismo	
API de usuários, banco de dados	
Medida de resposta	
A senha deve ser armazenada na forma de hash, e o e-mail e senha criptografados	
Considerações sobre a arquitetura	
Riscos	O armazenamento de senhas, e-mail e celular de forma aberta é de grande risco em caso de invasões à base de dados e pode violar termos da LGPD (Lei Geral de Proteção de Dados)
Ponto de Sensibilidade	Armazenamento inseguro das chaves criptográficas e transmissão de dados sem utilização de HTTPS
Tradeoff	Maior complexidade de desenvolvimento para leitura e gravação de dados criptografados e mais consumo de CPU

- Evidências do cenário 1:

Figura 3 – Cadastro de um novo usuário

Cadastrar Usuário

Dados cadastrais

Nome *

E-mail *

Senha *

Celular *

Módulos permitidos

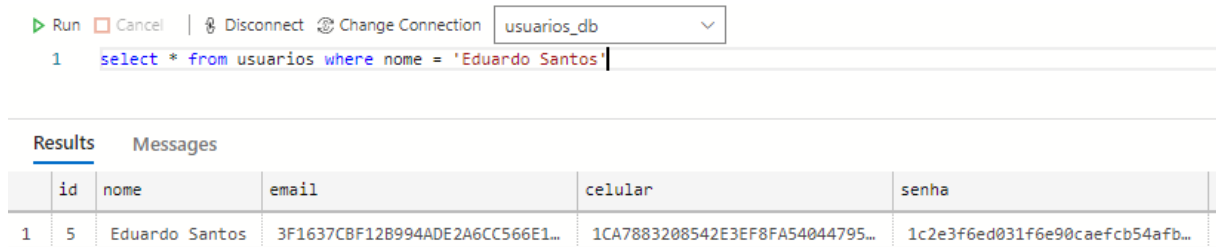
☐ Gerenciamento de Usuários

☒ Normas Técnicas

☒ Assessorias e Consultorias

Cadastrar

Fonte: autor

Figura 4 – Consulta na tabela de usuários evidenciando os dados criptografados


The screenshot shows a database query interface. At the top, there are buttons for 'Run', 'Cancel', 'Disconnect', and 'Change Connection'. A dropdown menu shows 'usuarios_db'. Below this, a SQL query is entered: `1 select * from usuarios where nome = 'Eduardo Santos'`. The results are displayed in a table with columns: id, nome, email, celular, and senha. The first row shows data for 'Eduardo Santos', with the email and celular fields truncated.

	id	nome	email	celular	senha
1	5	Eduardo Santos	3F1637CBF12B994ADE2A6CC566E1...	1CA7883208542E3EF8FA54044795...	1c2e3f6ed031f6e90caefcb54afb...

Fonte: autor

- **Cenário 2**

Atributo de Qualidade	Interoperabilidade e Segurança
Requisito de Qualidade	Comunicação entre os módulos e sistemas legados deve utilizar um broker de comunicação preferencialmente
Preocupação	
Diminuir acoplamento dos sistemas e segurança	
Cenário	
Cenário 2	
Ambiente	
Produção, carga normal	
Estímulo	
Usuário tentando se autenticar no sistema	
Mecanismo	
Frontend, API de Usuários, Broker e Serviço de Notificações	
Medida de resposta	
É enviado um SMS com um código de verificação para o celular do usuário	
Considerações sobre a arquitetura	
Riscos	Indisponibilidade do serviço de SMS
Ponto de Sensibilidade	Usuário pode estar sem acesso ao celular
Tradeoff	Processo de autenticação mais seguro porém mais moroso que autenticação simples e custos com SMS

- **Evidências do cenário 2**

Figura 5 – Tela de inserção do código de verificação para autenticação

IndTexBr - Login

Olá, **Murilo**! Foi enviado um código de verificação para seu celular terminado em **1339**

