**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):

Belo Horizonte

2020

*Dedico este trabalho a Deus, autor e consumador 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.*

**AGRADECIMENTOS**

O texto de “agradecimentos” é opcional. Se quiser, pode escrevê-lo sobre este texto. Se não, basta apagar o conteúdo desta página. (Formatação: fonte Arial 12, texto justificado, com espaçamento de 1,5.)

**RESUMO**

O resumo deve apresentar, de forma sucinta, os pontos relevantes do trabalho em um parágrafo único e espaço entre linhas de 1,5. O resumo deve conter a apresentação do problema, uma breve descrição da aplicação e os aspectos relevantes de sua arquitetura. O resumo também deve apresentar evidências da avaliação arquitetural. Ele deve conter de 100 a 250 palavras. Logo abaixo do resumo deve-se listar as palavras-chave que representam o conteúdo do estudo, separadas entre si por vírgula (ver exemplo a seguir).

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

**SUMÁRIO**

[1. Objetivos do trabalho 7](#_Toc35370935)

[2. Descrição geral da solução 8](#_Toc35370936)

[2.1. Apresentação do problema 8](#_Toc35370937)

[2.2. Descrição geral do software (Escopo) 8](#_Toc35370938)

[3. Definição conceitual da solução 9](#_Toc35370939)

[3.1. Requisitos Funcionais 9](#_Toc35370940)

[3.2 Requisitos Não-Funcionais 12](#_Toc35370941)

[3.3. Restrições Arquiteturais 15](#_Toc35370942)

[3.4. Mecanismos Arquiteturais 15](#_Toc35370943)

[4. Modelagem e projeto arquitetural 16](#_Toc35370944)

[4.1. Modelo de casos de uso 16](#_Toc35370945)

[4.2. Descrição resumidade de casos de uso 19](#_Toc35370946)

[4.3. Modelo de componentes 26](#_Toc35370947)

[4.4. Modelo de implantação 29](#_Toc35370948)

[4.5. Modelo de dados (opcional) 31](#_Toc35370949)

[5. Prova de Conceito (POC) / protótipo arquitetural 31](#_Toc35370950)

[5.1. Implementação e Implantação 31](#_Toc35370951)

[5.2 Interfaces/ APIs 32](#_Toc35370952)

[6. Avaliação da Arquitetura 32](#_Toc35370953)

[6.1. Análise das abordagens arquiteturais 32](#_Toc35370954)

[6.2. Cenários 32](#_Toc35370955)

[6.3. Avaliação 32](#_Toc35370956)

[6.4. Resultado 33](#_Toc35370957)

[7. Conclusão 33](#_Toc35370958)

[REFERÊNCIAS 34](#_Toc35370959)

[APÊNDICES 35](#_Toc35370960)

[CHECKLIST PARA VALIDAÇÃO DOS ITENS E ARTEFATOS DO TRABALHO 36](#_Toc35370961)

## 1. Objetivos do trabalho

O propósito deste trabalho é apresentar uma proposta arquitetural para o desenvolvimento de uma plataforma de gestão de 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 estapa 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 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 cumprimeto 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, apontando a ocorrência de incidentes e problemas ocorridos durante o processo.
* **Engenheiro**: responsável por cadastrar os workfows 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 incidentes e problemas ocorridos, acompanhando sua resolução. É de responsabilidade do Gerente de Qualidade a aprovação da resolução de uma não conformidade cadastrada.
* **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 não confirmidades encontradas. O cadastro será feito por um Gerente de Qualidade após o recebimento de uma notificação que será enviada após a geração de um Registro de Não Conformidade (RNC) no Módulo de Controle de Processos Automotivos.
  + Os usuários com perfil Engenheiro ou Gerente de Qualidade podem consultar os RNC cadastrados no sistema.
  + 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, um Engenheiro deverá corrigir a resolução do RNC.
  + 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 uma lista de atividades diárias (workflow) que deverá ser executada por um Operador.
  + O sistema deve permitir que Operadores, Gerentes de Qualidade e Engenheiros possam consultar os workflows em andamento.
  + O sistema deve permitir que um Engenheiro altere um workflow cadastrado.
  + O sistema deve permitir que o Operador preencha o andamento das atividades realizadas do workflow, sendo possível informar sobre incidentes e problemas, caso ocorram. O sistema deve emitir uma notificação após o cadastramento de um incidente ou problema através do Serviço de Notificação.
* **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 auxíliar 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 núvem 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 confimidades 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 suguir são apresentados os requisitos não funcionais do sistema descritos no padrão estímulo-resposta:

**Usabilidade**:

* 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.

|  |  |
| --- | --- |
| **Estimulo** | 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 funcionamem 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 |

**Intepoperabilidade:**

* 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 *Compliance* |
| **Ambiente** | Funcionamento, carga normal |
| **Artefato** | Módulo de C*ompliance* |
| **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 |

**Segurança:**

* O sistema deve restringir acesso aos usuários não autenticados no sistema.

|  |  |
| --- | --- |
| **Estimulo** | Acessar uma página privada do sistema pela URL sem estar logado no sistema |
| **Fonte do Estimulo** | 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 |

**Confiabilidade:**

* O sistema deve ter alta disponibilidade, operando initerruptamente ou com poucas paradas.

