

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

Luiz Augusto Kill Bernardo

SISTEMA DE GESTÃO DA QUALIDADE

Belo Horizonte

2020

Luiz Augusto Kill Bernardo

SISTEMA DE GESTÃO DA QUALIDADE

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

2020

*Para meu filho Gael e minha esposa Flávia,
sem os quais nada teria sentido à essa altura.*

AGRADECIMENTOS

Agradeço à minha esposa Flávia pela compreensão e ajuda, aos meus pais por tudo que sempre fizeram e ainda fazem por mim e aos meus colegas de trabalho pelas trocas de idéias e pelos leões abatidos.

RESUMO

Este projeto diz respeito a uma solução de gestão da qualidade de uma empresa do setor automotivo. Mais especificamente, trata-se de uma aplicação para suporte ao *Sistema de Gestão da Qualidade* da empresa, contendo funcionalidades *core* nesse contexto, como cadastro de não conformidades e produtos não conformes.

Por sua natureza distribuída e responsiva ela pode ser acessada de qualquer lugar do mundo através de navegadores e dispositivos móveis. Sua arquitetura baseada em microserviços habilita o escalonamento horizontal de todas os componentes, de modo a lidar de forma flexível com diferentes cargas de trabalho.

A organização deste projeto segue a seguinte lógica: definição dos requisitos funcionais e não funcionais, modelagem e projeto da arquitetura, prova de conceito e avaliação arquitetural. Espera-se ao fim comprovar a viabilidade da arquitetura proposta, apontando suas principais características e pontos de aderência ao escopo levantado.

Palavras-chave: arquitetura de software, projeto de software, requisitos arquiteturais, microserviços, cloud-native, Java, Spring Boot, Angular, Typescript, PWA, JWT, API, JSON, RESTful, MariaDB, Hazelcast, Netflix OSS, Docker, Swarm, Swagger, Maven, Liquibase, Bootstrap, Elasticsearch, Logstash, Kibana, 12-factor, JHipster.

SUMÁRIO

1. Objetivos do trabalho.....	7
2. Descrição geral da solução.....	8
2.1. Apresentação do problema.....	8
2.2. Descrição geral do software (Escopo).....	9
3. Definição conceitual da solução.....	10
3.1. Requisitos Funcionais.....	10
3.2 Requisitos Não-Funcionais.....	15
3.3 Restrições Arquiteturais.....	20
3.4. Mecanismos Arquiteturais.....	21
4. Modelagem e projeto arquitetural.....	23
4.1. Modelo de componentes.....	23
4.2. Modelo de implantação.....	25
4.3. Modelo de dados.....	29
5. Prova de Conceito (POC) / protótipo arquitetural.....	30
5.1. Implementação e Implantação.....	30
5.2 Código.....	39
5.3 Interfaces/ APIs.....	42
6. Avaliação da Arquitetura.....	42
6.1. Análise das abordagens arquiteturais.....	42
6.2. Cenários.....	43
6.3. Avaliação.....	44
6.4. Resultado.....	61
7. Conclusão.....	62
APÊNDICE I - Links.....	63
APÊNDICE II – Diagramas de casos de uso.....	64
CHECKLIST PARA VALIDAÇÃO DOS ITENS E ARTEFATOS DO TRABALHO...	71

1. Objetivos do trabalho

O objetivo geral deste projeto é apresentar um protótipo arquitetural de uma aplicação de gestão da qualidade de uma empresa do setor automotivo. A solução aqui detalhada permite que colaboradores e gestores da qualidade geograficamente dispersos estejam inseridos na mesma ferramenta de trabalho diário, de modo que erros sejam identificados e tratados antes da entrega para o cliente. Em última análise ela possibilita à empresa a incorporação da cultura de garantia da qualidade a todos os seus processos de forma integrada.

Os objetivos específicos do projeto são:

1. Construir um módulo de incidentes e problemas, no qual colaboradores e gestores da qualidade interagem para registrar não conformidades, produtos não conformes, ações corretivas e preventivas. Este módulo permite também planejar e registrar os controles de qualidade que a empresa julga mais pertinentes (auditorias);
2. Construir um módulo de controle de processos, no qual padrões de qualidade pré-estabelecidos são cadastrados e verificados com periodicidade variável (checklists). Este módulo permite também o registro de eventos operacionais relevantes (paradas da produção);
3. Construir um módulo de divulgação e transparência, no qual gestores da qualidade podem registrar campanhas de recall, boletivos informativos e feeds de notícias;
4. *Descrever* um módulo de BI, responsável por agregar e pré processar as informações geradas nos demais módulos da aplicação para posterior geração de relatórios *táticos e estratégicos*. Este módulo permite a confecção de relatórios pelos usuários da aplicação, com base nos *datamarts* previamente construídos;
5. Construir um mecanismo de integração com um catálogo externo de normas do setor automotivo. A consulta ao catálogo é acessível por qualquer usuário logado na aplicação;

6. Construir um mecanismo para integração com empresas de consultoria. Este módulo permite que o gestor da qualidade solicite análises externas de não conformidades e planos de ação;

7. *Descrever* um módulo de relatórios operacionais, no qual colaboradores têm feedback imediato sobre o estado da operação;

8. Disponibilizar ferramentas de monitoramento da aplicação para que a equipe de operação tenha acesso a todo momento a informações sobre erros e métricas de performance.

2. Descrição geral da solução

2.1. Apresentação do problema

Empresas do setor automotivo sofrem grande pressão da sociedade e dos governos para que seus produtos apresentem elevados níveis de confiança (qualidade). Afinal, veículos e componentes defeituosos implicam acidentes, desgaste da imagem, ações judiciais e etc. Essa é a razão pela qual o setor é tão fortemente regulado e normatizado. Muitas filosofias de trabalho e normas de qualidade são aplicadas ao setor como forma de garantir que a cultura da qualidade esteja presente em todos os momentos da cadeia produtiva.

Nesse contexto, muitas empresas implementam *Sistemas de Gestão da Qualidade – SGQ*. Por sistema não deve-se entender software, mas o conjunto de normas, procedimentos, processos e valores que a empresa mantém para que a qualidade seja premissa desde a aquisição de insumos até a entrega do produto final ao cliente. A norma internacional que regulamenta SGQs é a ISO 9001, traduzida para o português pela ABNT. Esta norma traz uma extensa série de requisitos que a empresa deve atender antes de obter o selo de certificação.

Ainda que, em teoria, um SGQ possa ser mantido sem o apoio de ferramentas informatizadas, na prática isso é inviável. Empresas de grande porte dependem de softwares especializados para fazer a gestão da qualidade. Tais ferramentas devem ser capazes de atender a usuários dispersos por muitas localidades físicas, além de suportar cargas variáveis

de trabalho. Isso decorre da sazonalidade de alguns mercados, de modo que a escalabilidade dessas aplicações não deve ser um aspecto limitante.

2.2. Descrição geral do software (Escopo)

A aplicação desenhada possui arquitetura baseada em microserviços e tem como público-alvo os colaboradores e gestores da qualidade. Resumidamente, trata-se de um frontend web que se comunica com um microserviço responsável pelos dados e pelas regras de negócio do SGQ. Entretanto, outras peças de arquitetura estão presentes para sanar dificuldades impostas pela arquitetura de microserviços, como descoberta de serviços, balanceamento de carga, agregação de logs, monitoramento, orquestração e etc.

A arquitetura da aplicação também prevê a integração com serviços externos. A integração com o serviço de normas é feita pelo consumo da API fornecida pelo SaaS em questão, de modo que a conexão é feita diretamente do frontend sem a participação do backend (gateway). Por outro lado, a integração com o serviço de consultoria é feita partindo da premissa que as empresas contratadas devem implementar o contrato de API fornecido pela dona do SGQ. Junte-se a isso o fato de que a análise da consultoria é assíncrona, ou seja, é realizada em momento posterior à solicitação. Dessa forma, a integração com as empresas de consultoria é feita de modo agendado no backend, com o envio de solicitações registradas pelo usuário e o recebimento das análises feitas pelos auditores externos.

deve ser possível informar qual é a ação planejada para saná-lo, tarefa executada pelo gestor da qualidade. Deve ser possível concluir o PNC após executada a ação de correção.

- **Controle de ações do SGQ**

A aplicação deve permitir que o gestor da qualidade registre ações, sendo elas correções, ações corretivas ou preventivas. Deve ser possível informar o usuário responsável pela execução, a data prevista de conclusão e o resultado pós implementação.

Módulo de controle de processos

- **Controle de auditorias**

A aplicação deve permitir o registro de auditorias pelo gestor da qualidade. Deve ser possível informar a data de início da auditoria, o auditor responsável, o tipo (interna / externa) e os itens de auditoria verificados.

- **Controle de eventos operacionais**

A aplicação deve permitir o registro de eventos operacionais por qualquer usuário logado. Deve ser possível informar a data e a duração do evento, seu tipo, sua descrição detalhada e se houve parada de produção.

- **Controle de resultados de checklists**

A aplicação deve permitir o registro dos resultados encontrados quando da verificação dos checklists de qualidade. Deve ser possível informar a data da verificação e os resultados individuais de cada item do checklist em questão.

Módulo de divulgação e transparência

- **Controle de boletins informativos**

A aplicação deve permitir ao gestor da qualidade o registro e a manutenção de boletins informativos da qualidade. Deve ser possível informar a descrição do boletim, a data de publicação, seu público-alvo e suas categorias.

- **Controle de campanhas de recall**

A aplicação deve permitir ao gestor da qualidade o registro e a manutenção das campanhas de recall organizadas pela empresa. Deve ser possível informar a descrição da campanha, as datas de início e fim e o produto em questão.

- **Controle de feeds de notícias**

A aplicação deve permitir ao gestor da qualidade o cadastro de feeds de notícias, para consumo via web pelos diversos públicos-alvo. Deve ser possível informar o tipo do feed (Atom / RSS), seu título, descrição e URI. Um endpoint específico da aplicação deve expor as informações do feed para que qualquer pessoa da Internet possa salvá-lo e acompanhar suas publicações.

- **Controle de publicações de feed**

A aplicação deve permitir ao gestor da qualidade o registro de publicações de feed de notícias. Deve ser possível informar o título da publicação, seu autor, URI, conteúdo e categorias. Um endpoint específico da aplicação deve disponibilizar todas as publicações de um determinado feed.

Módulo de compliance

- **Consulta de normas**

A aplicação deve permitir que qualquer usuário logado consulte um catálogo de normas externo, fornecido por uma empresa especializada na modalidade SaaS. Deve ser possível consultar os detalhes de cada norma, bem como realizar o download do arquivo da norma, caso disponível.

- **Solicitações de análise**

A aplicação deve permitir que o gestor da qualidade dispare solicitações de análise de não conformidades e planos de ação por empresas de consultoria. Essa integração visa fornecer ao gestor da qualidade informações sobre a eficácia do plano de ação elaborado. Uma vez que a análise é feita por consultores especialistas alocados nas empresas contratadas, as solicitações devem ter caráter assíncrono. A aplicação deve verificar de tempos em tempos quais análises estão finalizadas no sistema externo e sincronizá-las para consulta pelo gestor da qualidade.

Módulo de relatórios

- Relatório de análises de consultoria
- Relatório de auditoria
- Relatório de boletim informativo
- Relatório de controle de processo (checklist)
- Relatório de eventos operacionais
- Relatório de não conformidades
- Relatório de produtos não conformes

Módulo de configuração

- **Controle de usuários**

A aplicação deve permitir que o gestor da qualidade administre os usuários da aplicação, bem como os perfis de acesso de cada um.

- **Controle de empresas**

A aplicação deve permitir que o gestor da qualidade administre as empresas habilitadas para operação, bem como seus setores, processos e produtos.

- **Controle de planos de auditoria**

A aplicação deve permitir que o gestor da qualidade gerencie os planos anuais de auditoria. Deve ser possível registrar itens de auditoria (requisitos previstos para verificação) e distribuí-los no tempo para compor cada plano.

- **Controle de checklists de processos**

A aplicação deve permitir que o gestor da qualidade gerencie os checklists de processos operacionais. Deve ser possível informar o título, a periodicidade e os itens ordenados que compõem cada checklist.

- **Controle de empresas de consultorias**

A aplicação deve permitir que o gestor da qualidade gerencie as empresas de consultoria habilitadas para integração. Deve ser possível informar o nome, a URL e o token de acesso para a integração.

- **Controle de públicos-alvo**

A aplicação deve permitir que o gestor da qualidade gerencie os públicos-alvo para uso nas publicações (boletins, feeds, campanhas de recall).

- **Controle de categorias de publicação**

A aplicação deve permitir que o gestor da qualidade gerencie as categorias de publicações (boletins, feeds, campanhas de recall).

Módulo de conta

- **Meus dados**

A aplicação deve permitir que o usuário edite seus dados próprios: nome, sobrenome, e-mail e idioma.

- **Alterar senha**

A aplicação deve permitir que o usuário altere sua senha de acesso.

A tabela a seguir apresenta as prioridades de cada requisito funcional, tendo por critério a importância para o negócio da empresa. (Essencial / Importante / Desejável)

Módulo	RF	Prioridade
Incidentes e problemas	Controle de não conformidades (NC)	Essencial
	Controle de produtos não conformes (PNC)	Essencial
	Controle de ações do SGQ	Essencial
Controle de processos	Controle de auditorias	Essencial
	Controle de eventos operacionais	Essencial
	Controle de resultados de checklists	Essencial
Divulgação e transparência	Controle de boletins informativos	Importante
	Controle de campanhas de recall	Essencial

	Controle de feeds de notícias	Desejável
	Controle de publicações de feed	Desejável
Compliance	Consulta de normas	Importante
	Solicitações de análise	Importante
Relatórios	Relatório de análises de consultoria	Importante
	Relatório de auditoria	Importante
	Relatório de boletim informativo	Importante
	Relatório de controle de processo (checklist)	Importante
	Relatório de eventos operacionais	Importante
	Relatório de não conformidades	Importante
	Relatório de produtos não conformes	Importante
Configuração	Controle de usuários	Essencial
	Controle de empresas	Desejável
	Controle de planos de auditoria	Importante
	Controle de checklists de processos	Essencial
	Controle de empresas de consultorias	Importante
	Controle de públicos-alvo	Desejável
	Controle de categorias de publicação	Desejável
Conta	Meus dados	Desejável
	Alterar senha	Desejável

3.2 Requisitos Não-Funcionais

A seguir encontram-se descritos os requisitos não funcionais da aplicação.

- **Acessibilidade** – A aplicação deve suportar ambientes Web responsivos e ambiente *mobile*

Estímulo	Consulta de não conformidades.
Fonte do estímulo	Usuário acessando o sistema a partir de um smartphone.
Ambiente	Funcionamento com carga normal.
Artefato	Módulo de incidentes e problemas.
Resposta	O frontend da aplicação ajustou corretamente as propriedades CSS para que o melhor layout fosse selecionado para o dispositivo em questão.
Medida da resposta	Identidade visual da aplicação preservada em todos os modos de exibição, com objetos redimensionados e reposicionados para a melhor experiência do usuário.

- **Usabilidade** – A aplicação deve ser de fácil utilização

Estímulo	Registro de não conformidade.
Fonte do estímulo	Usuário acessando a funcionalidade de registro de não conformidade.
Ambiente	Funcionamento com carga normal.
Artefato	Módulo de incidentes e problemas.
Resposta	O frontend permite uma utilização flúida e intuitiva.
Medida da resposta	Usuário foi capaz de realizar o registro de uma não conformidade e seu plano de ação em até 3 minutos.

- **Desempenho** – A aplicação deve apresentar performance elevada

Estímulo	Acesso à funcionalidade de eventos operacionais.
Fonte do estímulo	Usuário acessando a funcionalidade de consulta de eventos operacionais.
Ambiente	Funcionamento com carga normal.
Artefato	Módulo de controle de processos.
Resposta	O frontend obteve e exibiu a lista de eventos operacionais cadastrados.
Medida da resposta	A primeira página da lista de eventos operacionais foi exibida em até de 3 segundos.

● **Testabilidade – A aplicação deve ser fácil de testar**

Estímulo	Execução de testes da aplicação.
Fonte do estímulo	Analista desenvolvedor realizando testes na aplicação antes da disponibilização de uma release.
Ambiente	Ambiente de desenvolvimento.
Artefato	Aplicação
Resposta	O processo de desenvolvimento realizou testes automatizados em todas as funcionalidades da aplicação.
Medida da resposta	O processo de desenvolvimento reportou quantos testes foram executados, quais executaram com sucesso, quais falharam e quais foram os erros.

● **Disponibilidade – A aplicação deve ser resiliente a falhas**

Estímulo	Acesso a um serviço externo offline
Fonte do estímulo	Usuário acessando um serviço externo que

	está inacessível
Ambiente	Acesso a algum serviço externo comprometido
Artefato	Módulo de compliance
Resposta	O frontend deve ser resiliente à falha do serviço e retornar ao funcionamento normal de forma transparente tão logo o acesso seja reestabelecido.
Medida da resposta	O frontend exibe uma mensagem de erro para cada chamada à API que não pode ser efetuada, sem contudo impedir a navegação do usuário para outras telas.