Verificar

© Murilo E B Santos - 2021

Fonte: autor

Figura 6 – Recebimento do SMS de verificação

Fonte: autor

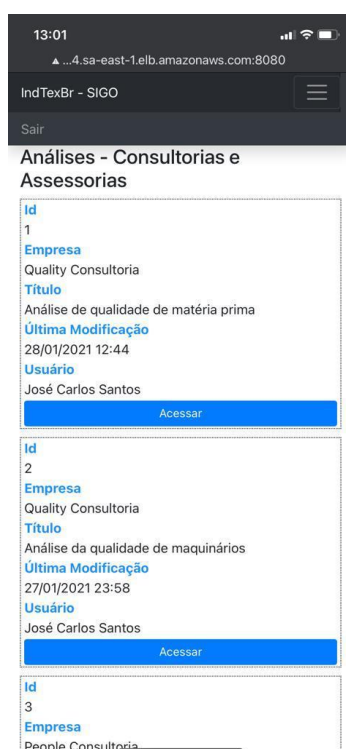
- **Cenário 3**

Atributo de Qualidade	Usabilidade
Requisito de Qualidade	O sistema deve possuir interface responsiva
Preocupação	
Boa usabilidade do sistema em dispositivos móveis	
Cenário	
Cenário 3	
Ambiente	
Produção, carga normal	

Estímulo	
Usuário utilizando o sistema do celular	
Mecanismo	
Frontend	
Medida de resposta	
As páginas devem ser apresentadas de forma adequada para visualização no celular	
Considerações sobre a arquitetura	
Riscos	Lentidão ou incompatibilidades de HTML/CSS
Ponto de Sensibilidade	Diferenças entre tamanho de telas, resoluções, navegador e sistema operacional
Tradeoff	Solução mais barata e mais simples que um aplicativo nativo ou híbrido, porém não é tão performática e 100% adequada para processos de manipulação de arquivos

Evidências do cenário 3

Figura 7 – Listagem de análises



Fonte: autor

Figura 8 – Criação de nova análise

13:04

IndTexBr - SIGO

Sair

Análise Nova

Título *

Resumo *

Anexos

Selecionar Arquivo nenhum arquiv...selecionado

Nome do arquivo	Data de inclusão
Nenhum arquivo adicionado	

Salvar Análise Voltar

Figura 9 – Criação de um novo usuário

13:00

IndTexBr - SIGO

Sair

Cadastrar Usuário

Dados cadastrais

Nome *

E-mail *

Senha *

Celular *

Módulos permitidos

- ☐ Gerenciamento de Usuários
- ☐ Normas Técnicas
- ☐ Assessorias e Consultorias

Cadastrar

Fonte: autor

- **Cenário 4**

Atributo de Qualidade	Desempenho
Requisito de Qualidade	O sistema deve possuir bom desempenho
Preocupação	
O sistema deve apresentar bom desempenho	
Cenário	
Cenário 4	
Ambiente	
Produção, carga normal	
Estímulo	
Consulta de análises, inclusão de usuário ou login	
Mecanismo	
Frontend, serviço de usuários, serviço de consultorias	
Medida de resposta	
O sistema deve responder as requisições em até 3 segundos	
Considerações sobre a arquitetura	
Riscos	Não há
Ponto de Sensibilidade	Latência, velocidade da internet, alta demanda de requisições
Tradeoff	O aumento na quantidade de injeções de dependências, camadas ou na quantidade middlewares reduz o tempo de resposta.

- **Evidências do cenário 4**

Figura 10 – Tempo de resposta das requisições de autenticação em menos de 1 segundo

Url	Method	Stat...	Size	Time
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/usuarios/auth	OPTIONS	204	0 B	43 ms
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/usuarios/auth	POST	200	249 B	285 ms
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/usuarios/auth/verificacao	OPTIONS	204	0 B	137 ms
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/usuarios/auth/verificacao	POST	200	624 B	783 ms
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/usuarios/auth/permissoes?modulo=home	OPTIONS	204	0 B	47 ms
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/usuarios/auth/permissoes?modulo=home	GET	200	148 B	105 ms

Fonte: autor

Figura 11 – Tempo de resposta na criação de um novo usuário em menos de 1 segundo

Url	Method	Status	Size	Time
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/usuarios/usuarios	OPTIONS	204	0 B	97 ms
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/usuarios/usuarios	POST	200	196 B	498 ms

