

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

**André Lopes Russo**  
**Vinícius Lopes Russo**

**SISTEMA INTEGRADO DE GESTÃO DE QUALIDADE VOLTADO AO RAMO  
AUTOMOTIVO**

São Paulo  
2020

**André Lopes Russo**  
**Vinícius Lopes Russo**

**SISTEMA INTEGRADO DE GESTÃO DE QUALIDADE VOLTADO AO 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): Pedro Alves de Oliveira

São Paulo  
2020

*“With great power comes great responsibility”*

**(Uncle Ben)**

## **AGRADECIMENTOS**

Em agradecimento a nossos pais Franco Russo e Raquel Ermelinda Lopes Russo, à namorada do Vinícius: Francielly Alves da Silva, e a esposa e filho de André: Raquel de Vitto Baldin Russo e Giovanni de Vitto Baldin Russo, sem os quais não teríamos apoio para o desenvolvimento desse projeto.

## RESUMO

Este trabalho tem como objetivo a criação de um projeto arquitetural que vai desde a descrição do problema, até a construção de uma prova de conceito, passando pelos principais temas e passos necessários para comprovar que a arquitetura escolhida é a melhor, dentre as outras possíveis arquiteturas existentes. O tema escolhido para a criação do documento e desenvolvimento da prova de conceito foi Gestão de Qualidade, pois atualmente no mercado é importante que as empresas se preocupem com a experiência satisfatória de seus clientes, devido à alta competitividade. O objetivo do sistema é melhorar continuamente os produtos, serviços e processos de uma empresa, auxiliando na organização e na auditoria diária.

**Palavras-Chave:** gestão de qualidade, arquitetura de software, projeto de software, requisitos arquiteturais.

## SUMÁRIO

<b>1. Objetivos do trabalho.....</b>	<b>7</b>
<b>2. Descrição geral da solução .....</b>	<b>7</b>
2.1. Apresentação do problema .....	7
2.2. Descrição geral do software.....	8
<b>3. Definição conceitual da solução .....</b>	<b>8</b>
3.1. Requisitos Funcionais.....	8
3.2 Requisitos Não-Funcionais.....	11
3.3. Restrições Arquiteturais .....	15
3.4. Mecanismos Arquiteturais .....	15
<b>4. Modelagem e projeto arquitetural.....</b>	<b>16</b>
4.1. Modelo de componentes.....	16
4.2. Modelo de componentes.....	19
4.3. Modelo de implantação .....	20
4.4. Modelo de dados .....	21
<b>5. Prova de Conceito (POC) / protótipo arquitetural .....</b>	<b>22</b>
5.1. Implementação e Implantação .....	22
5.2 Código .....	24
5.3. Interfaces/ APIs.....	28
<b>6. Avaliação da Arquitetura .....</b>	<b>40</b>
6.1. Análise das abordagens arquiteturais .....	40
6.2. Cenários .....	40
6.3. Avaliação .....	42
6.4. Resultado .....	51
7. Conclusão .....	52
<b>REFERÊNCIAS.....</b>	<b>53</b>
<b>APÊNDICES .....</b>	<b>53</b>

## **1. Objetivos do trabalho**

O objetivo deste trabalho é apresentar um projeto arquitetural para uma aplicação de Gestão de Qualidade, com foco na indústria automotiva.

Os objetivos específicos são:

- Criar o módulo de Cadastro de Incidentes e Problemas: módulo que permite cadastrar e gerir incidentes e problemas relacionados a não conformidades dos produtos da empresa, tais como falhas em veículos e seus insumos (peças).
- Criar o módulo de Controle de Processos Automotivos: módulo que permite a gestão dos processos envolvidos com a atividade automotiva no que se refere à qualidade dos processos e dos produtos, garantindo a qualidade comparando aos padrões pré-estabelecidos.
- Criar o módulo de Divulgação e Transparência: módulo que cuida de tornar as informações de qualidade disponíveis, tanto para o público interno quanto externo.
- Criar o módulo de Inteligência do negócio: módulo que gera informações estratégicas de negócio, com foco na qualidade, consistindo em uma aplicação de Business Intelligence (BI).
- Criar o módulo de Compliance: módulo integrado aos sistemas externos de Gestão de Normas e de Consultorias e Assessorias, este módulo visa garantir a aderência da empresa às normas nacionais e internacionais do setor automotivo, assim como aprimorar os controles de governança corporativa existentes.

## **2. Descrição geral da solução**

### **2.1. Apresentação do problema**

Com um mercado competitivo é importante que as empresas se preocupem com a experiência satisfatória de seus clientes. Existe a necessidade de melhorar continuamente seus produtos, serviços e processos, para obter resultados sempre melhores.

Atualmente existem diversas empresas que se utilizam de métodos não digitais ou não controlados como forma de estabelecer um padrão entre seus procedimentos, no cenário citado acima podemos ver como exemplo empresas que utilizam um servidor FTP no qual qualquer um pode editar, adicionar e excluir um documento, causando com o tempo uma diferença no conhecimento dos funcionários sobre determinados processos, levando a um possível produto de má qualidade e clientes insatisfeitos.

## 2.2. Descrição geral do software

Esse software tem o objetivo fornecer uma maneira de organizar, de padronizar e também auditar os processos de uma empresa automotiva.

Os colaboradores poderão acessar ao sistema a partir de qualquer dispositivo com acesso à internet, a qualquer hora do dia e de dentro da intranet da empresa.

Possibilita a retirada de métricas para análise do negócio em tempo real

Gerentes e RH terão acesso a um controle completo dos funcionários e suas áreas de domínio e aptidão.

Possibilita o acompanhamento em tempo real, os impeditivos e os riscos identificados pelos colaboradores e responsáveis, também provê uma interface.

Colaboradores terão acesso a diversas documentações relacionadas aos processos da empresa, e pessoas selecionadas irão realizar as alterações e controle de versão do documento.

## 3. Definição conceitual da solução

### 3.1. Requisitos Funcionais

#### Módulo de Cadastro de Incidentes e Problemas

- **Cadastro de Incidentes e Problema:** O sistema deve permitir que usuários autenticados cadastrem incidentes e ou problemas, informando a Não Conformidade.
- **Cadastro de Não Conformidade:** O sistema deve permitir que usuários autenticados cadastrem as não conformidades.
- **Consulta de Incidentes e Problema:** O sistema deve permitir que usuários autenticados possam consultar incidentes e ou problemas.
- **Consulta de Não Conformidade:** O sistema deve permitir que usuários autenticados possam consultar não conformidades.
- **Altera Situação de Incidentes e Problema:** O sistema deve permitir que usuários autenticados alterem o status do incidente ou problema.



### **Módulo de Controle de Processos Automotivos**

- **Cadastro de Atividades:** O sistema deve permitir que usuários autenticados cadastrem atividades diárias programadas para os setores operacionais.
- **Consulta de Atividades:** O sistema deve permitir que usuários autenticados possam consultar atividades diárias programadas para os setores operacionais.
- **Alteração de Atividades:** O sistema deve permitir que usuários autenticados possam alterar atividades diárias programadas para os setores operacionais.
- **Exclusão de Atividades:** O sistema deve permitir que usuários autenticados possam excluir atividades diárias programadas para os setores operacionais.
- **Cadastro de Atividades Realizadas:** O sistema deve permitir que usuários autenticados possam inserir uma data em que a atividade foi realizada.
- **Cadastro de Paradas:** O sistema deve permitir que usuários autenticados cadastrem paradas realizadas na produção.
- **Consulta de Paradas:** O sistema deve permitir que usuários autenticados possam consultar paradas realizadas na produção.
- **Alteração de Paradas:** O sistema deve permitir que usuários autenticados possam alterar paradas realizadas na produção.
- **Exclusão de Paradas:** O sistema deve permitir que usuários autenticados possam excluir paradas realizadas na produção.

### **Módulo de Divulgação e Transparência**

- **Consulta de Orçamento da Despesa:** O sistema deve permitir que usuários não autenticados consultem o orçamento da despesa.
- **Cadastro de Orçamento da Despesa:** O sistema deve permitir que usuários autenticados cadastrem despesas.
- **Consulta de Orçamento da Receita:** O sistema deve permitir que usuários não autenticados consultem o orçamento da receita.
- **Cadastro de Orçamento da Receita:** O sistema deve permitir que usuários autenticados cadastrem receitas.
- **Configuração Robô de Compartilhamento:** O sistema deve permitir que usuários autenticados controlem o horário e a frequência com que é gerado um relatório, que

é disponibilizado para o público automaticamente, no site da empresa para usuários não autenticados e em outras redes sociais.