- **Interoperabilidade – A aplicação deve se comunicar com serviços através do consumo de APIs**

Estímulo	Consulta ao catálogo de normas
Fonte do estímulo	Usuário acessando o catálogo de normas do setor automotivo.
Ambiente	Funcionamento com carga normal
Artefato	Módulo de compliance
Resposta	A aplicação deve acessar o catálogo de normas através do consumo da API do serviço externo (SaaS).
Medida da resposta	O frontend exibe a listagem de normas do setor automotivo, bem como links para downloads dos arquivos das normas.

- **Segurança – A aplicação deve proteger dados e funcionalidades com um mecanismo de segurança confiável**

Estímulo	Acesso a uma página privada sem login
-----------------	---------------------------------------

	prévio
Fonte do estímulo	Usuário acessando uma funcionalidade protegida sem a especificação do token de segurança.
Ambiente	Funcionamento com carga normal.
Artefato	Todos os módulos da aplicação, excetuada a funcionalidade de consulta de planos de auditoria do módulo de controle de processos.
Resposta	O acesso deve ser bloqueado.
Medida da resposta	O frontend impede o acesso à funcionalidade e exibe o formulário de login, após o qual o acesso é fornecido automaticamente sem a necessidade de um novo estímulo (clique).

● **Disponibilidade – A aplicação deve estar disponível 24/7**

Estímulo	Manutenção em um dos nós do cluster Swarm
Fonte do estímulo	Analista DevOps realizando manutenção na infraestrutura de containerização onde a aplicação é executada.
Ambiente	Funcionamento com carga normal, possivelmente com muitos usuários acessando o sistema simultaneamente.
Artefato	Aplicação
Resposta	O funcionamento da aplicação e o acesso dos usuários não deve interrompido por conta da manutenção, uma vez que o cluster redireciona as requisições para outro nó worker.
Medida da resposta	A aplicação continua operacional para todos os usuários, sem impacto nos tempos de resposta.

- **Escalabilidade** – A aplicação deve possibilitar o escalonamento horizontal dos componentes de arquitetura

Estímulo	Aumento de demanda
Fonte do estímulo	Muitos usuários acessando a aplicação simultaneamente
Ambiente	Funcionamento com carga acima do normal, com muitos usuários realizando operações de alto custo operacional
Artefato	Aplicação
Resposta	Os tempos de resposta não devem ser impactados pelo aumento da demanda
Medida da resposta	A aplicação permite o escalonamento horizontal dos componentes de arquitetura para tratar o aumento de demanda

3.3 Restrições Arquiteturais

- A aplicação deve ser desenvolvida usando os frameworks Angular (frontend) e Spring Boot (backend);
- A arquitetura da aplicação deve ser baseada em microserviços;
- A aplicação deve ser totalmente *stateless*;
- O frontend da aplicação deve ser responsivo;
- As integrações com serviços externos devem ser feitas de acordo com o padrão RESTful, trafegando dados serializados no formato JSON;
- A autenticação da aplicação deve se dar por meio de tokens JWT;
- O frontend deve ter o comportamento PWA (Progressive Web Application);
- Os bancos de dados dos componentes de backend devem ser MariaDB;

- O gateway da aplicação deve utilizar as ferramentas Netflix Zuul (proxy), Netflix Hystrix (circuit breaker) e Netflix Ribbon (load balancing);
- O registro de serviços deve utilizar as ferramentas Netflix Eureka (service discovery) e Spring Cloud Config (configuração distribuída);
- A stack de monitoramento deve ser ELK (Elasticsearch, Logstash, Kibana);
- O ambiente de execução do backend deve ser o Docker ou Docker Swarm;
- A ferramenta de carga de dados deve ser o Microsoft SQL Integration Services;
- A ferramenta de modelagem de data marts e relatórios deve ser o IBM Cognos;
- A ferramenta de confecção e execução de relatórios operacionais deve ser o Microsoft SQL Server Reporting Services;

3.4. Mecanismos Arquiteturais

Mecanismo de análise	Mecanismo de design	Mecanismo de implementação
Comunicação entre processos	Gateway, microserviço SGQ, registro de serviços, stack de monitoramento, cache	Chamadas HTTP
Integrações com sistemas externos	Interfaces padronizadas (OpenAPI)	APIs Restful JSON
Log	Agregação de log distribuído	Logstash
Monitoramento	Bashboards	Elasticsearch + Kibana
Configuração	Configuração distribuída	Spring Cloud Config
Cache	Cache distribuído	Hazelcast
Build	Automação do processo de empacotamento da aplicação	Maven, npm, Google jib
Deploy	Orquestração da stack da solução	Docker compose

Front end	Interface usuário x backend	Angular Typescript
Versionamento	Controle das alterações na aplicação	Git
Autenticação e autorização	Garantia da segurança de dados e funcionalidades	Tokens JWT gerados no gateway e verificados nos microserviços (<i>secret</i> compartilhado via registro)
Alta disponibilidade	Balanceamento de carga	Netflix Ribbon (gateway)
Alta disponibilidade	Orquestração de containers	Docker Swarm
Descoberta de serviços	Registro das instâncias de microserviços disponíveis	Netflix Eureka (o registro é feito no boot de cada instância)
Documentação	Documentação de APIs	Swagger (OpenAPI)
Controle de esquemas de banco	Migrations	Liquibase
Sistema operacional	Sistema base para as imagens das aplicações	Debian

4. Modelagem e projeto arquitetural

4.1. Modelo de componentes

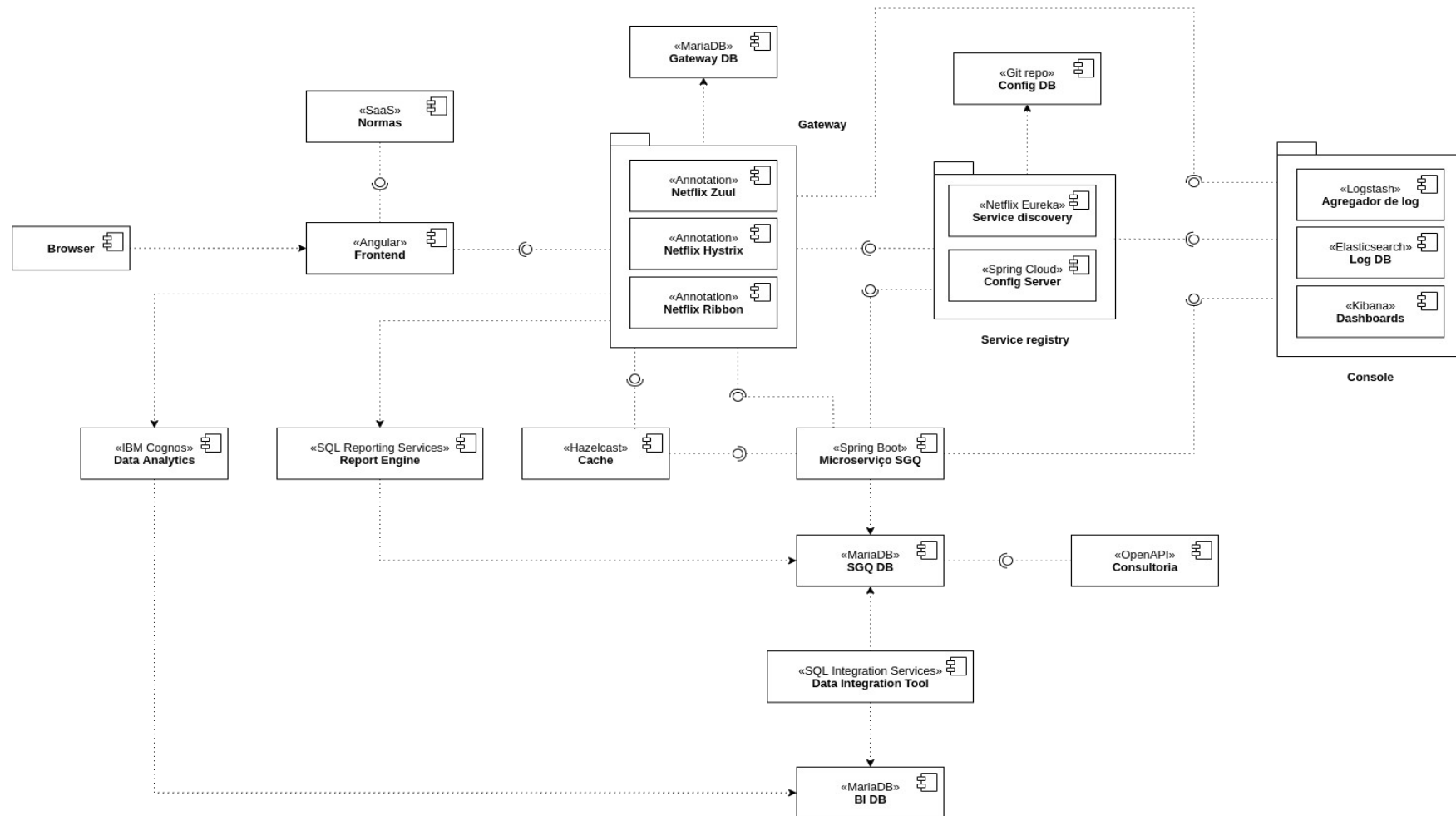


Figura 2: Modelo de componentes

A seguir são descritos os componentes principais da arquitetura.

Frontend: aplicação Angular com comportamento PWA, implementado por um service worker. Esta aplicação é servida pelo gateway da solução no primeiro acesso feito pelo usuário, a partir do qual apenas dados são trafegados via chamadas de API (json). O frontend também executa chamadas ao SaaS de normas sem a participação do backend (gateway). Este componente deve ser desenvolvido.

Gateway: aplicação Spring Boot/Angular que serve como ponto de entrada para todas as chamadas de API feitas pelo frontend. Esse componente se comunica ativamente com o service registry com o intuito de obter as urls das instâncias de microserviços disponíveis para roteamento. Internamente é utilizado o Netflix Zuul como roteador e o Netflix Hystrix como circuit breaker. O gateway também fornece o serviço de load balancing através do Ribbon (Netflix OSS), proteção contra ataques de DoS (rate limiting) e os serviços de autenticação stateless (JWT) e controle de acesso nas rotas (ACL). Este componente deve ser desenvolvido.

Service registry: aplicação Spring Boot/Angular responsável por manter o registro de todas as instâncias ativas de microserviços. Na arquitetura proposta toda instância de microserviço deve registrar-se no registry assim que inicia. Um mecanismo de heartbeat é executado pelo registry a fim de identificar instâncias inativas. O componente utilizado internamente para tanto é o Netflix Eureka. O service registry também tem a função de manter as configurações das aplicações em um repositório central, facilitando assim a distribuição de configurações entre as instâncias e componentes, bem como a alteração de valores *on the fly* (Spring Cloud Config). Várias outras funcionalidades secundárias são disponibilizadas pelo registry, como a listagem de rotas disponíveis nos microserviços e a configuração dos níveis de log. Este componente deve ser desenvolvido.

Console: aplicação que serve como portal de monitoramento da solução. Nela é possível explorar os logs de todos os componentes da arquitetura de forma centralizada, bem como visualizar métricas e dados de performance em dashboards. Essa funcionalidade é obtida através do consagrado stack ELK (Elasticsearch, Logstash, Kibana). Este componente deve ser desenvolvido.

Microserviço SGQ: aplicação Spring Boot que detém os dados e as regras de negócio do SGQ. Uma vez que os microserviços fazem o registro ativo no service registry, podem haver múltiplas instâncias desta aplicação de acordo com as necessidades de escalonamento da empresa (escalabilidade horizontal). O microserviço de SGQ é responsável por chamar os endpoints de integração das empresas de consultoria a intervalos regulares (cron), a fim de obter as análises externas de não conformidades e planos de ação. Uma função secundária do microserviço é a de validar o token JWT enviado na requisição, possivelmente gerado pelo gateway no login do usuário. Este componente deve ser desenvolvido.

Cache: componente distribuído cuja função é armazenar temporariamente os dados mais requisitados nos microserviços e no gateway. A ferramenta escolhida é o cache Hazelcast. Este componente deve ser configurado a partir de alguma opção de ferramenta open source disponível.

Data Integration Tool: componente responsável pela execução agendada de mapas de carga de dados (ETL), necessário para que cópias dos dados de produção estejam disponíveis para posterior processamento pela ferramenta OLAP. Este componente deve ser adquirido.

Report Engine: componente responsável pela execução de relatórios operacionais a partir de dados disponíveis no banco de produção. Este componente deve ser adquirido.

Data Analytics: componente onde são modelados os data marts relacionados ao processo de negócio da empresa. Essa ferramenta também permite a construção e a execução de relatórios pelos usuários da aplicação. Este componente deve ser adquirido.

4.2. Modelo de implantação

O advento de tecnologias como o Docker simplificou enormemente o trabalho de implantação e operação de aplicações. Dessa forma, o modelo de implantação apresentado é bastante simplificado pela presença de um ambiente de containerização. Pode-se adicionalmente implantar a stack da solução em um ambiente de alta disponibilidade sem maiores mudanças nos artefatos existentes (Docker Swarm). Em ambos os casos, a execução pode ser feita em ambiente de nuvem, *on-premises* ou em nuvens híbridas, no que pese mais o critério financeiro da empresa dona do SGQ. Do ponto de vista técnico, cada peça de

arquitetura pode estar alocada em um nó de infraestrutura diferente, responsabilidade delegada ao gerenciador do cluster de containerização.

Obs: os serviços externos (normas / consultoria) foram suprimidos do diagrama.

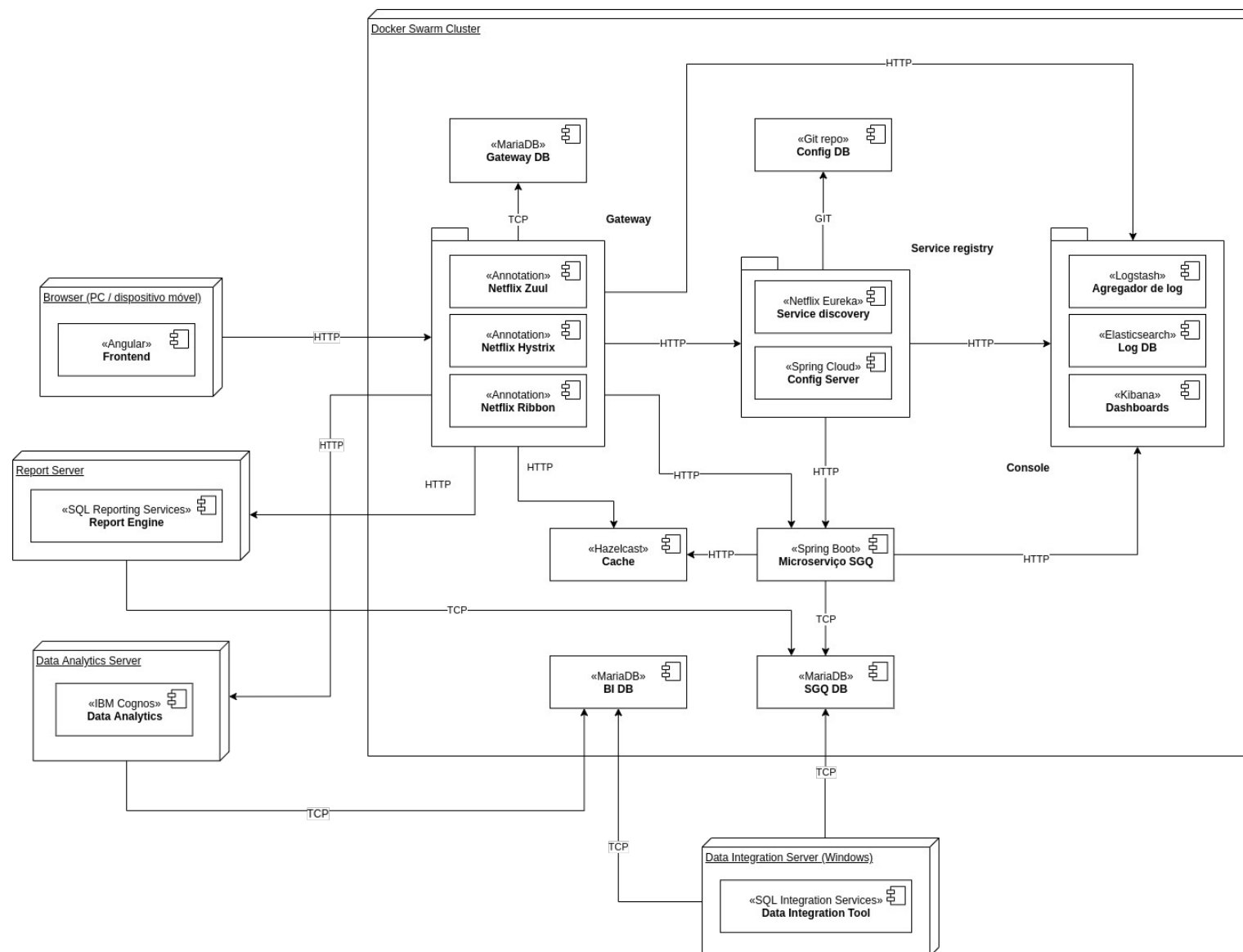


Figura 3: Modelo de implantação

A seguir são descritos os elementos contidos no modelo de implantação, bem como recomendações mínimas de hardware.