|  |  |
| --- | --- |
| **Estimulo** | Monitoramento do sistema a cada 30 segundos para verificar quedas |
| **Fonte do Estimulo** | Funcionalidade de *Healthy Check* |
| **Ambiente** | Funcionamento, carga normal |
| **Artefato** | Qualquer módulo do sistema |
| **Resposta** | Um desenvolvedor acessa o *Elmah.io* e verifica que o sistema não apresenta quedas constantes |
| **Medida da resposta** | Ao detectar uma indisponibilidade do *front-end*, 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 *Elmah.io* |

* 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 mantem 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 mantem 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 continínua.
* O sistema deve conter automação de testes no pipeline de integração contínua.
* O sistema deve permitir a sua implantação em nuvem ou *on-primise*.
* 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 |  |
| Notificações | Notificação de mensagens aos usuários via email |  |
| 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 Kubernets |
| 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.

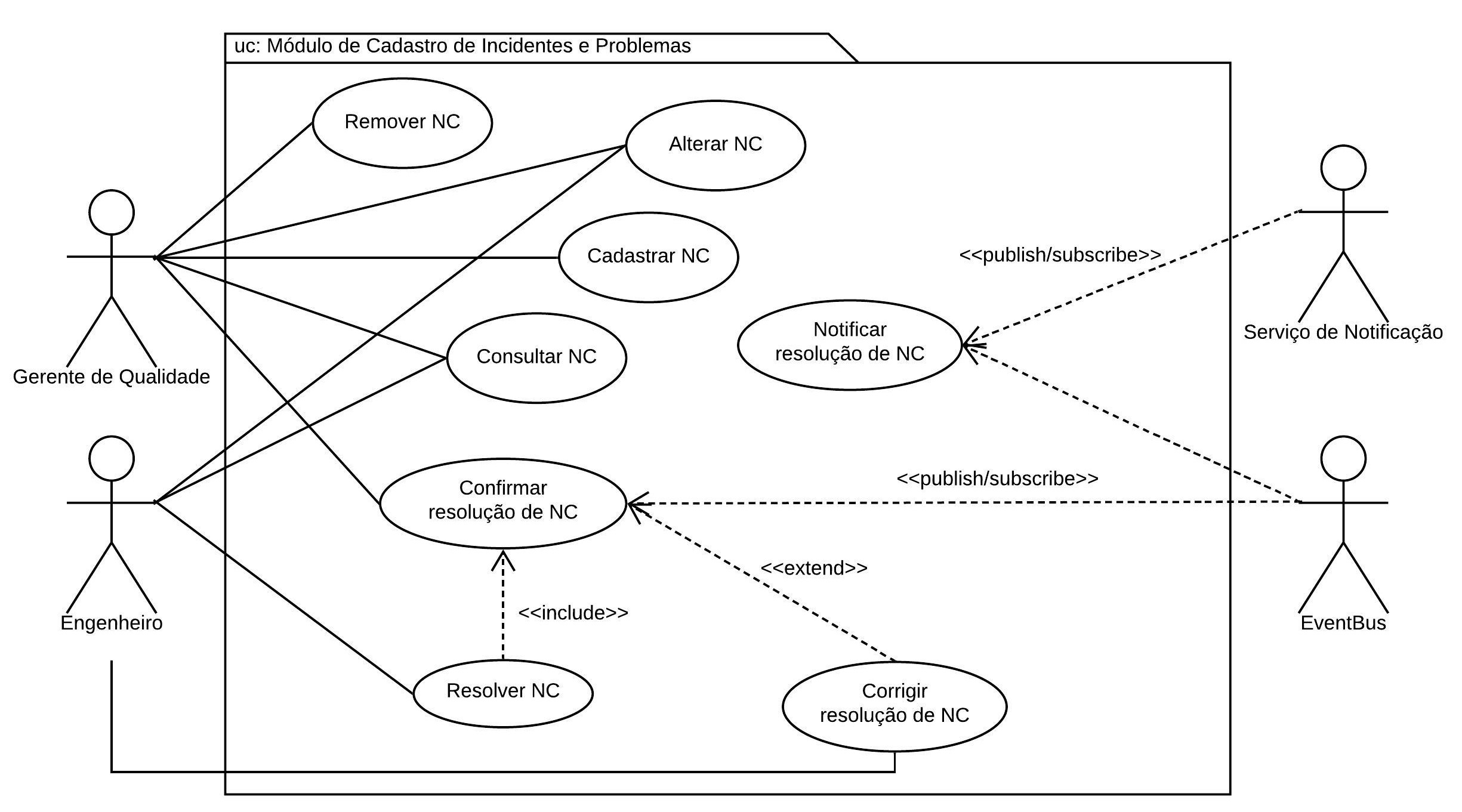


Figura - Diagrama de Casos de Uso: Módulo de Cadastro de Incidentes e Problemas

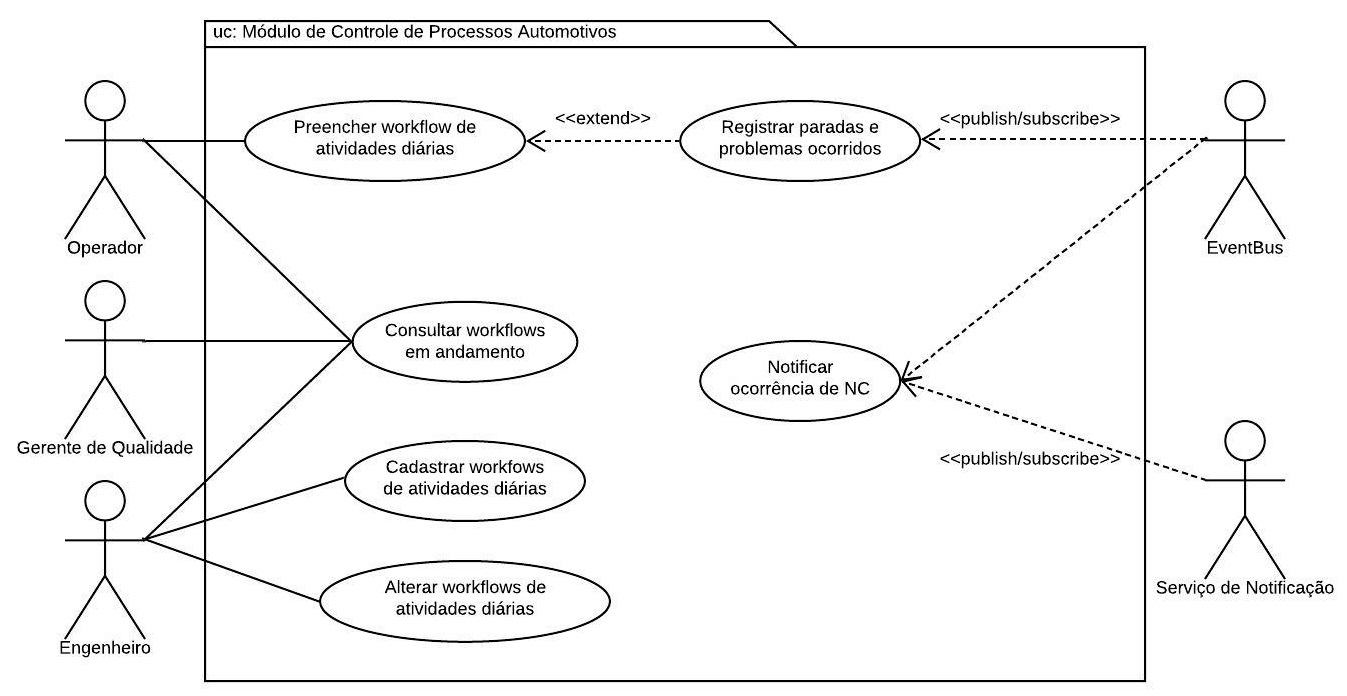


Figura - Diagrama de Casos de Uso: Módulo de Controle de Processos Automotivos

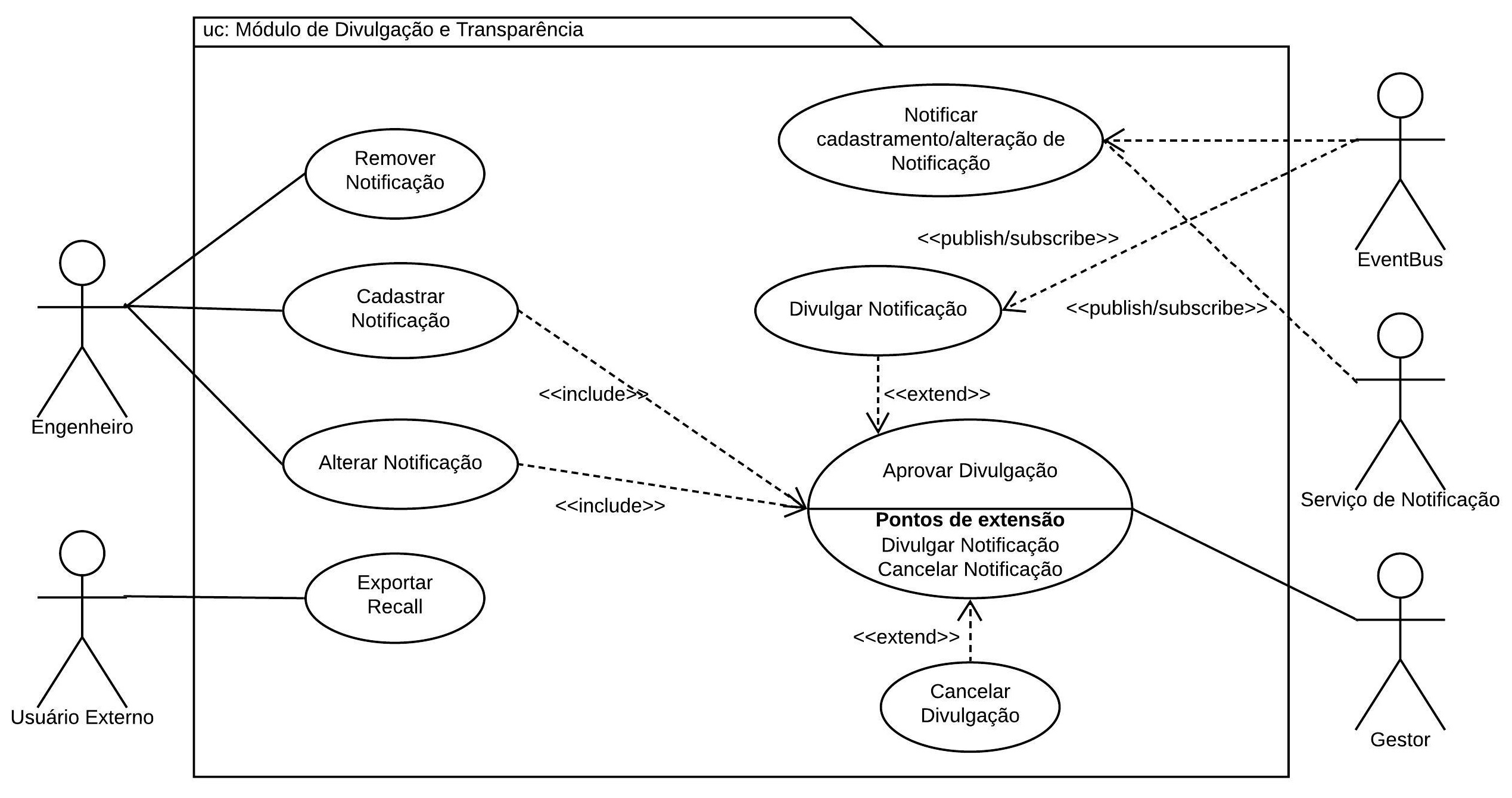


