

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

Alessandro Vasconcelos de Oliveira

SISTEMA DE GESTÃO DE QUALIDADE PARA RAMO AUTOMOTIVO

Belo Horizonte

2020

Alessandro Vasconcelos de Oliveira

SISTEMA DE GESTÃO DE QUALIDADE PARA RAMO AUTOMOTIVO

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

Orientador(a): MSc. Luiz Alberto Ferreira Gomes

Belo Horizonte

2020

*Dedico este trabalho a Deus, autor e consumidor da minha fé, que me
permitiu cumprir mais esta etapa em minha vida.
Dedico também à minha esposa Andréa e ao meu filho Josué, meus
grandes incentivadores e companheiros de todos os momentos.*

RESUMO

Este projeto aborda uma proposta arquitetural para um sistema de gestão da qualidade do ramo automotivo que utiliza recursos provenientes de serviços Web de terceiros e que pode ser acessado por meio da Internet utilizando dispositivos móveis ou desktop. Gestão da qualidade é um conceito cada vez mais exigido das empresas e muitas dessas exigências são provenientes de normas legais. Fazer a gestão correta da qualidade exige sistemas capazes de se comunicar com sistemas provenientes das mais diversas tecnologias, além de ser de fácil manutenção para adaptar-se as constantes mudanças que ocorrem no mercado. Um ponto importante é o correto e eficiente controle das atividades que devem ser executadas diariamente, além da rápida resolução dos problemas ocorridos durante estas atividades. Este projeto apresenta uma solução arquitetural utilizando microsserviços do lado servidor e um projeto SPA (*Single Page Application*) do lado cliente, juntamente com uma prova de conceito para validação dos requisitos arquiteturais mais relevantes.

Palavras-chave: arquitetura de software, projeto de software, requisitos arquiteturais, gestão da qualidade, microsserviços, SPA.

SUMÁRIO

1. Objetivos do trabalho	6
2. Descrição geral da solução	7
2.1. Apresentação do problema.....	7
2.2. Descrição geral do software (Escopo)	7
3. Definição conceitual da solução	8
3.1. Requisitos Funcionais	8
3.2 Requisitos Não-Funcionais	11
3.3. Restrições Arquiteturais	15
3.4. Mecanismos Arquiteturais	15
4. Modelagem e projeto arquitetural.....	16
4.1. Modelo de casos de uso	16
4.2. Descrição resumida de casos de uso	19
4.3. Modelo de componentes	29
4.4. Modelo de implantação	31
4.5. Modelo de dados (opcional).....	33
5. Prova de Conceito (POC) / protótipo arquitetural	34
5.1. Implementação e Implantação	34
5.2 Interfaces/ APIs.....	37
6. Avaliação da Arquitetura.....	45
6.1. Análise das abordagens arquiteturais	45
6.2. Cenários	45
6.3. Avaliação	47
6.4. Resultado.....	60
7. Conclusão.....	61
REFERÊNCIAS.....	62
APÊNDICES.....	63

1. Objetivos do trabalho

O propósito deste trabalho é apresentar uma proposta arquitetural para o desenvolvimento de uma plataforma de gestão da qualidade para indústrias do ramo automotivo. O sistema visa fornecer mecanismos para um melhor controle dos processos envolvidos na fabricação de produtos automotivos em conformidade com as normas nacionais e internacionais criadas para o setor, auxiliando no cadastro de incidentes e problemas relacionados a não conformidades encontradas durante esses processos.

Os objetivos específicos são:

1. Criar um módulo de cadastro de incidentes e problemas relacionados a não conformidades dos produtos da empresa, tais como falhas encontradas em veículos e suas peças (insumos).
2. Criar módulo para controle dos processos automotivos que permita a gestão de todas as etapas envolvidas na gestão da qualidade dos produtos produzidos e processos envolvidos na atividade automotiva. Este módulo deve permitir a comparação da qualidade com padrões pré-estabelecidos, permitindo que os problemas encontrados sejam registrados via integração com o módulo de cadastro de incidentes e problemas.
3. Criar módulo de divulgação e transparência para tornar as informações de qualidade disponíveis, tanto para o público interno quanto externo. As informações deste módulo são provenientes do módulo para cadastro de incidentes e também do módulo para controle de processos.
4. Criar módulo de inteligência do negócio voltado para gerar informações estratégicas de negócio, com foco na qualidade.
5. Criar módulo de *compliance* que deve se integrar aos sistemas externos de Gestão de Normas e de Consultorias e Assessorias. Este módulo visa garantir a aderência da empresa às normas do setor automotivo, tanto nacionais quanto internacionais, bem como aprimorar os controles de governança corporativa existentes.
6. Criar módulo de relatório de acompanhamento que permita a geração de relatórios e consultas sob demanda, para acompanhar a real situação das atividades desenvolvidas pela empresa.

2. Descrição geral da solução

Esta seção se destina a descrever a solução arquitetural definida para o Sistema de Gestão de Qualidade para o Ramo Automotivo.

2.1. Apresentação do problema

Atualmente o termo qualidade é mencionado em vários lugares onde haja atividade humana e nos processos produtivos do ramo automotivo isso tem se tornado uma exigência cada vez mais acentuada. Muitos são os requisitos de qualidade exigidos por normas nacionais e internacionais, e o seu cumprimento exige muita organização e disciplina por parte das empresas, além de conhecimento profundo de muitas normas que estão em constante mudança e evolução.

Outro ponto esperado hoje em dia é a transparência em relação aos problemas encontrados, assim como a resposta rápida aos problemas e incidentes reportados. Isso exige que a empresa tenha mecanismos que possibilitem uma precisa comunicação, tanto interna (quando a comunicação é feita entre os diversos setores da empresa), quando externa (quando a comunicação é feita com seus fornecedores de produtos e serviços, assim como com seus clientes).

O gerenciamento das atividades das empresas automotivas e a correta gestão da qualidade exige sistemas flexíveis e com auto disponibilidade, que permitam o acesso a inúmeros usuários espalhados ao redor do mundo, além de possuir integração com sistemas externos e suas bases de dados.

2.2. Descrição geral do software (Escopo)

O desenvolvimento deste software tem por objetivo prover uma ferramenta integrada de gestão e controle da qualidade para empresas do setor automotivo, possibilitando que as atividades desenvolvidas pela empresa sejam registradas, monitoradas e comparadas com normas nacionais e internacionais de qualidade.

As normas estarão disponíveis no sistema através de integração com sistemas externos, possibilitando que a informação esteja sempre atualizada. Outro ponto proveniente de sistemas externos é a integração com empresas terceirizadas de consultorias e assessorias.

As informações geradas pelo sistema ficarão disponíveis a seus interessados através de controle de acesso, em que cada informação estará acessível apenas aos seus respectivos responsáveis, garantindo que informações sensíveis sejam protegidas contra acesso não autorizado.

O controle de acesso será feito através dos seguintes perfis de usuário:

- **Operador:** responsável por preencher o workflow de atividades diárias, cadastrando as paradas ocorridas durante o processo.
- **Engenheiro:** responsável por cadastrar os workflows de atividades diárias e acompanhá-los, resolvendo os problemas ocorridos durante a execução dos mesmos. Também é de responsabilidade do Engenheiro o cadastramento de Recall e a geração de notificações técnicas para serem disponibilizadas publicamente.
- **Gerente de Qualidade:** responsável pelo cadastramento de não conformidades, acompanhando sua resolução. É de responsabilidade do Gerente de Qualidade a aprovação da resolução de uma não conformidade ou incidente cadastrado.
- **Gestor:** responsável pelas tomadas de decisões estratégicas da empresa. Para isso, terá acesso aos módulos de relatórios e BI. O Gestor terá a atribuição de aprovar tanto um Recall quanto as notificações técnicas emitidas por um Engenheiro antes que as mesmas sejam disponibilizadas para o público.

3. Definição conceitual da solução

3.1. Requisitos Funcionais

- **Módulo de Cadastro de Incidentes e Problemas**
 - O sistema deve permitir o cadastro de uma RNC (Registro de Não conformidade). O cadastro será feito por um gerente de qualidade após o registro de uma parada no Módulo de Controle de Processos Automotivos. Um gerente de qualidade também poderá cadastrar não conformidades que não estejam relacionadas as paradas ocorridas, permitindo o cadastro de melhorias ou antecipação de eventuais problemas.

- Os usuários com perfil Engenheiro ou Gerente de Qualidade podem listar os RNC cadastrados no sistema, bem como ver os detalhes de cada RNC.
- O sistema deve permitir que um Engenheiro ou Gerente de Qualidade possa alterar um RNC para corrigir possíveis erros no cadastro ou para atualizar informações.
- Os RNC cadastrados devem ser resolvidos por um Engenheiro. A resolução precisa ser confirmada por um usuário com perfil Gerente de Qualidade. Após a confirmação da resolução, o sistema deve enviar uma notificação de resolução de RNC através do Serviço de Notificação, caso a resolução seja aprovada. Em caso de não aprovação, a RNC precisará ser novamente resolvida por um Engenheiro.
- Após cadastrado, um RNC só poderá ser removido por um Gerente de Qualidade e o sistema deverá informá-lo que a remoção será feita de forma definitiva.

- **Módulo de Controle de Processos Automotivos**

- O sistema deve permitir que um Engenheiro cadastre um modelo (template) de workflow contendo uma lista de atividades diárias que deverão ser executadas por um Operador.
- O sistema deve permitir que Operadores e Engenheiros possam listar os modelos de workflow existentes, no entanto, apenas os engenheiros poderão ver os detalhes do mesmo.
- O sistema deve permitir que um Engenheiro possa alterar ou excluir um modelo de workflow.
- O sistema deve permitir que um Operador possa iniciar a execução de um workflow criando-o a partir de um modelo pré-cadastrado.
- O sistema deve permitir que o Operador preencha o andamento das atividades realizadas do workflow, sendo possível informar sobre paradas, caso ocorram. O sistema deve emitir uma notificação após o cadastramento de um incidente ou problema através do Serviço de Notificação.
- O sistema deve permitir que Operadores, Gerentes de Qualidade e Engenheiros possam listar os workflows em andamento.

- **Módulo de Divulgação e Transparência**

- O sistema deve permitir que um Engenheiro cadastre notificações, que podem ser notificações técnicas destinadas ao público geral ou campanhas de Recall.
- As notificações cadastradas devem ser aprovadas por um Gestor antes de ficarem disponíveis para o público. Após aprovação, o sistema deve emitir um comunicado ao público, mantendo a notificação acessível. Caso a notificação não seja aprovada, a mesma será cancelada pelo sistema, podendo ser reaberta por um Engenheiro.
- O sistema deve permitir que qualquer usuário possa exportar as notificações publicadas, podendo acessá-las no formato json ou xml.
- O sistema deve permitir que um Engenheiro possa remover uma notificação caso a mesma ainda não tenha sido aprovado por um Gestor.
- **Módulo de Inteligência do Negócio**
 - O sistema deve indicar os pontos de maior probabilidade de ocorrência de incidentes com base em simulações realizadas com dados históricos de falhas, possibilitando ações preventivas.
 - O sistema deve realizar simulações com base em dados provenientes de mídias sociais para identificar qual a aceitação dos produtos da empresa pelo público geral.
 - O sistema deve prover dashboards gerenciais para auxiliar os Gestores nas tomadas de decisão.
- **Módulo *Compliance***
 - O sistema deve prover acesso a um catálogo de normas nacionais e internacionais que forneçam base para a gestão da qualidade. Esse acesso se dará através de integração com um repositório de normas disponível em nuvem como um serviço (SaaS).
 - O sistema deve permitir a contratação de assessorias e consultorias terceirizadas, visando adequar a empresa às normas e padrões da sua área de negócio. Essa contratação deve ser por meio de integração com APIs de serviços de terceiros providas para esse fim.
- **Módulo de Relatórios de Acompanhamento**
 - O sistema deve prover um relatório de não conformidades contendo as seguintes informações sobre um RNC: número do RNC, atividade que

gerou o RNC, risco, causa e gravidade da não conformidade, além das ações necessárias para sua solução e status. O usuário deve ter a opção de trazer no relatório apenas as Não conformidades que ainda não foram resolvidas.

- O sistema deve prover um relatório das ações realizadas por cada usuário no sistema contendo data e hora da ação, o login do usuário e a ação realizada.
- O sistema deve prover um relatório das notificações emitidas pelo sistema contendo a data e a hora da aprovação, o nome do usuário que emitiu a notificação, o nome do usuário que aprovou, o título da notificação, o tipo da notificação e a data final de divulgação.

3.2 Requisitos Não-Funcionais

A seguir são apresentados os requisitos não funcionais do sistema descritos no padrão estímulo-resposta:

Segurança:

- O sistema deve apresentar altos padrões de segurança.

Estímulo	Acessar uma página privada do sistema pela URL sem estar logado no sistema
Fonte do Estímulo	Qualquer usuário não autenticado
Ambiente	Funcionamento, carga normal
Artefato	Qualquer módulo do sistema
Resposta	O sistema redireciona o usuário para a tela de autenticação do sistema
Medida da resposta	O sistema não permite acesso ao sistema sem autenticação e redireciona o usuário para uma tela de autenticação

Usabilidade:

- O sistema deve prover boa usabilidade.

Estímulo	O usuário navega pelo sistema tentando entender o funcionamento geral do sistema
-----------------	--

Fonte do Estimulo	Qualquer usuário logado
Ambiente	Funcionamento, carga normal
Artefato	Qualquer módulo do sistema
Resposta	O sistema fornece uma visão clara de sua utilidade e finalidade a partir de funcionalidades bem definidas e de fácil navegação
Medida da resposta	Identidade visual padronizada em todos os seus módulos, com fontes, cores e tamanhos coerentes e redimensionados por igual

- Ao navegar pelo sistema, o usuário deve ter conhecimento de todas as ações que ele pode executar e ter um *feedback* do que foi realizado.

Estimulo	O usuário executa as funcionalidades que deseja e o sistema informa se a operação foi realizada com sucesso ou não
Fonte do Estimulo	Qualquer usuário logado
Ambiente	Funcionamento, carga normal
Artefato	Qualquer módulo do sistema
Resposta	Após a execução de uma funcionalidade, o sistema informa o resultado de sua execução, sendo bem sucedida ou não
Medida da resposta	Mensagens padronizadas que informem com clareza se a operação foi bem sucedida ou que apresentem todos os problemas ocorridos em caso de insucesso.

Acessibilidade:

- O sistema deve possuir design responsivo que funcione bem em ambientes Web e móveis.

Estimulo	O usuário acessa uma funcionalidade do sistema usando um dispositivo móvel (celular ou tablet)
Fonte do Estimulo	Qualquer usuário logado
Ambiente	Funcionamento, carga normal
Artefato	Qualquer módulo do sistema
Resposta	A usabilidade do sistema deve ser similar, independente do dispositivo usado
Medida da resposta	O funcionamento do sistema deve apresentar o mesmo resultado,

	independente de qual dispositivo está sendo usado pelo usuário
--	--

Interoperabilidade:

- O sistema deve comunicar-se com serviços externos de terceiros de forma transparente ao usuário.

Estimulo	Consulta ao catálogo de normas nacionais e internacionais disponível através de serviço externo
Fonte do Estimulo	Consulta a API de <i>Compliance</i>
Ambiente	Funcionamento, carga normal
Artefato	Módulo de <i>Compliance</i>
Resposta	O sistema externo de normas deve responder a requisição com sucesso
Medida da resposta	O sistema deve apresentar as normas disponíveis no serviço externo

Desempenho:

- O sistema deve ter alto desempenho mesmo com alto volume de usuários simultâneos.

Estimulo	Aumento do volume de requisições aos serviços que compõem o sistema devido ao maior volume de usuários simultâneos
Fonte do Estimulo	Aumento na quantidade de usuários simultâneos
Ambiente	Funcionamento, carga acima do normal
Artefato	Qualquer módulo do sistema
Resposta	O usuário utiliza o sistema normalmente
Medida da resposta	O cluster de instâncias dos microserviços detecta automaticamente o aumento de requisições e ativa novas instâncias afim de compensar o aumento da demanda

- O sistema deve ter alto desempenho no acesso aos dados.