### **Módulo de Inteligência do Negócio**

- **Consulta de dados:** O sistema deve permitir que ferramentas de BI se conectem e tenham informações atualizadas para o cálculo das métricas do negócio.
- **Cadastro de dados:** O sistema deve coletar informações em tempo real para o cálculo das métricas do negócio.

### **Módulo de Compliance**

- **Monitoramento de atividades:** O sistema deve permitir que usuários autenticados monitorem as atividades da empresa e validem se estão adequadas aos processos de compliance.
- **Cadastro de Contratos:** O sistema deve permitir que usuários autenticados cadastrem contratos de serviços e materiais.
- **Consulta de Contratos:** O sistema deve permitir que usuários autenticados possam consultar contratos de serviços e materiais.
- **Alteração de Contratos:** O sistema deve permitir que usuários autenticados possam alterar contratos de serviços e materiais.
- **Exclusão de Contratos:** O sistema deve permitir que usuários autenticados possam excluir contratos de serviços e materiais.
- **Cadastro de Processos de Auditoria:** O sistema deve permitir que usuários autenticados cadastrem processos de auditoria.
- **Consulta de Processos de Auditoria:** O sistema deve permitir que usuários autenticados possam consultar processos de auditoria.
- **Alteração de Processos de Auditoria:** O sistema deve permitir que usuários autenticados possam alterar processos de auditoria.
- **Exclusão de Processos de Auditoria:** O sistema deve permitir que usuários autenticados possam excluir processos de auditoria.
- **Cadastro de Processos de Auditoria Realizados:** O sistema deve permitir que usuários autenticados possam inserir uma data em que a auditoria foi realizada.

### 3.2 Requisitos Não-Funcionais

- Usabilidade - O sistema deve ser de fácil utilização.

<b>Estímulo</b>	Usuário precisando cadastrar um incidente.
<b>Fonte do Estímulo</b>	Usuário acessando o módulo de Cadastro de Incidentes e Problemas, cadastrando um problema com uma peça.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo Cadastro de Incidentes e Problemas.
<b>Resposta</b>	A interface apresenta simplicidade e objetividade.
<b>Medida da resposta</b>	O usuário consegue realizar o cadastro do problema.

- Desempenho - O sistema deve possuir bom desempenho.

<b>Estímulo</b>	Usuário gera um relatório no Power BI
<b>Fonte do Estímulo</b>	Usuário acessando o módulo de Inteligência do Negócio, gerando relatório empresarial.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo de Inteligência do Negócio.
<b>Resposta</b>	O sistema dispõe de um banco não relacional para otimizar o tempo de resposta.
<b>Medida da resposta</b>	O usuário gera o relatório em menos de 2 minutos.

- Manutenibilidade - O sistema deve ser de fácil manutenção.

<b>Estímulo</b>	O usuário encontra uma falha no sistema.
<b>Fonte do Estímulo</b>	O usuário identifica uma possível falha em algum dos módulos.
<b>Ambiente</b>	Funcionamento parcialmente, carga normal.
<b>Artefato</b>	Todos os módulos.
<b>Resposta</b>	As modificações no código são realizadas isoladamente no módulo problemático.
<b>Medida da resposta</b>	Apenas a correção de um modulo é enviada a produção, sem a necessidade de interromper o funcionamento dos demais módulos

- Testabilidade - O sistema deve ser passível de ser testado em todas as funcionalidades.

<b>Estímulo</b>	Desenvolvedor testando o sistema.
<b>Fonte do Estímulo</b>	Desenvolvedor utiliza um framework de testes para estressar o sistema.
<b>Ambiente</b>	Funcionamento, carga alta.
<b>Artefato</b>	Todos os módulos, ambiente de desenvolvimento.
<b>Resposta</b>	O sistema realiza todo o teste sem apresentar falhas.
<b>Medida da resposta</b>	É disponibilizada a release testada.

- Acessibilidade - O sistema deve suportar ambientes web e móveis.

<b>Estímulo</b>	Usuário avaliando a qualidade dos acabamentos utilizando um PDA.
<b>Fonte do Estímulo</b>	Usuário acessando o sistema utilizando um dispositivo móvel.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo de Controle de Processos Automotivos.
<b>.Resposta</b>	O sistema se adequa ao tamanho da tela do usuário.
<b>Medida da resposta</b>	O usuário tem uma experiência otimizada em dispositivos móveis.

- Disponibilidade - O sistema deve ser confiável e robusto, se recuperando no caso da ocorrência de erro.

<b>Estímulo</b>	Um dos nós do cluster apresenta falha.
<b>Fonte do Estímulo</b>	O servidor apresenta falha generalizada.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Todos os módulos.
<b>Resposta</b>	O serviço de kubernetes do Azure realiza a orquestração dos containers.
<b>Medida da resposta</b>	O sistema continua funcionando.

- Interoperabilidade - O sistema deve se comunicar com sistemas externos via APIs de integração.

<b>Estímulo</b>	O sistema recebe a informação de um lote defeituoso.
-----------------	--

<b>Fonte do Estímulo</b>	O usuário cadastra um problema com um lote de peças produzidas.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo de Controle de Processos Automotivos.
<b>Resposta</b>	O sistema realiza um cadastro no banco relacional.
<b>Medida da resposta</b>	Os relatórios são atualizados em tempo real.

- Segurança - O sistema deve apresentar segurança adequada à atividade, que é de alto risco.

<b>Estímulo</b>	O usuário tenta acessar o sistema.
<b>Fonte do Estímulo</b>	O usuário insere a credencial errada no sistema.
<b>Ambiente</b>	Funcionamento, carga normal.
<b>Artefato</b>	Módulo Cadastro de Incidentes e Problemas, módulo de Controle de Processos Automotivos.
<b>Resposta</b>	O sistema mostra uma página de senha incorreta.
<b>Medida da resposta</b>	Apenas usuários que possuem credenciais cadastradas utilizam o sistema.

- Disponibilidade - O sistema deve estar disponível 24 horas por dia, nos sete dias da semana.

<b>Estímulo</b>	O sistema apresenta instabilidade no tempo de resposta.
-----------------	---

<b>Fonte do Estímulo</b>	Muitos acessos simultâneos.
<b>Ambiente</b>	Funcionamento, carga alta.
<b>Artefato</b>	Todos os módulos.
<b>Resposta</b>	É adicionado mais um nó Cloud no cluster híbrido.
<b>Medida da resposta</b>	O sistema tem seu tempo de resposta normalizado.

### 3.3. Restrições Arquiteturais

- Apresentar características de aplicações distribuídas, tais como abertura, portabilidade e uso extensivo de recursos de rede.
- Ser hospedado parte em nuvem e parte no Data Center da empresa.
- Ser modular e implantável por módulos, de acordo com a prioridade e necessidade da empresa.
- Utilizar arquitetura baseada em serviços.

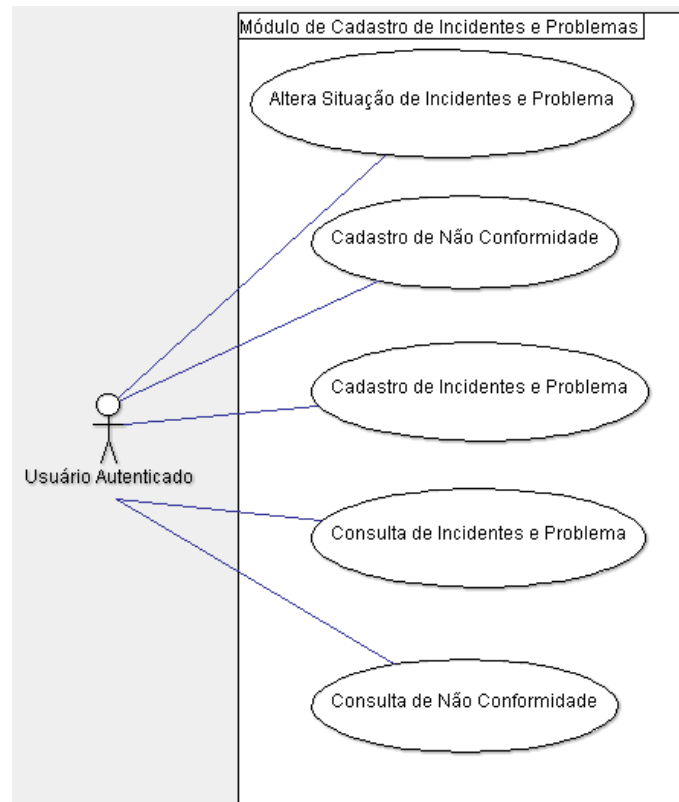
### 3.4. Mecanismos Arquiteturais

<b>Mecanismo de análise</b>	<b>Mecanismo de design</b>	<b>Mecanismo de implementação</b>
Comunicação entre processos	Contêiner Web e Aplicação	Docker
Front-end	Interface de comunicação com o usuário do sistema	ASP.NET Core MVC, JavaScript e Bootstrapper
Back-end	Regras de negócio da aplicação	ASP.NET Core
Integrações com outros módulos e sistemas	Interfaces utilizando JSON	Rest
Persistência	Banco de dados relacional	Microsoft SQL Server
Log	Framework de Log	log4Net
Autenticação e autorização	Verificação das credenciais para execução de ações	JWT - JSON Web Token
Exposição de API's	Exposição de Restful API's	Azure WebSites
CI/CD	Ferramenta para pipeline de integração e entrega contínua	Jenkins
Build	Geração de artefatos para publicação nos servidores de aplicação	MSBuild e Nuget
Teste	Teste automatizado	MSTest