Figura - Diagrama de Casos de Uso: Módulo de Divulgação e Transparência

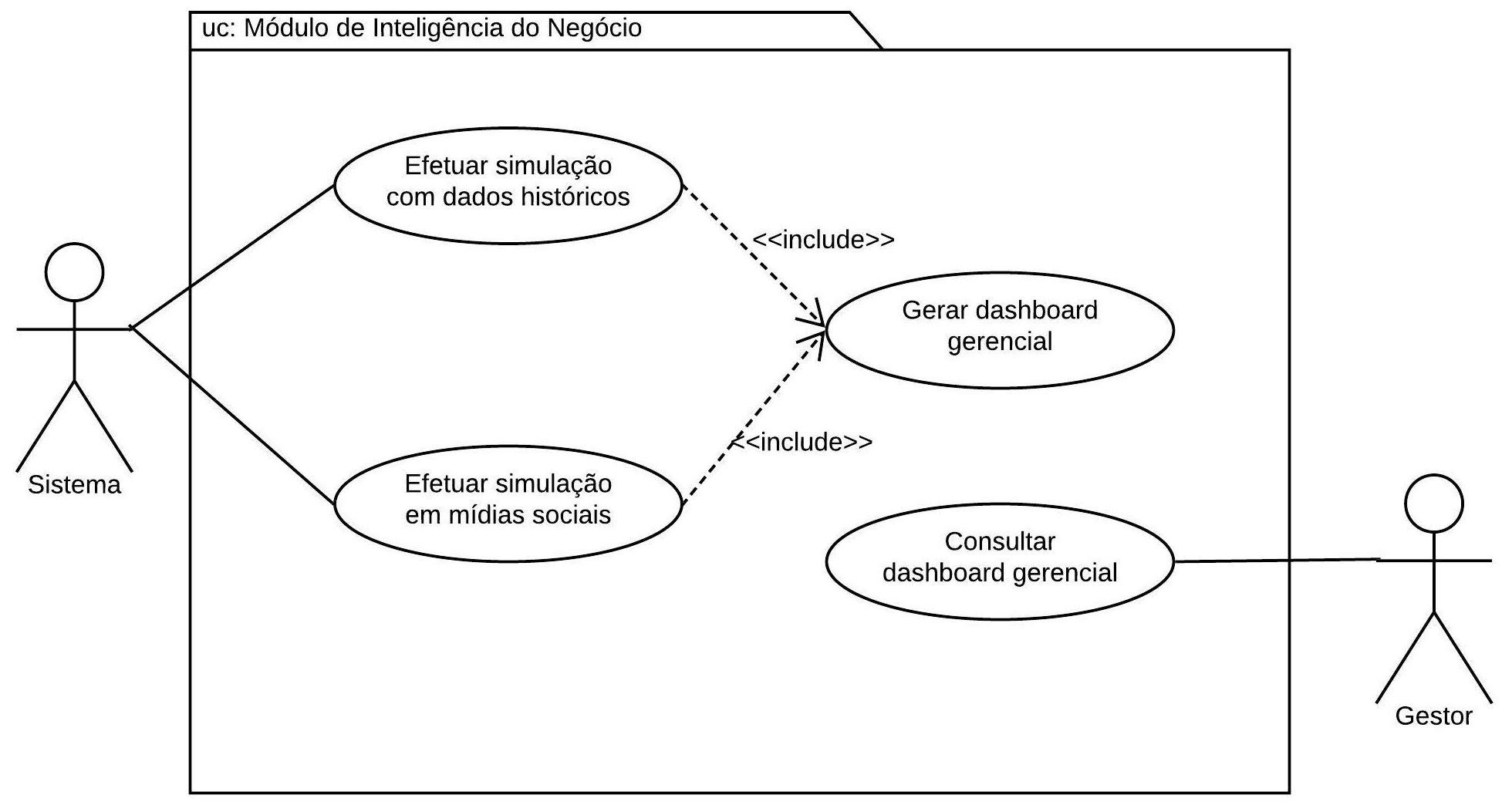


Figura - Diagrama de Casos de Uso: Módulo de Inteligência do Negócio

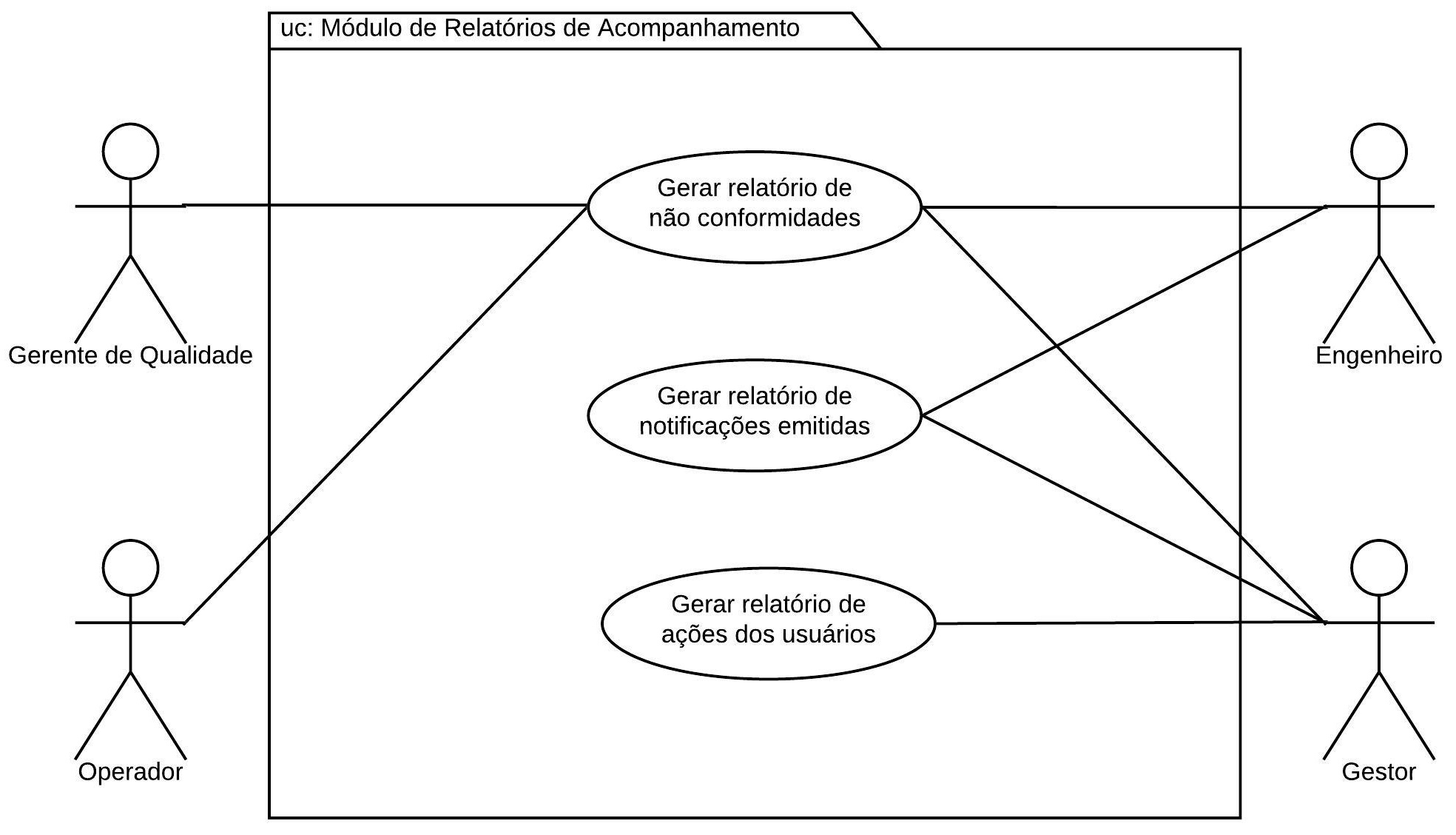


Figura 5 - Diagrama de Casos de Uso: Módulo de Relatórios de Acompanhamento

## 4.2. Descrição resumidade de casos de uso

Caso de uso UC001: cadastrar NC (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 NC** |
| **Atores:** Gerente de Qualidade | |
| **Descrição:** Como Gerente de Qualidade, eu quero cadastrar os incidentes (não conformidades) ocorridos durante a execução dos processos automotivos que foram apontados por um Operador. | |
| **Pré-Condições:** Operador registra um incidente 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 NC (Não conformidade)

**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 NC** |
| **Atores:** Gerente de Qualidade | |
| **Descrição:** Como Gerente de Qualidade, eu quero remover os incidentes (não conformidades) cadastrados. | |
| **Pré-Condições:** A não conformidade não pode ter sido resolvida por um Engenheiro. | |
| **Valor do negócio:** Para uma correta manutenção das não conformidades no sistema. | |
| **Prioridade:** Baixa | **Estimativa:** 2 |

Caso de uso UC003: consultar NC (Não conformidade)

**Descrição resumida:**

Este caso de uso deve permitir ao Gerente de Qualidade ou Engenheiro consultar as não conformidades cadastradas no sistema.