Fonte: autor

Figura 12 – Tempo de resposta na criação de uma nova análise com anexos em menos de 1 segundo

Url	Method	Status	Size	Time
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/consultorias/analises	OPTIONS	204	0 B	107 ms
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/consultorias/analises	POST	201	201 B	152 ms
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/consultorias/analises/4/anexos	OPTIONS	204	0 B	51 ms
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/consultorias/analises/4/anexos	OPTIONS	204	0 B	93 ms
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/consultorias/analises/4/anexos	POST	200	148 B	578 ms

Fonte: autor

Figura 13 – Tempo de resposta na consulta de análises em menos de 1 segundo

Url	Method	Status	Size	Time
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/usuarios/auth/permissoe...	OPTIONS	204	0 B	85 ms
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/usuarios/auth/permissoe...	GET	200	148 B	92 ms
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/consultorias/analises	GET	200	842 B	640 ms
<input type="checkbox"/> http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com/consultorias/analises	OPTIONS	204	0 B	47 ms

Fonte: autor

Tempo de resposta consulta de análises

- **Cenário 5**

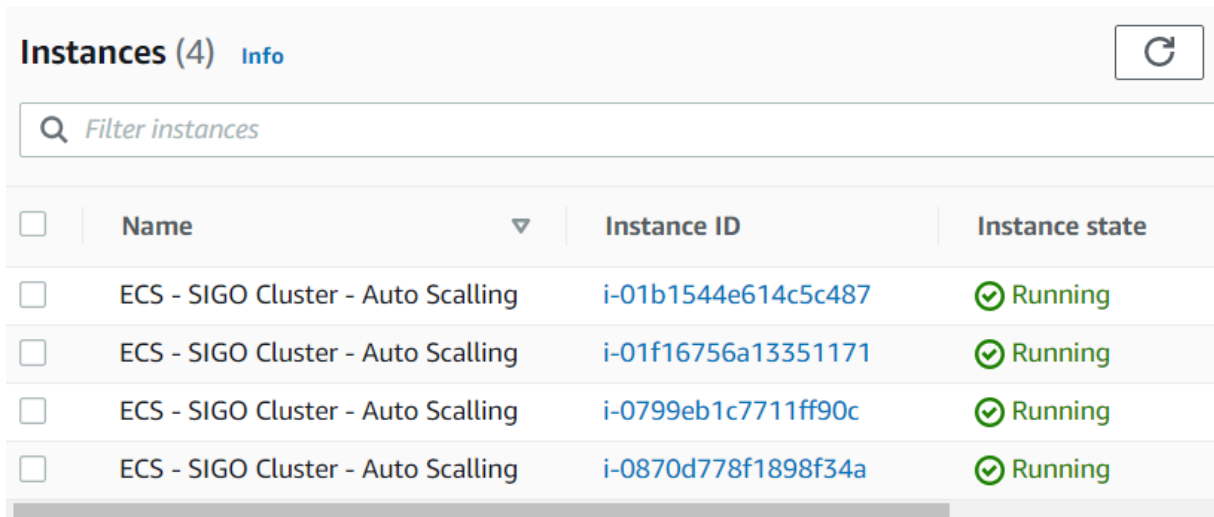
Atributo de Qualidade	Disponibilidade
Requisito de Qualidade	O sistema deve estar disponível 24 horas por dia
Preocupação	
Tolerância a falhas e alta disponibilidade	
Cenário	
Cenário 5	
Ambiente	
Produção, carga normal	
Estímulo	
Desligamento de uma instância do cluster	

Mecanismo	
Cluster de máquinas/containers	
Medida de resposta	
O sistema deve continuar operando e nova instâncias devem ser provisionada automaticamente	
Considerações sobre a arquitetura	
Riscos	Indisponibilidade regional do serviço de nuvem
Ponto de Sensibilidade	Demora ou falha no provisionamento de novos recursos
Tradeoff	Maior disponibilidade resulta em custo com recursos redundantes

- **Evidências do cenário 5**

Abaixo segue imagens para evidenciar o cenário 5, porém no vídeo disponibilizado para a prova de conceito é possível visualizar melhor esse cenário.