Estimulo	Recuperação de dados
Fonte do Estimulo	Usuário solicita a mesma informação ao sistema
Ambiente	Funcionamento, carga normal

Artefato	Qualquer módulo do sistema
Resposta	O usuário recebe os dados solicitados a partir da segunda requisição mais rapidamente
Medida da resposta	Ao recuperar dados o sistema mantém estes em memória afim de otimizar novas consultas

Confiabilidade:

- O sistema deve ter alta disponibilidade, operando ininterruptamente ou com poucas paradas.

Estimulo	Monitoramento do sistema a cada 30 segundos para verificar quedas
Fonte do Estimulo	Funcionalidade de <i>Healthy Check</i>
Ambiente	Funcionamento, carga normal
Artefato	Qualquer módulo do sistema
Resposta	Um desenvolvedor acessa o <i>Elmah.io</i> e verifica que o sistema não apresenta quedas constantes
Medida da resposta	Ao detectar uma indisponibilidade do <i>front-end</i> , de uma das APIs do sistema, ou de um dos banco de dados utilizados, o sistema envia a informação para o serviço de monitoramento do <i>Elmah.io</i>

- O sistema deve ser tolerante a falhas e informar a ocorrência de erros.

Estimulo	Cadastrar novo incidente no sistema
Fonte do Estimulo	Gerente de Qualidade autenticado no sistema
Ambiente	Funcionamento, carga normal
Artefato	Módulo de Cadastro de Incidentes e Problemas
Resposta	O sistema apresenta mensagem de falha caso ocorra erro
Medida da resposta	Ao detectar um erro durante o cadastro (como a falta de conexão com o banco de dados) o sistema apresenta uma mensagem informando a ocorrência do erro

- O sistema deve recuperar-se de falhas inesperadas.

Estimulo	Cadastrar novo incidente no sistema
-----------------	-------------------------------------

Fonte do Estimulo	Gerente de Qualidade autenticado no sistema
Ambiente	Funcionamento, carga normal
Artefato	Módulo de Cadastro de Incidentes e Problemas
Resposta	O sistema mantém o estado correto após ocorrência de falha
Medida da resposta	Ao detectar um erro durante o cadastro (como a falta de conexão com o banco de dados) o sistema mantém o último estado correto anterior a ocorrência falha, voltando ao seu funcionamento normal após a falha

3.3. Restrições Arquiteturais

- O sistema deve ser desenvolvido utilizando .NET Core.
- O sistema deve ser dividido em módulos a fim de facilitar a implantação e manutenção.
- O sistema deve realizar integração com APIs de terceiros.
- O sistema deve ser implantado utilizando recursos de integração contínua.
- O sistema deve conter automação de testes no pipeline de integração contínua.
- O sistema deve permitir a sua implantação em nuvem ou *on-prmise*.
- O sistema deve ser responsivo, de forma que possa ser usando em aparelhos menores, como celulares ou tablets.

3.4. Mecanismos Arquiteturais

Mecanismo de Análise	Mecanismo de Design	Mecanismo de Implementação
Comunicação entre processos	Contêiner Web e Aplicação	Docker
Integração com outros módulos e/ou sistemas	Interfaces utilizando JSON	ASP.NET Web API
Front-end	Interface de interação do usuário com o sistema	Angular, Bootstrap e PowerBI
Back-end	APIs REST de serviços	ASP.NET Web API e .NET Framework Core
Versionamento	Versionamento do código-fonte da aplicação	Git
Repositório de código	Repositório do código-fonte da aplicação	GitHub

Log	Framework de Log	Log4net, Elmah.io
Cache	Framework de cache	Redis
Build	Geração de artefatos para servidor e aplicação	MSBuild e Nuget
Deploy	Deploy da aplicação no servidor e testes automatizados	Azure DevOps
CI/CD	Pipeline de integração contínua	Azure DevOps
Automação de testes	Execução de testes unitários automatizados	MSTest
Autenticação e Autorização	Verificação de credenciais e tentativas de conexão	AspNetCore Identity e JWT
Persistência	Banco de dados relacional e acesso a dados	Azure SQL e Entity Framework Core
Documentação	Documentação das APIs de serviço	Swagger
Alta disponibilidade	Balanceamento de carga dos serviços	Azure Kubernetes
Relatórios	Gerador de relatórios	Microsoft PowerBI

4. Modelagem e projeto arquitetural

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

4.1. Modelo de casos de uso

O diagrama de caso de uso oferece uma visão global dos casos de uso e dos atores que dele participam. Para uma melhor análise arquitetural do projeto, separei os casos de uso por módulos de acordo com os requisitos informados acima.

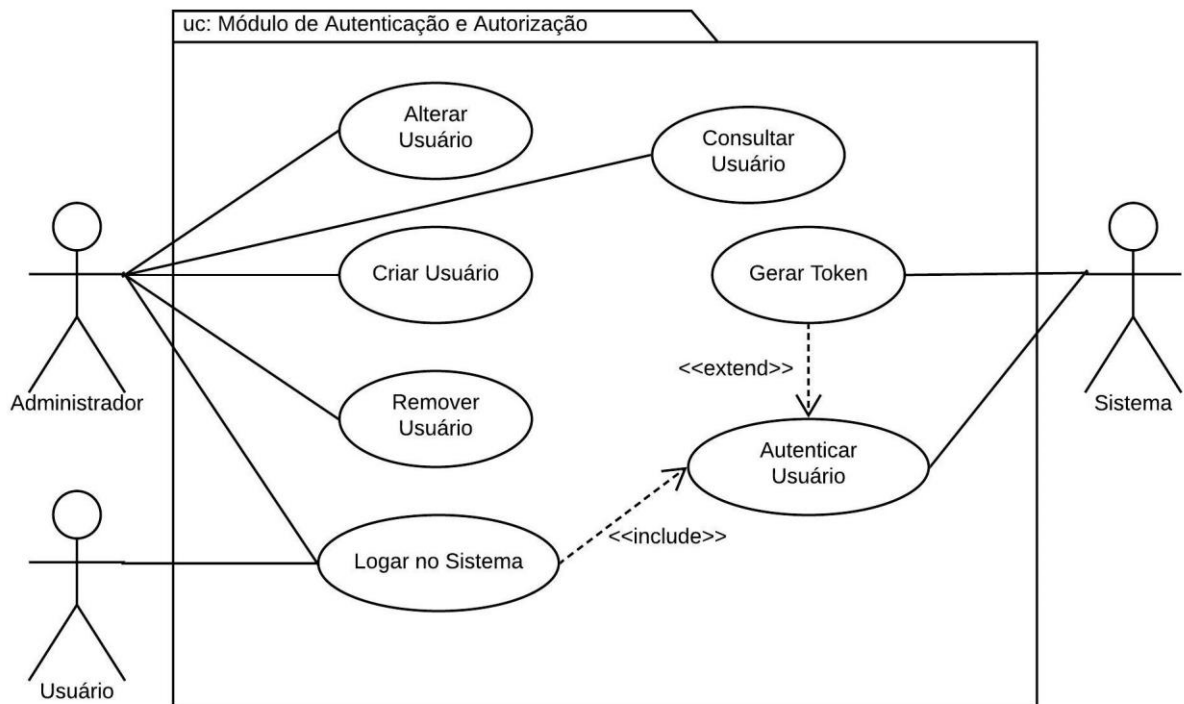


Figura 1 - Diagrama de Casos de Uso: Módulo de Autenticação e Autorização

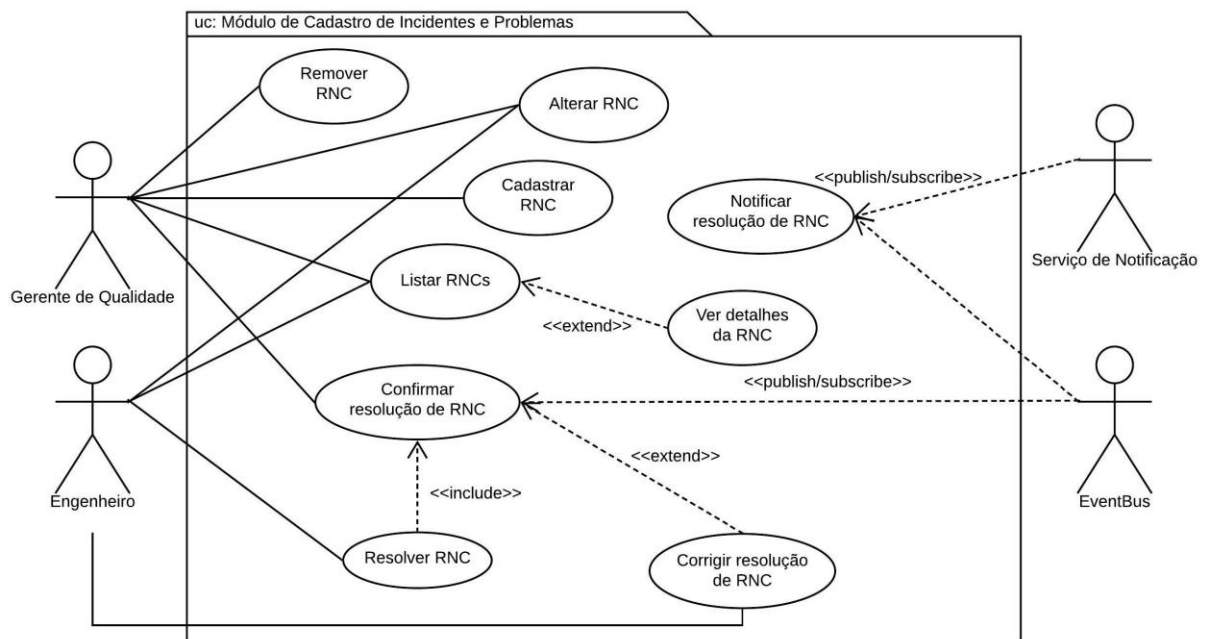


Figura 2 - Diagrama de Casos de Uso: Módulo de Cadastro de Incidentes e Problemas