Docker Swarm Cluster: elemento responsável pela execução dos principais componentes da arquitetura em alta disponibilidade. Em outras palavras, esse elemento garante que sempre há instâncias dos componentes disponíveis para tratar requisições dos usuários, mesmo quando manutenções são feitas na infraestrutura. Dada as características operacionais do Docker Swarm, espera-se no cluster uma quantidade ímpar de nós, pois esse é um requisito do algoritmo de eleição de líder (master node). Dessa forma, o cluster deve conter no mínimo três nós. Diferentemente de outras tecnologias como Kubernetes, em Docker Swarm o master node também age como worker. Recomendação mínima de hardware por nó: 8GB de memória DDR4-2666, 4 cores Intel Xeon. Sistema operacional: Debian 10 (Buster).

Data Integration Server: elemento responsável pela execução da ferramenta de ETL da arquitetura (Microsoft SQL Integration Services). Recomendação mínima de hardware: 8GB de memória DDR4-2666, 4 cores Intel Xeon. Sistema operacional: Windows Server 2016.

Report Server: elemento responsável pela execução e renderização de relatórios operacionais (Microsoft SQL Reporting Services). Recomendação mínima de hardware: 16GB de memória DDR4-2666, 8 cores Intel Xeon. Sistema operacional: Windows Server 2016.

Data Analytics Server: elemento responsável pela execução da ferramenta OLAP da arquitetura (IBM Cognos). Recomendação mínima de hardware: 32GB de memória DDR4-2666, 16 cores Intel Xeon. Sistema operacional: Windows Server 2016.

4.3. Modelo de dados

Obs: as tabelas de relacionamento com a tabela *Anexo* foram suprimidas com o intuito de simplificar o diagrama.

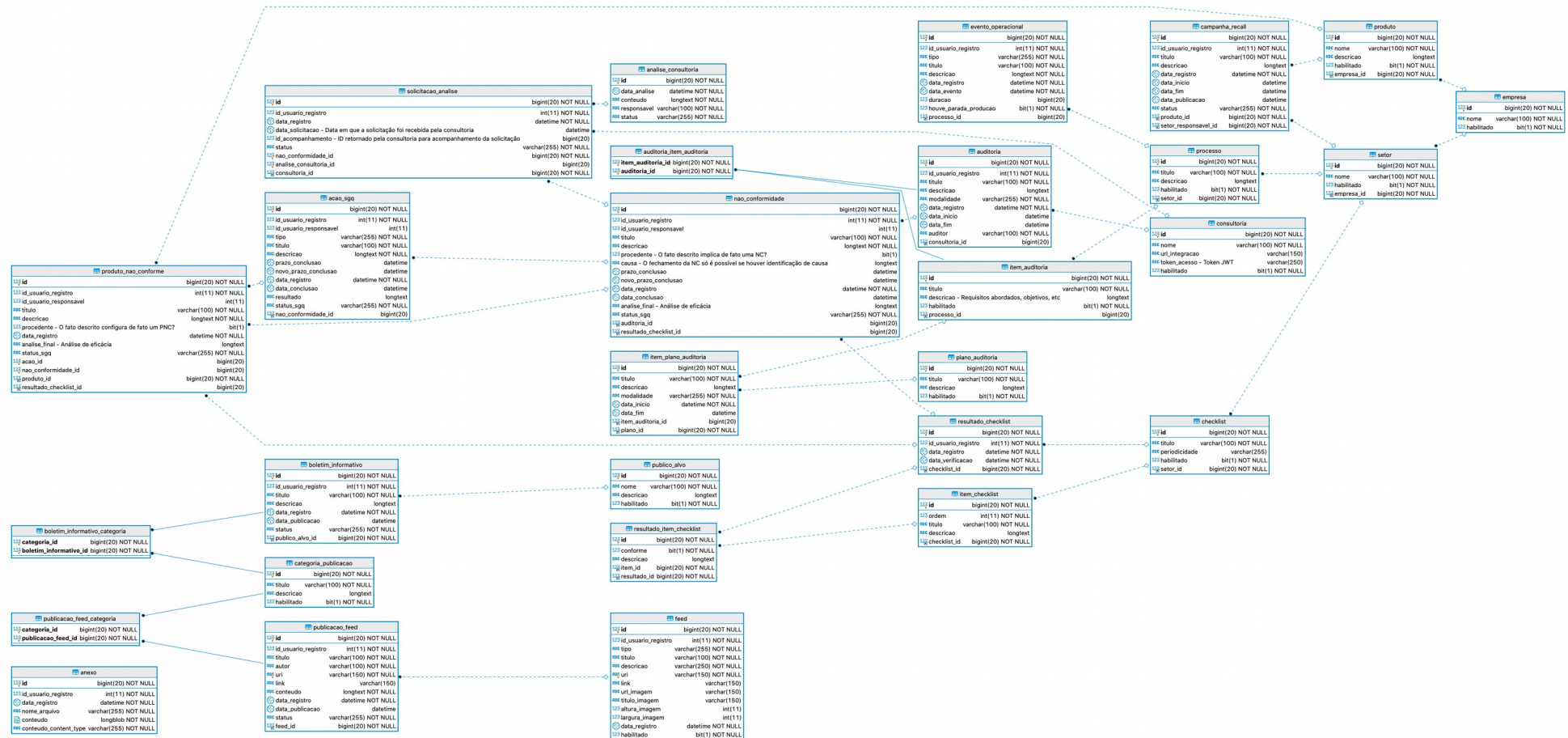


Figura 4: Diagrama ER da aplicação (microserviço SGQ)

5. Prova de Conceito (POC) / protótipo arquitetural

5.1. Implementação e Implantação

Conforme dito anteriormente, o protótipo de aplicação construído para validar a arquitetura proposta consiste basicamente em um frontend web que se comunica com um microserviço através de chamadas RESTful. Entretanto, várias outras peças de arquitetura foram incorporadas ao protótipo para resolver questões inerentes à arquitetura de microserviços. As tecnologias utilizadas foram as seguintes:

- **Frontend:** Angular
- **Microserviço SGQ:** Spring Boot
- **Banco de dados do SGQ:** MariaDB
- **Gateway de aplicação:** Spring Boot + Netflix OSS (Zuul, Hystrix, Ribbon)
- **Banco de dados do gateway:** MariaDB
- **Registro de serviços:** Spring Boot + Netflix Eureka + Spring Cloud Config
- **Banco de dados do registro:** repositório Git
- **Cache:** Hazelcast
- **Ferramenta ETL:** Microsoft SLQ Integration Services
- **Ferramenta OLAP:** IBM Cognos
- **Engine de relatórios operacionais:** Microsoft SQL Reporting Services
- **Banco de dados do BI:** MariaDB
- **Stack de monitoramento:** ELK (Elasticsearch, Logstash, Kibana)
- **Mecanismo de autenticação / autorização:** JWT

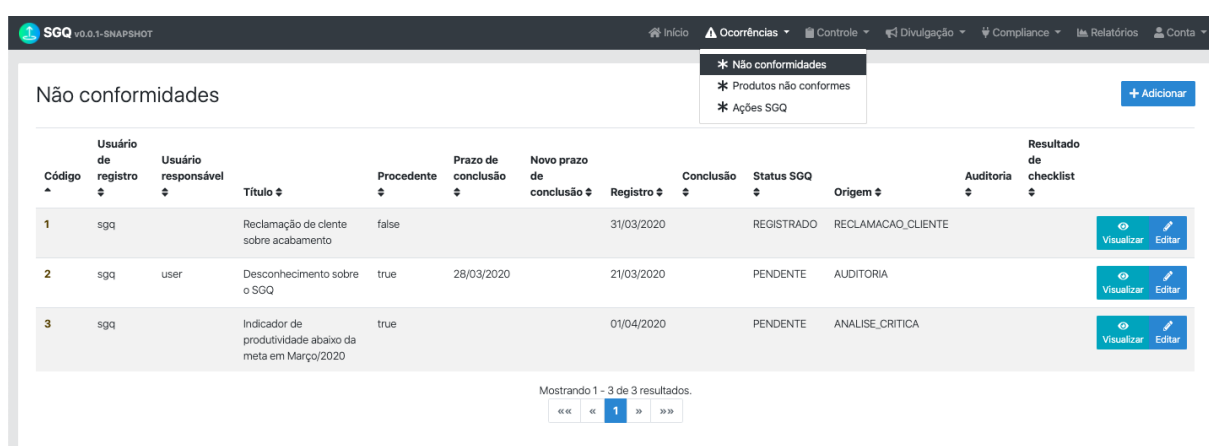
A seção 4 deste documento fornece um detalhamento maior em relação a cada componente da arquitetura.

Com base nos requisitos funcionais levantados e utilizando as tecnologias listadas anteriormente, o protótipo de aplicação construído tem por objetivo comprovar a viabilidade da arquitetura proposta para suportar os requisitos de negócio da empresa associados ao SGQ.

Isso é feito pela análise de requisitos não funcionais durante a execução dos casos de uso da aplicação. Os casos de uso analisados são os seguintes (vide diagramas no APÊNDICE II):

● Módulo de incidentes e problemas – Consulta de não conformidades

Esse requisito permite ao usuário consultar em uma lista paginada as Ncs registradas na aplicação. Deve ser possível visualizar os detalhes de cada NC, bem como editar ou excluir uma NC específica (apenas gestor do SGQ).



Não conformidades

- * Não conformidades
- * Produtos não conformes
- * Ações SGQ

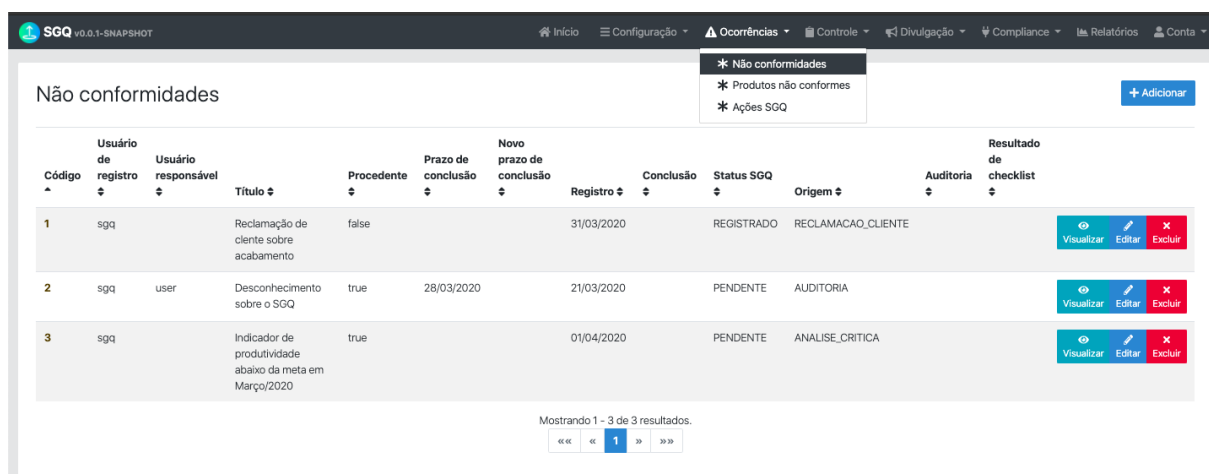
+ Adicionar

Código	Usuário de registro	Usuário responsável	Título	Procedente	Prazo de conclusão	Novo prazo de conclusão	Registro	Conclusão	Status SGQ	Origem	Auditoria	Resultado de checklist
1	sgq		Reclamação de cliente sobre acabamento	false			31/03/2020		REGISTRADO	RECLAMACAO_CLIENTE		Visualizar Editar
2	sgq	user	Desconhecimento sobre o SGQ	true	28/03/2020	21/03/2020			PENDENTE	AUDITORIA		Visualizar Editar
3	sgq		Indicador de produtividade abaixo da meta em Março/2020	true			01/04/2020		PENDENTE	ANALISE_CRITICA		Visualizar Editar

Mostrando 1 - 3 de 3 resultados.

<< < 1 > >>

Figura 5: Consulta de NCs vista por um usuário comum do sistema



Não conformidades

- * Não conformidades
- * Produtos não conformes
- * Ações SGQ

+ Adicionar

Código	Usuário de registro	Usuário responsável	Título	Procedente	Prazo de conclusão	Novo prazo de conclusão	Registro	Conclusão	Status SGQ	Origem	Auditoria	Resultado de checklist
1	sgq		Reclamação de cliente sobre acabamento	false			31/03/2020		REGISTRADO	RECLAMACAO_CLIENTE		Visualizar Editar Excluir
2	sgq	user	Desconhecimento sobre o SGQ	true	28/03/2020	21/03/2020			PENDENTE	AUDITORIA		Visualizar Editar Excluir
3	sgq		Indicador de produtividade abaixo da meta em Março/2020	true			01/04/2020		PENDENTE	ANALISE_CRITICA		Visualizar Editar Excluir

Mostrando 1 - 3 de 3 resultados.

<< < 1 > >>

Figura 6: Consulta de NCs vista pelo gestor do SGQ

- **Módulo de incidentes e problemas – Registro de PNCs**

Esse requisito permite ao usuário registrar produtos não conformes. Deve ser possível informar o título, a descrição, a data de registro, o status e o produto relacionado ao PNC. Opcionalmente deve-se poder adicionar arquivos anexos ao PNC e relacioná-lo a um resultado de checklist de processo. Para o gestor do SGQ devem ser exibidos campos adicionais: usuário responsável, precedente (booleano), análise final, a ação de mitigação e uma não conformidade associada.

A imagem mostra a interface de usuário para o registro de um Produto Não Conforme (PNC) no sistema SGQ. No topo, há uma barra de navegação com o logotipo 'SGQ v0.0.1-SNAPSHOT' e menus para 'Início', 'Ocorrências', 'Controle', 'Divulgação', 'Compliance', 'Relatórios' e 'Conta'. Um menu suspenso sob 'Ocorrências' está aberto, mostrando as opções: 'Não conformidades', 'Produtos não conformes' (destacado) e 'Ações SGQ'. O formulário principal, intitulado 'Criar ou editar produto não conforme', contém os seguintes campos: 'Usuário de registro' (campo de texto com ícone de lupa), 'Título' (campo de texto), 'Descrição' (campo de texto com ícone de lupa), 'Registro' (campo de data no formato dd/mm/yyyy), 'Status SGQ' (campo de seleção com seta para baixo), 'Produto' (campo de seleção com seta para baixo), 'Anexo' (área de upload de arquivos, mostrando exemplos como 'Diagrama elétrico Up!', 'Laudo técnico v1', 'Folder Plano de Auditorias 2020' e 'Catálogo de peças Gol'), e 'Resultado de checklist' (campo de seleção com seta para baixo).

Figura 7: Registro de PNC visto pelo usuário comum da aplicação

SGQ v0.0.1-SNAPSHOT

[Início](#)
[Configuração](#)
[Ocorrências](#)
[Controle](#)
[Divulgação](#)
[Compliance](#)
[Relatórios](#)
[Conta](#)

Criar ou editar produto não conforme

* Não conformidades
 * **Produtos não conformes**
 * Ações SGQ

Usuário de registro

Usuário responsável

Título

Descrição

Procedente

Registro

Análise final

Status SGQ

Ação

Não conformidade

Produto

Anexo
 Diagrama elétrico Up!
 Laudo técnico V1
 Folder Plano de Auditorias 2020
 Catálogo de peças Gol

Resultado de checklist

Figura 8: Registro de PNC visto pelo gestor do SGQ

- **Módulo de controle de processos – Consulta de itens de planos de auditorias**

Esse requisito permite que usuários visualizem a lista de itens do plano de auditoria atual na página inicial da aplicação. Devem ser exibidas as 5 próximas auditorias programadas no plano de auditorias ativo, ordenadas de forma crescente pela data de início.



Figura 9: Consulta de itens de plano de auditoria na página inicial da aplicação

● Módulo de controle de processos – Consulta de eventos operacionais

Esse requisito permite ao usuário consultar em uma lista paginada os eventos operacionais registrados na aplicação. Deve ser possível visualizar os detalhes de cada evento, bem como editar ou excluir um evento específico (apenas gestor do SGQ).