Figura 14 – Cluster operando com sua capacidade total de instâncias



The screenshot shows the 'Instances (4)' page in the AWS Management Console. It displays a table with four instances, all in a 'Running' state. The table has columns for Name, Instance ID, and Instance state. Each instance is named 'ECS - SIGO Cluster - Auto Scalling' and has a unique ID. The 'Instance state' column shows a green checkmark and the word 'Running' for each instance.

	Name	Instance ID	Instance state
<input type="checkbox"/>	ECS - SIGO Cluster - Auto Scalling	i-01b1544e614c5c487	✓ Running
<input type="checkbox"/>	ECS - SIGO Cluster - Auto Scalling	i-01f16756a13351171	✓ Running
<input type="checkbox"/>	ECS - SIGO Cluster - Auto Scalling	i-0799eb1c7711ff90c	✓ Running
<input type="checkbox"/>	ECS - SIGO Cluster - Auto Scalling	i-0870d778f1898f34a	✓ Running

Fonte: autor

Figura 15 – Serviços operando com sua capacidade total as 13h50

Create Update Delete Actions ▾

Last updated on January 28, 2021 1:50:10 PM (0m ago)

Filter in this page Launch type ALL Service type ALL


<input type="checkbox"/>	Service Name	Status	Desired tasks ▾	Running tasks ▾	Launch type	Platform vers
<input type="checkbox"/>	consultorias-service	ACTIVE	2	2	EC2	--
<input type="checkbox"/>	frontend-service	ACTIVE	2	2	EC2	--
<input type="checkbox"/>	usuarios-service	ACTIVE	2	2	EC2	--
<input type="checkbox"/>	notificacao-service	ACTIVE	2	2	EC2	--
<input type="checkbox"/>	api-gateway-service	ACTIVE	2	2	EC2	--

Fonte: autor

Figura 16 – Encerramento forçado de uma instância do cluster

Stop instance? ×

Instance IDs

 i-01b1544e614c5c487 (ECS - SIGO Cluster - Auto Scalling)

The following instances are attached to an Auto Scaling group:

- i-01b1544e614c5c487 (EC2ContainerService-sigo-cluster-EcsInstanceAsg-1P4SNUCDNAR8S)

If you stop the instances, Amazon EC2 Auto Scaling might launch replacement instances automatically. If you do not want Amazon EC2 Auto Scaling to launch replacement instances, first detach the instances from the Auto Scaling group.

To confirm that you want to stop the instance, choose the *Stop* button below.

Cancel **Stop**

Fonte: autor

Figura 17 – Serviços afetados pelo desligamento da instância

Create Update Delete Actions ▾

Last updated on January 28, 2021 1:58:14 PM (0n

Filter in this page Launch type ALL Service type ALL

<input type="checkbox"/>	Service Name	Status	Service type...	Task Definiti...	Desired tas...	Running tas...	Launch typ...
<input type="checkbox"/>	consultorias-service	ACTIVE	REPLICA	sigo-consulto...	2	1	EC2
<input type="checkbox"/>	frontend-service	ACTIVE	REPLICA	sigo-frontend:2	2	2	EC2
<input type="checkbox"/>	usuarios-service	ACTIVE	REPLICA	sigo-usuarios:6	2	2	EC2
<input type="checkbox"/>	notificacao-service	ACTIVE	REPLICA	sigo-notificac...	2	2	EC2
<input type="checkbox"/>	api-gateway-service	ACTIVE	REPLICA	sigo-api-gate...	2	1	EC2

Fonte: autor

Figura 18 – Após alguns minutos os serviços afetados estão operando com capacidade total

Create	Update	Delete	Actions ▾	Last updated on January 28, 2021 2:06:42 PM (0m ago)				
Filter in this page		Launch type	ALL ▾	Service type	ALL ▾			
<input type="checkbox"/>	Service Name	Status	Service type...	Task Definiti...	Desired tas...	Running tas...	Launch typ...	Platfo
<input type="checkbox"/>	consultorias-service	ACTIVE	REPLICA	sigo-consulto...	2	2	EC2	--
<input type="checkbox"/>	frontend-service	ACTIVE	REPLICA	sigo-frontend:2	2	2	EC2	--
<input type="checkbox"/>	usuarios-service	ACTIVE	REPLICA	sigo-usuarios:6	2	2	EC2	--
<input type="checkbox"/>	notificacao-service	ACTIVE	REPLICA	sigo-notificac...	2	2	EC2	--
<input type="checkbox"/>	api-gateway-service	ACTIVE	REPLICA	sigo-api-gate...	2	2	EC2	--