Deploy	Deploy de artefatos para os servidores de aplicação	Jenkins
Versionamento	Controle de código-fonte	Git e Github

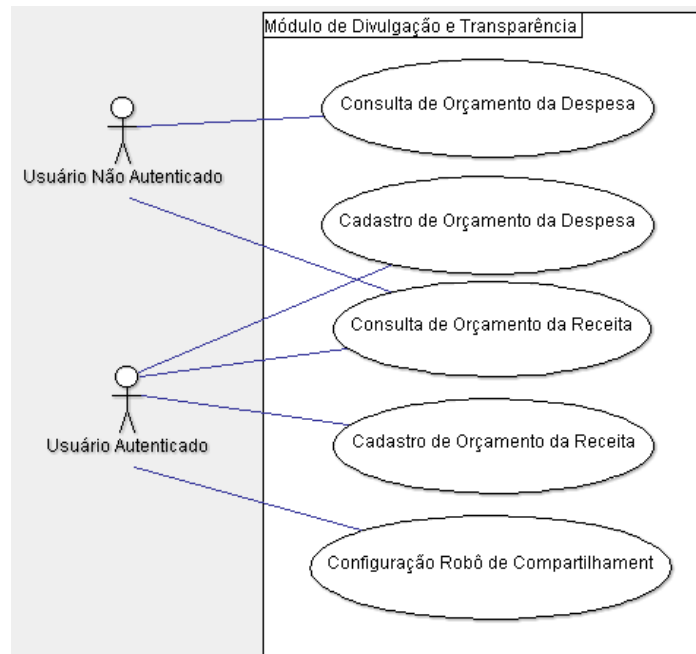
## 4. Modelagem e projeto arquitetural

### 4.1. Modelo de componentes

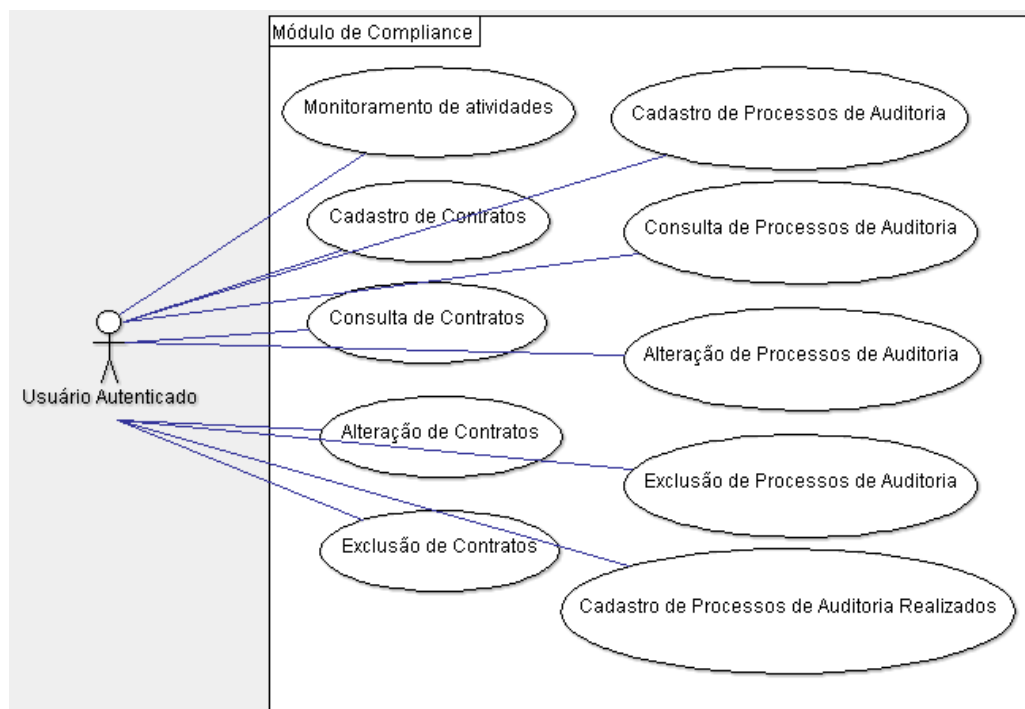


**Figura 1 – Diagrama de Casos de Uso – Módulo de Cadastro de Incidentes e Problemas**

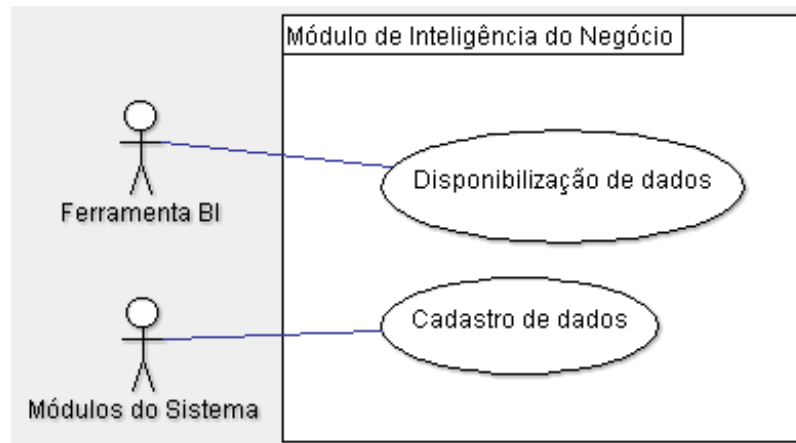




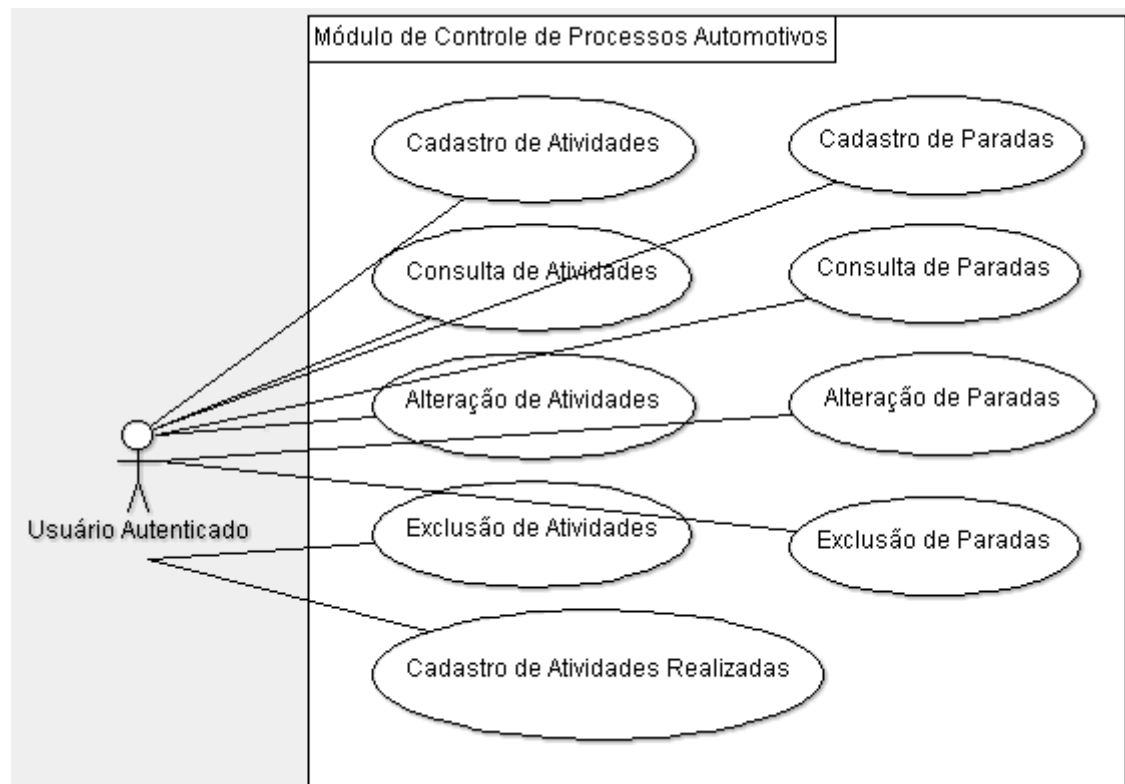
**Figura 2 – Diagrama de Casos de Uso – Módulo de Divulgação e Transparência**