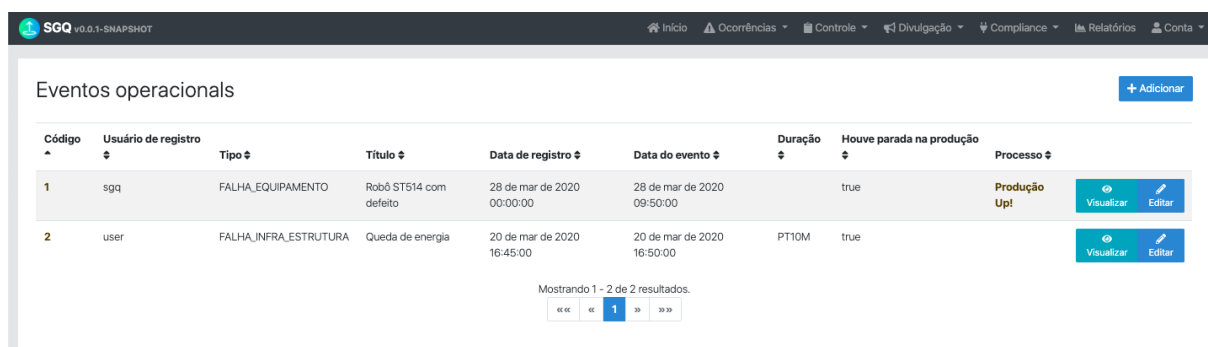


Figura 10: Consulta de eventos operacionais vista por um usuário comum da aplicação

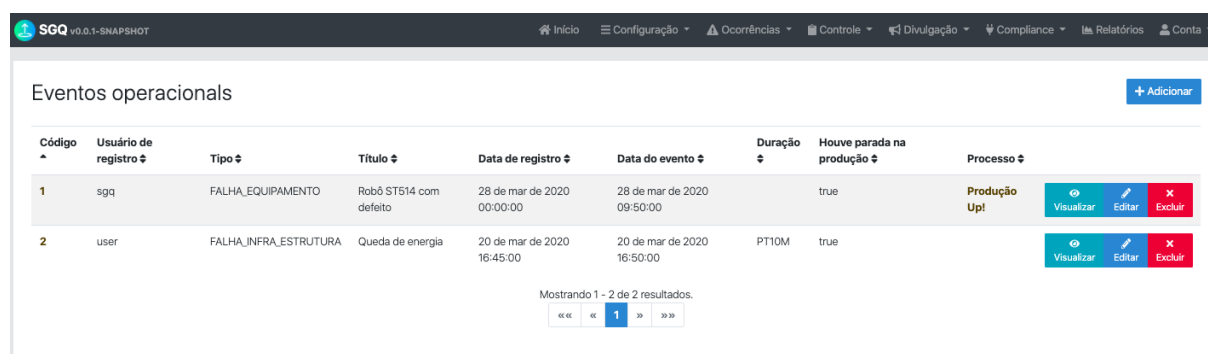


Figura 11: Consulta de eventos operacionais vista pelo gestor do SGQ

● Módulo de compliance – Registro de solicitações de análise

Esse requisito permite ao gestor do SGQ solicitar análises de não conformidades e seus planos de ação às empresas de consultoria habilitadas. Deve ser possível informar a data de registro, o status da solicitação, a não conformidade em análise e a empresa de consultoria.

Criar ou editar solicitação de análise

Usuário de registro

Registro

Solicitação

ID de acompanhamento

Status

Não conformidade

Consultoria

[Cancelar](#) [Salvar](#)

Normas
Solicitações de análise
Análises de consultoria

Figura 12: Solicitação de análise de consultoria

● Módulo de compliance – Consulta de normas

Esse requisito permite ao usuário consultar em uma lista paginada as normas contidas no catálogo externo de normas (SaaS). Deve ser possível visualizar os detalhes das normas e realizar o download do arquivo completo de cada norma.

Normas

Código	Órgão	Título	Versão	Número de edição	Edição	Início de validade	Categoria	
1	ABNT	NBR ISO 9001	2015	3	29/09/2015	29/10/2015	QUALIDADE	Visualizar Download
4	ABNT	NBR 14284	1999	1	09/06/1999	09/07/1999	TECNICA	Visualizar Download
5	ABNT	NBR 14481	2008	1	09/07/2008	09/08/2008	TECNICA	Visualizar Download
6	ABNT	NBR 14482	2000	2	10/01/2000	10/02/2000	TECNICA	Visualizar Download
7	ABNT	NBR 14752	2001	2	10/02/2001	10/03/2001	TECNICA	Visualizar Download
8	ABNT	NBR 14753	2001	4	10/03/2001	10/04/2001	TECNICA	Visualizar Download
9	ABNT	NBR 14754	2001	2	10/04/2001	10/05/2001	TECNICA	Visualizar Download
10	ABNT	NBR 14755	2001	1	10/05/2001	10/06/2001	TECNICA	Visualizar Download
11	ABNT	NBR 14777	2001	5	10/06/2001	10/07/2001	TECNICA	Visualizar Download
12	ABNT	NBR 14778	2001	1	10/07/2001	10/08/2001	TECNICA	Visualizar Download
13	ABNT	NBR 14779	2001	2	10/08/2001	10/09/2001	TECNICA	Visualizar Download
14	ABNT	NBR 14780	2001	1	10/09/2001	10/10/2001	TECNICA	Visualizar Download

Normas
Solicitações de análise
Análises de consultoria

Figura 13: Consulta de normas

Os requisitos *não funcionais* analisados durante a execução de tais requisitos funcionais são os seguintes:

- **Acessibilidade – Layouts responsivos para dispositivos móveis**

Esse requisito foi escolhido para garantir que os usuários não tenham a experiência de uso da aplicação prejudicada se o acesso é feito a partir de dispositivos distintos. Isso é especialmente importante nas funcionalidades acessadas possivelmente a partir de smartphones, como o registro de eventos operacionais e produtos não conformes (PNCs). Os critérios de aceite para a análise deste requisito são:

- As telas da aplicação devem se adaptar dinamicamente às diferentes resoluções dos dispositivos de acesso, reorganizando e redimensionando seus componentes da melhor forma possível;
- As telas devem ser intuitivas e proporcionar facilidade de navegação, mesmo quando em layouts adaptados para resoluções menores;
- A identidade visual da aplicação deve se manter em todas as resoluções;
- A aplicação deve ser acessível a partir dos navegadores web mais modernos (Chrome, Firefox, Edge, Safari), bem como de navegadores de dispositivos móveis.

- **Segurança – Bloqueio de acessos não autenticados/autorizados**

Esse RNF foi escolhido devido à preocupação com a segurança dos dados corporativos, pois determinadas informações nas mãos da concorrência podem se traduzir em grandes perdas financeiras. Os critérios de aceite para a análise deste requisito são:

- Proteger páginas privadas de acessos não autenticados e/ou autorizados;
- Em acessos não autenticados a aplicação deve exibir o formulário de login. Após a autenticação o usuário deve ser redirecionado à página inicialmente desejada;
- O sistema deve permitir a exibição de dados e de telas públicas (sem autenticação).

- **Desempenho – A aplicação deve apresentar performance elevada**

Esse requisito foi selecionado dada a necessidade de manter a performance uniforme mesmo para usuários geograficamente dispersos. Os critérios de aceite para a análise deste requisito são:

- As telas de consulta paginada da aplicação devem exibir uma página de dados em até 3 segundos;

- **Disponibilidade – A aplicação deve ser resiliente a falhas**

Esse requisito foi selecionado dada a importância de manter a aplicação disponível mesmo quando algum recurso estiver inacessível. A resiliência a falhas é uma característica essencial de aplicações distribuídas, pois erros ocorrem de forma inesperada em qualquer nó da arquitetura. Pode-se listar exemplos como bancos de dados inacessíveis, APIs retornando erros, chamadas de componentes falhando por timeout. Os critérios de aceite para a análise deste requisito são:

- A aplicação deve exibir uma mensagem de erro quando algum serviço externo estiver inacessível, mas sem que isso represente uma quebra total da aplicação. Tão logo o recurso esteja novamente disponível o comportamento esperado deve ser retomado.

- **Escalabilidade – A aplicação deve permitir o escalonamento horizontal**

Esse requisito foi selecionado dada a necessidade de garantir o atendimento da demanda de usuários. Em outras palavras, quando do aumento do número de requisições para o backend a aplicação deve permitir o escalonamento horizontal (mais instâncias) dos componentes saturados. Os critérios de aceite para a análise deste requisito são:

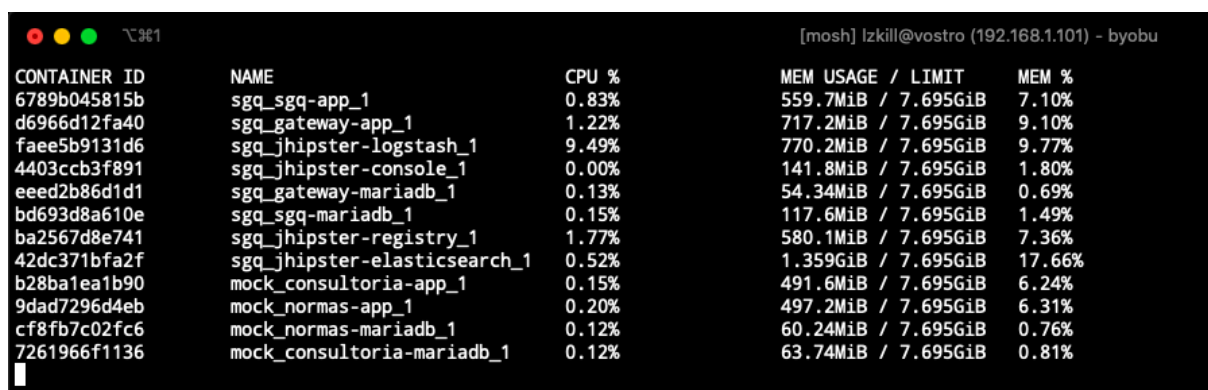
- O ambiente de execução da aplicação deve permitir que o escalonamento de um componente qualquer seja realizado de forma simples e direta, podendo inclusive ser automatizado por outros componentes da arquitetura responsáveis pelo monitoramento da demanda.

O protótipo completo foi implantado em um cluster Docker Swarm local, contendo duas stacks distintas: uma da aplicação e outra para os mocks de serviços externos. A stack de monitoramento é opcional e sua ausência não impede a execução da stack principal. (veja <https://12factor.net/backing-services>)

Stack principal: https://github.com/lzkill/tcc_puc-minas/blob/develop/docker-compose/docker-compose.yml

Stack de serviços externos (mock):

https://github.com/lzkill/tcc_puc-minas/blob/develop/mock/docker-compose/docker-compose.yml

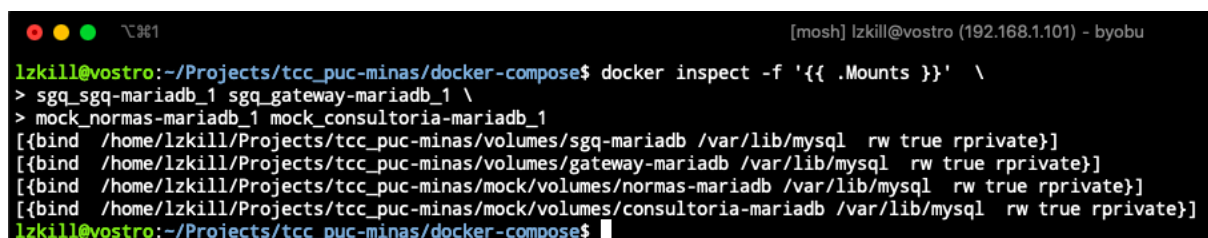


[mosh] lzkill@vostro (192.168.1.101) - byobu

CONTAINER ID	NAME	CPU %	MEM USAGE / LIMIT	MEM %
6789b045815b	sgq_sgq-app_1	0.83%	559.7MiB / 7.695GiB	7.10%
d6966d12fa40	sgq_gateway-app_1	1.22%	717.2MiB / 7.695GiB	9.10%
faee5b9131d6	sgq_jhipster-logstash_1	9.49%	770.2MiB / 7.695GiB	9.77%
4403ccb3f891	sgq_jhipster-console_1	0.00%	141.8MiB / 7.695GiB	1.80%
eeed2b86d1d1	sgq_gateway-mariadb_1	0.13%	54.34MiB / 7.695GiB	0.69%
bd693d8a610e	sgq_sgq-mariadb_1	0.15%	117.6MiB / 7.695GiB	1.49%
ba2567d8e741	sgq_jhipster-registry_1	1.77%	580.1MiB / 7.695GiB	7.36%
42dc371bfa2f	sgq_jhipster-elasticsearch_1	0.52%	1.359GiB / 7.695GiB	17.66%
b28ba1ea1b90	mock_consultoria-app_1	0.15%	491.6MiB / 7.695GiB	6.24%
9dad7296d4eb	mock_normas-app_1	0.20%	497.2MiB / 7.695GiB	6.31%
cf8fb7c02fc6	mock_normas-mariadb_1	0.12%	60.24MiB / 7.695GiB	0.76%
7261966f1136	mock_consultoria-mariadb_1	0.12%	63.74MiB / 7.695GiB	0.81%

Figura 14: Stacks do protótipo em execução

Os bancos de dados foram mapeados para volumes locais Docker, de modo que não há perda em caso de restart dos componentes. Isso é essencial em aplicações containerizadas, pois containers devem ser voláteis para permitir o escalonamento horizontal.



[mosh] lzkill@vostro (192.168.1.101) - byobu

```
lzkill@vostro:~/Projects/tcc_puc-minas/docker-compose$ docker inspect -f '{{ .Mounts }}' \
> sgq_sgq-mariadb_1 sgq_gateway-mariadb_1 \
> mock_normas-mariadb_1 mock_consultoria-mariadb_1
[{"bind": "/home/lzkill/Projects/tcc_puc-minas/volumes/sgq-mariadb /var/lib/mysql", "rw": true, "rprivate": true}, {"bind": "/home/lzkill/Projects/tcc_puc-minas/volumes/gateway-mariadb /var/lib/mysql", "rw": true, "rprivate": true}, {"bind": "/home/lzkill/Projects/tcc_puc-minas/mock/volumes/normas-mariadb /var/lib/mysql", "rw": true, "rprivate": true}, {"bind": "/home/lzkill/Projects/tcc_puc-minas/mock/volumes/consultoria-mariadb /var/lib/mysql", "rw": true, "rprivate": true}]
lzkill@vostro:~/Projects/tcc_puc-minas/docker-compose$
```

Figura 15: Volumes do protótipo (persistência)

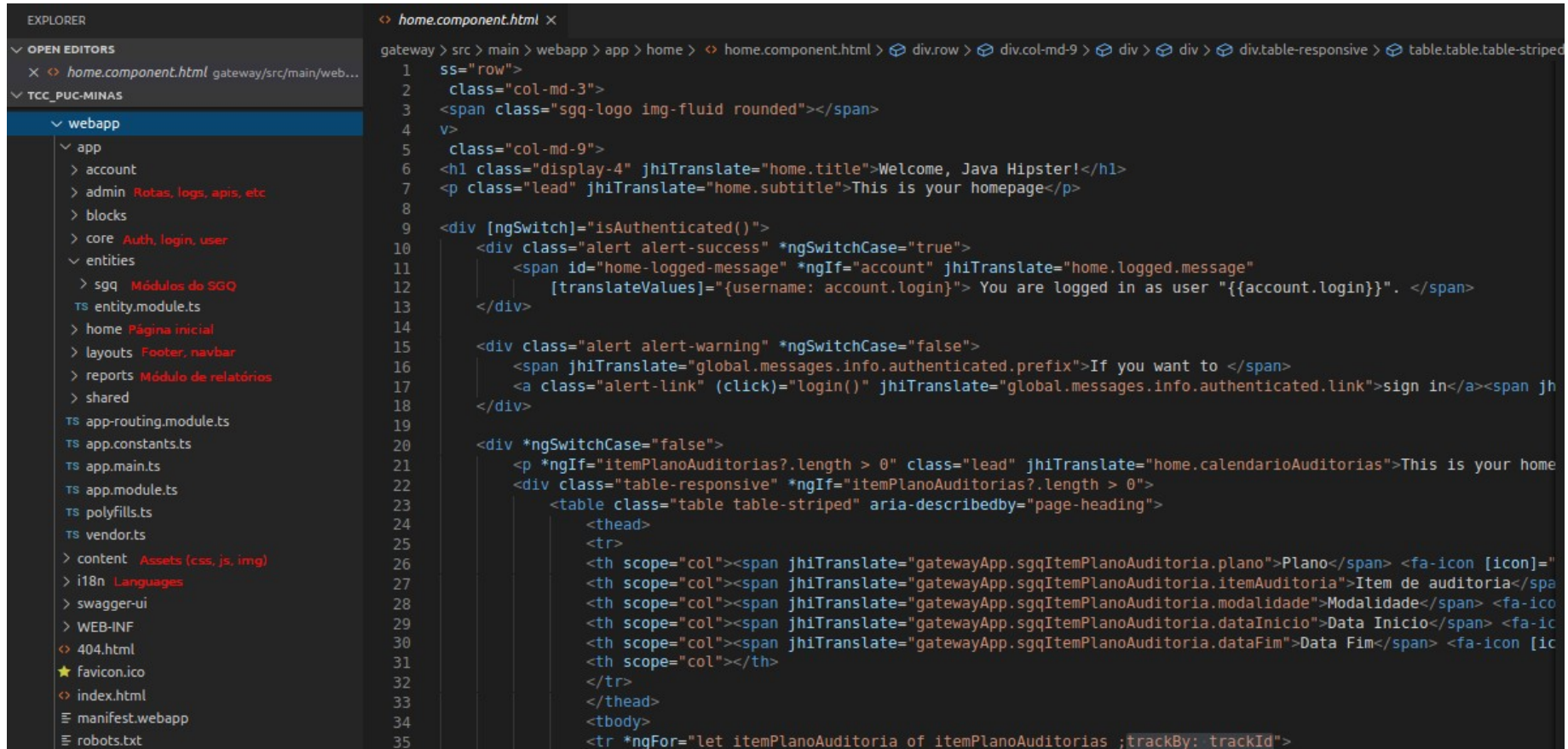
5.2 Código

Repositório do projeto: https://github.com/lzkill/tcc_puc-minas

O código da aplicação frontend segue em grande parte as recomendações de estrutura para projetos Angular. Módulos e componentes são organizados em seus respectivos diretórios e o processo de *bootstrapping* da aplicação é responsável pelo carregamento em memória. Essa organização permite que módulos e componentes sejam facilmente reutilizados em outras aplicações, pois estes são construídos desde o início com a idéia de isolamento das dependências.