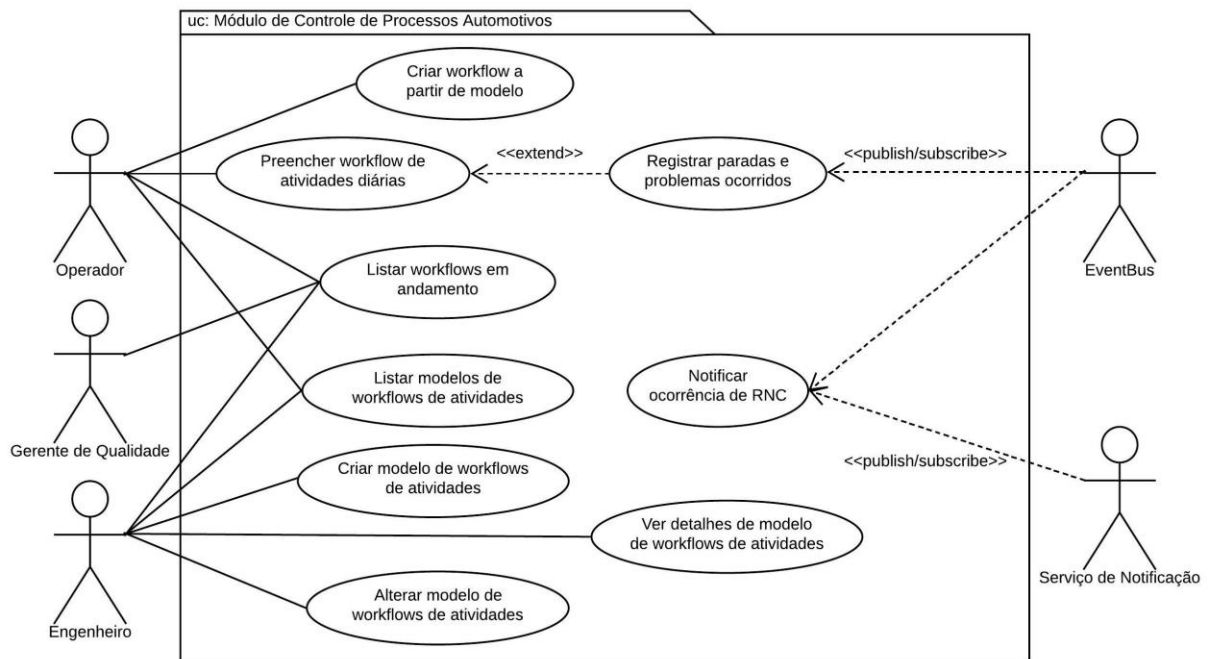


Figura 3 - Diagrama de Casos de Uso: Módulo de Controle de Processos Automotivos

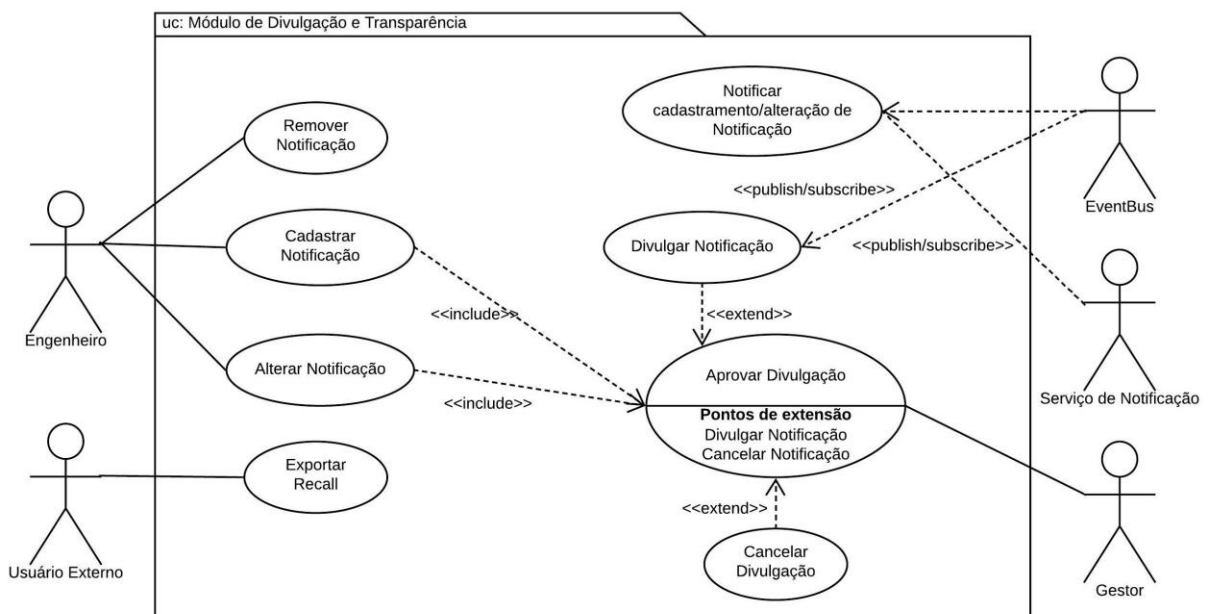


Figura 4 - Diagrama de Casos de Uso: Módulo de Divulgação e Transparência

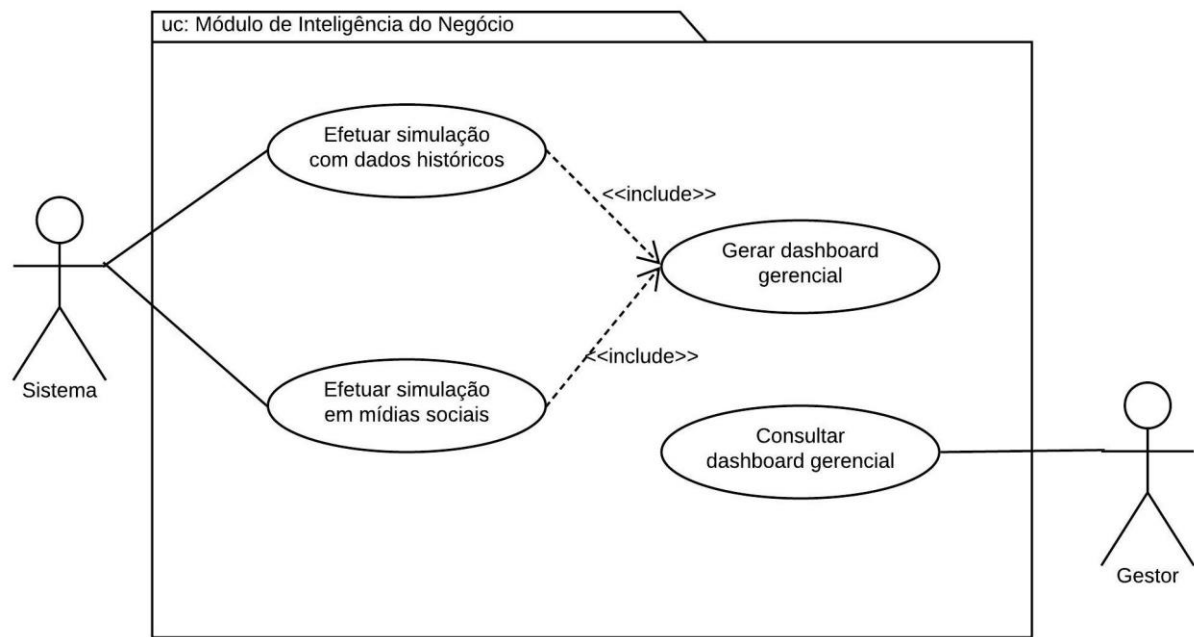


Figura 5 - Diagrama de Casos de Uso: Módulo de Inteligência do Negócio

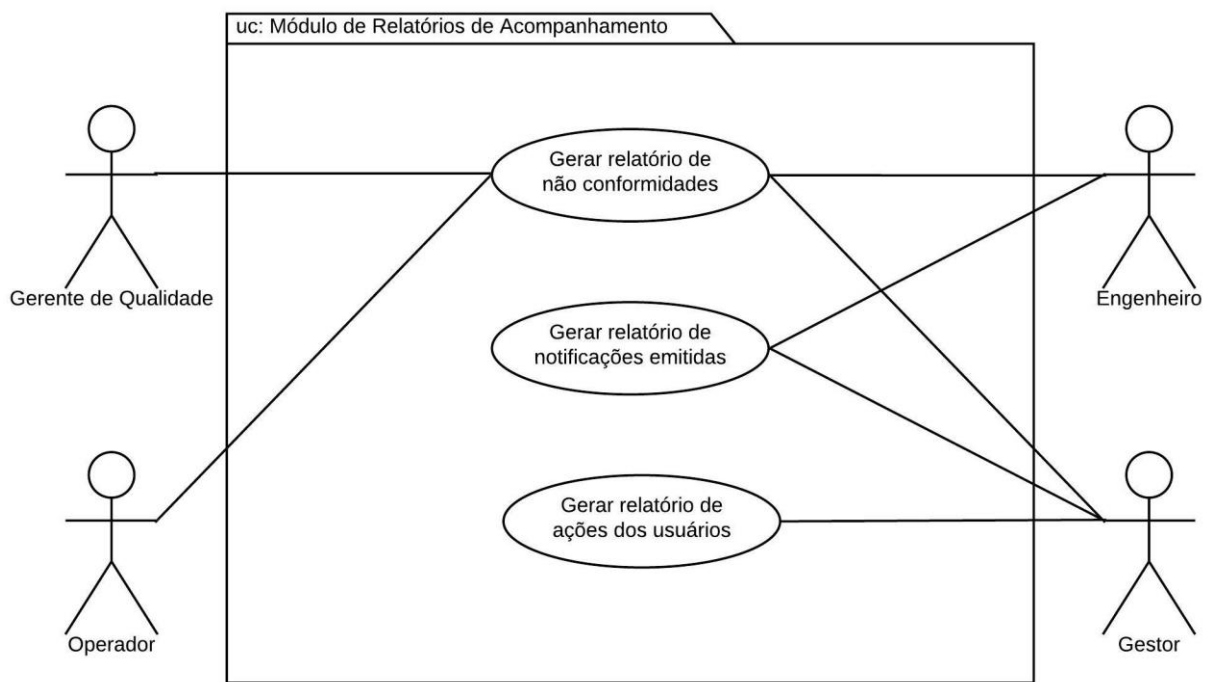


Figura 6 - Diagrama de Casos de Uso: Módulo de Relatórios de Acompanhamento

4.2. Descrição resumida de casos de uso

Caso de uso UC001: cadastrar RNC (Registro de Não conformidade)

Descrição resumida:

Este caso de uso deve permitir ao Gerente de Qualidade o cadastramento de um incidente apontado por um Operador durante a execução dos processos automotivos diários.

UC001	Cadastrar RNC
Atores: Gerente de Qualidade	
Descrição: Como Gerente de Qualidade, eu quero cadastrar os incidentes (registros de não conformidade) ocorridos durante a execução dos processos automotivos que foram apontados por um Operador.	
Pré-Condições: Operador registra uma parada no Módulo de Controle de Processos Automotivos.	
Valor do negócio: Para uma rápida e correta resolução dos problemas ocorridos durante a execução das atividades diárias da empresa.	
Prioridade: Alta	Estimativa: 4

Caso de uso UC002: remover RNC

Descrição resumida:

Este caso de uso deve permitir ao Gerente de Qualidade a remoção de uma não conformidade cadastrada desde que a mesma não tenha sido resolvida por um Engenheiro.

UC002	Remover RNC
Atores: Gerente de Qualidade	
Descrição: Como Gerente de Qualidade, eu quero remover os incidentes (registros de não conformidade) cadastrados.	
Pré-Condições: O registro de não conformidade não pode ter sido resolvida por um Engenheiro.	
Valor do negócio: Para uma correta manutenção dos registros de não conformidade no sistema.	
Prioridade: Baixa	Estimativa: 2

Caso de uso UC003: listar RNCs

Descrição resumida:

Este caso de uso deve permitir ao Gerente de Qualidade ou Engenheiro listar os registros de não conformidade cadastrados no sistema.

UC003	Listar RNCs
Atores: Gerente de Qualidade, Engenheiro	
Descrição: Como Gerente de Qualidade ou Engenheiro, eu quero consultar os registros de não conformidade cadastrados.	
Valor do negócio: Para uma correta manutenção dos registros de não conformidade do sistema.	
Prioridade: Alta	Estimativa: 2

Caso de uso UC004: ver detalhes da RNC

Descrição resumida:

Este caso de uso deve permitir ao Gerente de Qualidade ou Engenheiro ver os detalhes de um registro de não conformidade listado pela UC003.

UC004	Ver Detalhes da RNC
Atores: Gerente de Qualidade, Engenheiro	
Descrição: Como Gerente de Qualidade ou Engenheiro, eu quero ver os detalhes de um registro de não conformidade dentre os que foram listados no caso de uso 3 (UC003).	
Valor do negócio: Para uma correta manutenção dos registros de não conformidade no sistema.	
Prioridade: Média	Estimativa: 4

Caso de uso UC005: alterar RNC

Descrição resumida:

Este caso de uso deve permitir ao Gerente de Qualidade ou Engenheiro alterar os registros de não conformidade cadastrados no sistema.

UC005	Alterar RNC
Atores: Gerente de Qualidade, Engenheiro	
Descrição: Como Gerente de Qualidade ou Engenheiro, eu quero alterar os registros de não	

conformidade cadastrados.

Valor do negócio: Para uma correta manutenção dos registros de conformidade no sistema.

Prioridade: Alta

Estimativa: 4

Caso de uso UC006: resolver RNC

Descrição resumida:

Este caso de uso deve permitir ao Engenheiro resolver os registros de não conformidade cadastrados no sistema.

UC006	Resolver RNC
Atores: Engenheiro	
Descrição: Como Engenheiro, eu quero resolver os registros de não conformidade cadastrados.	
Valor do negócio: Para uma rápida e correta resolução dos problemas ocorridos durante a execução das atividades diárias da empresa.	
Prioridade: Alta	Estimativa: 2

Caso de uso UC007: confirmar resolução de RNC

Descrição resumida:

Este caso de uso deve permitir ao Gerente de Qualidade confirmar a resolução de um registro de não conformidade resolvido por um Engenheiro.

UC007	Confirmar Resolução de RNC
Atores: Gerente de Qualidade	
Descrição: Como Gerente de Qualidade, eu quero confirmar a resolução dos registros de não conformidade resolvidos pelos Engenheiro.	
Pré-Condições: O registro de não conformidade deve ter sido resolvido por um Engenheiro.	
Valor do negócio: Para uma rápida e correta resolução dos problemas ocorridos durante a execução das atividades diárias da empresa.	
Prioridade: Alta	Estimativa: 2

Caso de uso UC008: corrigir resolução de RNC

Descrição resumida:

Este caso de uso deve permitir ao Engenheiro corrigir a resolução de um registro de não conformidade anteriormente resolvido que não tenha sido confirmado (aprovado) por um Gerente de Qualidade.

UC008	Corrigir Resolução de RNC
Atores: Engenheiro	
Descrição: Como Engenheiro, eu quero corrigir a resolução dos registros de não conformidade resolvidas, mas não confirmados (aprovados).	
Pré-Condições: O registro de não conformidade deve ter sido resolvido por um Engenheiro e não confirmado (aprovado) por um Gerente de Qualidade.	
Valor do negócio: Para uma correta manutenção dos registros de não conformidade no sistema.	
Prioridade: Média	Estimativa: 2

Caso de uso UC009: criar modelos de workflows de atividades

Descrição resumida:

Este caso de uso deve permitir ao Engenheiro criar modelos (templates) de workflows de atividades diárias que serão usados pelos Operadores para criar workflows.

UC009	Criar Modelos de Workflows de Atividades
Atores: Engenheiro	
Descrição: Como Engenheiro, eu quero criar modelos de workflows de atividades diárias para servirem de template na criação de workflows.	
Valor do negócio: Para uma maior agilidade no processo de criação das atividades que devem ser executadas diariamente.	
Prioridade: Alta	Estimativa: 4

Caso de uso UC010: alterar modelo de workflows de atividades

Descrição resumida:

Este caso de uso deve permitir ao Engenheiro alterar os modelos de workflows de atividades cadastrados no sistema.

UC010	Alterar Modelo de Workflows de Atividades
Atores: Engenheiro	
Descrição: Como Engenheiro, eu quero alterar os modelos de workflows de atividades diárias criados no sistema.	
Valor do negócio: Para uma maior agilidade no processo de criação das atividades que devem ser executadas diariamente.	
Prioridade: Média	Estimativa: 2

Caso de uso UC011: listar modelos de workflows de atividades

Descrição resumida:

Este caso de uso deve permitir que um Operador ou Engenheiro possa listar os modelos de workflows de atividades existentes.

UC011	Listar Modelos de Workflows de Atividades
Atores: Operador, Engenheiro	
Descrição: Como Operador ou Engenheiro, eu quero listar os modelos de workflows de atividades existentes.	
Valor do negócio: Para uma maior agilidade no processo de criação das atividades que devem ser executadas diariamente.	
Prioridade: Alta	Estimativa: 2

Caso de uso UC012: ver detalhes de modelo de workflow

Descrição resumida:

Este caso de uso deve permitir ao Operador ver detalhes dos modelos de workflows de atividades existentes.

UC012	Ver Detalhes de Modelo de Workflow
Atores: Engenheiro	
Descrição: Como Engenheiro, eu quero ver os detalhes de um modelo de workflow de	

atividades.

Valor do negócio: Para uma maior agilidade no processo de criação das atividades que devem ser executadas diariamente.

Prioridade: Média

Estimativa: 2

Caso de uso UC013: criar workflow a partir de modelo

Descrição resumida:

Este caso de uso deve permitir ao Operador criar um workflow de atividades diárias a partir de um modelo existente.

UC013	Criar Workflow a Partir de um Modelo
Atores: Operador	
Descrição: Como Operador, eu quero criar um workflow de atividades a partir de um modelo existente no sistema.	
Valor do negócio: Para controle e acompanhamento das atividades que devem ser executadas diariamente e para uma rápida e correta resolução dos problemas ocorridos durante a execução dessas atividades.	
Prioridade: Alta	Estimativa: 2

Caso de uso UC014: preencher workflow de atividades diárias

Descrição resumida:

Este caso de uso deve permitir ao Operador preencher os workflows de atividades diárias durante a execução de suas tarefas.

UC014	Preencher Workflow de Atividades Diárias
Atores: Operador	
Descrição: Como Operador, eu quero preencher o workflow de atividades durante a execução das tarefas diárias.	
Valor do negócio: Para controle e acompanhamento das atividades que devem ser executadas diariamente e para uma rápida e correta resolução dos problemas ocorridos durante a execução dessas atividades.	

Prioridade: Alta	Estimativa: 2
-------------------------	----------------------

Caso de uso UC015: registrar paradas e problemas ocorridos

Descrição resumida:

Este caso de uso deve permitir ao Operador registrar as paradas e problemas ocorridos durante a execução das atividades cadastradas no workflow.

UC015	Registrar Paradas e Problemas Ocorridos
Atores: Operador	
Descrição: Como Operador, eu quero registrar as paradas e problemas ocorridos durante a execução das atividades diárias cadastradas no workflow.	
Valor do negócio: Para uma rápida e correta resolução dos problemas ocorridos durante a execução das atividades diárias.	
Prioridade: Alta	Estimativa: 2

Caso de uso UC016: consultar workflows em andamento

Descrição resumida:

Este caso de uso deve permitir ao Operador, Gerente de Qualidade e Engenheiro consultar o andamento das atividades do workflow.

UC016	Consultar Workflows em Andamento
Atores: Operador, Gerente de Qualidade e Engenheiro	
Descrição: Como Operador, Gerente de Qualidade ou Engenheiro eu quero consultar o andamento das atividades do workflow.	
Pré-Condições: A execução das atividades do workflow devem ter sido iniciadas.	
Valor do negócio: Para controle e acompanhamento das atividades que devem ser executadas diariamente.	
Prioridade: Média	Estimativa: 2

Caso de uso UC017: cadastrar notificação

Descrição resumida:

Este caso de uso deve permitir ao Engenheiro cadastrar notificações que podem ser apenas informações técnicas ou recall de produtos.

UC017	Cadastrar Notificação
Atores: Engenheiro	
Descrição: Como Engenheiro eu quero cadastrar notificações para serem divulgadas ao público externo. Estas podem ser informações técnicas ou recall de produtos.	
Valor do negócio: Para divulgação de informações importantes relacionadas aos produtos da empresa.	
Prioridade: Alta	Estimativa: 2

Caso de uso UC018: alterar notificação

Descrição resumida:

Este caso de uso deve permitir ao Engenheiro alterar notificações já cadastradas no sistema antes que as mesmas sejam divulgadas ao público externo.

UC018	Alterar Notificação
Atores: Engenheiro	
Descrição: Como Engenheiro eu quero alterar notificações já cadastradas no sistema antes delas serem divulgadas ao público externo.	
Pré-Condições: A notificação deve estar cadastrada no sistema e ainda não ter sido divulgada ao público externo.	
Valor do negócio: Para correção das informações que serão divulgadas ao público externo.	
Prioridade: Alta	Estimativa: 2

Caso de uso UC019: remover notificação

Descrição resumida:

Este caso de uso deve permitir ao Engenheiro remover notificações já cadastradas no sistema antes que as mesmas sejam divulgadas ao público externo.