**Figura 3 – Diagrama de Casos de Uso – Módulo de Compliance**



**Figura 4 – Diagrama de Casos de Uso – Módulo de Inteligência do Negócio**



**Figura 5 – Diagrama de Casos de Uso – Módulo de Controle de Processos Automotivos**

## 4.2. Modelo de componentes

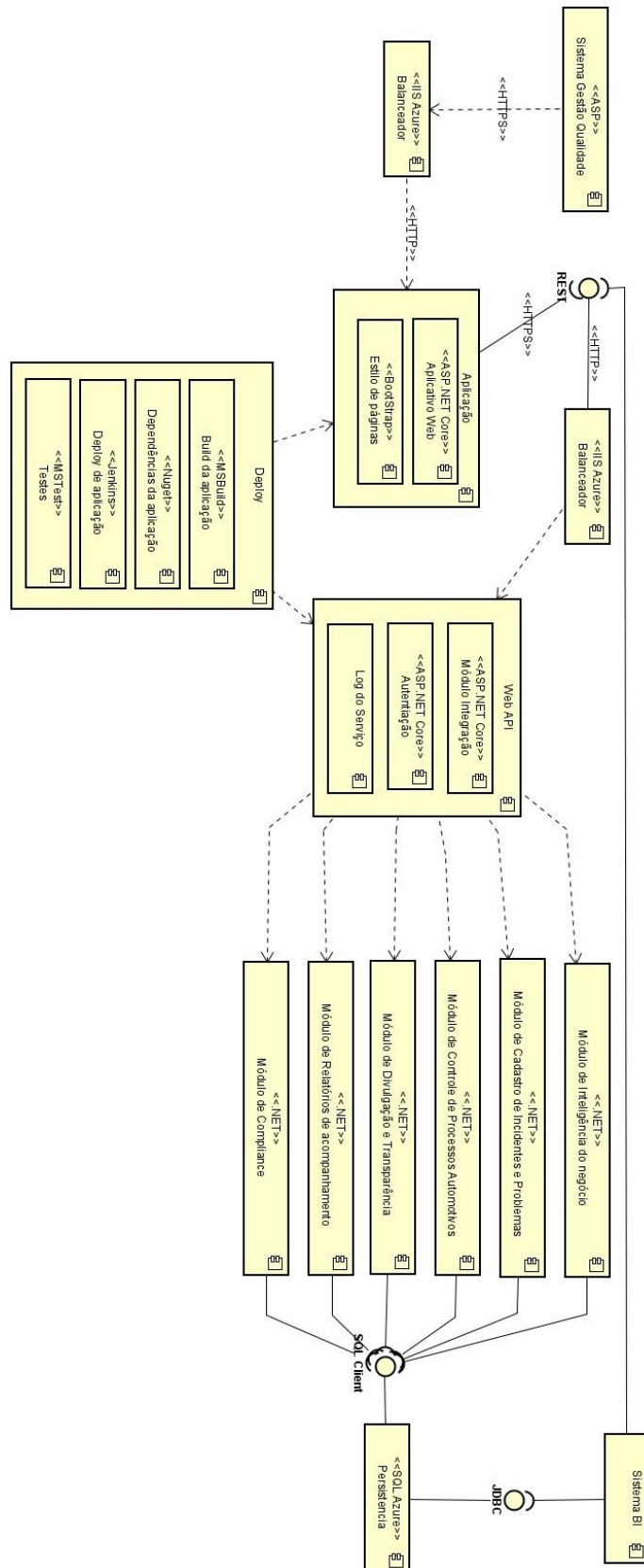
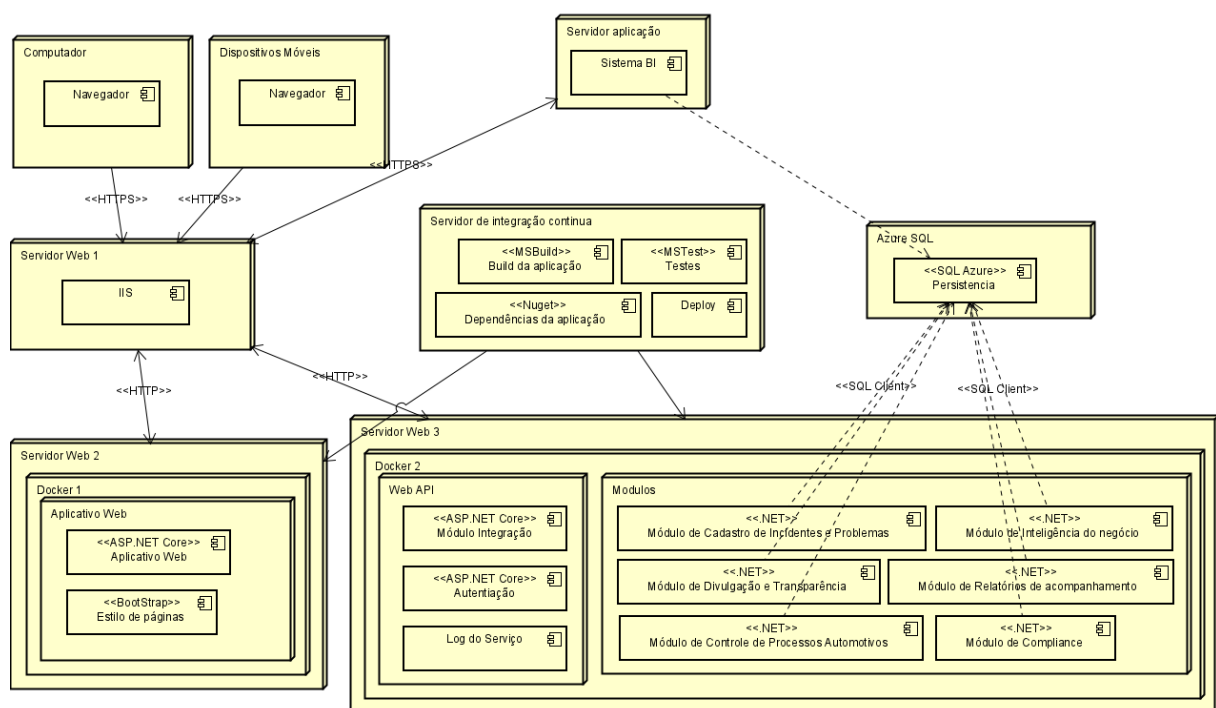


Figura 6 – Diagrama de componentes

Componente	Descrição
<b>Sistema Gestão Qualidade</b>	Representa o site em qualquer dispositivo móvel ou computador
<b>Balanceador</b>	Recurso utilizado para aumentar a disponibilidade dos componentes acessados
<b>Aplicação</b>	Componentes utilizados para gerar e disponibiliza os conteúdos a serem acessados nos navegadores.
<b>Deploy</b>	Um conjunto de componentes utilizados para testar, depurar, controlar a versão, e realiza releases.
<b>Web API</b>	Componentes utilizados para expor as operações que serão invocadas por outros módulos do sistema.
<b>Sistema BI</b>	Representa qualquer sistema externo de BI
<b>Persistência</b>	Componente utilizado para persistir qualquer informação utilizada pelo sistema.

### 4.3. Modelo de implantação



**Figura 7 – Modelo de Implantação**

Acima é mostrada a relação dos componentes com os seus respectivos servidores, e como o modelo de implantação deve ser implantado. No modelo acima não está explícito a arquitetura de cluster para os servidores Web e nem SQL, porém nessa arquitetura devemos considerar um sistema de cluster híbrido, e a utilização de kubernetes para a alocação de mais recursos conforme a demanda ou falha dos nós do sistema.