|  |  |
| --- | --- |
| **UC003** | **Consultar NC** |
| **Atores:** Gerente de Qualidade, Engenheiro | |
| **Descrição:** Como Gerente de Qualidade ou Engenheiro, eu quero consultar as não conformidades cadastradas. | |
| **Valor do negócio:** Para uma correta manutenção das não conformidades do sistema. | |
| **Prioridade:** Alta | **Estimativa:** 2 |

Caso de uso UC004: alterar NC (Não conformidade)

**Descrição resumida:**

Este caso de uso deve permitir ao Gerente de Qualidade ou Engenheiro alterar as não conformidades cadastradas no sistema.

|  |  |
| --- | --- |
| **UC004** | **Alterar NC** |
| **Atores:** Gerente de Qualidade, Engenheiro | |
| **Descrição:** Como Gerente de Qualidade ou Engenheiro, eu quero alterar as não conformidades cadastradas. | |
| **Valor do negócio:** Para uma correta manutenção das não conformidades do sistema. | |
| **Prioridade:** Alta | **Estimativa:** 4 |

Caso de uso UC005: resolver NC (Não conformidade)

**Descrição resumida:**

Este caso de uso deve permitir ao Engenheiro resolver as não conformidades cadastradas no sistema.

|  |  |
| --- | --- |
| **UC005** | **Resolver NC** |
| **Atores:** Engenheiro | |
| **Descrição:** Como Engenheiro, eu quero resolver as não conformidades cadastradas. | |
| **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 UC006: confirmar resolução de NC (Não conformidade)

**Descrição resumida:**

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

|  |  |
| --- | --- |
| **UC006** | **Confirmar Resolução de NC** |
| **Atores:** Gerente de Qualidade | |
| **Descrição:** Como Gerente de Qualidade, eu quero confirmar a resolução das não conformidades resolvidas pelos Engenheiro. | |
| **Pré-Condições:** A não conformidade deve ter sido resolvida 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 UC007: corrigir resolução de NC (Não conformidade)

**Descrição resumida:**

Este caso de uso deve permitir ao Engenheiro corrigir a resolução de uma não conformidade anteriormente resolvida que não tenha sido confirmada por um Gerente de Qualidade.

|  |  |
| --- | --- |
| **UC007** | **Corrigir Resolução de NC** |
| **Atores:** Engenheiro | |
| **Descrição:** Como Engenheiro, eu quero corrigir a resolução das não conformidades resolvidas, mas não confirmadas. | |
| **Pré-Condições:** A não conformidade deve ter sido resolvida por um Engenheiro e não confirmada por um Gerente de Qualidade. | |
| **Valor do negócio:** Para uma correta manutenção das não conformidades do sistema. | |
| **Prioridade:** Média | **Estimativa:** 2 |

Caso de uso UC008: cadastrar workflows de atividades diárias

**Descrição resumida:**

Este caso de uso deve permitir ao Engenheiro cadastrar workflows das atividades diárias que serão executadas pelos Operadores da empresa.

|  |  |
| --- | --- |
| **UC008** | **Cadastrar Workflows de Atividades Diárias** |
| **Atores:** Engenheiro | |
| **Descrição:** Como Engenheiro, eu quero cadastrar workflows de atividades diárias para serem executadas pelos Operadores. | |
| **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 UC009: alterar workflows de atividades diárias

**Descrição resumida:**

Este caso de uso deve permitir ao Engenheiro alterar os workflows de atividades diárias cadastradas no sistema.

|  |  |
| --- | --- |
| **UC009** | **Alterar Workflows de Atividades Diárias** |
| **Atores:** Engenheiro | |
| **Descrição:** Como Engenheiro, eu quero alterar workflows de atividades diárias cadastradas no sistema. | |
| **Valor do negócio:** Para uma correta manutenção dos workflows de atividades diárias já cadastrados no sistema. | |
| **Prioridade:** Média | **Estimativa:** 2 |

Caso de uso UC010: 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.

|  |  |
| --- | --- |
| **UC010** | **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 UC011: 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.

|  |  |
| --- | --- |
| **UC011** | **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 UC012: 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.

|  |  |
| --- | --- |
| **UC012** | **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 UC013: cadastrar notificação

**Descrição resumida:**

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

|  |  |
| --- | --- |
| **UC013** | **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 UC014: 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.

|  |  |
| --- | --- |
| **UC014** | **Alterar Notificação** |
| **Atores:** Engenheiro | |
| **Descrição:** Como Engenheiro eu quero alterar notificações já cadastradas no sistema antes delas serem divulgadas ao público externo. | |
| **Valor do negócio:** Para correção das informações que serão divulgadas ao público externo. | |
| **Prioridade:** Alta | **Estimativa:** 2 |

Caso de uso UC015: 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.

|  |  |
| --- | --- |
| **UC015** | **Remover Notificação** |
| **Atores:** Engenheiro | |
| **Descrição:** Como Engenheiro eu quero remover notificações já cadastradas no sistema antes delas serem divulgadas ao público externo. | |
| **Valor do negócio:** Para correção das informações que serão divulgadas ao público externo. | |
| **Prioridade:** Média | **Estimativa:** 2 |

Caso de uso UC016: aprovar divulgação

**Descrição resumida:**

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

|  |  |
| --- | --- |
| **UC016** | **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 UC017: exportar recall