UC019	Remover Notificação
Atores: Engenheiro	

Descrição: Como Engenheiro eu quero remover notificações já cadastradas no sistema antes delas serem divulgadas ao público externo.	
Pré-Condições: A notificação deve estar cadastrada no sistema e ainda não ter sido divulgada ao público externo.	
Valor do negócio: Para correção das informações que serão divulgadas ao público externo.	
Prioridade: Média	Estimativa: 2

Caso de uso UC020: aprovar divulgação

Descrição resumida:

Este caso de uso deve permitir ao Gestor aprovar a divulgação das notificações cadastradas pelos engenheiros.

UC020	Aprovar Divulgação
Atores: Gestor	
Descrição: Como Gestor eu quero aprovar as notificações que serão divulgadas ao público externo.	
Valor do negócio: Para divulgação de informações importantes relacionadas aos produtos da empresa.	
Prioridade: Alta	Estimativa: 2

Caso de uso UC021: exportar recall

Descrição resumida:

Este caso de uso deve permitir ao Usuário Externo exportar os recalls emitidos pelo sistema.

UC021	Exportar Recall
Atores: Usuário Externo	
Descrição: Como Usuário Externo eu quero exportar os recall emitidos pelo sistema nos formatos json ou xml.	
Valor do negócio: Para divulgação de informações importantes relacionadas aos produtos da empresa.	
Prioridade: Alta	Estimativa: 2

4.3. Modelo de componentes

O diagrama de componentes do sistema, os quais impactam no design da arquitetura e seleção das tecnologias. Foram organizados para serem reutilizáveis e fornecerem interfaces bem definidas de acordo com suas responsabilidades.

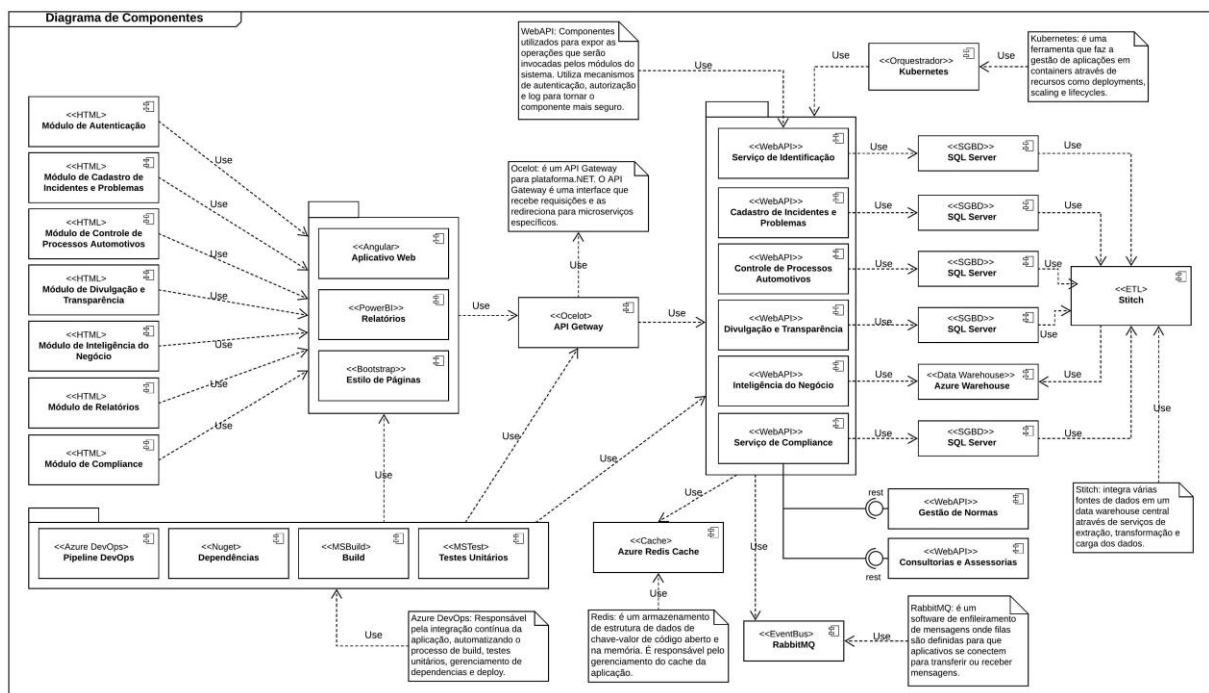


Figura 7 - Diagrama de Componentes

A seguir é apresentado um detalhamento dos componentes e módulos envolvidos no modelo de componentes. Nessa arquitetura devemos considerar a divisão do sistema em aplicativo Web, composto por Angular, Bootstrap e PowerBI, e microserviços criados para cada módulo do sistema. Cada microserviço é um WebAPI que funciona de forma independente dos demais e possuem sua própria base de dados SQL Server, que irão fornecer os dados necessários para exibir as informações nas páginas acessadas via navegador.

O aplicativo Web utiliza componentes responsivos e padronizados, garantindo conformidade com os requisitos não funcionais de usabilidade e portabilidade. Além disso, utilizará o conceito de *single-page application* ou SPA, em que todo o código necessário ao funcionamento da página é obtido em um único carregamento, fazendo com que as demais interações do usuário tragam apenas recursos adicionais para serem incluídos dinamicamente, sendo importante na garantia do requisito não funcional de desempenho.

O sistema provê um módulo separado, composto por um Web API e um banco de dados Sql Server, para lidar com autenticação e autorização de usuários. O controle de acesso aos recursos do sistema será feito por meio de tokens gerados no momento da autenticação do usuário através do padrão JWT – Jason Web Token. Além disso, cada recurso só poderá ser acessado por determinados perfis de usuário, garantindo conformidade ao requisito não funcional de segurança.

A integração entre os microserviços será feita por meio de eventos utilizando o padrão publish/subscribe, onde os serviços enviarão notificações uns aos outros através de filas de mensagens, garantindo que a informação sempre chegue ao seu destino, mesmo que ocorram falhas, garantindo o requisito não funcional de confiabilidade. A ferramenta escolhida para implementar esse EventBus foi a RabbitMQ, que é um servidor de mensageria de código aberto.

Ainda para garantir conformidade ao requisito não funcional de confiabilidade será utilizado o recurso de cache, ajudando também na otimização da performance do sistema. Com isso, os dados frequentemente acessados pelos serviços serão mantidos em memória, reduzindo tráfego de rede e economizando acesso à base de dados. O Azure Redis Cache será usado para este cenário.

Os artefatos executáveis serão produzidos utilizando a plataforma .NET e versionados no servidor GitHub através do padrão Git. O código fonte no GitHub será integrado ao Azure DevOps, onde será construído o pipeline de integração contínua para construção dos artefatos usando o MSBuild, execução de testes unitários com o MSTest, versionamento das bibliotecas usando o Nuget e deploy da aplicação em ambiente de produção, garantindo os RNF de manutenibilidade e testabilidade.

Componente	Descrição
Azure DevOps (Integração contínua da aplicação)	Componente utilizado para construir, gerenciar as dependências, executar testes unitários de forma automatizada e implantar os módulos da aplicação nos respectivos containers.
Aplicativo Web Angular (<i>SPA – Single-Page Application</i>)	Componente utilizado para gerar o conteúdo HTML, CSS e javascript dos módulos do sistema a serem renderizados nos navegadores.

WebAPI (Microserviços)	Componentes utilizados para expor as operações que serão invocadas por outros módulos do sistema.
Ocelot (API Getway)	Componente utilizado como ponto central para receber as requisições da aplicação, recebendo-as e redirecionando-as aos seus respectivos microserviços de destino.
Kubernetes	Componente utilizado para gerenciamento do cluster de containers. Possui recursos de balanceamento de carga, escalabilidade e resiliência.
Azure Redis Cache	Recurso utilizado para armazenar em memória os recursos mais utilizados para diminuir a latência das requisições.
Azure Warehouse (Data warehouse)	Componente utilizado para fornecer informações para o módulo de inteligência do negócio. Funciona como um banco para armazenamento das informações relevantes às tomadas de decisões.
Stitch (ETL)	Componente responsável por colher os dados de todos os módulos do sistema e concentrá-los em um data warehouse centralizado.
RabbitMQ (EventBus)	Componente utilizado para a troca de mensagens entre os diversos microserviços da aplicação.
Gestão de Normas	Interface WebAPI REST, responsável por expor um catálogo de normas nacionais e internacionais que fornecem toda base para a gestão da qualidade. É disponibilizado em nuvem com um serviço (SaaS).
Consultorias e Acessorias	Interface WebAPI REST, responsável por expor operações que permitem a contratação de serviços de consultoria e acessorias no mercado, visando adequar a empresa às normas e padrões da sua área de negócio. É disponibilizado em nuvem com um serviço (SaaS).

4.4. Modelo de implantação

O modelo de implantação auxilia no entendimento de como os componentes de software estarão fisicamente implantados e como a comunicação entre eles deve ocorrer. Os componentes foram organizados para serem reutilizáveis, fornecendo interfaces bem definidas de acordo com suas responsabilidades. Este modelo de implantação da arquitetura é apresentado abaixo.

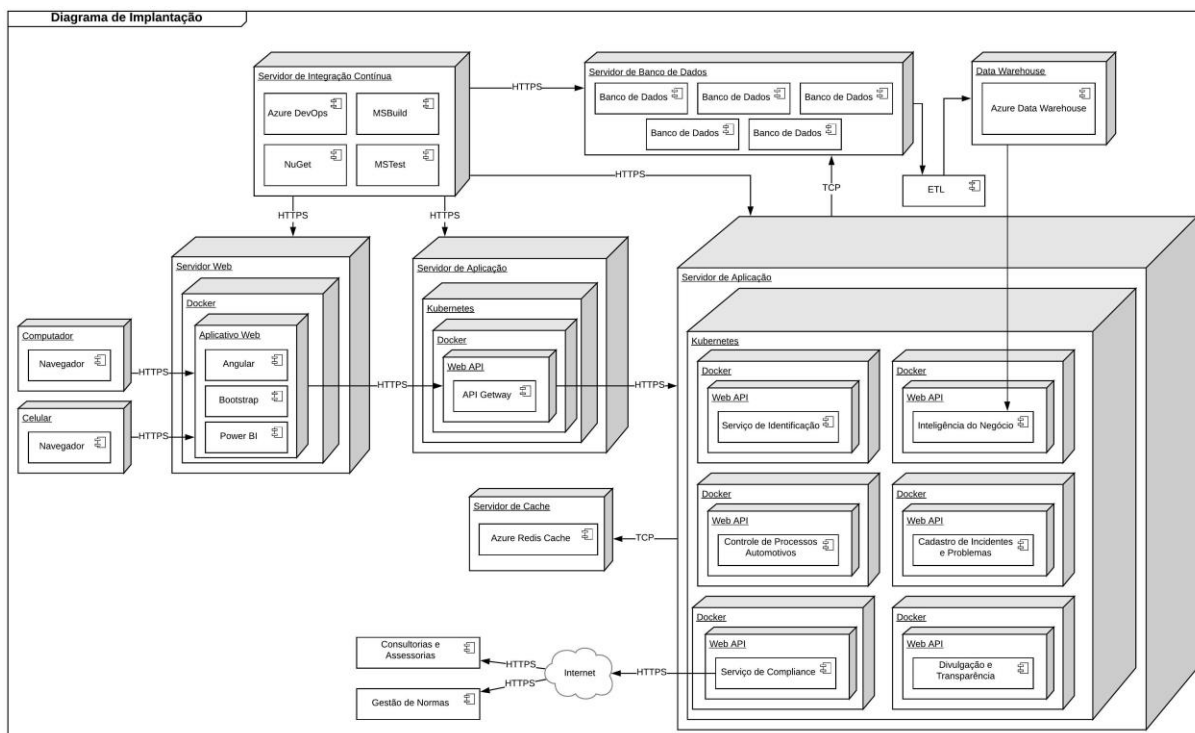


Figura 8 - Diagrama de Implantação

A seguir é apresentada a relação dos componentes com os seus respectivos servidores, e como o modelo de implantação deve ser implantado. Na arquitetura explicitada no modelo acima, devemos considerar um sistema de cluster balanceado onde podemos garantir que no caso de uma sobrecarga em um dos servidores, o processo de balanceamento possa redirecionar a requisição para outro servidor que esteja com uma carga menor no momento.

Componente	Descrição
Navegador	Representa o navegador a ser utilizado pelos usuários para interagir com os recursos do sistema. Pode ser utilizado a partir de um computador pessoal ou dispositivo móvel.
Servidor de integração contínua	Responsável pela construção, gerenciamento de dependências, execução de testes unitários e implantação da aplicação nos servidores de produção.
Servidor Web	Responsável por disponibilizar as páginas da aplicação que serão acessadas pelos usuários através de um navegador.
Servidor de aplicação	Responsável por prover toda a infraestrutura necessária para que a aplicação funcione no servidor, gerenciando o processamento das requisições e lidando com as regras de

	negócio da aplicação.
Servidor de cache	Responsável por armazenar as requisições mais acessadas em memória e sincronizar os dados com a base de dados do sistema. Esse componente tem o objetivo de melhorar o desempenho e permitir que alguns dados retornem de forma mais rápida para o usuário.
Servidor de banco de dados	Representa o servidor que conterá os repositórios de dados de cada microserviço da aplicação.
Data warehouse	Componente utilizado para fornecer as informações necessárias ao módulo de inteligência do negócio. Esse componente é utilizado para conter as informações que serão usadas no processo de tomada de decisões por parte dos gestores.
ETL	Componente responsável por transportar os dados de todos os bancos de dados da aplicação para um data warehouse centralizado.

4.5. Modelo de dados

Cada módulo do sistema é implantado como um microserviço onde cada um possui sua própria base de dados independente das demais.