Fonte: autor

- **Cenário 6**

Atributo de Qualidade	Testabilidade
Requisito de Qualidade	O desenvolvimento do sistema deve utilizar testes automatizados
Preocupação	
Garantia da testabilidade e manutenibilidade do sistema	
Cenário	
Cenário 6	
Ambiente	
Desenvolvimento	
Estímulo	
Codificação do sistema	
Mecanismo	
Código fonte	
Medida de resposta	
Os principais componentes do sistema devem possuir 75% do código testado	
Considerações sobre a arquitetura	
Riscos	Falta de experiência do time de desenvolvimento
Ponto de Sensibilidade	Classes de acesso a dados e acesso a serviços externos são difíceis de testar
Tradeoff	O desenvolvimento com testes unitários é mais demorado e exige maior qualificação

	técnica, porém garante uma maior qualidade e proporciona maior segurança na evolução do software
--	--------------------------------------------------------------------------------------------------

- **Evidências do cenário 6**

Figura 19 – Relatório de cobertura de testes unitários do serviço de usuários

Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks) ▼
msantos_WKSJUN000209 2021-...	440	28,85%	1085	71,15%
▷ sigo.usuarios.entities.dll	0	0,00%	34	100,00%
▷ sigo.usuarios.security.dll	0	0,00%	76	100,00%
▷ sigo.usuarios.application.dll	2	0,96%	206	99,04%
▷ sigo.usuarios.data.dll	257	28,15%	656	71,85%
▷ sigo.usuarios.api.dll	149	56,87%	113	43,13%
▷ sigo.usuarios.integrations.dll	32	100,00%	0	0,00%

Fonte: autor

Figura 20 – Relatório de cobertura de testes unitários do serviço de consultorias

Hierarchy	Not Covered (Blocks)	Not Covered (% Blocks)	Covered (Blocks)	Covered (% Blocks) ▼
msantos_WKSJUN000209 2021-01-29 12_15_27...	602	37,84%	989	62,16%
▷ sigo.consultorias.application.dll	2	0,75%	264	99,25%
▷ sigo.consultorias.entities.dll	3	6,00%	47	94,00%
▷ sigo.consultorias.data.dll	359	35,65%	648	64,35%
▷ sigo.consultorias.api.dll	202	87,07%	30	12,93%
▷ sigo.consultorias.attachmentmanagement.dll	36	100,00%	0	0,00%

Fonte: autor

6.4. Resultado

A arquitetura proposta atende com êxito os requisitos de qualidade identificados. É uma arquitetura robusta e resiliente, pois utiliza massivamente serviços de computação em nuvem, tais como cluster de instâncias de computação distribuída, containers, balanceadores de carga e autoescalonamento. Porém na implantação do cluster de containers foi utilizado o serviço ECS, que é um serviço proprietário do provedor AWS, o qual eu já tinha um pouco de familiaridade. Essa abordagem traz consigo um pouco de vendor lock-in (dependência com o fornecedor), porém em contrapartida, como é utilizado Docker dentro desse serviço, se necessário a aplicação ainda pode ser migrada para outro ambiente de nuvem com um pouco de planejamento.

Não foi utilizado o API Gateway proprietário do provedor AWS levando em conta essa questão de vendor lock-in e devido minha falta de experiência nessa ferramenta no

momento da elaboração dessa arquitetura. A ferramenta escolhida, Ocelot, atendeu bem a arquitetura proposta.

As integrações entre os novos módulos e sistemas legados são feitas na sua maioria através do RabbitMQ, sendo esse gerenciado pelo serviço AWS MQ, o que não chega trazer um aprisionamento de fornecedor, pois o RabbitMQ pode ser instalado em qualquer provedor de nuvem, porém essa abordagem garante a disponibilidade desse serviço tão importante de uma forma bem simples.