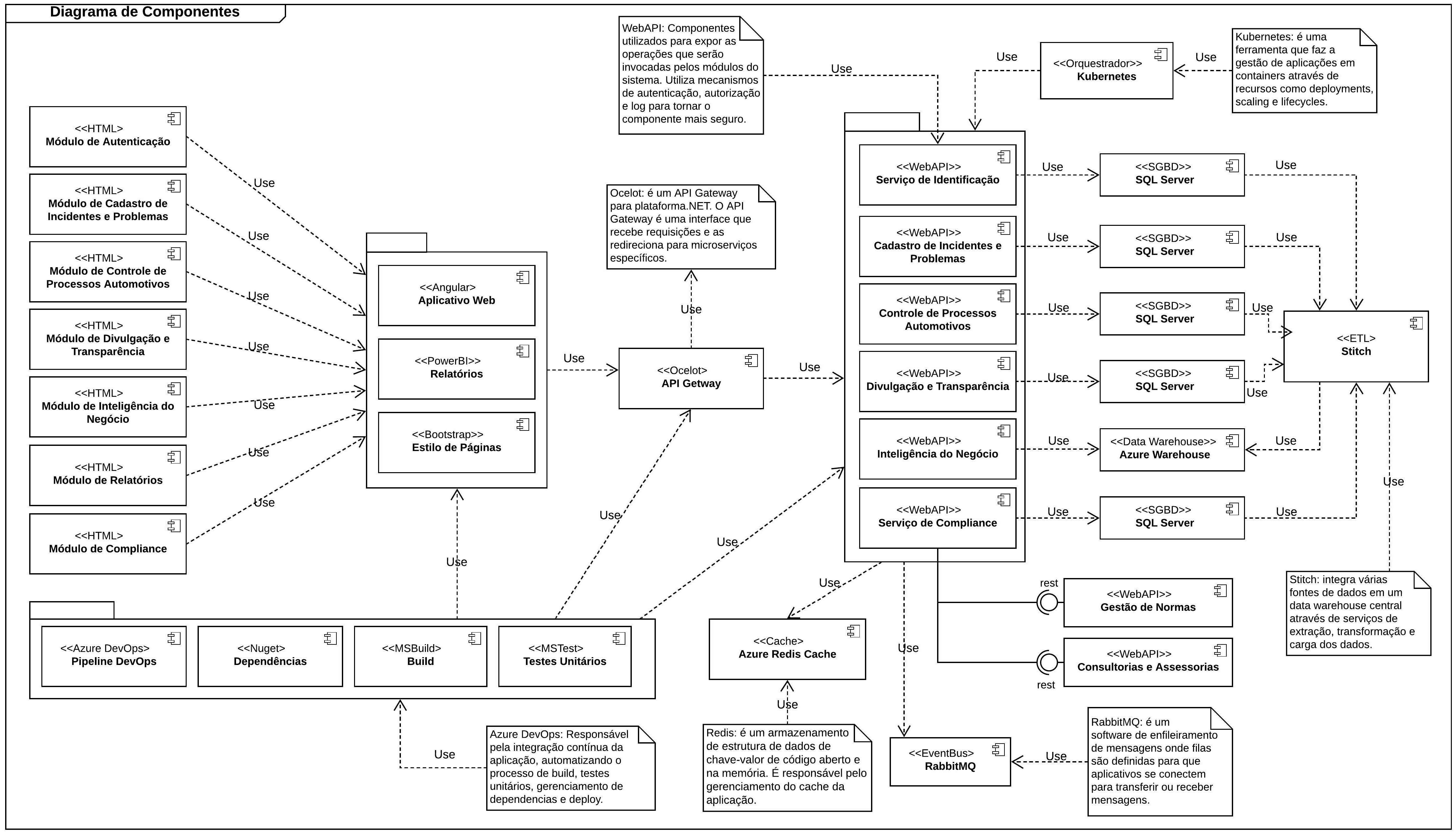
**Descrição resumida:**

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

|  |  |
| --- | --- |
| **UC017** | **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 fornecer interfaces bem definidas de acordo com suas responsabilidades.



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 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. A ferramenta escolhida para implementar esse EventBus foi a RabbitMQ, que é um servidor de mensageria de código aberto.