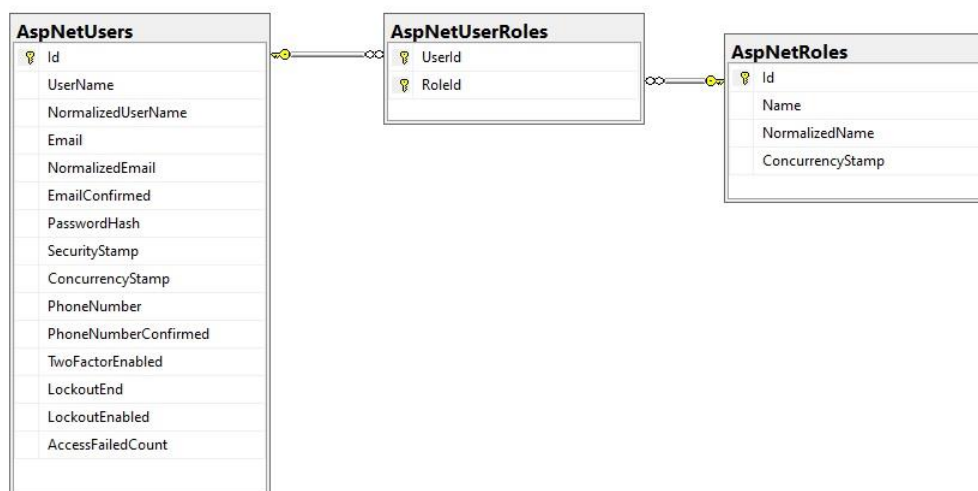


Figura 9 - Modelo de Dados do Módulo de Autenticação e Autorização

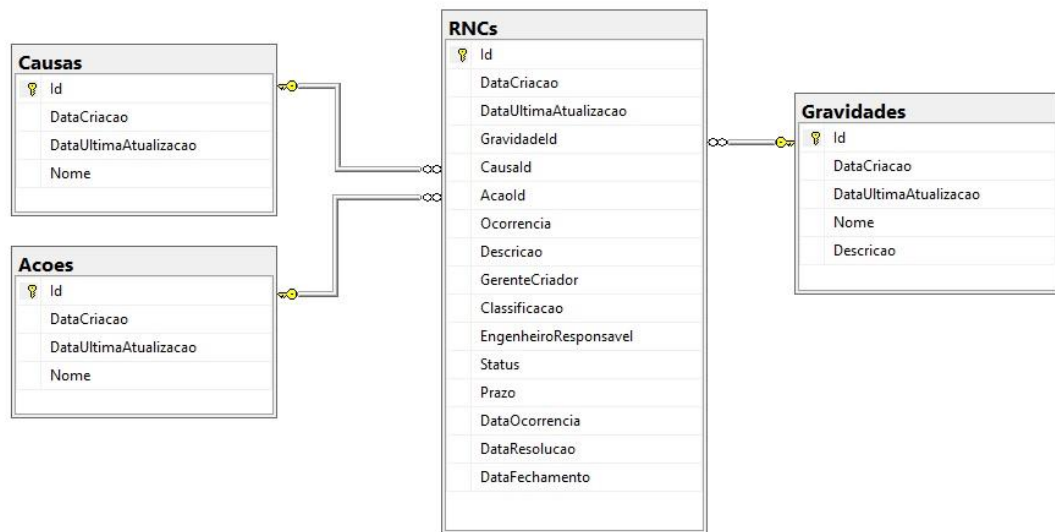


Figura 10 - Modelo de Dados do Módulo de Cadastro de Incidentes e Problemas

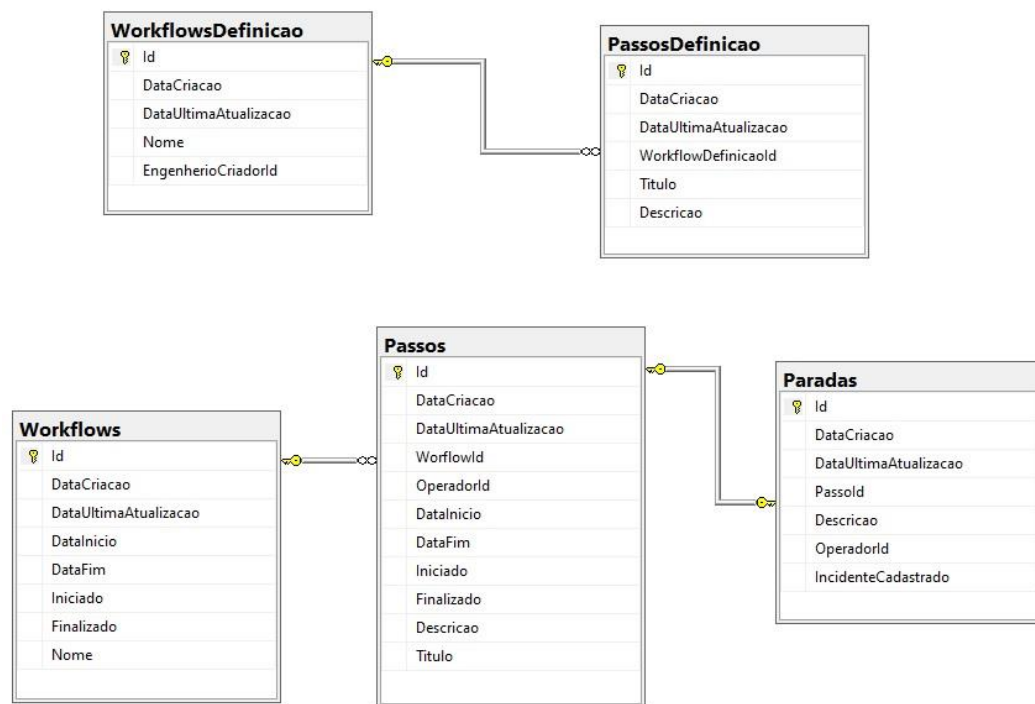


Figura 11 - Modelo de Dados do Módulo de Controle de Processos Automotivos

5. Prova de Conceito (POC) / protótipo arquitetural

5.1. Implementação e Implantação

A prova de conceito desse projeto visa validar as escolhas arquiteturais e garantir que os requisitos funcionais e não funcionais do projeto estão sendo atendidos de forma adequada.

O objetivo desse protótipo é verificar se o processo de autenticação dos usuários, assim como a validação dos perfis de acesso estão sendo atendidos. Foi validado se os módulos de cadastro de incidentes e problemas e também o controle de processos automatizados estão atendendo todas as necessidades do usuário relacionado aos requisitos de qualidade, afim de minimizar riscos e maximizar ganhos de produtividade na sequência do projeto. Os requisitos não funcionais validados estão listados abaixo:

- **Segurança:** este requisito não funcional foi escolhido devido à necessidade e preocupação em manter os dados seguros.
 - Critérios de aceite:
 - Impedir que usuários não autenticados possam acessar as páginas do sistema ou utilizar os recursos das APIs.
 - Redirecionar o usuário não autenticado para uma tela de autenticação.
 - Permitir a livre navegação do usuário através de telas públicas, bem como permitir o acesso aos recursos não privados das APIs.
 - Garantir que os usuários autenticados só terão acesso às páginas e recursos destinados ao seu perfil.
 - Apresentar no sistema apenas os recursos aos quais o usuário tem acesso.
- **Usabilidade:** esse requisito não funcional foi escolhido devido a importância em manter um sistema com boa usabilidade, e que possa garantir uma navegação simples e objetiva:
 - Os critérios de aceite são:
 - A navegação no sistema deve ser visível, clara, reduzida e natural.
 - O sistema deve possuir uma linguagem de fácil entendimento, de forma que o usuário compreenda facilmente o sistema e possa usá-lo corretamente mesmo sem treinamento prévio.
 - Ao navegar pelo sistema, o usuário deve ter conhecimento de todas as ações que ele pode executar e ter um feedback do que foi realizado.

- **Acessibilidade:** esse requisito não funcional foi escolhido para garantir que o sistema atenda todas as exigências arquiteturais de um sistema responsivo que se adapte à diferentes dispositivos, como celulares tablets e desktops:
 - Os critérios de aceite são:
 - Os componentes da tela devem se adaptar com a resolução identificada, sejam eles provenientes de celulares, tablets ou desktops.
 - O sistema deve manter um padrão de cores, fontes e outros objetos.
 - O sistema deve ser compatível com os principais navegadores do mercado, como Internet Explorer, Firefox, Chrome e Opera.

5.1.1 Tecnologias utilizadas

As tecnologias utilizadas nessa PoC são:

Módulo	Tecnologias
Interface do Usuário	Angular, Bootstrap, NPM
API Getway	ASP.NET Core Web API, NuGet
Serviço de Autenticação	ASP.NET Core Web API, ASP.NET Core Identity, Entity Framework Core, SQL Server, JWT, Auto Mapper, Swagger, Elmah.io, NuGet
Cadastro de Incidentes e Problemas	ASP.NET Core Web API, ASP.NET Core Identity, Entity Framework Core, SQL Server, JWT, Auto Mapper, Swagger, Elmah.io, NuGet
Controle de Processos Automotivos	ASP.NET Core Web API, ASP.NET Core Identity, Entity Framework Core, SQL Server, JWT, Auto Mapper, Swagger, Elmah.io, NuGet
Divulgação e Transparência	ASP.NET Core Web API, ASP.NET Core Identity, Entity Framework Core, SQL Server, JWT, Auto Mapper, Swagger, Elmah.io, NuGet

5.1.2 Casos de Uso

Para esta PoC foram implementados casos de uso dos módulos de serviço de autenticação, módulo de cadastro de incidentes e problemas e módulo de controle de processos automotivos, conforme quadro abaixo:

Módulo	Caso de Uso
Serviço de Autenticação	Gerar Token, Logar no Sistema, Autenticar Usuário
Cadastro de Incidentes e Problemas	Cadastrar RNC, Alterar RNC, Listar RNC, Ver Detalhes da RNC
Controle de Processos Automotivos	Cadastrar Definição de Workflow, Listar Definições de Workflow, Ver Detalhes da Definição de Workflow, Alterar Definição de Workflow, Criar workflow a partir de Modelo, Preencher Workflow de Atividades, Listar Workflows em Andamento, Registrar Paradas e Problemas Ocorridos

5.1.3 Implantação

A prova de conceito desenvolvida foi totalmente implantada na nuvem através do serviço Azure.

Implementação	Recurso
Todos os microserviços (APIs)	Azure App Services
Todos os bancos de dados dos microserviços	Azure SQL
Sistema SPA (User Interface)	Azure App Services

5.2 Interfaces/ APIs

API	API REST de Autenticação e Autorização
URL	https://pucminassggidentityservice.azurewebsites.net/swagger
Documentação	/swagger
/api/v1/registrar	
Métodos	POST
Request Content Type	Application/json

Response Content Type	Application/json
Exemplo de Requisição	<pre>{ "userName": "string", "email": "string", "password": "string", "confirmPassword": "string" }</pre>
Exemplo de Resposta	<pre>{ "success": true, "data": { "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiI1ZTYzZjhhMy01YjI4L TRhZjEtOWMlOS0yYzE1NzA1MGJmMzkiLCJlbWFPbCI6InRlc3RlQHRlc3RlLmNvbSIs Imp0aSI6ImM1YmJmYzcxLTc4ZGQtNDI1MiliYjllWEYODMyYWZiMmNlOSIsIm5iZiI6 MTU4Njc3MTQxMCwiaWF0IjoxNTg2NzcxNDEwLCJleHAiOjE1ODY3Nzg2MTAsImZcy I6IlNHUS5JZGVudGl0eVNiLnZpY2UiLCJhdWQiOiJodHRwOi8vbG9jYWxob3N0In0.d BYLHlInFvHYNxzWBZuABqezgRHrJRNZerd_Kw4nytI", "expiresIn": 7200, "userToken": { "id": "5e63f8d3-5b28-4af1-9c59-2c157050bf39", "email": "teste@teste.com", "claims": [{ "value": "5e63f8d3-5b28-4af1-9c59-2c157050bf39", "type": "sub" }, { "value": "teste@teste.com", "type": "email" }, { "value": "c5bbfc71-78dd-4252-bb9e-a2832afb2ce9", "type": "jti" }, { "value": "1586771411", "type": "nbf" }, { "value": "1586771411", "type": "iat" }] } } }</pre>
/api/v1/entrar	
Métodos	POST
Request Content Type	Application/json
Response Content Type	Application/json
Exemplo de Requisição	<pre>{ "userName": "string", "password": "string" }</pre>
Exemplo de Resposta	<pre>{ "success": true, "data": { "accessToken": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9. eyJzdWIiOi-</pre>

	<pre> iI2ZGQ2OGQ2Ni1kOGU5LTQ5MTItODFlMi04ZTc2ZmI1MDU3YmIiLCJlbWFpbCI 6ImdlcmVudGVAdGVzdGUuY29tIiwianRpIjoInjA0ZDljYTgtZGIyYy00ZDI1LWJ iM- jAtY2U4ZjU1M2JlZjE5IiwibmJmIjoxNTg1NzA0MDQwLCJpYXQiojE1ODU3MDQwN DAsInJvbGUi- OiJnZXJlbnRlIiwizXhwIjoxNTg1NzExMjQwLCJpc3MiOiJTR1EuSWRlbnRpdH1T ZXJ2aWNl IiwiYXVki- joiaHR0cDovL2xvY2FsaG9zdCJ9.2qlmqwK7LXuI2s2cwM8hZgB_cmfc89 VhoEoDjhK9zOM", "expiresIn": 7200, "userToken": { "id": "6dd68d66-d8e9-4912-81e2-8e76fb5057bb", "email": "gerente@teste.com", "claims": [{ "value": "6dd68d66-d8e9-4912-81e2-8e76fb5057bb", "type": "sub" }, { "value": "gerente@teste.com", "type": "email" }, { "value": "604d9ca8-db2c-4d25-bb20-ce8f553bef19", "type": "jti" }, { "value": "1585704040", "type": "nbf" }, { "value": "1585704040", "type": "iat" }, { "value": "gerente", "type": "role" }] } } } </pre>
--	--

API	API REST de Controle de Incidentes e Problemas
URL	https://pucminassggincidentes.azurewebsites.net/swagger
Documentação	/swagger

/api/v1/acoes	
Métodos	GET, POST, PUT e DELETE
Request Content Type	Sem parâmetros
Response Content Type	Application/json
Exemplo de Requisição	Sem parâmetros
Exemplo de Resposta	<pre>[{ "id": "e7405e99-1f5b-4720-95db-149622b0f84b", "nome": "Treinamento de conscientização do uso correto do EPI" }, { "id": "d54f25ee-b8ec-48b8-959c-3b5b4c2889a1", "nome": "Fazer a substituição do equipamento por um mais moderno" }, { "id": "42a3b688-5888-4991-a0f7-611938addae9", "nome": "Fazer manutenção preventiva de tempos em tempos para garantir a segurança dos funcionários" }, { "id": "2b0b0561-e5d7-4326-a651-9933d709d56f", "nome": "Campanha de segurança no trabalho com o uso correto do EPI" }, { "id": "fe3a89b2-5282-4391-85ed-b55ecb15d6d2", "nome": "Nenhuma medida de controle" }]</pre>
/api/v1/causas	
Métodos	GET, POST, PUT e DELETE
Request Content Type	Sem parâmetros
Response Content Type	Application/json
Exemplo de Requisição	Sem parâmetros
Exemplo de Resposta	<pre>[{ "id": "40db6681-4e73-4508-894a-552805aa5573", "nome": "Matérias primas de baixa qualidade" }, { "id": "517ef504-7630-4084-a557-7df2d6b618f2", "nome": "Não utilização do EPI" }, { "id": "a204d59b-e1af-491a-acb3-953a7b0196d8", "nome": "Equipamento sem proteção" }, { "id": "a3c32927-a122-4ff5-ae70-a74e372605d3", "nome": "Defeito na ponte rolante" }, { </pre>

	<pre> "id": "12e4bd0e-6a88-4f35-a556-c22bed1c58f0", "nome": "Operador sem capacitação" }, { "id": "2ca53256-c85c-43f5-9f06-fe52fda4bb85", "nome": "Utilização de andaime inadequada" }] </pre>
/api/v1/gravidade	
Métodos	GET, POST, PUT, DELETE
Request Content Type	Sem parâmetros
Response Content Type	Application/json
Exemplo de Requisição	Sem parâmetros
Exemplo de Resposta	<pre> [{ "id": "03a12fa6-4c19-4ced-9d03-3fefa2abb9b0", "nome": "Baixa", "descricao": "O risco gerado na execução da atividade pode ocasionar pequenas perdas em termos de saúde e segurança do colaborador" }, { "id": "a217248b-e3fc-491a-a49a-5712e8d689f4", "nome": "Elevada", "descricao": "O risco gerado na execução da atividade pode ocasionar grandes perdas em termos de saúde e segurança do colaborador" }, { "id": "f3d08139-1f50-4731-9384-88c2dedbfb4a", "nome": "Ausente", "descricao": "O risco gerado na execução da atividade não ocasiona perdas em termos de saúde e segurança do colaborador" }, { "id": "821efdf1-b04f-49df-86cd-e09ed4473fe8", "nome": "Moderada", "descricao": "O risco gerado na execução da atividade pode ocasionar perdas moderadas em termos de saúde e segurança do colaborador" }] </pre>
/api/v1/rncs	
Métodos	GET, POST, PUT, DELETE
Request Content Type	Sem parâmetros
Response Content Type	Application/json
Exemplo de Requisição	Sem parâmetros
Exemplo de Resposta	<pre> [{ "id": "4aadle67-04c2-4f36-55f5-08d7e017a5a0", "ocorrencia": "Problema na máquina da funilaria", "descricao": "Foi necessário trocar a máquina da funilaria." }] </pre>