No âmbito de banco de dados foi proposto a abordagem de um banco de dados por serviço, padrão conhecido por *Database per service*. Essa escolha traz independência entre os serviços, podendo escalar os bancos individualmente, porém em contrapartida é necessário a redundância de dados em alguns momentos.

No desenvolvimento, foi utilizado .NET Core em toda a solução, pois essa é uma tecnologia multiplataforma, leve, que vem ganhando espaço no mercado, além de ser do meu domínio. Embora eu tenha feito essa escolha, seria possível a utilização de NodeJS ou até mesmo Java na criação de algum microserviço da solução.

Na parte de segurança é indicado nos requisitos de qualidade e nos diagramas a utilização de protocolos seguros como HTTPS e AMQPS. Porém na elaboração da PoC foi utilizado apenas AMQPS. Na parte de hash há espaço para utilização de “salt” e na parte de criptografia a utilização de chaves dinâmicas.

Na implementação das provas de conceito, foi tomado o cuidado de não expor dados sensíveis no Github. Para isso, toda informação sensível é obtida através de variáveis de ambiente, sendo essas definidas na máquina de desenvolvimento e nas definições de tarefas do ECS. Por último, toda implementação das PoC foi baseada na Clean Architecture do Robert Cecil Martin, para garantir ao máximo a aplicação dos princípios SOLID, tentando manter ao máximo a alta-coesão e baixo-acoplamento dos componentes.

Em geral, a arquitetura proposta se mostra robusta, segura, de fácil manutenção e performática. Ainda há bastante espaço para melhorias nos âmbitos de DevOps, segurança, integrações e usabilidade.

7. Conclusão

Esse projeto abordou a elaboração da arquitetura de software do projeto SIGO, sendo alguns dos principais objetivos a identificação e definição dos requisitos de qualidade e implementação de uma prova de conceito para validação geral da arquitetura. Esses objetivos foram atendidos com sucesso. Uma dificuldade encontrada foi que a grande quantidade de componentes envolvidos na solução tornaria inviável ou inelegível um detalhamento muito grande nos diagramas de componentes e implantação, assim foi detalhado o suficiente para entendimento geral da solução e início do desenvolvimento sem gerar grandes mudanças arquiteturais na evolução do sistema.

REFERÊNCIAS

MICROSERVICE ARCHITECTURE. **Database per service**. Disponível em:
<<https://microservices.io/patterns/data/database-per-service.html>> Acesso em 29 de janeiro de 2021

CLEAN CODER BLOG. **The Clean Architecture**. Disponível em:
<<https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html>> Acesso em 29 de janeiro de 2021

BLOG TREINA WEB. **Criando um API Gateway com ASP.NET Core e Ocelot**. Disponível em <<https://www.treinaweb.com.br/blog/criando-um-api-gateway-com-asp-net-core-e-ocelot/>> Acesso em 29 de janeiro de 2021

RABBITMQ. **RabbitMQ tutorial - Hello World**. Disponível em
<<https://www.rabbitmq.com/tutorials/tutorial-one-dotnet.html>> Acesso em 29 de janeiro de 2021

TWILLIO. **Programmable SMS Quickstart for C# with .NET Framework**. Disponível em
<<https://www.twilio.com/docs/sms/quickstart/csharp-dotnet-framework>> Acesso em 29 de janeiro de 2021

AWS. **Fazer upload de um objeto usando o AWS SDK para .NET**. Disponível em
<https://docs.aws.amazon.com/pt_br/AmazonS3/latest/dev/UploadObjSingleOpNET.html> Acesso em 29 de janeiro de 2021

AWS. **Enviar uma imagem do Docker**. Disponível em
<https://docs.aws.amazon.com/pt_br/AmazonECR/latest/userguide/docker-push-ecr-image.html>. Acesso em 29 de janeiro de 2021

APÊNDICES

URL do repositório do código fonte da PoC: <https://github.com/mbacagini-tcc-puc>

URL do vídeo da apresentação da PoC: <https://youtu.be/Qasfa-5T6Fg>

URL da PoC: <http://sigo-loadbalancer-1170437824.sa-east-1.elb.amazonaws.com:8080/>