#### 4.4. Modelo de dados

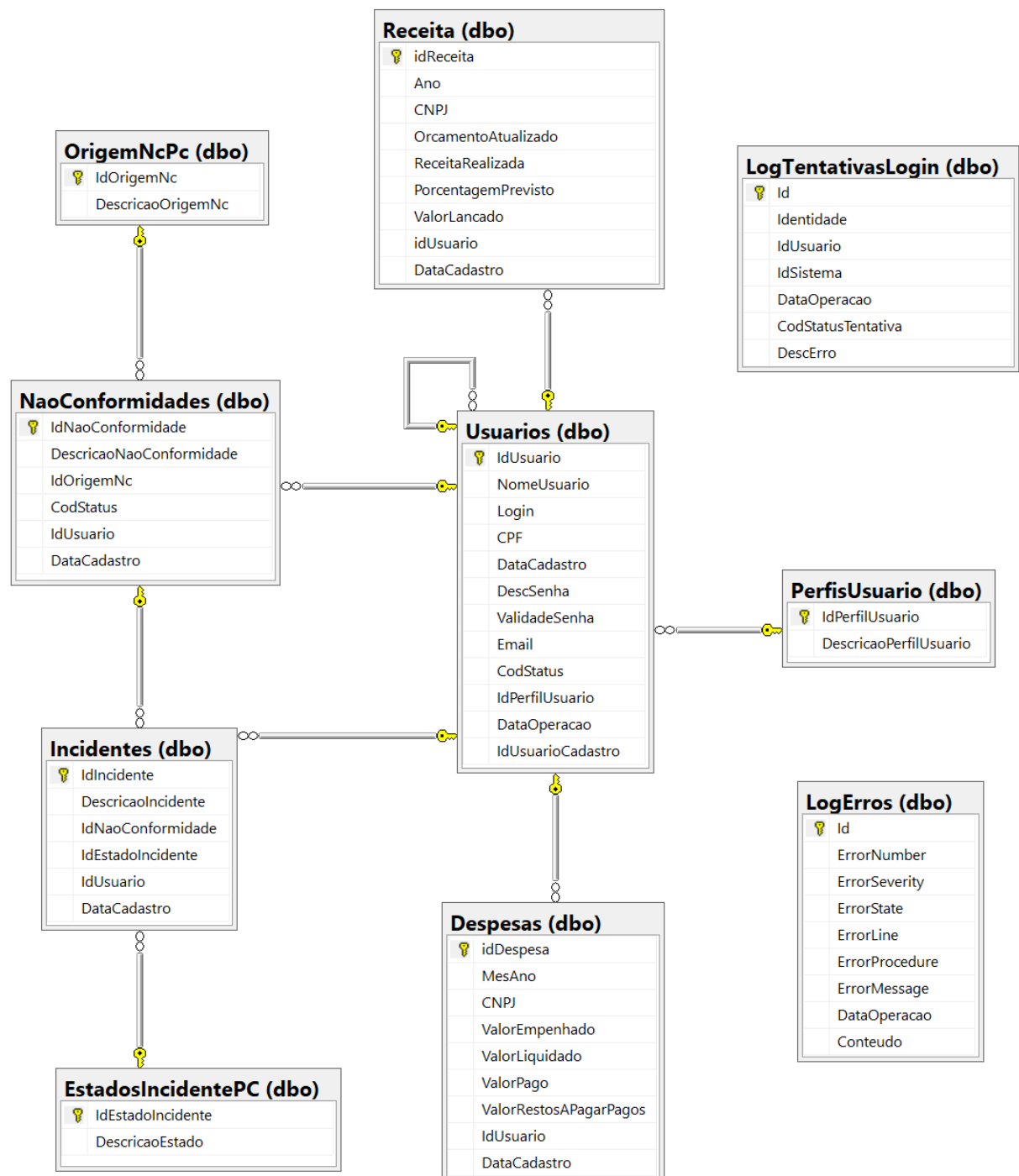


Figura 8 – Modelo Relacional

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

A prova de conceito desse projeto visa atender as necessidades dos casos de uso do módulo de Cadastro de Incidentes e Problemas e Módulo de Divulgação e Transparência. O objetivo desse protótipo é atender os requisitos funcionais e não funcionais especificados nos respectivos módulos.

### 5.1. Implementação e Implantação

As tecnologias usadas na POC são as descritas no item de mecanismos arquitetuais.

A telas da aplicação podem ser vistas nas evidências de cenário.

A url do vídeo de apresentação da POC bem como o link do código fonte da mesma pode ser encontrado no apêndice deste trabalho

Nessa POC, pretende-se validar os seguintes requisitos não funcionais:

- **Segurança**

Esse requisito não funcional foi escolhido devido à criticidade e a preocupação em manter os dados sensíveis seguros.

Os critérios de aceite são:

- Não permitir que usuários possam acessar páginas privadas sem estar autenticados no sistema.
- Ao identificar que um acesso a área segura está sendo feito sem autenticação, sistema deverá redirecionar para tela de autenticação.
- O sistema deverá permitir que usuários naveguem em telas públicas sem estar autenticado.

- **Usabilidade**

Esse requisito não funcional foi escolhido pois a facilidade de navegação e utilização. É importantíssimo que usuários possam acessar funcionalidades de maneira simples e objetiva, auxiliando no processo de tomada de decisões.

Os critérios de aceite são:

- A tela do sistema deve apresentar facilidade de navegação.
- O usuário deve ser capaz de cadastrar um incidente em no máximo cinco minutos.
- O acesso às funcionalidades deve apresentar objetividade e não ser confuso.

- **Acessibilidade**

Esse requisito não funcional foi escolhido devido a necessidade de a plataforma funcionar em ambientes responsivos, como celulares e tablets.

Os critérios de aceite são:

- Os componentes das interfaces devem se adaptar de forma que nenhuma funcionalidade seja perdida.
- A navegação deve continuar sendo simples e direta.
- A identidade visual da aplicação deve ser a mesma (fontes e cores).

- **Interoperabilidade**

Esse requisito não funcional foi escolhido pois a comunicação com sistemas de terceiros é uma das peças-chave do sistema, para que o sistema seja modular e de fácil desenvolvimento

Os critérios de aceite são:

- O sistema deve conseguir se comunicar com tecnologias heterogêneas dos sistemas terceiros.
- O sistema deve prover uma interface de comunicação nos padrões atuais de tecnologia.

Para a realização da prova de conceito desse projeto, vários casos de uso foram implementados visando fornecer insumos para validação dos requisitos não funcionais priorizados.

São eles:

<b>Módulo</b>	<b>Caso de Uso</b>	<b>Requisito não funcional</b>
	Efetuar Login	Segurança
Incidentes	Cadastrar Incidentes	Segurança, Usabilidade e Acessibilidade
	Consultar Incidentes	Segurança, Usabilidade e Acessibilidade
	Cadastrar Não Conformidades	Segurança, Usabilidade e Acessibilidade
	Consultar Não Conformidades	Segurança, Usabilidade e Acessibilidade
Divulgação e Transp.	Cadastrar Receita	Segurança, Usabilidade e Acessibilidade
	Consultar Receita	Segurança, Usabilidade e Acessibilidade
	Cadastrar Despesa	Segurança, Usabilidade e Acessibilidade
	Consultar Despesa	Segurança, Usabilidade e Acessibilidade
Integração (API)	Cadastrar Incidentes	Interoperabilidade
	Consultar Incidentes	Interoperabilidade
	Cadastrar Não Conformidades	Interoperabilidade
	Consultar Não Conformidades	Interoperabilidade
	Cadastrar Receita	Interoperabilidade
	Consultar Receita	Interoperabilidade
	Cadastrar Despesa	Interoperabilidade
	Consultar Despesa	Interoperabilidade
	Cadastrar Usuários	Interoperabilidade
	Consultar Usuários	Interoperabilidade

## 5.2 Código

A Arquitetura da POC da solução proposta foi construída baseada em DDD (Domain Driven Design) e obedecendo os princípios do SOLID, que é um acrônimo criado por Michael Feathers, que representa os 5 princípios da programação orientada a objetos:

S — Single Responsibility Principle (Princípio da responsabilidade única)

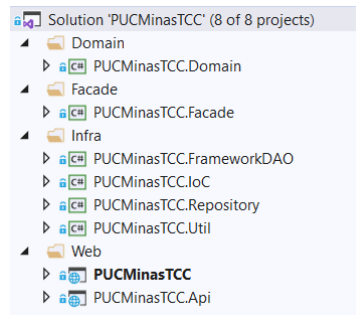
O — Open-Closed Principle (Princípio Aberto-Fechado)

L — Liskov Substitution Principle (Princípio da substituição de Liskov)

I — Interface Segregation Principle (Princípio da Segregação da Interface)