	<pre> "classificacao": 1, "status": 1, "gravidade": { "id": "03a12fa6-4c19-4ced-9d03-3fef2abb9b0", "nome": "Baixa", "descricao": "O risco gerado na execução da atividade pode ocasionar pequenas perdas em termos de saúde e segurança do colaborador" }, "causa": { "id": "40db6681-4e73-4508-894a-552805aa5573", "nome": "Matérias primas de baixa qualidade" }, "acao": { "id": "d54f25ee-b8ec-48b8-959c-3b5b4c2889a1", "nome": "Fazer a substituição do equipamento por um mais moderno" }, "prazo": "2020-04-16T02:00:45.763", "dataOcorrencia": "2020-04-14T02:00:46.1970834" }] </pre>
--	---

API	API REST de Controle de Processos Automotivos
URL	https://pucminassggprocessos.azurewebsites.net/swagger
Documentação	/swagger
	/api/v1/workflowdefinicao
Métodos	GET, POST, PUT
Request Content Type	Sem parâmetros
Response Content Type	Application/json
Exemplo de Requisição	Sem parâmetros
Exemplo de Resposta	<pre> [{ "id": "116c65ca-1977-4467-8505-139491221263", "nome": "Fixação das Portas", "engenheiroCriadorId": "2f4ce2c2-359a-491d-99b9-4237012d152a", "passosDefinicao": [{ "id": "d8050dea-0cb3-4772-8360-08d7e013d0ba", "workflowDefinicaoId": "116c65ca-1977-4467-8505-139491221263", "titulo": "Fixação das Portas", "descricao": "As portas são instaladas após a colocação dos itens de maior volume no carro, como assentos, painel de instrumentos, vidros laterais e dianteiros e acabamento do teto e do interior. Só então elas são parafusadas, num processo manual realizado pelos operários." }] }, { "id": "2422f860-c912-45ae-bbca-4821f4538ad0", "nome": "Produção diária", "engenheiroCriadorId": "2f4ce2c2-359a-491d-99b9- </pre>

	<pre> 4237012d152a", "passosDefinicao": [{ "id": "ad421047-641e-44c7-8359-08d7e013d0ba", "workflowDefinicaoId": "2422f860-c912-45ae-bbca-4821f4538ad0", "titulo": "Estamparia", "descricao": "As chapas de aço chegam em bobinas ou \"blanks\", com etiquetas que apontam para quais modelos se destinam. Depois de passar por um controle de qualidade, elas são \"recortadas\" em prensas de até 3.000 toneladas, programadas com as dimensões específicas das peças que formarão o veículo." }, { "id": "79bfebf1-51a2-460b-835a-08d7e013d0ba", "workflowDefinicaoId": "2422f860-c912-45ae-bbca-4821f4538ad0", "titulo": "Estruturação", "descricao": "Também chamada de \"body shop\", a fase responsável pela montagem da carroceria só começa quando a parte dianteira recebe sua etiqueta de identificação, que já o credencia como um novo automóvel. A parte lateral do veículo é trazida por um robô e se junta à parte dianteira e ao assoalho." }, { "id": "006670c6-882e-40ad-835b-08d7e013d0ba", "workflowDefinicaoId": "2422f860-c912-45ae-bbca-4821f4538ad0", "titulo": "Funilaria", "descricao": "\tApós a estruturação da carroceria, ela passa por um acabamento de funilaria, em que pequenos defeitos são corrigidos. São testadas as aberturas de portas, capôs e porta-malas. As junções são conferidas visualmente uma a uma pelos técnicos." }], { "id": "1468598d-9866-42d8-a91c-517a3d56fccf", "nome": "Testar cadastro", "engenheiroCriadorId": "2f4ce2c2-359a-491d-99b9-4237012d152a", "passosDefinicao": [{ "id": "ae1cf743-39c1-41ed-835c-08d7e013d0ba", "workflowDefinicaoId": "1468598d-9866-42d8-a91c-517a3d56fccf", "titulo": "Teste de cadastro pelo mobile", "descricao": "verificar se tela é responsiva." }] }] </pre>
/api/v1/workflow	
Métodos	GET
Request Content Type	Sem parâmetros
Response Content Type	Application/json
Exemplo de Requisição	Sem parâmetros

Exemplo de Resposta	<pre>[{ "id": "2422f860-c912-45ae-bbca-4821f4538ad0", "nome": "Fixação das Portas", "iniciado": true, "finalizado": true, "passos": [{ "id": "79bfebf1-51a2-460b-835a-08d7e013d0ba", "workflowId": "2422f860-c912-45ae-bbca-4821f4538ad0", "titulo": "Produção diária", "descricao": "", "iniciado": true, "finalizado": true, "parada": { "id": "2f4ce2c2-359a-491d-99b9-4237012d152a", "passoId": "79bfebf1-51a2-460b-835a-08d7e013d0ba", "descricao": "Hou um problema durante a fixação da porta", "incidenteCadastrado": true } }] }]</pre>
/api/v1/workflow/totolist	
Métodos	GET, PUT
Request Content Type	Sem parâmetros
Response Content Type	Application/json
Exemplo de Requisição	Sem parâmetros
Exemplo de Resposta	<pre>[{ "id": "424fbbbd-d86c-4755-bc7f-bf89d2165754", "nome": "Produção diária", "iniciado": false, "finalizado": false, "passos": [] }]</pre>
/api/v1/workflow/registrar-parada	
Métodos	POST
Request Content Type	Application/json
Response Content Type	Application/json
Exemplo de Requisição	<pre>{ "id": "5bc0b22a-d9de-4c1f-9618-ea2214499d08", "passoId": "424fbbbd-d86c-4755-bc7f-bf89d2165754", "descricao": "Quebra de EPI durante a fixação da porta", "incidenteCadastrado": true }</pre>
Exemplo de Resposta	<pre>{ "id": "5bc0b22a-d9de-4c1f-9618-ea2214499d08", "passoId": "424fbbbd-d86c-4755-bc7f-bf89d2165754", "descricao": "Quebra de EPI durante a fixação da porta", "incidenteCadastrado": true }</pre>

6. Avaliação da Arquitetura

6.1. Análise das abordagens arquiteturais

A arquitetura proposta baseou-se no uso de microserviços, de forma que cada módulo do sistema disponibiliza uma variedade de recursos relacionados e funcionam de forma independente dos demais, com cada microserviço possuindo a sua própria base de dados. Apesar de todos os módulos estarem implantados da mesma forma, os mesmos foram concebidos para poderem ser implantados de diversas formas, podendo ser em nuvem, on premise, self-hosted, em containers, dentre outras.

6.2. Cenários

Os atributos relacionados são referentes aos requisitos de segurança, usabilidade e acessibilidade, listados na seção 5.1.

Cenário 1: ao realizar o acesso a uma página privada, o sistema deve garantir que somente usuários autenticados consigam acesso, redirecionando os usuários não autenticados para uma tela de autenticação para que os mesmos informem login e senha de acesso. O sistema deve permitir o acesso as páginas publicas sem necessidade de autenticação. Caso o usuário informe parâmetros inválidos, o sistema deve mostrar mensagens informado o ocorrido. O cumprimento deste cenário garante o requisito não funcional de segurança.

Cenário 2: Ao acessar um recurso protegido em uma API, o sistema deve garantir que apenas os usuários autenticados tenham acesso mediante utilização de token gerado pela API de autenticação e segurança. O sistema deve retornar um erro que deixe claro que o recurso é acessível apenas com autenticação prévia. O cumprimento deste cenário garante o requisito não funcional de segurança.

Cenário 3: Quando um usuário acessar o sistema, o mesmo só poderá ter acesso aos recursos disponíveis para ele, um **Operador** terá acesso aos recursos: listar as definições de workflow, consultar os workflows em andamento, preencher workflow e registrar paradas e problemas ocorridos. Um **Gerente** terá acesso aos recursos: cadastrar RNC, alterar RNC, listar RNC, ver detalhes da RNC, remover RNC e listar workflows em andamento. Um **Engenheiro** terá acesso aos recursos: alterar RNC, listar RNC, ver detalhes da RNC, cadastrar definição de workflows, listar definições de workflows, alterar definição de workflows e listar workflows

em andamento. O cumprimento deste cenário garante os requisitos não funcionais de segurança e usabilidade.

Cenário 4: Ao acessar o sistema, o usuário deverá ver de forma clara, todos os recursos que o seu perfil poderá utilizar, de forma que a navegação seja facilitada. O sistema deverá apresentar mensagens informando a conclusão de requisições, apresentando erros ocorridos, assim como, informando sobre possíveis problemas ocasionados por ações indevidas, como sair de uma página de cadastro preenchida antes de salvar ou remover um recurso acidentalmente. O cumprimento deste cenário garante o requisito não funcional de usabilidade.

Cenário 5: Ao utilizar o sistema através de um dispositivo móvel, o sistema deve adaptar-se ao tamanho da tela e continuar apresentando boa usabilidade. O sistema deverá manter a mesma identidade visual, tendo seus controles ajustados de acordo com o dispositivo ou navegador utilizado. O cumprimento deste cenário garante o requisito não funcional acessibilidade.

Na priorização foi utilizado o método de árvore de utilidades reduzida com prioridades. Cada atributo foi categorizado de acordo com os requisitos de qualidade a que está relacionado e então classificado em função de sua importância e complexidade, considerando a percepção de negócio e arquitetura. As duas variáveis de priorização “Importância” e “Complexidade” são representadas respectivamente como (IMP) e (COM), sendo classificados como alta (A), média (M) e baixa (B) de acordo com as características do requisito.

Atributos de Qualidade		Cenários	IMP	COM
Funcionalidade	Segurança	Cenário 1: o acesso as páginas do sistema devem ser feitos mediante autenticação através de login e senha.	A	M
	Segurança	Cenário 2: o acesso aos recursos privados das APIs devem exigir autenticação via token.	A	A
	Segurança e usabilidade	Cenário 3: o acesso as funcionalidades do sistema devem ser feitos mediante autorização por perfil e o sistema deverá apresentar ao usuário apenas os recursos que ele tem acesso.	M	B
	Usabilidade	Cenário 5: o sistema deve prover boa usabilidade.	M	B

	Acessibilidade	Cenário 5: o sistema deve ser responsivo, adaptando-se aos mais diferentes dispositivos e navegadores.	M	M
--	----------------	--	---	---

6.3. Avaliação

Processo de avaliação dos cenários apresentados no item 5.1 são analisados; O objetivo é determinar os riscos, não riscos, pontos de sensibilidade e *trade-offs* e as evidências mostrando o requisito de qualidade sendo atendido.

Cenário 1 e 2

Atributo de Qualidade:	Segurança
Requisito de Qualidade:	O acesso as páginas do sistema devem ser feitos mediante autenticação através de login e senha a o acesso aos recursos privados das APIs devem exigir autenticação via token
Preocupação:	
Impossibilitar o acesso aos recursos protegidos do sistema	
Cenário(s):	
Cenário 1 e 2	
Ambiente:	
Sistema em operação normal	
Estímulo:	
Usuário tentando acessar páginas privadas do sistema ou recursos das APIs sem autenticação prévia.	
Mecanismo:	
Criar mecanismo de autenticação e autorização centralizado através de API, redirecionando os usuários não autenticados para tela de login e impedindo acesso aos recursos privados das APIs.	
Medida de Resposta:	
O usuário deve ser redirecionado para tela de autenticação na interface do usuário ao tentar acessar uma tela protegida sem autenticação ou receber o status HTTP 401 informando que o usuário não está autorizado a acessar o recurso ao tentar acessar as APIs privadas. Os usuários devem conseguir acessar páginas públicas sem necessidade de autenticação.	
Considerações sobre a arquitetura:	
Riscos:	O gerenciamento de autenticação e autorização é um ponto crítico para a segurança do sistema e falhas neste recurso podem provocar acesso a

	informações privadas e dados sensíveis dos usuários. A utilização de um recurso de autenticação centralizado é uma maneira simples de proteger todos os serviços que compõem o sistema.
Pontos de Sensibilidade:	Servidor de aplicação operando em modo HTTP e HTTPS.
Tradeoff:	Não há.

Evidências do cenários 1

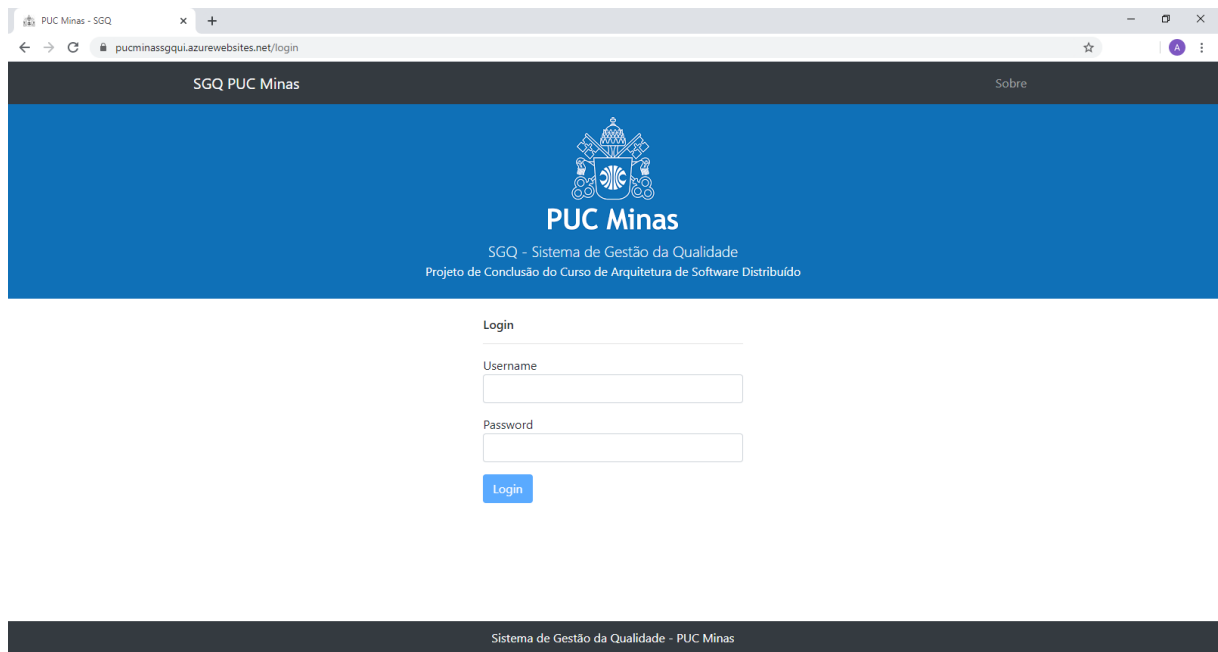


Figura 12 – Usuário é redirecionado para tela de login ao tentar acessar página privada do sistema sem autenticação.

A imagem acima apresenta uma tela de login exibida após a tentativa de acesso a uma página privada do sistema sem autenticação.



Figura 13 - Usuário acessa tela inicial do sistema após realização de login.

Após realizar a autenticação, o usuário tem acesso ao sistema, como pode ser visto na imagem acima.

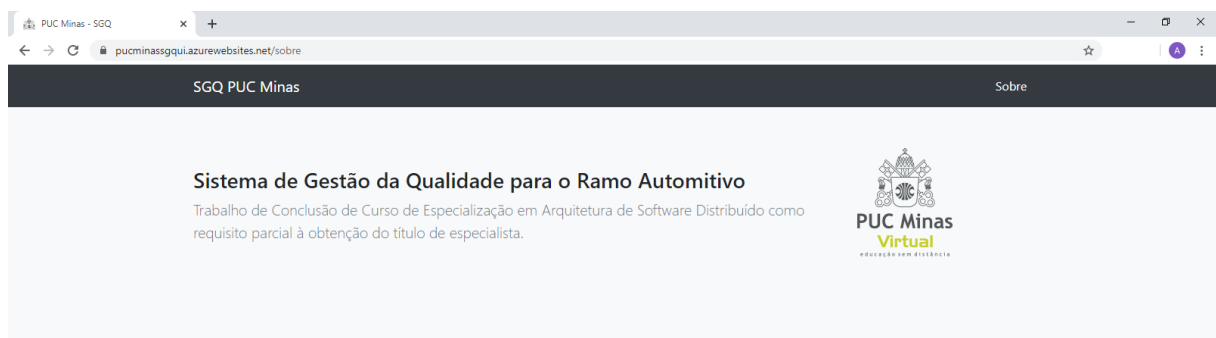


Figura 14 - Usuário acessa a página publica "sobre" sem a necessidade de autenticação.