O código de backend, mais especificamente do microserviço de SGQ, segue a estrutura convencionada para projetos Spring Boot / Spring MVC. Há pacotes específicos para os controladores Web, recursos RESTful, classes de configuração e repositórios de dados (DAOs). Tal organização leva ao desenvolvimento de programas mais legíveis e por conseguinte alta manutenibilidade.

Estrutura do código frontend:



The image shows a code editor interface with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'webapp', 'app', 'account', 'admin', 'blocks', 'core', 'entities', 'reports', 'shared', and 'vendor'. The 'webapp' folder is expanded, showing subfolders like 'app', 'content', 'i18n', 'swagger-ui', and 'WEB-INF'. The 'app' folder is also expanded, showing files like 'app.constants.ts', 'app.main.ts', 'app.module.ts', 'polyfills.ts', and 'vendor.ts'. The 'home.component.html' file is selected in the file explorer. The code editor shows the content of 'home.component.html', which is a template for a home page. It includes a header with a logo and a title, a main content area with a message and a table, and a footer. The code uses Angular's syntax for components, including directives like 'ngSwitch' and 'ngIf', and data-binding expressions like 'jhiTranslate' and 'gatewayApp.sggItemPlanoAuditoria.plano'.

```
gateway > src > main > webapp > app > home > home.component.html > div.row > div.col-md-9 > div > div > div.table-responsive > table.table.table-striped
1  <div class="row">
2    <div class="col-md-3">
3      <span class="sgq-logo img-fluid rounded"></span>
4    </div>
5    <div class="col-md-9">
6      <h1 class="display-4" jhiTranslate="home.title">Welcome, Java Hipster!</h1>
7      <p class="lead" jhiTranslate="home.subtitle">This is your homepage</p>
8    </div>
9    <div [ngSwitch]="isAuthenticated()">
10      <div class="alert alert-success" *ngSwitchCase="true">
11        <span id="home-logged-message" *ngIf="account" jhiTranslate="home.logged.message"
12          [translateValues]="{username: account.login}"> You are logged in as user "{{account.login}}". </span>
13      </div>
14      <div class="alert alert-warning" *ngSwitchCase="false">
15        <span jhiTranslate="global.messages.info.authenticated.prefix">If you want to </span>
16        <a class="alert-link" (click)="login()" jhiTranslate="global.messages.info.authenticated.link">sign in</a><span jh
17      </div>
18    </div>
19    <div *ngSwitchCase="false">
20      <p *ngIf="itemPlanoAuditorias?.length > 0" class="lead" jhiTranslate="home.calendarioAuditorias">This is your home
21      <div class="table-responsive" *ngIf="itemPlanoAuditorias?.length > 0">
22        <table class="table table-striped" aria-describedby="page-heading">
23          <thead>
24            <tr>
25              <th scope="col"><span jhiTranslate="gatewayApp.sggItemPlanoAuditoria.plano">Plano</span> <fa-icon [icon]="
26              <th scope="col"><span jhiTranslate="gatewayApp.sggItemPlanoAuditoria.itemAuditoria">Item de auditoria</span> </th>
27              <th scope="col"><span jhiTranslate="gatewayApp.sggItemPlanoAuditoria.modalidade">Modalidade</span> <fa-ico
28              <th scope="col"><span jhiTranslate="gatewayApp.sggItemPlanoAuditoria.dataInicio">Data Inicio</span> <fa-ic
29              <th scope="col"><span jhiTranslate="gatewayApp.sggItemPlanoAuditoria.dataFim">Data Fim</span> <fa-icon [ic
30              <th scope="col"></th>
31            </tr>
32          </thead>
33          <tbody>
34            <tr *ngFor="let itemPlanoAuditoria of itemPlanoAuditorias ;trackBy: trackId">
```

Figura 16: Estrutura do código frontend

Estrutura do código backend (microserviço SGQ):

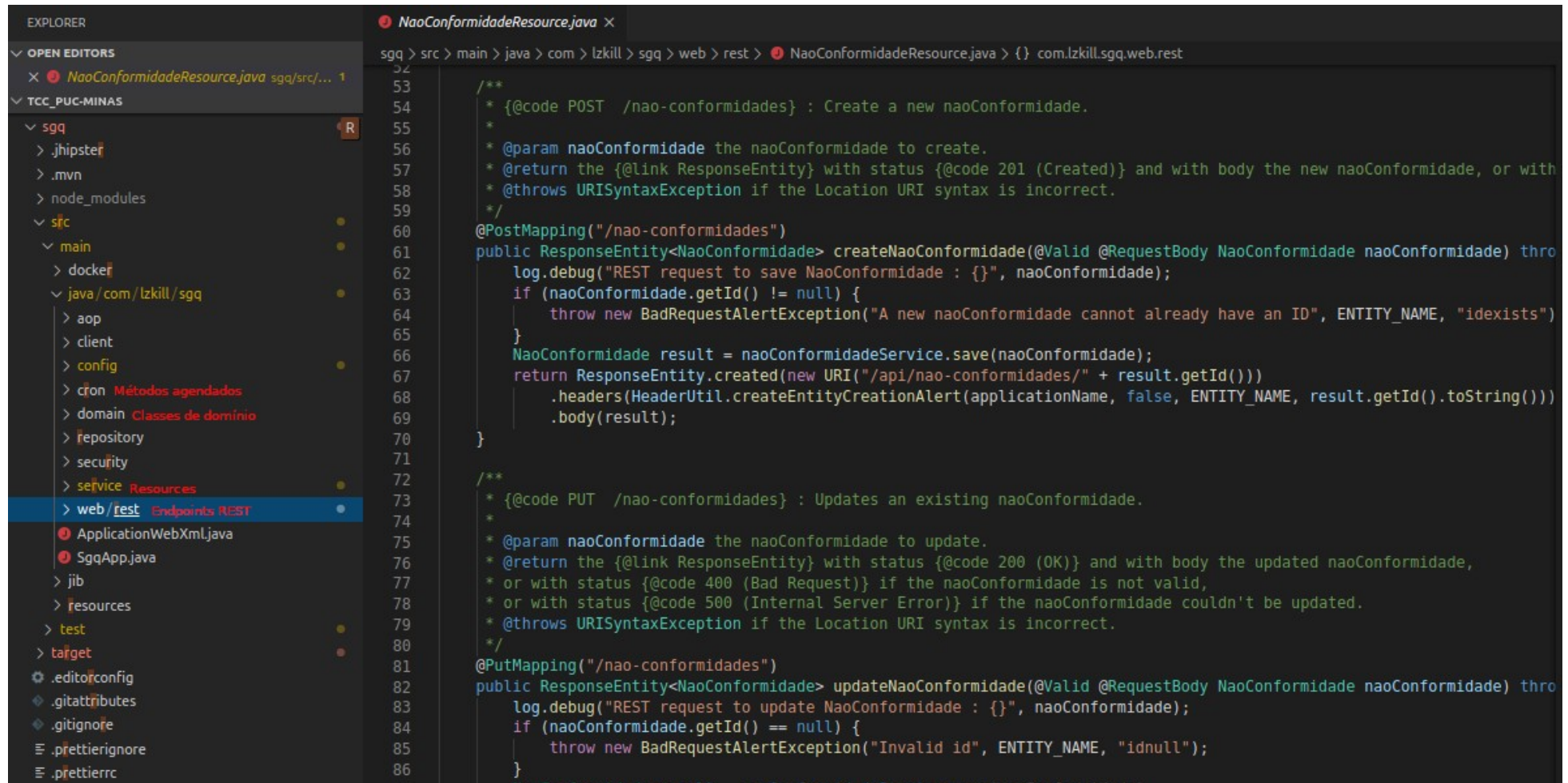


Figura 17: Estrutura do código backend (microserviço SGO)

5.3 Interfaces/ APIs

Dois contratos de API merecem destaque na arquitetura da solução: a interface do SaaS de normas e a interface fornecida para as empresas de consultoria. A primeira é consumida pelo frontend da aplicação, enquanto a segunda é implementada pelas empresas de consultoria contratadas e consumida pelo microserviço de SGQ. A seguir ambas são listadas utilizando o padrão OpenAPI.

Normas API:

https://editor.swagger.io/?url=https://raw.githubusercontent.com/lzkill/tcc_puc-minas/develop/normas.yml

Consultoria API:

https://editor.swagger.io/?url=https://raw.githubusercontent.com/lzkill/tcc_puc-minas/develop/consultoria.yml

6. Avaliação da Arquitetura

6.1. Análise das abordagens arquiteturais

A arquitetura de microserviços é vista hoje como uma evolução natural das arquiteturas para Web, em especial para aplicações que lidam com grandes volumes de tráfego (web scale). Nestes cenários é preciso que a arquitetura permita a escalabilidade horizontal, a manutenção sem downtime e a automação do processo de desenvolvimento e entrega. Mas microserviços não se aplicam apenas a aplicações dessa natureza. Sistemas de qualquer tamanho e escopo podem se beneficiar das vantagens inerentes à arquitetura distribuída baseada em microserviços. Neste projeto ficou claro que o SGQ adere perfeitamente a esta arquitetura e pode dela extrair qualidades para entrega de valor ao negócio.

Entretanto, como qualquer arquitetura, microserviços também impõem desafios. Tarefas antes simples como joins entre tables tornam-se mais complexas e elaboradas, demandando mais análise e expertise do arquiteto. Felizmente o mercado já avançou consideravelmente nesse campo, de modo que já existem soluções e ferramentas para todas as dificuldades que surgem na adoção de microserviços. Neste projeto pode-se ver em ação algumas delas, como o gateway de aplicação multi-tarefa, o registro de serviços com configuração distribuída e a stack de monitoramento com agregação de logs.

6.2. Cenários

Cenário 1: ao acessar a aplicação a partir de um dispositivo móvel com dimensões de tela reduzidas os componentes do frontend devem se ajustar da melhor forma para que a experiência do usuário não seja prejudicada. Eles devem ter suas proporções redimensionadas, posicionamentos recalculados, textos omitidos e etc. A aplicação deve manter sua identidade visual quando exibe layouts ajustados.

Cenário 2: ao executar os casos de uso da aplicação o usuário deve ter uma experiência fluida e intuitiva, de modo que a organização dos componentes e os fluxos das telas não sejam entraves para o seu trabalho. O usuário deve ser capaz de inserir um registro de não conformidade em até 3 minutos.

Cenário 3: ao acessar os casos de uso da aplicação o usuário deve ter uma experiência vívida, sem travamentos ou lentidão. O usuário deve visualizar a primeira página da consulta de eventos operacionais em no máximo 3 segundos. Este tempo deve incluir a chamada ao backend, o processamento no microserviço, o tráfego dos dados de retorno, o processamento pelo frontend e a renderização da lista.

Cenário 4: ao tentar acessar uma tela privada da aplicação sem autenticação prévia o usuário deve ser redirecionado ao formulário de login. Entretanto, o acesso a telas públicas deve ser permitido sem autenticação. Se uma chamada a uma rota privada do microserviço de SGQ for realizada um erro apropriado deve ser retornado. Chamadas a rotas públicas devem ser permitidas.

Cenário 5: ao acessar uma página da aplicação que obtém dados de um serviço externo, esta deve exibir uma mensagem de erro caso o serviço esteja offline. Isso deve ser feito sem que a aplicação como um todo seja impactada. Tão logo o serviço externo seja reestabelecido a página deve exibir os dados requisitados.

Cenário 6: a arquitetura da aplicação deve permitir o escalonamento horizontal de qualquer componente quando for percebido um aumento significativo do volume de requisições de usuários. Isso deve ser feito de maneira clara e simples, sem a necessidade de configuração manual de componentes de rede ou de quaisquer outro tipo.

6.3. Avaliação

- **Cenário 1**

Atributo de qualidade	Acessibilidade
Requisito de qualidade	A aplicação deve suportar ambientes Web responsivos e ambiente <i>mobile</i> .
Preocupação	Fornecer uma experiência inferior ao usuário no uso da aplicação.
Cenário	Cenário 1
Ambiente	Funcionamento com carga normal.
Estímulo	Usuário acessando a consulta de não conformidades.
Mecanismo	Utilizar um framework CSS que permita o desenvolvimento de telas responsivas.
Medida de resposta	A aplicação deve se ajustar a dispositivos com tamanhos de telas reduzidos.
Riscos	Uma experiência de uso inferior pode levar ao desuso da aplicação pelos usuários mais próximos da geração de não conformidades e produtos não conformes.
Pontos sensíveis	N/A

Trade-offs	N/A
-------------------	-----

Evidências

A aplicação apresentou o comportamento esperado, ajustando apropriadamente quando acessada a partir de um smartphone.

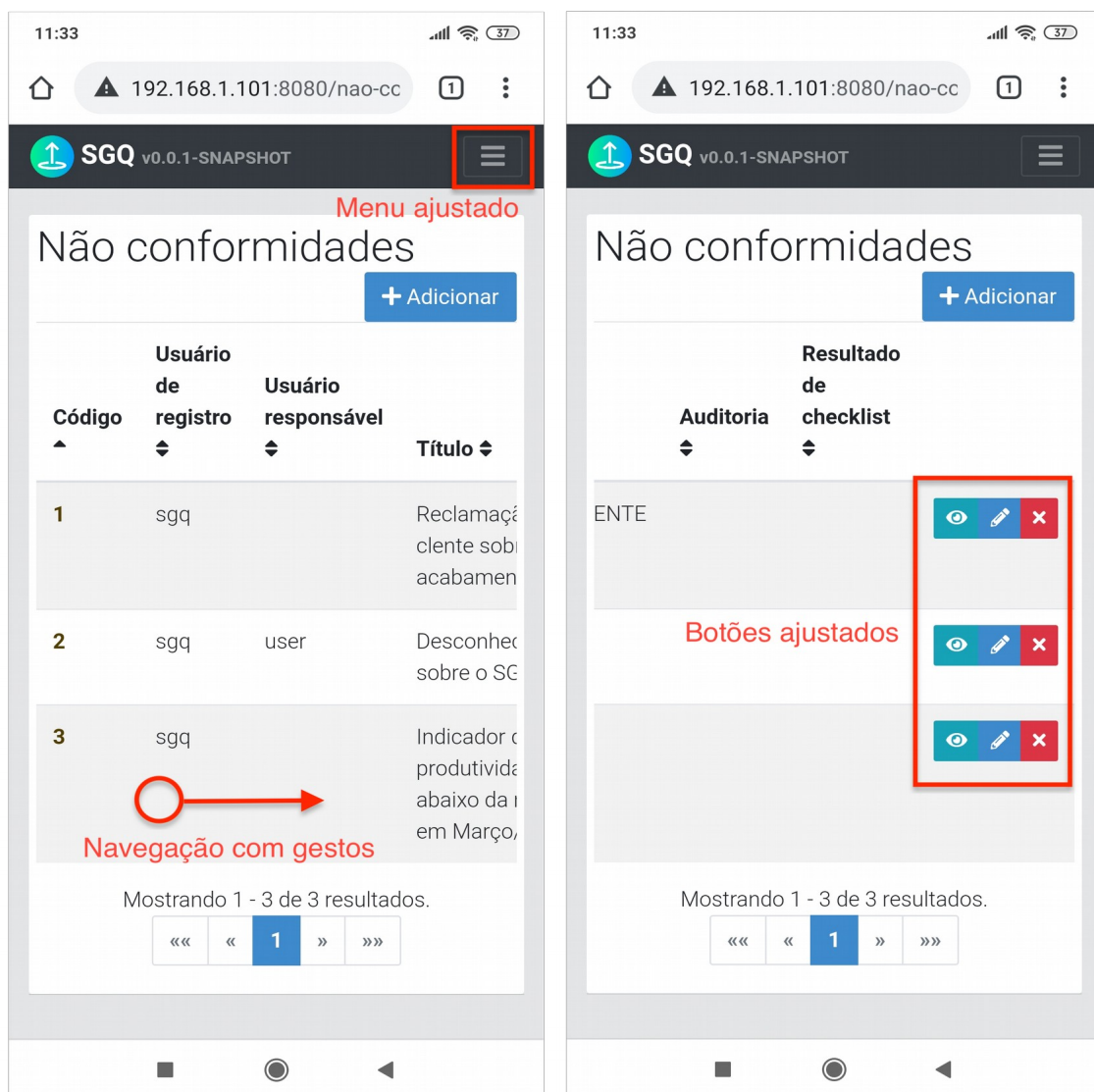


Figura 18: Evidências do cenário 1

● Cenário 2

Atributo de qualidade	Usabilidade
Requisito de qualidade	A aplicação deve ser de fácil utilização
Preocupação	Fornecer telas organizadas de forma que não dificultem a realização do trabalho do usuário
Cenário	Cenário 2
Ambiente	Funcionamento com carga normal.