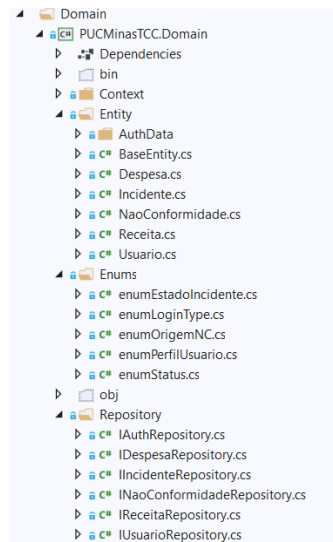
D — Dependency Inversion Principle (Princípio da inversão da dependência)





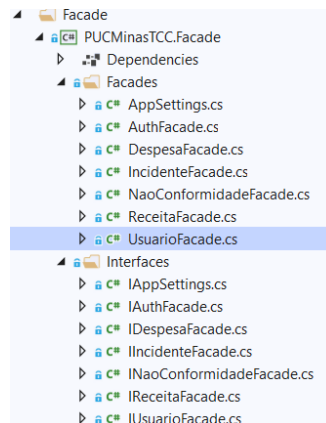
**Figura 9 – Estrutura Código**

No Domain (Domínio) temos as entidades, enumeradores, e interfaces de repositório utilizados em todo o sistema, o domínio é o projeto que pode ser referenciado por todas as camadas do projeto:



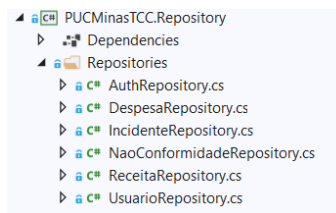
**Figura 10 – Estrutura Código**

Em Facade (Fachadas), temos a centralização das regras de negócio e sua intermediação com nosso repositório, sendo assim a Facade pode ser aproveitada pela aplicação Web e API, para validar as regras de negócio e obter acesso ao repositório.



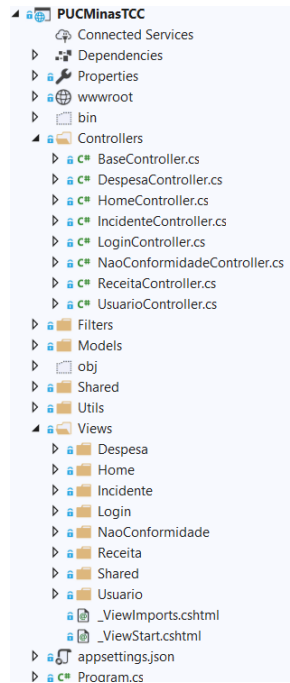
**Figura 11 – Estrutura Código**

Em Repository (Repositório), é a camada onde temos o acesso a nossa base de dados.



**Figura 12 – Estrutura Código**

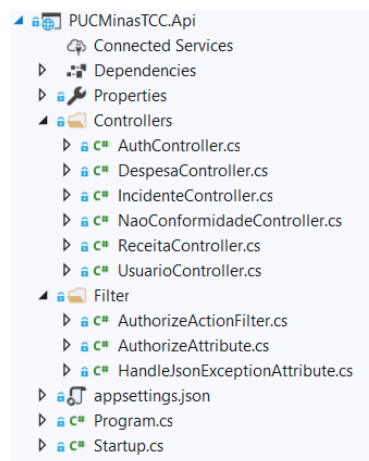
Projeto Web foi criado utilizando a tecnologia .net Core 3.0 adotando o padrão MVC (Model-View-Controller), utilizamos a facade para validação das regras de negócio e fazer a intermediação com o repository. Utilizamos a geração de um token (JWT) no login para que o usuário possa ter acesso as páginas privadas, para as páginas públicas não é necessário o token.



**Figura 13 – Estrutura Código**

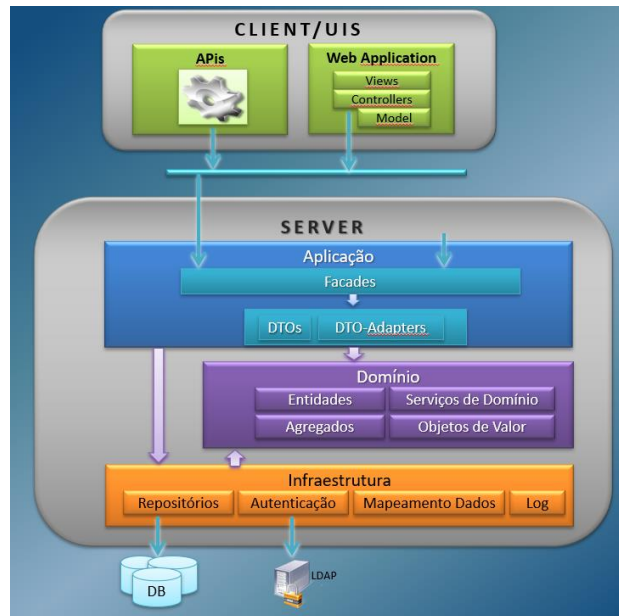
A API foi desenvolvida utilizando a tecnologia .net Core 3.0, também utiliza a facade, para validação das regras de negócio e intermediação ao repository.

Foi implementado filtros de autorização, para que na chamada efetuada pelo cliente, verificar se o mesmo necessita de autorização (JWT token) ou se o mesmo realizou a chamada para uma API pública.



**Figura 14 – Estrutura Código**

Para uma visão mais simples, o desenho a seguir representa a arquitetura da POC para a solução proposta:



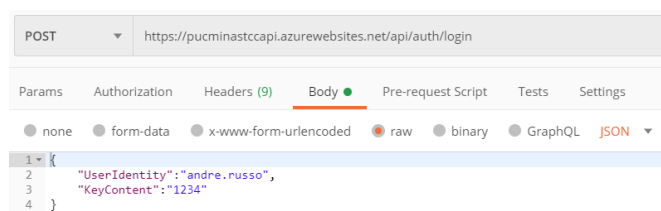
**Figura 15 – Estrutura Código**

### 5.3. Interfaces/ APIs

- **Login**

A API de Login é a principal do projeto, é com ela que obtemos o JWT token, para ter acesso as APIs privadas, ao obter sucesso na autenticação o “token” de ser informado no header de todas as APIs privadas

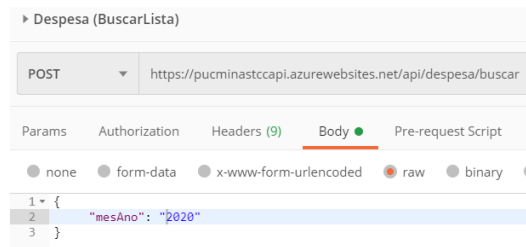
Requisição (POST) Login



**Figura 16 – Interface/APIs - Login**



### Requisição (POST) Busca por filtro



**Figura 20 – Interface/APIs - Despesa**

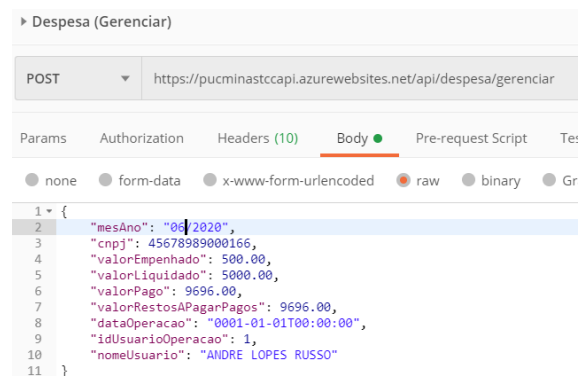
### Resposta (POST) Busca por filtro



**Figura 21 – Interface/APIs - Despesa**

Para Gerenciar (INSERT, UPDATE) a despesa é necessário informar o token, que é obtido no login, no header da aquisição.

### Requisição (POST) Gerenciar



**Figura 22 – Interface/APIs - Despesa**

Params	Authorization	Headers (10)	Body	Pre-request Script	Tests	Settings
Headers						
KEY						
VALUE						
token						

**Figura 23 – Interface/APIs - Despesa**

Resposta (POST) Gerenciar

Status: 200 OK Time: 122ms Size: 126 B

**Figura 24 – Interface/APIs - Despesa**

- Receita**

A API de Receita, é utilizada para ter acesso as receitas do sistema, podemos ter busca (por id ou filtro), e gerenciar (INSERT, UPDATE).

Nas buscas (por id ou filtro), por serem APIs públicas, não necessitamos de token.