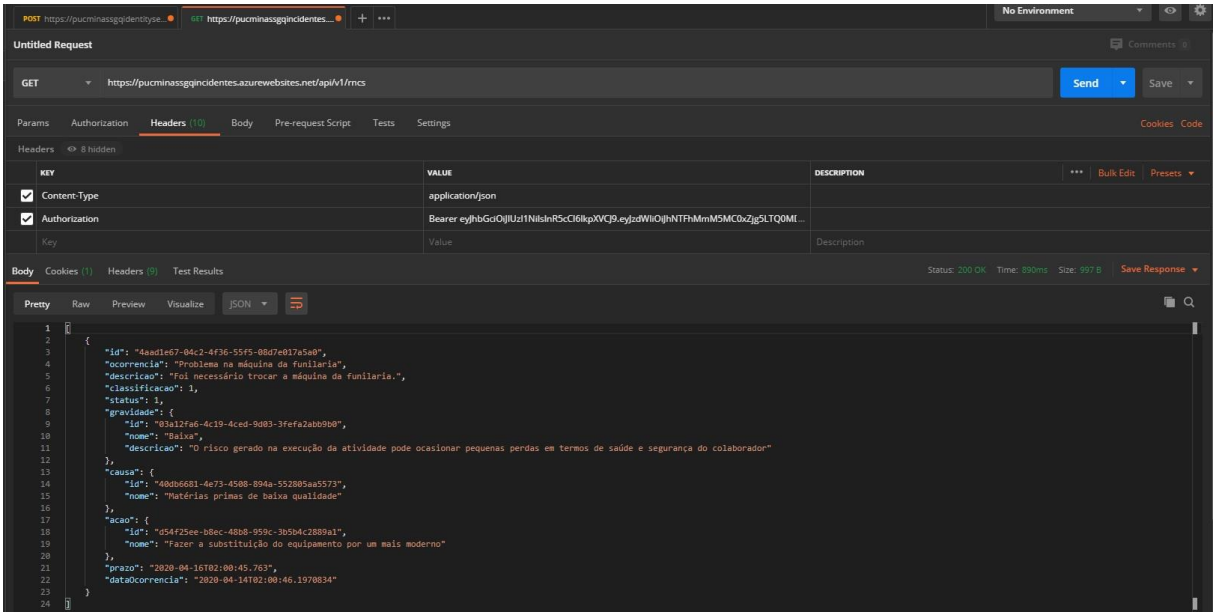


Figura 17 - Consulta realizada na API de controle de incidentes e problemas realizada com a autorização realizada com o token de acesso.

Após a realização de autenticação na API de autorização e autenticação, o usuário recebe um token de acesso que permite o acesso as outras APIs do sistema, como pode ser visto nas duas figuras anteriores.

Cenário 3:

Atributo de Qualidade:	Segurança e usabilidade
Requisito de Qualidade:	O acesso as funcionalidades do sistema devem ser feitos mediante autorização por perfil e o sistema deverá apresentar ao usuário apenas os recursos que ele tem acesso.
Preocupação:	
Possibilita o acesso apenas ao módulos e recursos aos quais o perfil do usuário tem acesso.	
Cenário(s):	
Cenário 3	
Ambiente:	
Sistema em operação normal	
Estímulo:	
Acessar o sistema com um usuário do perfil gerente	
Mecanismo:	
Vincular cada usuário do sistema a um ou mais perfis e fazer com que cada módulo ou recurso do	

sistema só possa ser acessado por um determinado grupo de perfis. Criar também um mecanismo de filtro para apresentar ao usuário apenas os recursos aos quais o perfil dele tem acesso na aplicação front-end.

Medida de Resposta:

O usuário gerente só tem acessos as funcionalidades de cadastrar RNC, alterar RNC, listar RNC e ver detalhes da RNC no módulo de cadastro de incidentes e problemas. No módulo de controle de processos automotivos, o gerente tem acesso apenas a funcionalidade de listar workflows em andamento.

Considerações sobre a arquitetura:

Riscos:	O gerenciamento de perfis de acesso é crítico para garantir que os recursos do sistema sejam acessados apenas pelos usuário com autorização de acesso, de acordo com suas funções estratégicas dentro da organização. Falhas nesse cenário podem comprometer informações sensíveis dentro da organização.
Pontos de Sensibilidade:	Servidor de autenticação operando em modo HTTP e HTTPS.
Tradeoff:	Não há.

Evidências do Cenário 3:

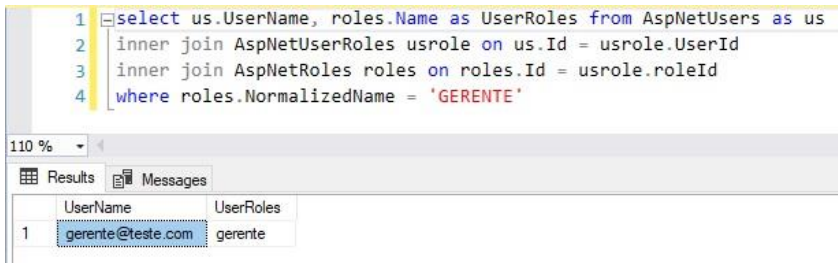


Figura 18 - Usuário gerente@teste.com vinculado ao perfil gerente na base de dados.

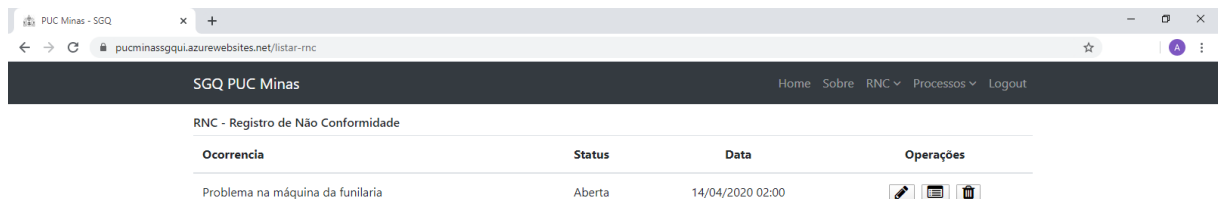
A figura acima mostra que o usuário gerente@teste.com está vinculado ao perfil “gerente” no banco de dados.



Sistema de Gestão da Qualidade - PUC Minas

Figura 19 - Ao entrar no sistema usando um usuário com perfil gerente é possível ver no menu apenas os recursos disponíveis a este perfil.

A figura acima mostra que ao logar no sistema com um usuário do perfil “gerente” o sistema apresenta no menu apenas as funcionalidades que este perfil tem acesso. As funcionalidades de listar RNC, alterar RNC, ver detalhes da RNC e remover RNC são acessíveis pelo menu através do caminho RNC -> Listar, visto na imagem abaixo.



Sistema de Gestão da Qualidade - PUC Minas

Figura 20 - Funcionalidades de alterar RNC, ver detalhes da RNC e remover RNC na tela de listar RNC.

Cenário 4:

Atributo de Qualidade:	Usabilidade
Requisito de Qualidade:	O sistema deve prover boa usabilidade.
Preocupação:	
Garantir que o usuário veja apenas os recursos aos quais tem acesso e que ele seja informado sobre o resultado de suas operações.	
Cenário(s):	
Cenário 4	
Ambiente:	
Sistema em operação normal	
Estímulo:	
Acessar o sistema com usuário do perfil engenheiro	
Mecanismo:	
Prover uma navegação facilitada ao usuário apresentando apenas os recursos que ele pode executar e mostrar através de mensagens de alertas o sucesso ou falha ao executar uma operação, bem como, mostrar modal de confirmação ao realizar operações perigosas.	
Medida de Resposta:	
O usuário deverá ver apenas os recursos aos quais tem acesso. Além disso, o sistema deve prover um mecanismo de mensagens ao executar as operações do sistema.	
Considerações sobre a arquitetura:	
Riscos:	A usabilidade do sistema é um fator crucial para o bom uso do sistema, evitando que sejam feitas operações indesejadas e dando a garantia ao usuário da realização de suas operações, sejam bem ou mal sucedidas.
Pontos de Sensibilidade:	Servidor de aplicação operando em modo HTTP e HTTPS.
Tradeoff:	Não há.

Evidências do Cenário 4:



Figura 21 - Acesso ao sistema com um usuário do perfil "engenheiro" apresentando apenas os recursos permitidos a este perfil.

A imagem acima apresenta os recursos disponíveis ao perfil engenheiro. As próximas três imagens apresentam exemplos de mensagens mostradas pelo sistema para deixar o usuário ciente do resultado de suas operações.

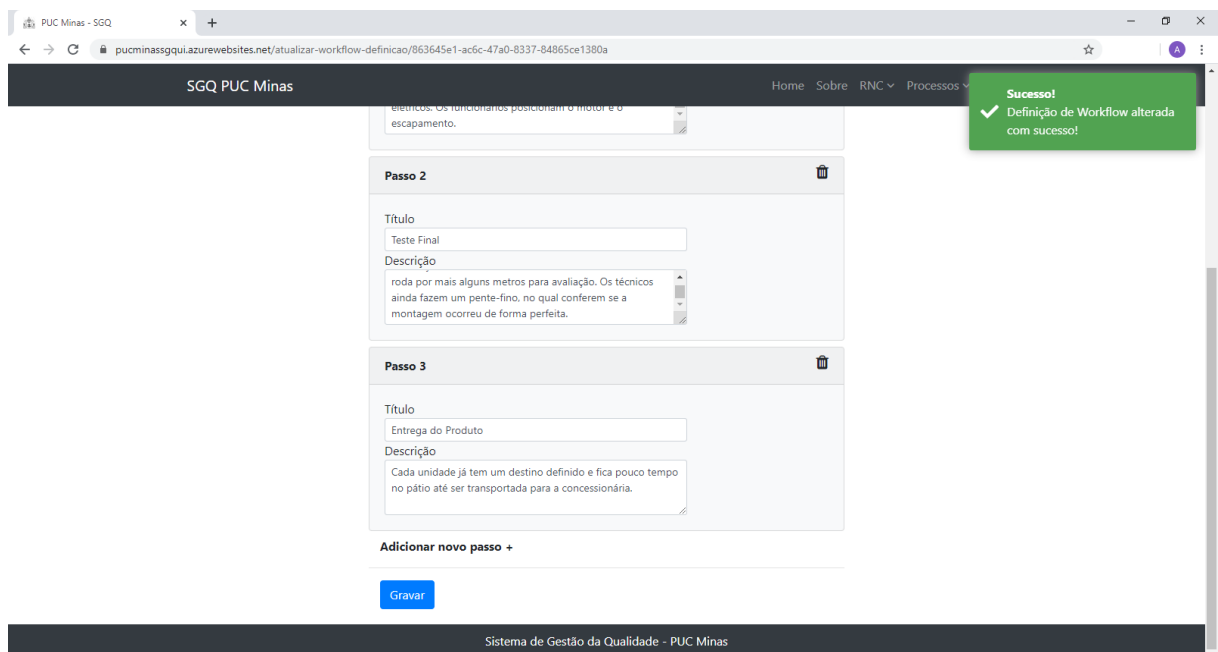


Figura 22 - Sistema apresentando mensagem informando o sucesso de uma operação.

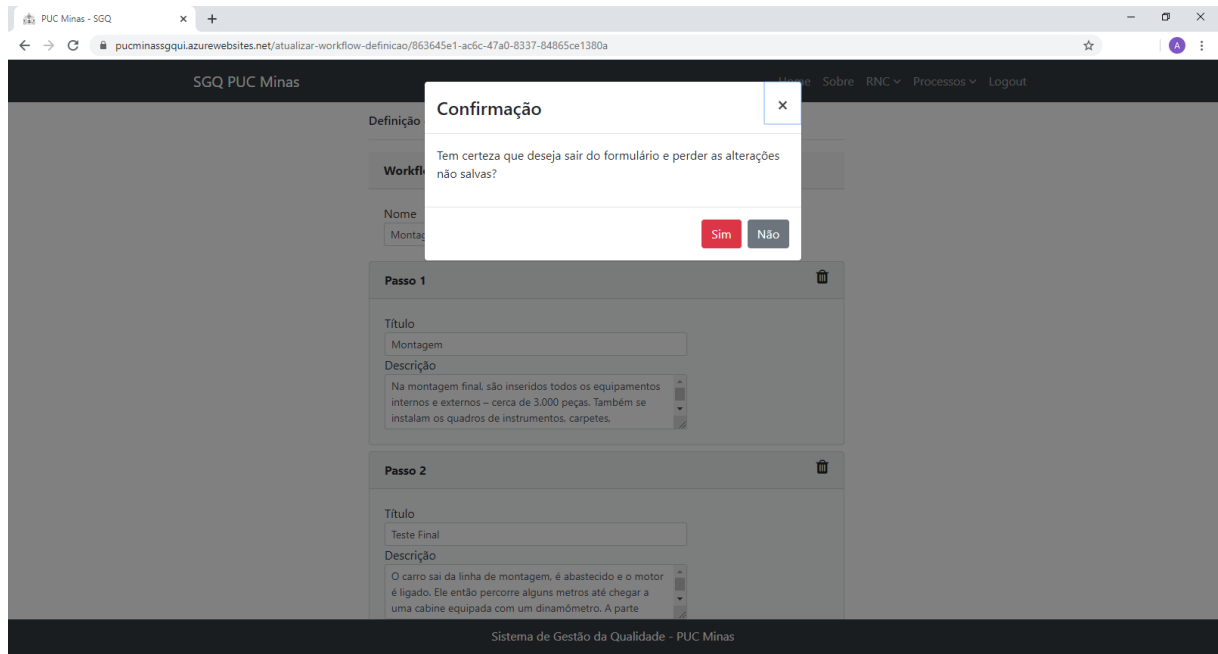


Figura 23- Mensagem de confirmação apresentada ao usuário ao tentar sair de um formulário antes de salvar as alterações.

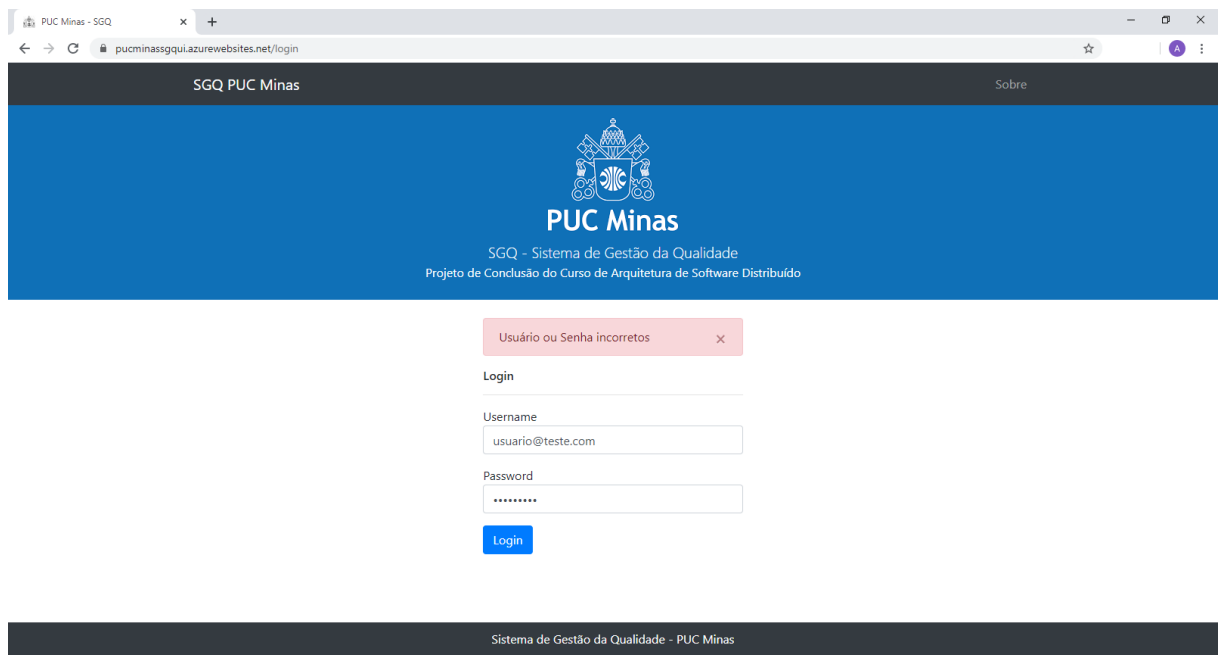


Figura 24 - Sistema apresenta mensagem de erro informando a ocorrência de problemas na operação.