Estímulo	Registro de não conformidade
Mecanismo	Menus organizados por módulo, telas estruturadas de forma intuitiva, campos associados a labels descritivos, área para exibição de mensagens de erro, padronização na estrutura (botões de ação sempre no final da página), campos obrigatórios assinalados, botões desativados até que seus requisitos sejam satisfeitos.
Medida de resposta	O usuário não deve levar mais que 3 minutos para registrar uma não conformidade.
Riscos	Uma experiência de uso inferior pode levar ao desuso da aplicação pelos usuários mais próximos da geração de não conformidades e produtos não conformes.
Pontos sensíveis	N/A
Trade-offs	N/A

Evidências

A aplicação apresentou navegação intuitiva e direta, permitindo o registro de uma não conformidade e seu plano de ação em poucos segundos. Tempos maiores serão observados se a não conformidade possuir um texto descritivo muito extenso ou um plano de ação com muitos itens.

SGQ v0.0.1-SNAPSHOT

Menus por módulo →

Início Ocorrências Controle Divulgação Compliance Relatórios Conta

* Não conformidades

* Produtos não conformes

* Ações SGQ

Criar ou editar não conformidade

Usuário de registro

O campo é obrigatório.

Labels →

Campos obrigatórios marcados →

Título

Reclamação de cliente sobre consumo

Descrição

O cliente Fulano de Tal fez reclamação através do Fale Conosco sobre o consumo elevado do veículo Gol, número de série CCY671761. Ele relata que...

Registro

02/04/2020 11:52

Status SGQ

REGISTRADO

Origem

Anexo

Comunicado interno 172

Boletim informativo 03-2020

Boletim informativo 18-2020

Campanha de recall 02-2020

Capacitação de colaboradores - v1

Auditoria

Resultado de checklist

Cancelar Salvar

Botões em local padronizado →

Figura 19: Evidências do cenário 2 – Registro de NC

Criar ou editar ação

Usuário de registro
Gestor SGQ

Usuário responsável
User User

Tipo
ACAO_CORRETIVA

Título
Substituição do módulo de injeção eletrônica

Descrição
Efetuar a substituição do módulo de injeção eletrônica

Prazo de conclusão
02/04/2020 17:00

Novo prazo de conclusão
dd/mm/yyyy --:--

Registro
02/04/2020 13:49

Conclusão
dd/mm/yyyy --:--

Resultado

Status SGQ
PENDENTE

Anexo
Diagrama elétrico Up!
Laudo técnico v1
Folder Plano de Auditorias 2020
Catálogo de peças Gol

Não conformidade
Reclamação de cliente sobre consumo

Cancelar Salvar

Figura 20: Evidências do cenário 2 – Registro de ação corretiva (plano de ação)

● Cenário 3

Atributo de qualidade	Desempenho
Requisito de qualidade	A aplicação deve apresentar performance elevada
Preocupação	Fornecer uma experiência de uso fluida, sem travamentos ou lentidão.
Cenário	Cenário 3
Ambiente	Funcionamento com carga normal.

Estímulo	Consulta de eventos operacionais
Mecanismo	Chamadas RESTFul trafegando dados serializados no formato JSON, backend construído com o uso de tecnologias e padrões de implementação eficientes, dados de retorno aglutinados para evitar o encadeamento de chamadas.
Medida de resposta	A primeira página da lista de eventos operacionais foi exibida em até 3 segundos.
Riscos	Uma experiência de uso inferior pode levar ao desuso da aplicação pelos usuários mais próximos da geração de não conformidades e produtos não conformes.
Pontos sensíveis	N/A
Trade-offs	N/A

Evidências

A aplicação apresentou o comportamento esperado, sendo capaz de renderizar a primeira página da listagem de eventos operacionais em pouco mais que 1 segundo.

SGQ v0.0.1-SNAPSHOT

Configuração Ocorrências Controle Divulgação Compliance Relatórios

Eventos operacionais

+ Adicionar

Código	Usuário de registro	Tipo	Título	Data de registro	Data do evento	Duração	Houve parada na produção	Processo	
1	sgq	FALHA_EQUIPAMENTO	Robô ST514 com defeito	28 de mar de 2020 00:00:00	28 de mar de 2020 09:50:00		true	Produção Up!	Visualizar
2	user	FALHA_INFRA_ESTRUTURA	Queda de energia	20 de mar de 2020 16:45:00	20 de mar de 2020 16:50:00	PT10M	true		Visualizar
3	sgq	FALHA_INFRA_ESTRUTURA	Interrupção dos dutos de resfriamento	2 de abr de 2020 17:05:00	1 de abr de 2020 09:17:00	PT10M	false		Visualizar
4	user	FALHA_PROCESSO	Máquina de corte a laser HD82 apresentou superaquecimento	2 de abr de 2020 17:06:00	1 de abr de 2020 06:01:00	PT2M30S	true	Produção Fox	Visualizar
5	sgq	FALHA_INFRA_ESTRUTURA	Queda no fornecimento de energia	25 de mar de 2020 23:00:00	25 de mar de 2020 22:30:00	PT2M	true		Visualizar

Mostrando 1 - 5 de 5 resultados.

« « 1 » »

Elements Console Sources Network Performance

Filter

Hide data URLs

All XHR JS CSS Img Media Font Doc WS Manifest Other

Only show requests with SameSite issues

500 ms 1000 ms 1500 ms 2000 ms

Name	Status	Protocol	T...	Size	Time	Waterfall
evento-operacional?page=...	200	http/1.1	d...	2.8 KB	15 ms	
loading.css	200	http/1.1	st...	972 B	11 ms	
main.07b91d0f25a5e3ef80...	200	http/1.1	st...	22.7 KB	26 ms	
global.c5875b4a3bbc1ca8...	200	http/1.1	st...	964 B	17 ms	
polyfills.5b54e70d46e48c5...	200	http/1.1	s...	22.4 KB	39 ms	
main.5b54e70d46e48c5c1...	200	http/1.1	s...	438 KB	372 ms	
global.5b54e70d46e48c5c...	200	http/1.1	s...	1.2 KB	41 ms	
logo-jhipster.png	200	http/1.1	png	845 B	6 ms	
pt-br.json?buildTimestamp...	200	http/1.1	xhr	8.1 KB	172 ms	
account	200	http/1.1	xhr	767 B	160 ms	
info	200	http/1.1	xhr	802 B	151 ms	
12.5b54e70d46e48c5c121...	200	http/1.1	s...	13.3 KB	126 ms	
users	200	http/1.1	xhr	1011 B	158 ms	
evento-operacionais?page=...	200	http/1.1	xhr	1.5 KB	155 ms	
a1a286736c98ee4d3ca70f...	200	http/1.1	png	20.6 KB	102 ms	

Tempo total da requisição

15 requests 536 KB transferred 2.0 MB resources Finish: 1.44 s DOMContentLoaded: 899 ms

Figura 21: Evidência do cenário 3

● Cenário 4

Atributo de qualidade	Segurança
Requisito de qualidade	A aplicação deve proteger dados e funcionalidades com um mecanismo de segurança confiável
Preocupação	Permitir que dados e telas de caráter privado sejam acessados por pessoas sem autorização
Cenário	Cenário 4
Ambiente	Funcionamento com carga normal.
Estímulo	Acesso a uma página privada sem login prévio / chamada direta a uma rota do microserviço de SGQ sem token de acesso
Mecanismo	Implementação de rotas protegidas no frontend Angular / implementação de rotas protegidas no backend Java (token JWT)
Medida de resposta	O frontend impede o acesso à página e exibe o formulário de login, após o qual o acesso é fornecido automaticamente sem a necessidade de um novo estímulo (clique); o acesso a uma rota do backend sem token de acesso retorna erro 401.
Riscos	O vazamento de dados confidenciais pode significar vantagem competitiva para os concorrentes ou ações judiciais contra a empresa.
Pontos sensíveis	N/A
Trade-offs	N/A

Evidências

A aplicação apresentou o comportamento esperado, exibindo o formulário de login quando é feito o acesso a uma página privada sem autenticação prévia; páginas públicas são exibidas sem a necessidade de login; a chamadas a rotas protegidas do backend sem token de acesso retorna erro 401.

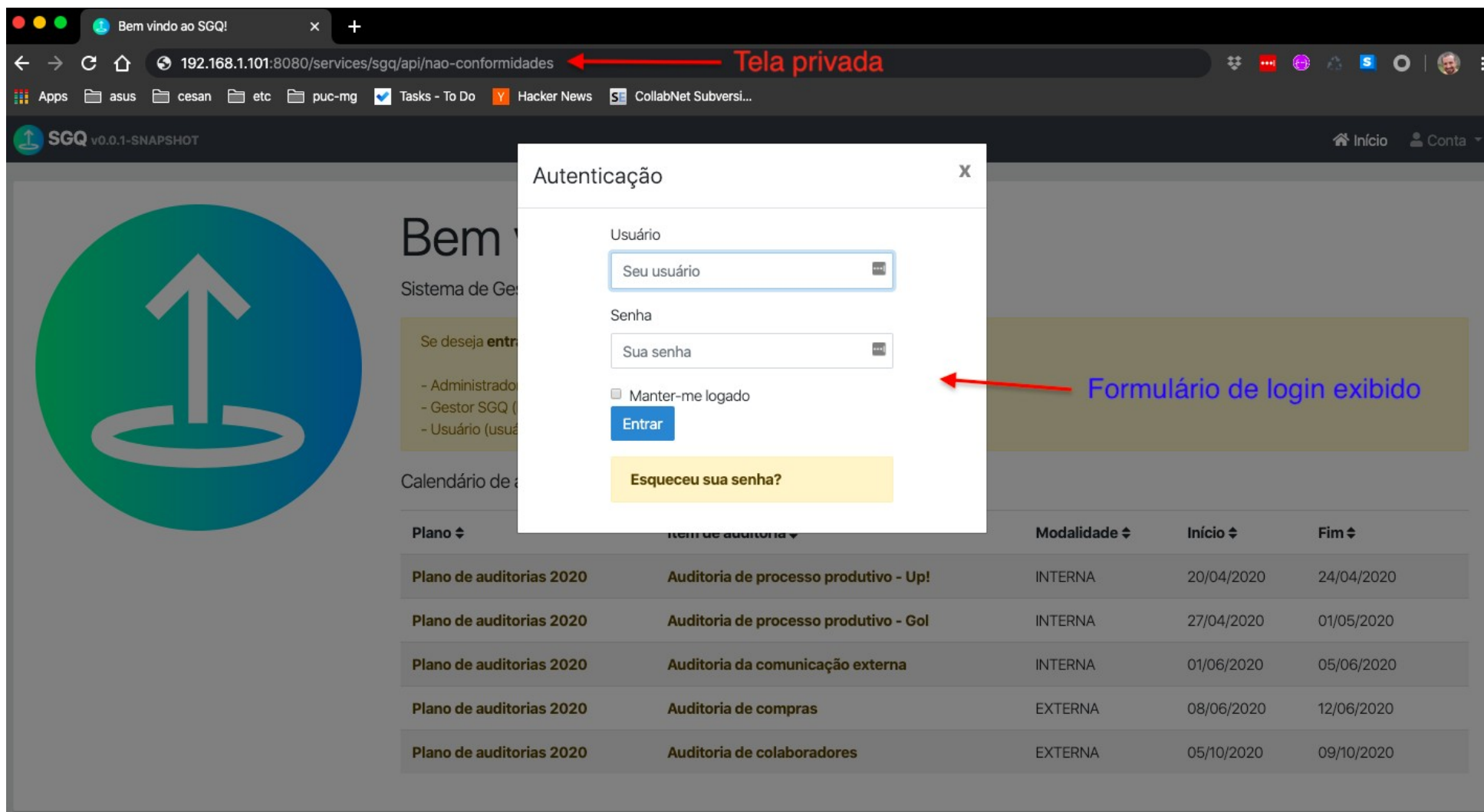


Figure 22: Evidência do cenário 4 - Formulário de login



Bem vindo ao SGQ!

Sistema de Gestão da Qualidade

Se deseja **entrar**, utilize as seguintes contas padrões:

- Administrador (usuário="admin" e senha="admin")
- Gestor SGQ (login="sgq" e senha="admin")
- Usuário (usuário="user" e senha="user").

Calendário de auditorias

Dados exibidos em uma página pública, sem login

Plano ↕	Item de auditoria ↕	Modalidade ↕	Início ↕	Fim ↕
Plano de auditorias 2020	Auditoria de processo produtivo - Up!	INTERNA	20/04/2020	24/04/2020
Plano de auditorias 2020	Auditoria de processo produtivo - Gol	INTERNA	27/04/2020	01/05/2020
Plano de auditorias 2020	Auditoria da comunicação externa	INTERNA	01/06/2020	05/06/2020
Plano de auditorias 2020	Auditoria de compras	EXTERNA	08/06/2020	12/06/2020
Plano de auditorias 2020	Auditoria de colaboradores	EXTERNA	05/10/2020	09/10/2020

Figura 23: Evidência do cenário 4 - página pública

```
[{"id": 1, "titulo": "Auditoria Up! 2020", "descricao": null, "modalidade": "INTERNA", "dataInicio": "2020-04-20T11:00:00Z", "dataFim": "2020-04-24T20:00:00Z", "itemAuditoria": {"id": 1, "titulo": "Auditoria de processo produtivo - Up!", "descricao": "ISO 9001 - 8.1.2, 8.1.3", "habilitado": true, "processo": {"id": 1, "titulo": "Produ\u00e7\u00e3o Up!", "descricao": "PS14-2020", "habilitado": true, "anexos": null, "setor": {"id": 4, "nome": "Setor de produ\u00e7\u00e3o", "habilitado": true, "checklists": null, "empresa": {"id": 1, "nome": "Volkswagen do Brasil Ltda", "habilitado": true, "produtos": null}}}}, {"anexos": null}]
```

Figura 24: Evid\u00eancia do cen\u00e1rio 4 - chamada a uma rota p\u00fablica do backend

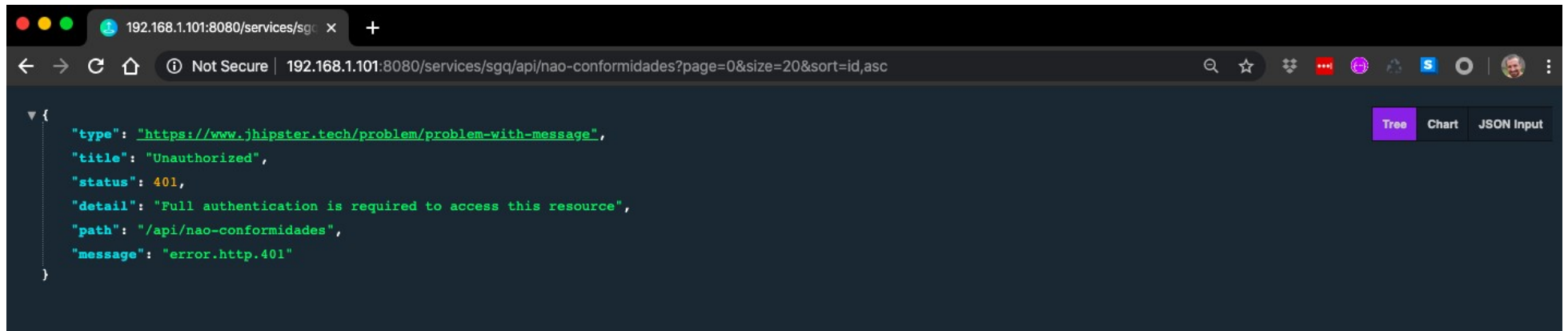


Figure 25: Evidência do cenário 4 - chamada a uma rota privada do backend sem token de acesso

- **Cenário 5**

Atributo de qualidade	Disponibilidade
Requisito de qualidade	A aplicação deve ser resiliente a falhas
Preocupação	Garantir o funcionamento da aplicação mesmo quando algum componente da arquitetura está em falha ou não disponível.
Cenário	Cenário 5
Ambiente	Funcionamento com carga normal, serviço externo de normas offline.
Estímulo	Acesso ao catálogo de normas
Mecanismo	Filtro dos códigos de retorno nas chamadas HTTP (4*, 5*), exibição de mensagens de erro
Medida de resposta	A aplicação continua operacional para todos os usuários, sem impacto nos tempos de resposta.
Riscos	A aplicação não pode ter seu funcionamento impactado por serviços externos, pois estes estão fora de controle da empresa detentora do SGQ.
Pontos sensíveis	N/A
Trade-offs	N/A