Requisição (GET) Busca por Id

Receita (Get por id)	
GET	https://pucminastccapi.azurewebsites.net/api/receita?id=1

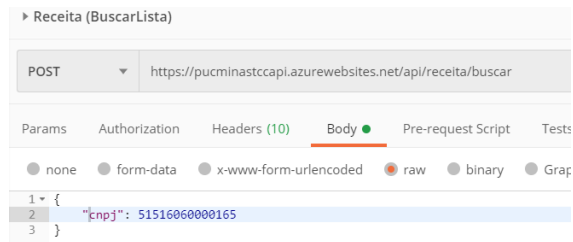
**Figura 25 – Interface/APIs - Receita**

Resposta (GET) Busca por Id

Pretty	Raw	Preview	Visualize	JSON	
<pre> 1  { 2    "idReceita": 1, 3    "ano": "2020", 4    "cnpj": 51516060000165, 5    "orcamentoAtualizado": 600.50, 6    "receitaRealizada": 300.00, 7    "porcentagemPrevisto": 50.65, 8    "valorLancado": 60.50, 9    "dataOperacao": "0001-01-01T00:00:00", 10   "idUsuarioOperacao": 0, 11   "nomeUsuario": "VINICIUS LOPES RUSSO" 12 }</pre>					

**Figura 26 – Interface/APIs - Receita**

## Requisição (POST) Busca por Filtro



**Figura 27 – Interface/APIs - Receita**

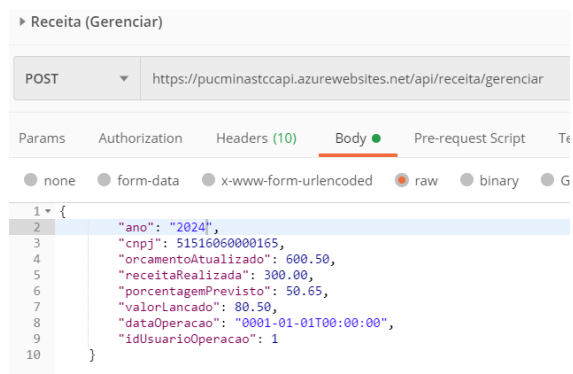
## Resposta (POST) Busca por Filtro



**Figura 28 – Interface/APIs - Receita**

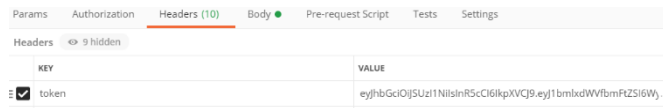
Para Gerenciar (INSERT, UPDATE) a receita é necessário informar o token, que é obtido no login, no header da aquisição

## Requisição (POST) Gerenciar



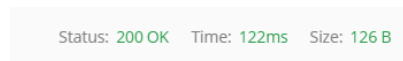
**Figura 29 – Interface/APIs - Receita**





**Figura 30 – Interface/APIs - Receita**

Resposta(POST) Gerenciar



**Figura 31 – Interface/APIs - Receita**

- **Não Conformidade**

A API de Não Conformidade, nos permite buscas (por id ou filtro) e gerenciar (INSERT, UPDATE) nas não conformidades utilizadas pelo sistema.

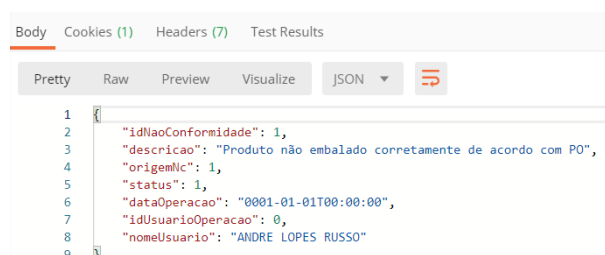
São APIs privadas que necessitam do token obtido no login, no header da requisição.

Requisição (GET) Busca por Id



**Figura 32 – Interface/APIs – Não Conformidade**

Resposta (GET) Busca por Id



**Figura 33 – Interface/APIs – Não Conformidade**

Requisição (POST) Busca por Filtro

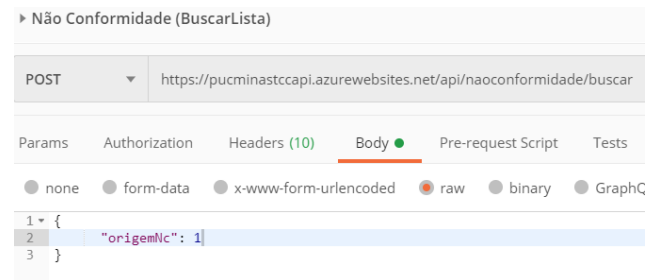


Figura 34 – Interface/APIs – Não Conformidade

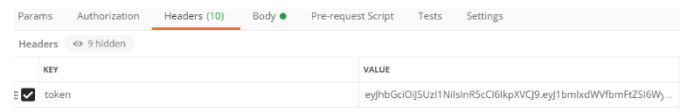


Figura 35 – Interface/APIs – Não Conformidade

Resposta (POST) Busca por Filtro

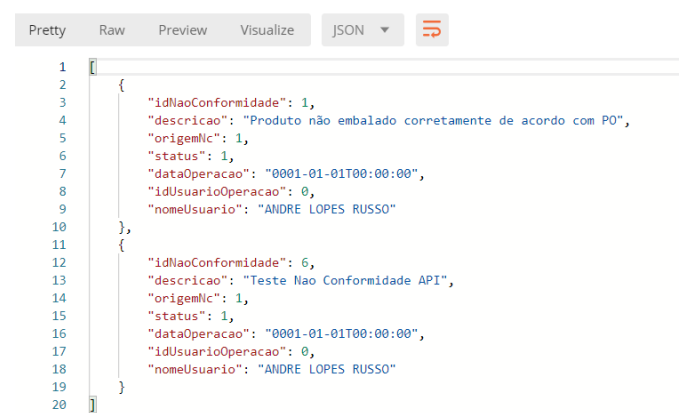
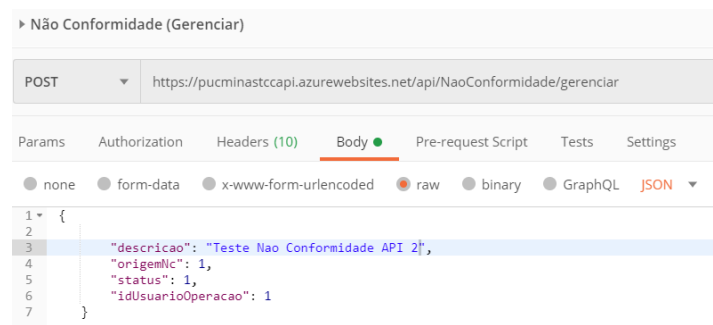


Figura 36 – Interface/APIs – Não Conformidade

Requisição (POST) Gerenciar





Requisição (POST) Busca por Filtro

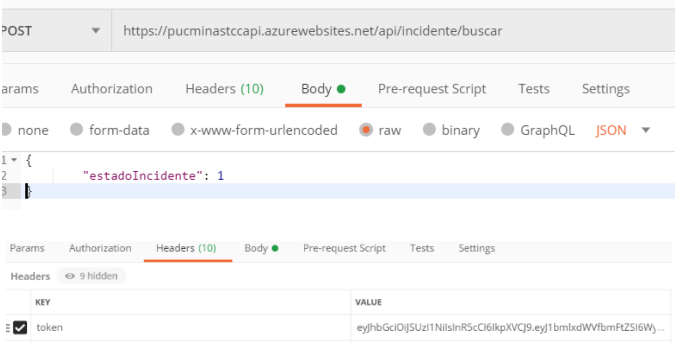


Figura 41 – Interface/APIs – Incidentes

Resposta (POST) Busca por Filtro

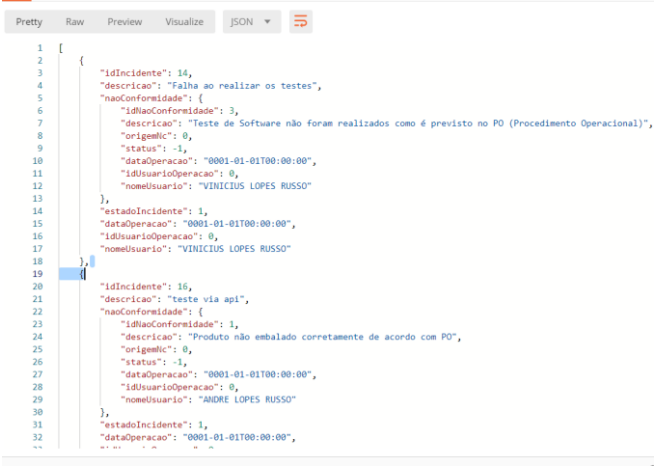
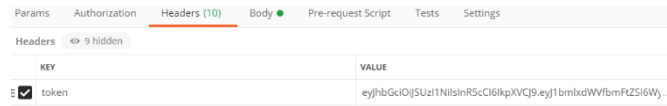


Figura 42 – Interface/APIs – Incidentes

Requisição (POST) Gerenciar



Figura 43 – Interface/APIs – Incidentes



**Figura 44 – Interface/APIs – Incidentes**

Resposta (POST) Gerenciar

Status: 200 OK Time: 122ms Size: 126 B

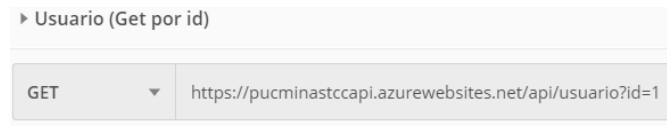
**Figura 45 – Interface/APIs – Incidentes**

- Usuários**

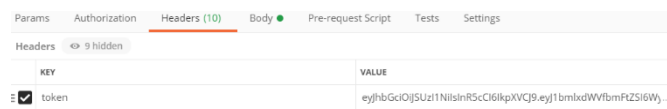
A API de Usuários, nos permite buscas (por id ou filtro) e gerenciar (INSERT, UPDATE) nos usuários do sistema.