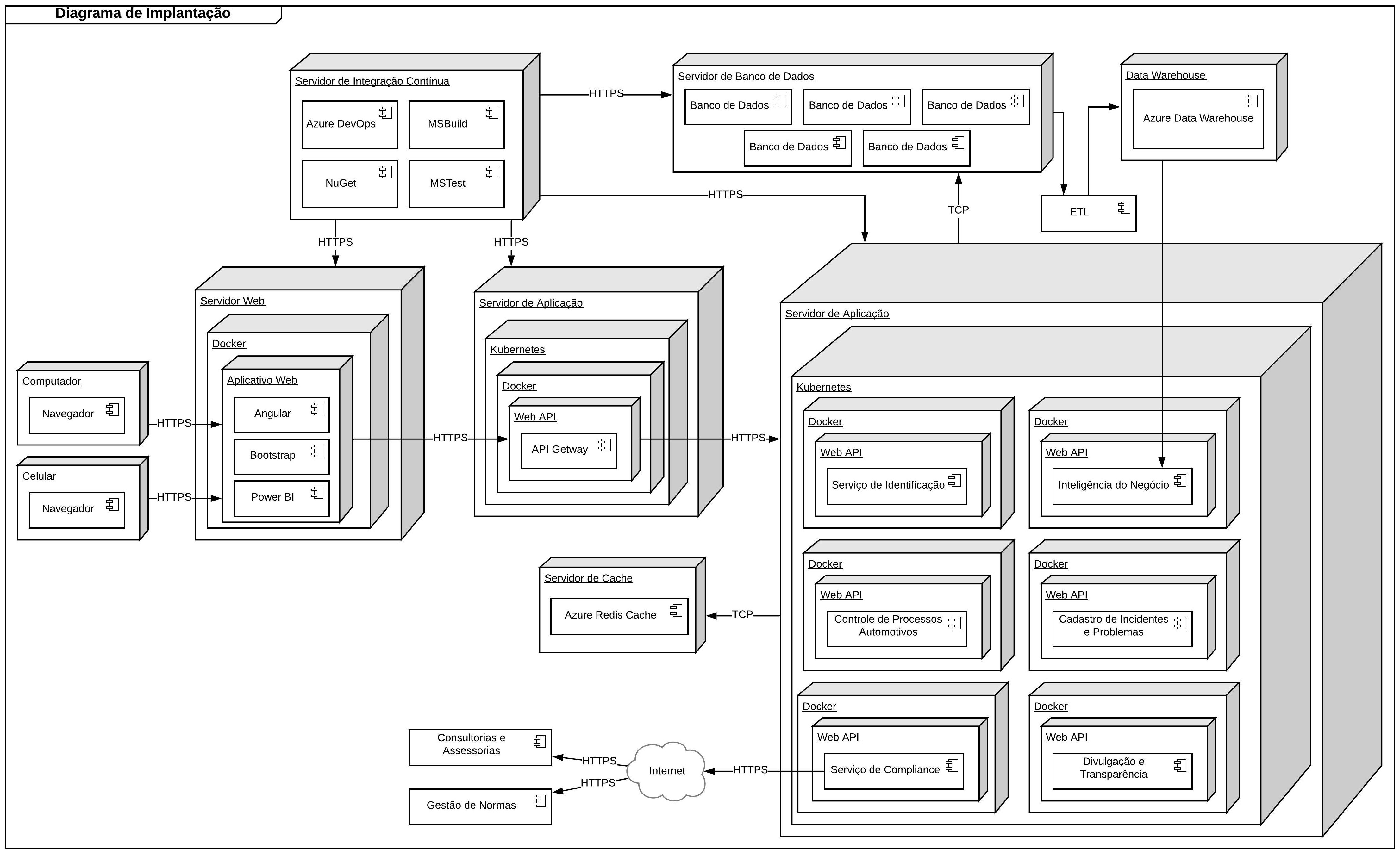
Para garantir conformidade ao requisito funcional de confiabilidade será utilizado o recurso de cache, otimizando a performance do sistema. Com isso, os dados frequentemente acessados pelos serviços serão mantidos em memória, reduzindo tráfego de rede e economizando acesso à base de dados. O Azure Redis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

Os artefatos executáveis sertão 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 contrução 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.

|  |  |
| --- | --- |
| **Componente** | **Descrição** |
| Azure DevOps (Integração contínua da aplicação) | Componentes utilizados para construir, gerenciar as dependências, executar testes unitários de forma automatizada e implantar os módulos da aplicação nos respectivos containers. |
| Aplicativo Web (*SPA – Single-Page Application*) | 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 mamória 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 núvem 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 núvem 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 componetes foram organizados para serem reutilizáveis, fornecendo interfaces bem definidas de acordo com suas responsabilidades. Este modelo de implantação da arquitetura é apresentado abaixo.



A seguir é apresentada a relação dos componentes com os seus respectivos servidores, e como o modelo de implantação deve ser implantado. A arquitetura de cluster para os servidores de aplicação não está explicita no modelo acima, apesar disso, 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 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 informação 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 (opcional)

Caso julgue necessário para explicar a arquitetura, apresente o diagrama de classes ou o diagrama de Entidade-Relacionamento ou as tabelas do banco de dados. Este modelo pode ser essencial caso a arquitetura utilize uma solução de banco de dados distribuídos ou um banco NoSQL.

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

## 5.1. Implementação e Implantação

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

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

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

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

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

## 5.2 Interfaces/ APIs

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

# 6. Avaliação da Arquitetura

A avaliação da arquitetura desenvolvida neste trabalho é abordada nesta seção, visando avaliar se atende ao que foi proposto.

## 6.1. Análise das abordagens arquiteturais

Apresente um breve resumo das principais características da proposta arquitetural.

## 6.2. Cenários

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

## 6.3. Avaliação

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

## 6.4. Resultado

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

## 7. Conclusão

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

# REFERÊNCIAS

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

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

# APÊNDICES

URL do front-end no Azure:

URL da API de autenticação no Azure:

URL da API de cadastro de incidentes no Azure:

URL da API de control de processos no Azure:

URL da API de ?? no Azure:

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

URL da apresentação da PoC no Youtube:

**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

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

# 

# CHECKLIST PARA VALIDAÇÃO DOS ITENS E ARTEFATOS DO TRABALHO

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

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