Evidências

A aplicação apresentou o comportamento esperado, exibindo uma mensagem de erro quando o serviço externo de normas está offline, sem contudo “quebrando” de forma irremediável quando a chamada não pôde ser concluída. Assim que o serviço é normalizado a aplicação volta ao funcionamento esperado.

```

lzkil@vostro ~$ docker-compose --project-name mock stop normas-app
Stopping mock_normas-app_1 ... done
lzkil@vostro ~$ docker ps -a
CONTAINER ID   NAMES                                PORTS                                STATUS
6789b045815b   sgq_sgq-app_1                       5701/udp, 8081/tcp                  Up 4 hours
d6966d12fa40   sgq_gateway-app_1                  5701/udp, 0.0.0.0:8080->8080/tcp    Up 27 hours
faee5b9131d6   sgq_jhipster-logstash_1            5044/tcp, 0.0.0.0:5000->5000/tcp, 0.0.0.0:5000->5000/udp, 9600/tcp    Up 27 hours
4403ccb3f891   sgq_jhipster-console_1             0.0.0.0:5601->5601/tcp              Up 27 hours
4e3cb28d2428   sgq_jhipster-import-dashboards_1   Exited (0) 4 hours ago
eed2b86d1d1    sgq_gateway-mariadb_1              0.0.0.0:3306->3306/tcp              Up 27 hours
bd693d8a610e   sgq_sgq-mariadb_1                  0.0.0.0:3307->3306/tcp              Up 27 hours
ba2567d8e741   sgq_jhipster-registry_1            0.0.0.0:8761->8761/tcp              Up 27 hours
42dc371bfa2f   sgq_jhipster-elasticsearch_1       9200/tcp, 9300/tcp                  Up 27 hours
b28ba1ea1b90   mock_consultoria-app_1              0.0.0.0:8083->8083/tcp              Up 27 hours
9dad7296d4eb   mock_normas-app_1                   Exited (137) 3 seconds ago
cf8fb7c02fc6   mock_normas-mariadb_1              0.0.0.0:3308->3306/tcp              Up 27 hours
7261966f1136   mock_consultoria-mariadb_1          0.0.0.0:3309->3306/tcp              Up 27 hours
lzkil@vostro ~$

```

Figura 26: Evidência do cenário 5 – Serviço externo offline

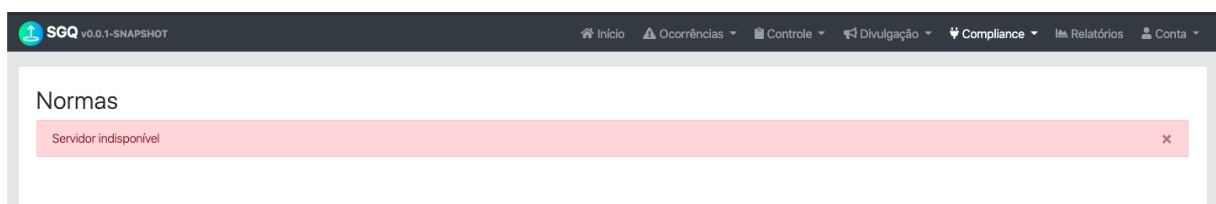


Figura 27: Evidência do cenário 5 - Mensagem de erro

● Cenário 6

Atributo de qualidade	Escalabilidade
Requisito de qualidade	A aplicação deve permitir o escalonamento horizontal
Preocupação	Garantir que a aplicação seja capaz de tratar grandes volumes de requisições, permitindo que recursos sejam liberados tão logo o volume diminua. Isso é particularmente importante em aplicações hospedadas em nuvem. (\$\$\$)
Cenário	Cenário 6
Ambiente	Funcionamento com carga acima do normal, com muitos usuários realizando operações de alto custo operacional.
Estímulo	Aumento do volume de requisições
Mecanismo	Implementação de uma arquitetura stateless, execução em ambiente de containers, utilização de ferramentas de orquestração

Medida de resposta	A aplicação permite o fácil escalonamento horizontal dos componentes de arquitetura para tratar o aumento de demanda
Riscos	Se a aplicação não é capaz de tratar volumes variáveis de requisições ela corre o risco de cair em descrédito pelos usuários, com o consequente desuso.
Pontos sensíveis	N/A
Trade-offs	N/A

```
lzkil@vostro ~/Projects/tcc_puc-minas/docker-compose develop ● docker ps
CONTAINER ID        NAMES                                PORTS                                STATUS
6789b045815b        sgq_sgq-app_1                       5701/udp, 8081/tcp                  Up 5 hours
d6966d12fa40        sgq_gateway-app_1                   5701/udp, 0.0.0.0:8080->8080/tcp     Up 27 hours
faee5b9131d6        sgq_jhipster-logstash_1             5044/tcp, 0.0.0.0:5000->5000/tcp, 0.0.0.0:5000->5000/udp, 9600/tcp Up 27 hours
4403ccb3f891        sgq_jhipster-console_1              0.0.0.0:5601->5601/tcp               Up 27 hours
eeed2b86d1d1        sgq_gateway-mariadb_1               0.0.0.0:3306->3306/tcp               Up 27 hours
bd693d8a610e        sgq_sgq-mariadb_1                   0.0.0.0:3307->3306/tcp               Up 27 hours
ba2567d8e741        sgq_jhipster-registry_1             0.0.0.0:8761->8761/tcp               Up 27 hours
42dc371bfa2f        sgq_jhipster-elasticsearch_1        9200/tcp, 9300/tcp                  Up 27 hours
b28ba1ea1b90        mock_consultoria-app_1              0.0.0.0:8083->8083/tcp               Up 27 hours
9dad7296d4eb        mock_normas-app_1                   0.0.0.0:8082->8082/tcp               Up About a minute
cf8fb7c02fc6        mock_normas-mariadb_1               0.0.0.0:3308->3306/tcp               Up 27 hours
7261966f1136        mock_consultoria-mariadb_1          0.0.0.0:3309->3306/tcp               Up 27 hours

lzkil@vostro ~/Projects/tcc_puc-minas/docker-compose develop ●
lzkil@vostro ~/Projects/tcc_puc-minas/docker-compose develop ● docker-compose --project-name sgq up -d --scale sgq-app=2
sgq_sgq-mariadb_1 is up-to-date
sgq_jhipster-registry_1 is up-to-date
sgq_jhipster-elasticsearch_1 is up-to-date
sgq_gateway-mariadb_1 is up-to-date
sgq_jhipster-logstash_1 is up-to-date
Starting sgq_sgq-app_1 ... done

sgq_jhipster-console_1 is up-to-date
Starting sgq_jhipster-import-dashboards_1 ... done
Creating sgq_sgq-app_2 ... done

lzkil@vostro ~/Projects/tcc_puc-minas/docker-compose develop ● docker ps
CONTAINER ID        NAMES                                PORTS                                STATUS
e70f06614a06        sgq_sgq-app_2                       5701/udp, 8081/tcp                  Up 2 seconds
6789b045815b        sgq_sgq-app_1                       5701/udp, 8081/tcp                  Up 5 hours
d6966d12fa40        sgq_gateway-app_1                   5701/udp, 0.0.0.0:8080->8080/tcp     Up 27 hours
faee5b9131d6        sgq_jhipster-logstash_1             5044/tcp, 0.0.0.0:5000->5000/tcp, 0.0.0.0:5000->5000/udp, 9600/tcp Up 27 hours
4403ccb3f891        sgq_jhipster-console_1              0.0.0.0:5601->5601/tcp               Up 27 hours
4e3cb28d2428        sgq_jhipster-import-dashboards_1    0.0.0.0:3306->3306/tcp               Up 2 seconds
eeed2b86d1d1        sgq_gateway-mariadb_1               0.0.0.0:3307->3306/tcp               Up 27 hours
bd693d8a610e        sgq_sgq-mariadb_1                   0.0.0.0:3307->3306/tcp               Up 27 hours
ba2567d8e741        sgq_jhipster-registry_1             0.0.0.0:8761->8761/tcp               Up 27 hours
42dc371bfa2f        sgq_jhipster-elasticsearch_1        9200/tcp, 9300/tcp                  Up 27 hours
b28ba1ea1b90        mock_consultoria-app_1              0.0.0.0:8083->8083/tcp               Up 27 hours
9dad7296d4eb        mock_normas-app_1                   0.0.0.0:8082->8082/tcp               Up About a minute
cf8fb7c02fc6        mock_normas-mariadb_1               0.0.0.0:3308->3306/tcp               Up 27 hours
7261966f1136        mock_consultoria-mariadb_1          0.0.0.0:3309->3306/tcp               Up 27 hours

lzkil@vostro ~/Projects/tcc_puc-minas/docker-compose develop ●
```

Uma instância do microserviço

Comando de scaling

Duas instâncias!

Figure 28: Evidência do cenário 6 - escalonamento horizontal

Evidências

A arquitetura apresentou o comportamento esperado, permitindo o rápido e fácil escalonamento horizontal do microserviço de SGQ. Essa operação poderia ser realizada para qualquer outro microserviço ou até mesmo para o gateway da aplicação. Isso só é possível pois a arquitetura foi concebida para ser totalmente stateless.

6.4. Resultado

De forma geral a arquitetura de microserviços é uma boa escolha para a implementação do SGQ. Suas vantagens para o projeto podem ser listadas facilmente:

- Facilita a construção e a manutenção de aplicações, pois tem por premissa que os serviços têm escopo bem definido e isolado;
- Facilita a implantação de novas funcionalidades, pois não é necessário derrubar a aplicação inteira para fazer um novo deploy;
- Permite que surjam novas organizações nas equipes (times, squads, etc);
- Permite um nível elevado de automação nos processos de construção e entrega;
- Permite a utilização de diversas tecnologias simultaneamente;
- Permite a escalabilidade horizontal, além da escalabilidade automática;
- Permite uma maior produtividade do desenvolvedor, que agora precisa conhecer apenas um subset do negócio;

Contudo, a escolha desta abordagem arquitetural deve levar em consideração alguns pontos não técnicos relevantes:

- A empresa deve entender o projeto como uma jornada e não apenas como um cronograma associado a um produto. Deve haver apoio total dos níveis hierárquicos superiores. Gestores devem estar cientes dos riscos, dos desafios e dos ganhos dessa jornada;

- Deve haver no projeto uma equipe madura nas habilidades de devops, pois esta arquitetura demanda muita automação, monitoramento e melhoria contínua. Caso não exista, ela deve ser formada;
- O escopo do projeto deve ser definido cuidadosamente, bem como a equipe que estará envolvida nele. Escopos muito extensos aumentam as chances de insucesso. Indivíduos multiplicadores devem ser selecionados e capacitados.

Mais especificamente sobre o protótipo construído e descrito neste projeto, um ponto de melhoria arquitetural interessante é a integração com os serviços externos de consultoria. O mecanismo atual se baseia no *polling* periódico da API da consultoria, de modo a obter as análises finalizadas. Uma forma mais eficiente de obter as análises seria por meio de *web hooks*. Neste esquema o SGQ não faria polling, mas seria acionado pela API da consultoria quando uma análise fosse realizada.

Outra melhoria que poderia ser implementada seria a adição de um *message broker* para comunicação assíncrona entre microserviços (RabbitMQ, Kafka, etc). No desenho atual isso não é um requisito pois há somente um microserviço (SGQ), mas em cenários com muitas instâncias de serviços distintos esta abordagem traz diversos ganhos.

7. Conclusão

O presente projeto teve por motivação a avaliação de uma opção de arquitetura para o problema proposto: o sistema de gestão da qualidade de uma empresa do setor automotivo. A arquitetura escolhida, baseada em microserviços, mostrou-se plenamente capaz de satisfazer os requisitos não funcionais da aplicação, conforme evidenciado anteriormente. Pôde-se verificar os pontos fortes e os pontos de atenção inerentes a esta arquitetura, bem como as características de várias ferramentas utilizadas na implementação. **Em resumo, consideramos que os objetivos do projeto foram atingidos.**

APÊNDICE I - Links

Repositório do projeto: https://github.com/lzkill/tcc_puc-minas

Vídeo de apresentação do projeto: <https://youtu.be/-Ogd5sOipvc>

APÊNDICE II – Diagramas de casos de uso



Figura 29: Modelo de casos de uso - Administração

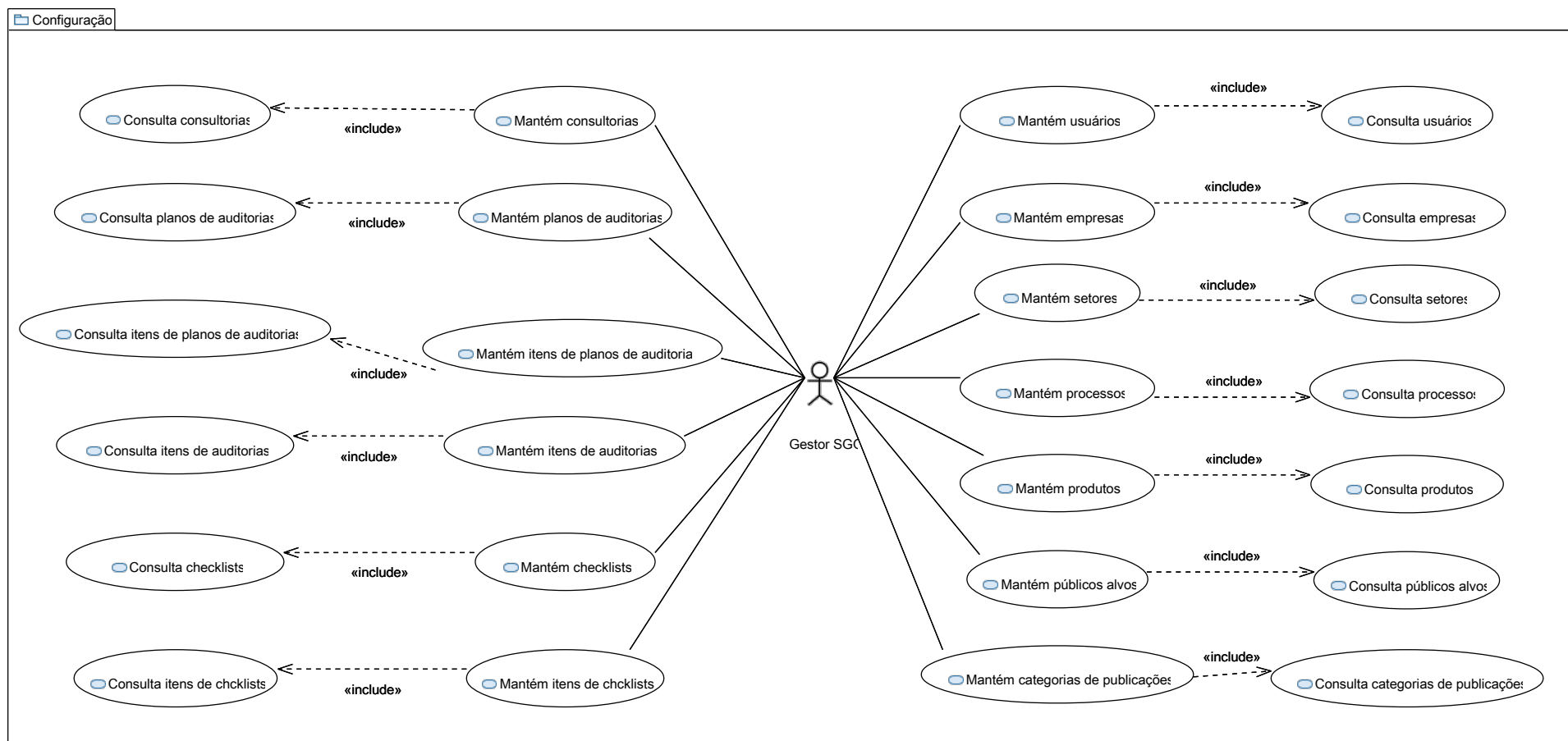
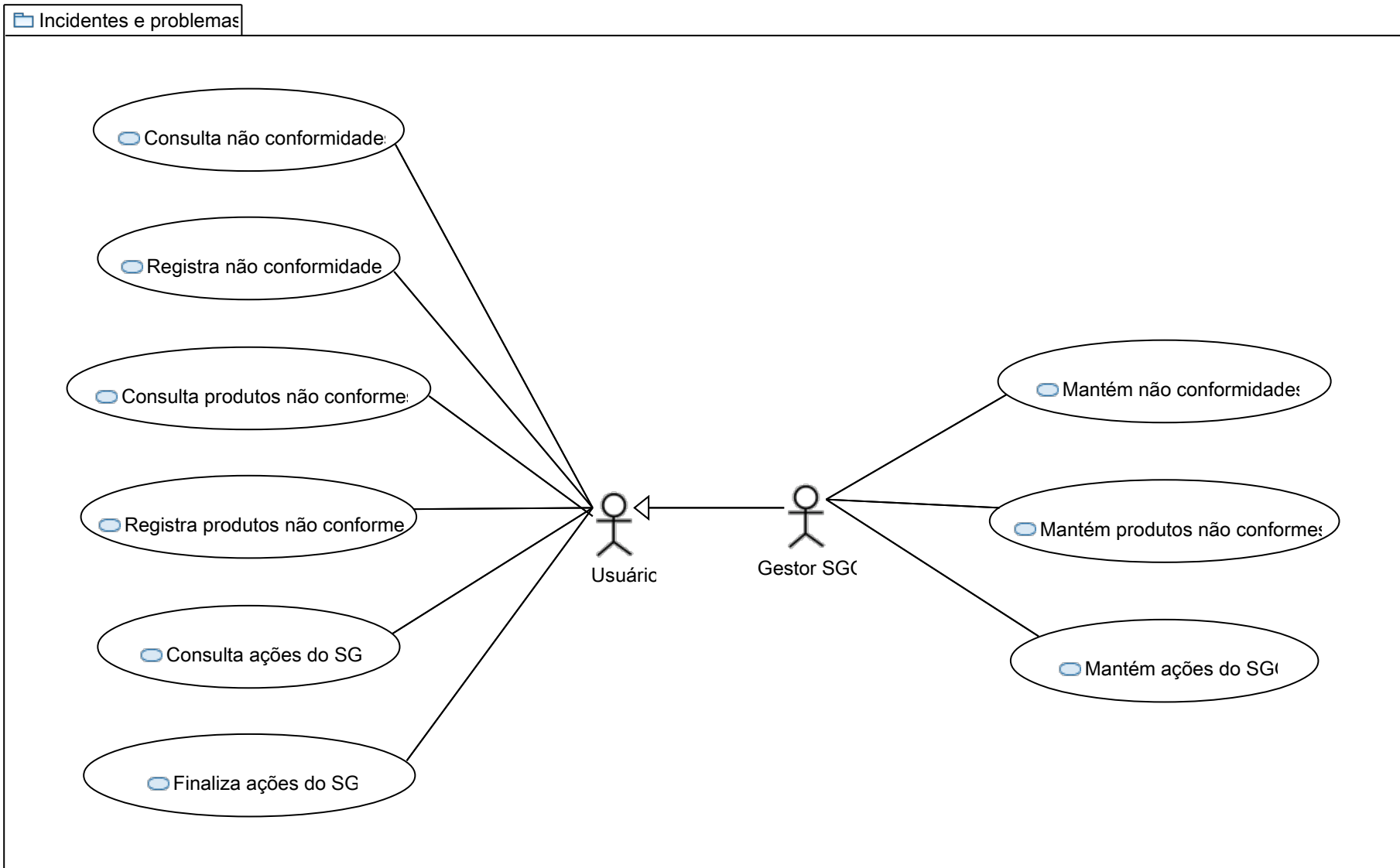