Cenário 5:

Atributo de Qualidade:	Acessibilidade
Requisito de Qualidade:	O sistema deve ser responsivo, adaptando-se aos mais diferentes dispositivos e navegadores.
Preocupação:	
Garantir que o usuário possa usar o sistema de diferentes dispositivos, com telas de diferentes	

tamanhos. Garantir também o uso do sistema através dos principais navegadores do mercado.

Cenário(s):

Cenário 5

Ambiente:

Sistema em operação normal

Estímulo:

Acessar o sistema a partir de diferentes dispositivos e navegadores

Mecanismo:

Criar interface de usuário responsiva, que adapte-se a difentes tamanhos de tela e navegadores.

Medida de Resposta:

O usuário deverá ter uma experiência similar ao usar o sistema a partir de diferentes dispositivos e navegadores.

Considerações sobre a arquitetura:

Riscos:	A acessibilidade do sistema é um fator muito importante para garantir que o usuário não ficará limitado aos dispositivos ou navegadores que poderá usar.
Pontos de Sensibilidade:	Não há.
Tradeoff:	Não há.

Evidências do Cenário 5:

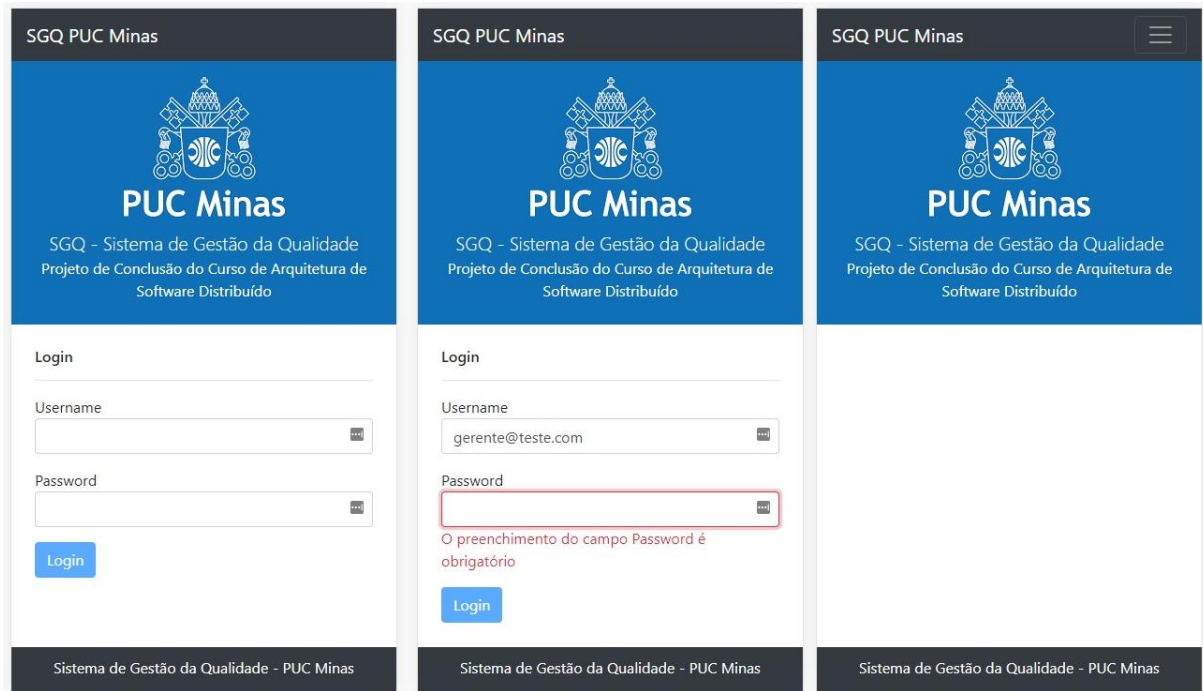


Figura 25 - Telas de Login e Home acessadas de um dispositivo móvel.

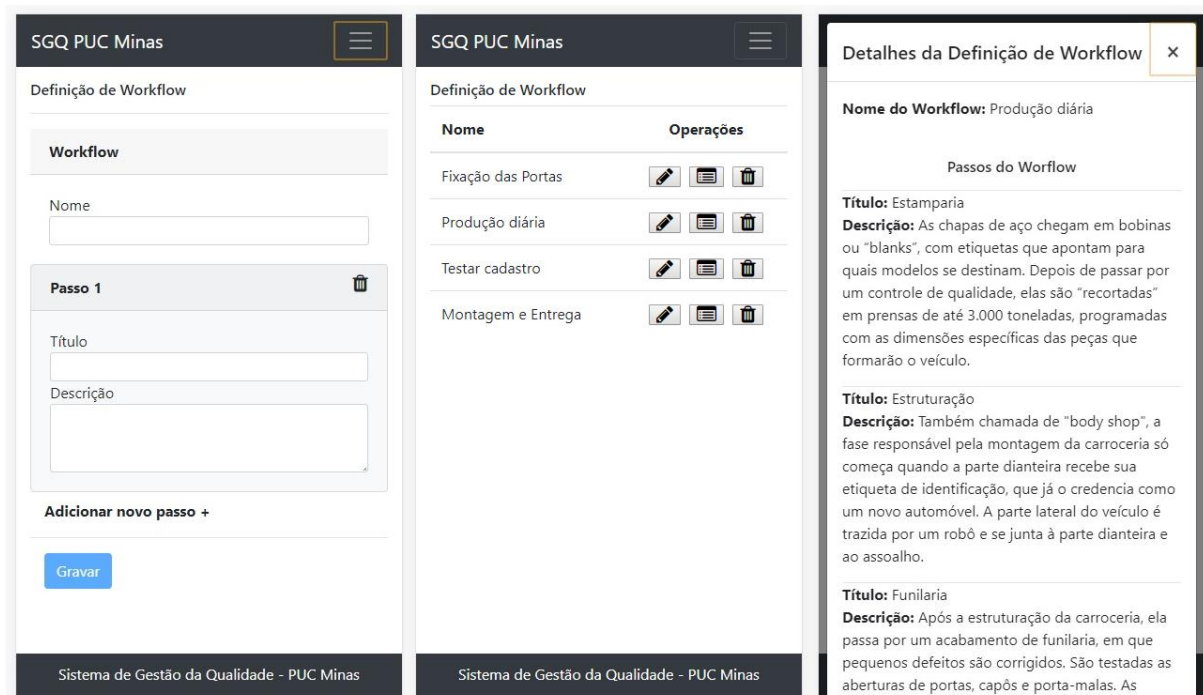


Figura 26 - Telas de cadastro de definição de workflow, listagem de definição de workflow e detalhes da definição de workflow acessada a partir de um dispositivo móvel.

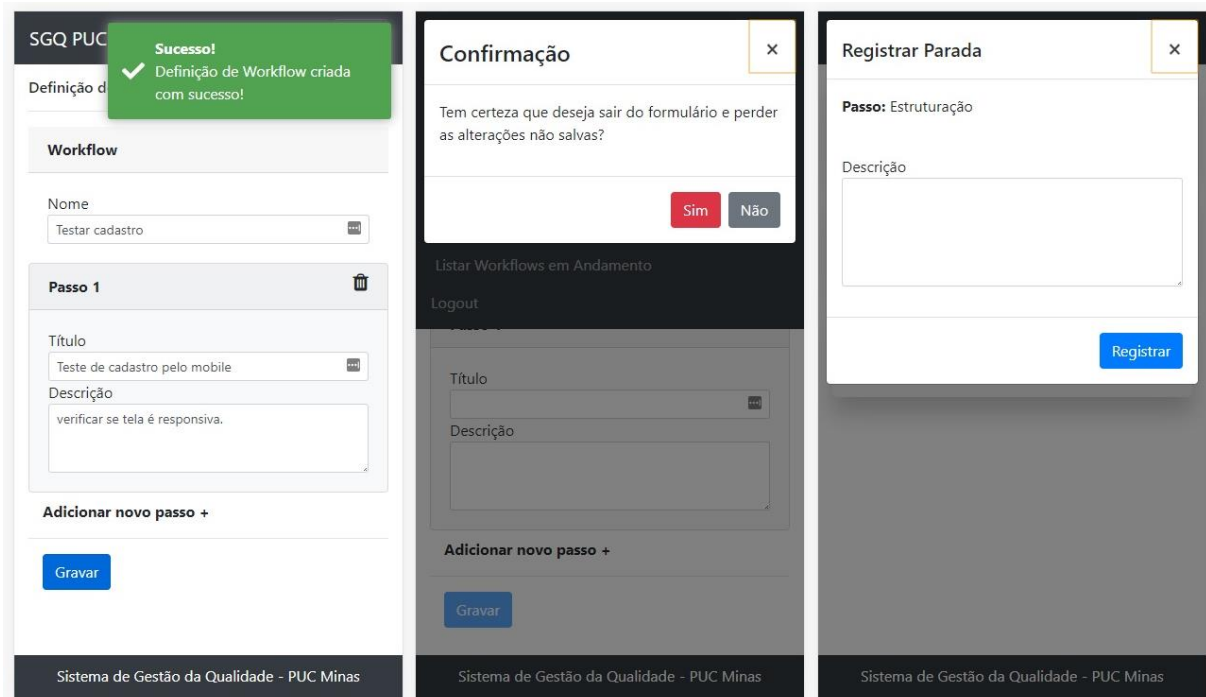


Figura 27 - Mensagem de sucesso, modal de confirmação e tela de registro de parada vistos em um dispositivo móvel.



Figura 28 - Telas de execução de workflow e tela sobre vistas de um dispositivo móvel.

A sequência de quatro imagens acima mostram o uso do sistema a partir de um dispositivo móvel.

6.4. Resultado

Dados os atributos de qualidade, foi possível realizar uma validação arquitetural com o objetivo de analisar esses atributos. Verifiquei que a arquitetura proposta atende bem as necessidade do projeto e foi possível ver a possibilidade de realização de melhorias. Os pontos fracos e fortes da proposta foram percebidos através da execução de diferentes cenários.

Os seguintes atributos de qualidade foram considerados e validados:

Requisitos Não Fncionais	Testado	Homologado
Segurança: O sistema deve apresentar altos padrões de segurança.	SIM	SIM
Usabilidade: O sistema deve prover boa usabilidade.	SIM	SIM
Acessibilidade: O sistema deve possuir design responsivo que funcione bem em ambientes Web e móveis.	SIM	SIM

Foi possível identificar pontos fortes que devem ser mantidos e alguns pontos que podem ser melhorados. Para cada módulo do sistema foi criado um microserviço, cada um contendo sua própria base de dados independente. Esta abordagem apresenta muitos pontos positivos, pois permite que cada microserviço seja criado, mantido e publicado separadamente. No entanto, essa separação também adiciona complexidade na comunicação e na realização de operações que envolvem vários microserviços. Outro ponto importante é que os microserviços foram desenvolvidos com o objetivo de separação de módulos e podem ser repensados futuramente para uma separação por contextos delimitados ou até por função, permitindo que o foco seja direcionado mais para o problema e menos para a arquitetura, além de permitir que pequenas partes do sistema possam ser escaladas separadamente.

A utilização de uma aplicação SPA utilizando Angular e Bootstrap facilitou muito o desenvolvimento de uma aplicação responsiva. O Angular permite a criação de uma aplicação composta por inúmeros componentes, assim, cada componente pode ser desenvolvido para realizar apenas uma função específica, facilitando o entendimento e manutenção. Outro ponto importante é o tempo de carregamento das páginas, que se torna muito reduzido devido ao fato da aplicação trazer todos os recursos necessários na primeira requisição e apenas adicionar os dados extras nas demais.

Foi proposto um serviço exclusivo para cuidar das autenticações e autorizações através de tokens de acesso, com isso, há um ponto único que provém acesso a todas as API do sistema. Embora o resultado desta abordagem tenha sido satisfatório, a manutenção de uma aplicação de segurança pode tornar-se problemática dependendo das necessidades que podem surgir. Como alternativa, pode-se passar essa funcionalidade para um serviço especializado pronto, como o Identity Server.

O controle de acesso da solução foi feito por perfil, que facilita a manutenção das atribuições do que cada usuário pode fazer, mas pode trazer problemas de segurança nos casos em que usuários que deveriam ter atribuições diferentes acabam sendo incluídos no mesmo perfil por não haver perfis suficientes que cubram cada especificidade de acesso que possa surgir. Uma possível melhoria é realizar o controle de acesso por claims, de forma que cada funcionalidade esteja acessível para uma claim específica. Desta forma, o acesso passa a ser individualizado, pois o usuário terá acesso de acordo com as claims que possui.

A arquitetura final mostrou-se bem eficiente, e embora tenham sido encontradas algumas limitações e pontos de melhoria, demonstrou ser uma arquitetura com possibilidade de crescimento e de fácil manutenção, de forma que as qualidades apresentadas por ela justificam o seu uso.

7. Conclusão

Este trabalho apresentou uma proposta arquitetural para uma aplicação de gestão da qualidade para o ramo automotivo. Entende-se que os objetivos arquiteturais foram atingidos e validados através de prova de conceito. Algumas limitações e pontos de melhoria foram apresentados, mas nada que impacte na aceitação da proposta.

REFERÊNCIAS

Angular. **Documentation**. Disponível em: <https://angular.io/docs>. Acesso em: 18 de março de 2020.

Bezerra, Eduardo. **Princípios de análise e projeto de sistemas com UML**. Rio de Janeiro: Elsevier, 2007.

Bootstrap. **Documentation**. Disponível em: <https://getbootstrap.com/docs/4.4/getting-started/introduction/>. Acesso em: 29 de março de 2020.

Medeiros, Ernani Sales de. **Desenvolvendo software com UML 2.0: definitivo**. São Paulo: Pearson Makron Books, 2004.

APÊNDICES

URL do front-end no Azure: <https://pucminassgqui.azurewebsites.net/>

URL das APIs no Azure:

- Autenticação e Autorização: <https://pucminassgqidentityservice.azurewebsites.net/swagger>
- Cadastro de Incidentes: <https://pucminassgqincidentes.azurewebsites.net/swagger>
- Controle de Processos: <https://pucminassgqprocessos.azurewebsites.net/swagger>

URL do repositório no GitHub: <https://github.com/alessandroibes/puc-minas-tcc>

URL da apresentação da PoC no Youtube: <https://www.youtube.com/watch?v=nwObo-htBWQ>

Dados de acesso:

- Operador
 - Usuário: operador@teste.com
 - Senha: Puc@2020
- Gerente de Qualidade
 - Usuário: gerente@teste.com
 - Senha: Puc@2020
- Engenheiro
 - Usuário: engenheiro@teste.com
 - Senha: Puc@2020
- Gestor
 - Usuário: gestor@teste.com
 - Senha: Puc@2020