Figura 30: Modelo de casos de uso - Configuração



Figura

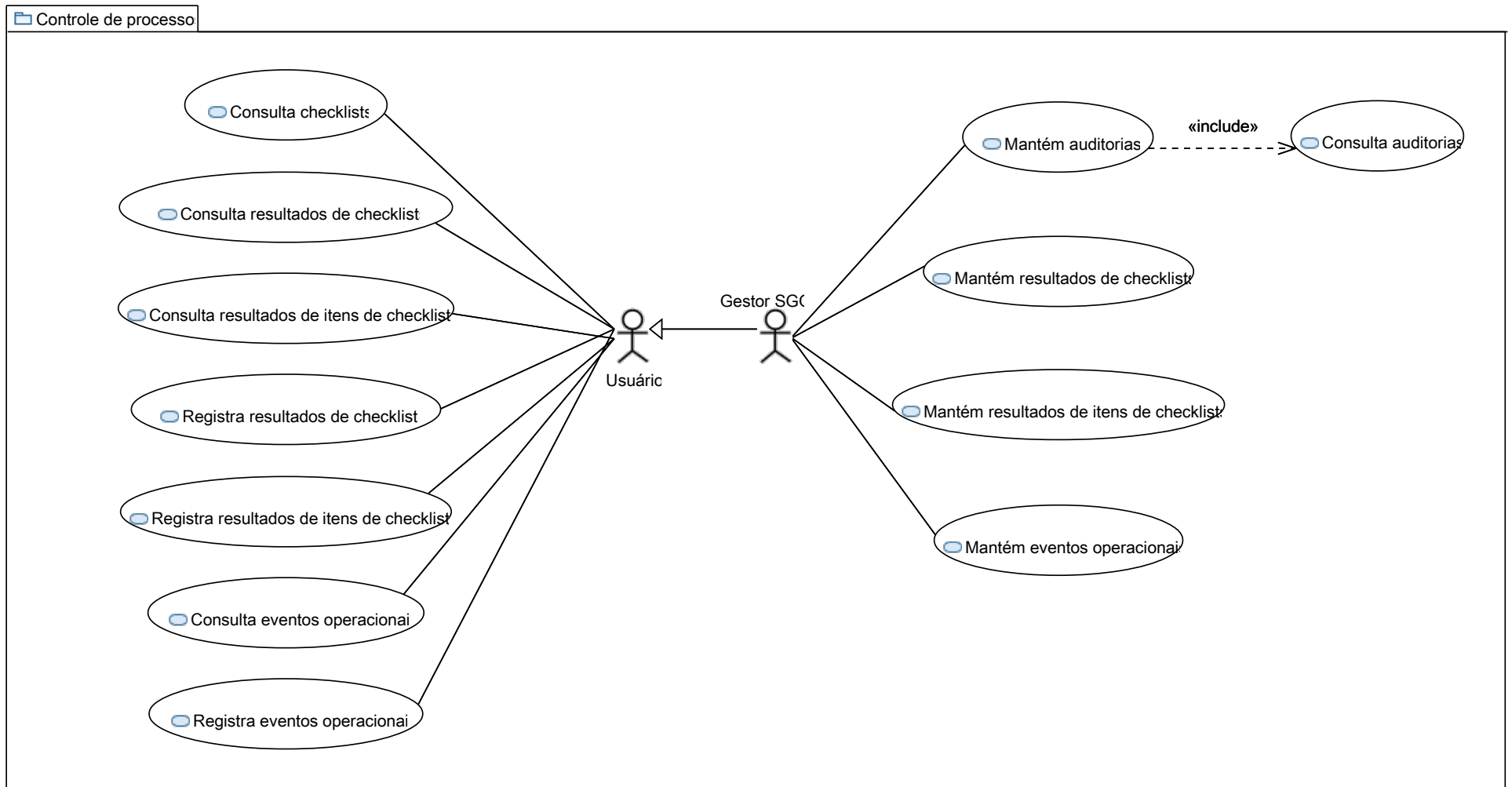


Figura 32: Modelo de casos de uso - Controle de processos

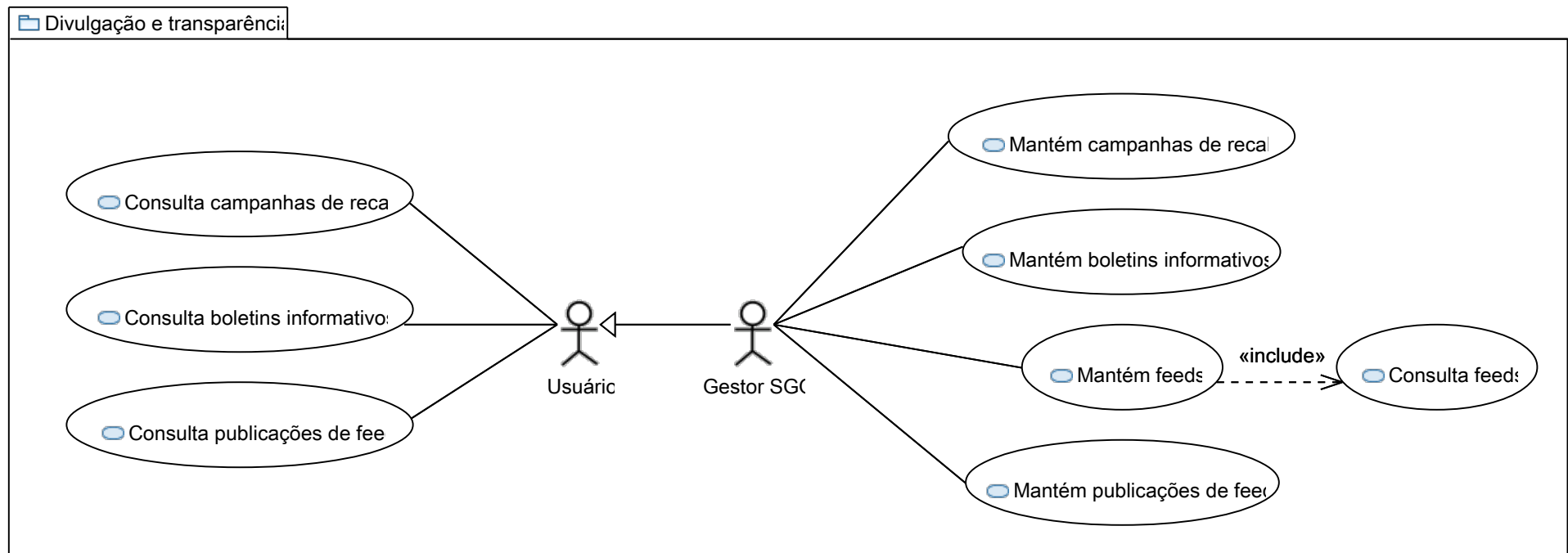


Figura 33: Modelo de casos de uso - Divulgação e transparência

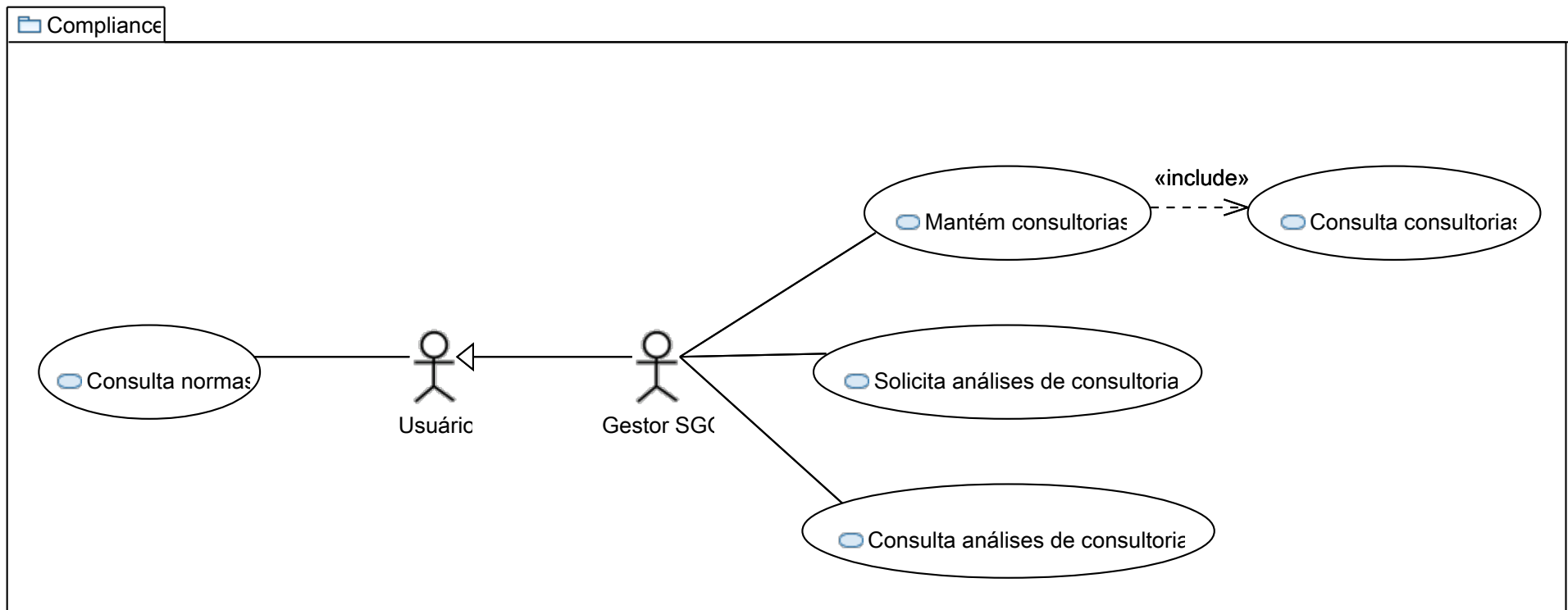


Figura 34: Modelo de casos de uso - Compliance

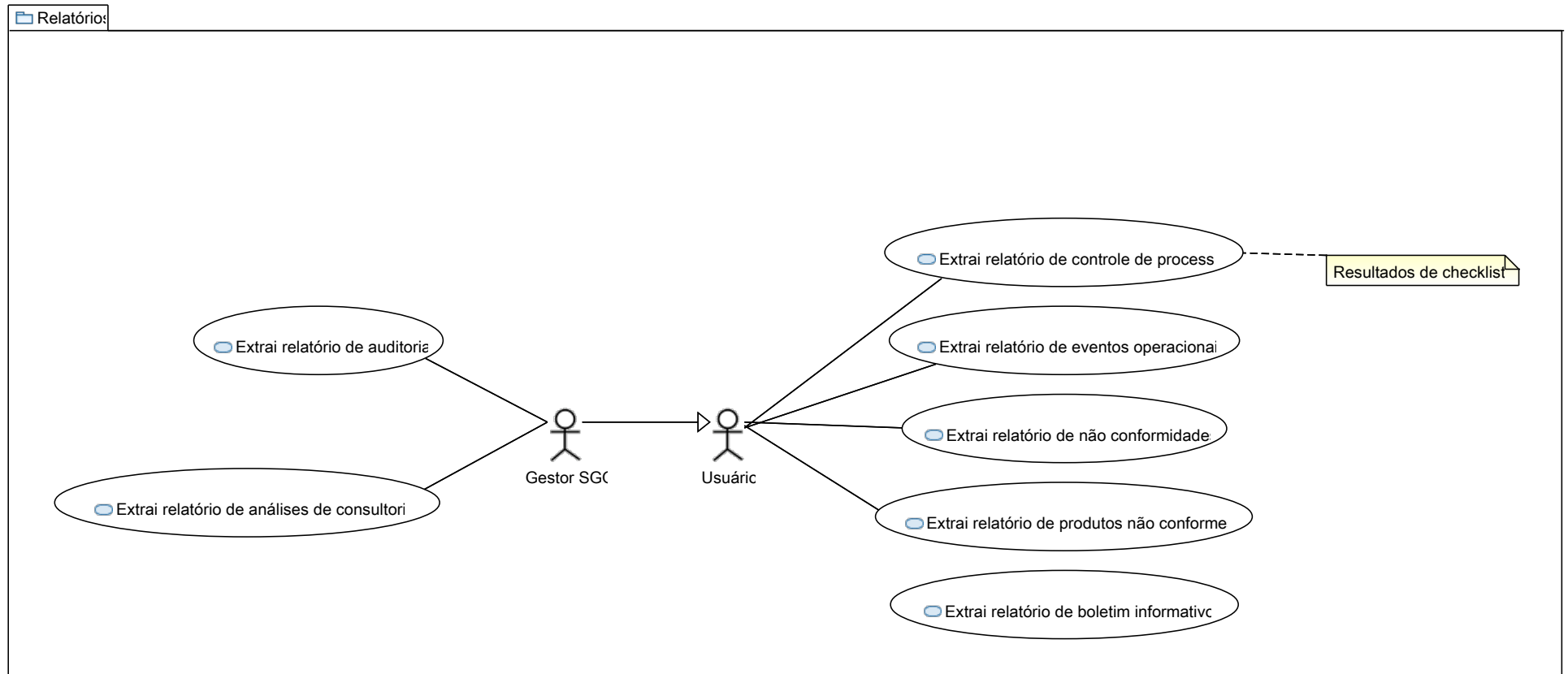


Figura 35: Modelo de casos de uso - Relatórios

CHECKLIST PARA VALIDAÇÃO DOS ITENS E ARTEFATOS DO TRABALHO

Nº	Item a ser cumprido	Sim	Não	Não se aplica	OBS
Completeza do documento					
1	Todos os elementos iniciais do documento (capa, contracapa, resumo, sumário...) foram definidos?	X			
2	Os objetivos do trabalho (objetivos gerais e pelo menos três específicos) foram especificados?	X			
3	Os requisitos funcionais foram listados e priorizados?	X			
4	Os requisitos não funcionais foram listados e identificados usando o estilo estímulo-resposta?	X			
5	As restrições arquiteturais foram definidas?	X			
6	Os mecanismos arquiteturais foram identificados?	X			
7	Um diagrama de caso de uso foi apresentado junto com uma breve descrição de cada caso de uso?			X	Apenas os casos de uso utilizados para avaliação da arquitetura foram descritos
8	Um modelo de componentes e uma breve descrição de cada componente foi apresentada?	X			
9	Um modelo de implantação e uma breve descrição de cada elemento de hardware foi apresentada?	X			
10	Prova de conceito: uma descrição da implementação foi feita?	X			
11	Prova de conceito: as tecnologias usadas foram listadas?	X			
12	Prova de conceito: os casos de uso e os requisitos não funcionais usados para validar a arquitetura foram listados?	X			

13	Prova de conceito: os detalhes da implementação dos casos de uso (telas, características, etc) foram apresentadas?	X			
14	Prova de conceito: foi feita a implantação da aplicação e indicado como foi feita e onde está disponível?	X			
15	As interfaces e/ou APIs foram descritas de acordo com um modelo padrão?	X			
16	Avaliação da arquitetura: foi feita uma breve descrição das características das abordagens da proposta arquitetural?	X			
17	Avaliação da arquitetura: Os atributos de qualidade e os cenários onde eles seriam validados foram apresentados?	X			
18	Avaliação da arquitetura: uma avaliação com as evidências dos testes foi apresentada?	X			
19	Os resultados e a conclusão foram apresentados?	X			
20	As referências bibliográficas foram listadas?		X		
21	As URLs com os códigos e com o vídeo da apresentação da POC foram listadas?	X			

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