São APIs privadas que necessitam do token obtido no login, no header da requisição.

Requisição (GET) Busca por Id

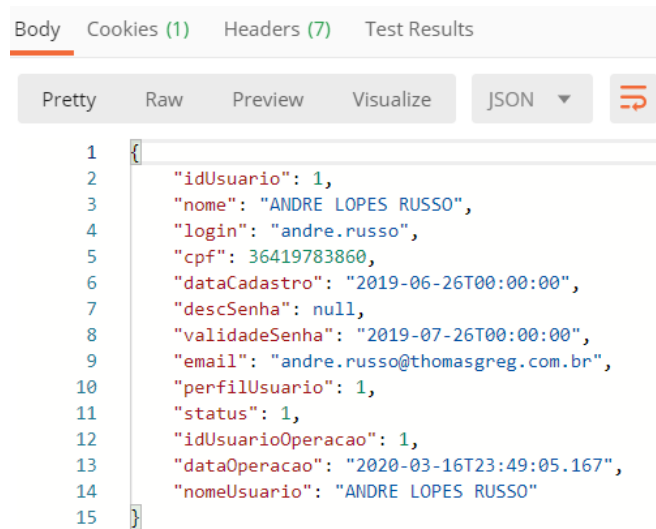


**Figura 46 – Interface/APIs – Usuário**



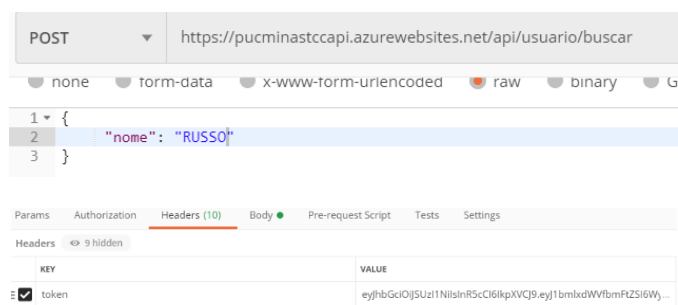
**Figura 47 – Interface/APIs – Usuário**

Resposta (GET) Busca por Id



**Figura 48 – Interface/APIs – Usuário**

### Requisição (POST) Busca por Filtro



**Figura 49 – Interface/APIs – Usuário**

### Resposta (POST) Busca por Filtro



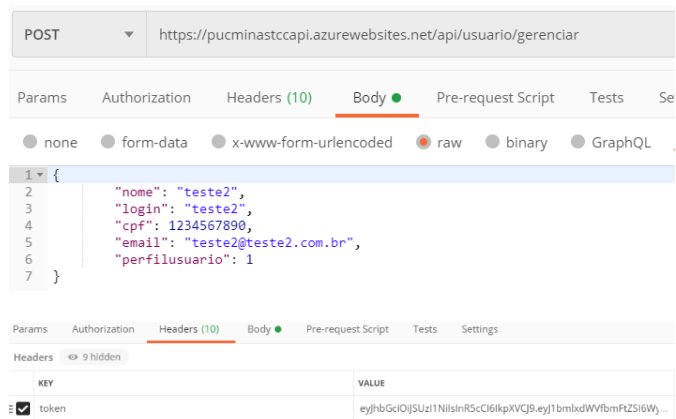
```

1  {
2    {
3      "idUsuario": 1,
4      "nome": "ANDRE LOPES RUSSO",
5      "login": "andre.russo",
6      "cpf": "36419783860",
7      "dataCadastro": "2019-06-26T00:00:00",
8      "descSenha": null,
9      "validadeSenha": "2019-07-26T00:00:00",
10     "email": "andre.russo@thomasmgreg.com.br",
11     "perfilUsuario": 1,
12     "status": 1,
13     "idUsuarioOperacao": 1,
14     "dataOperacao": "2020-03-16T23:49:05.167",
15     "nomeUsuario": "ANDRE LOPES RUSSO"
16   },
17   {
18     "idUsuario": 2,
19     "nome": "VINICIUS LOPES RUSSO",
20     "login": "vinicius.russo",
21     "cpf": "44235005820",
22     "dataCadastro": "2020-03-20T00:00:00",
23     "descSenha": null,
24     "validadeSenha": "2020-04-20T00:00:00",
25     "email": "vini@gmail.com",
26     "perfilUsuario": 1,
27     "status": 1,
28     "idUsuarioOperacao": 1,
29     "dataOperacao": "2020-03-20T00:04:45.02",
30     "nomeUsuario": "VINICIUS LOPES RUSSO"
31   }
32 }

```

**Figura 50 – Interface/APIs – Usuário**

### Requisição (POST) Gerenciar



POST <https://pucminastccapi.azurewebsites.net/api/usuario/gerenciar>

Params Authorization Headers (10) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

```

1 {
2   "nome": "teste2",
3   "login": "teste2",
4   "cpf": "1234567890",
5   "email": "teste2@teste2.com.br",
6   "perfilusuario": 1
7 }

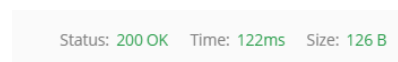
```

Params Authorization Headers (10) Body Pre-request Script Tests Settings

KEY	VALUE
token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmVxdWVibmFZS16Wj...

**Figura 51 – Interface/APIs – Usuário**

### Resposta (POST) Gerenciar



Status: 200 OK Time: 122ms Size: 126 B

**Figura 52 – Interface/APIs – Usuário**

## 6. Avaliação da Arquitetura

### 6.1. Análise das abordagens arquiteturais

A arquitetura proposta contempla uma série de componentes, todos modulares. Cada componente tem seu próprio conjunto de tecnologias e características de implantação. Apesar da utilização de vários componentes proprietários e de uma infraestrutura híbrida, toda a implementação foi feita de forma que seja o mais independente de plataforma possível. Foi ainda elaborada uma esteira DevOps para auxiliar no transporte de evoluções para os ambientes produtivos, com maior assertividade e garantia de qualidade através da execução de testes automatizados.

### 6.2. Cenários

**Cenário 1:** O sistema deve apresentar segurança adequada à atividade, que é de alto risco. Ao realizar o acesso a uma URL ou página, o sistema deve apresentar altos padrões de segurança necessário, garantindo que o usuário possa acessar as páginas privadas apenas autenticado no sistema. O sistema deve redirecionar o usuário para a tela de autenticação quando o mesmo tentar acessar uma página privada sem estar autenticado no sistema. O sistema deverá garantir que as páginas públicas possam ser acessadas sem estar autenticado. Garantindo assim a segurança e confidencialidade das informações estando em de acordo com um dos requisitos não funcionais.

**Cenário 2:** Ao navegar na tela, o sistema deve apresentar boa usabilidade. A navegação deve apresentar facilidade e o acesso às funcionalidades deve ser intuitivo e objetivo. O usuário deve conseguir cadastrar um incidente. Esta será a garantia de que o requisito não funcional de usabilidade foi satisfeito.

**Cenário 3:** Ao acessar a aplicação através de um dispositivo móvel ou desktop com resolução reduzida, a tela do usuário deverá se adaptar automaticamente, redimensionando seus componentes visuais de acordo com a resolução, porém sem perder funcionalidades ou complicar a navegação. Esta é a garantia de que o requisito não funcional de acessibilidade foi satisfeito.



**Cenário 4:** A aplicação deve prover recursos para que a comunicação entre sistemas terceiros caso queira utilizar um módulo de Incidentes e Problemas terceiro, o sistema terceiro deve-se comunicar com o sistema através API's REST, utilizando JSON. Caso não apresente o token de segurança nas aquisições, não será possível realizar as inclusões ou alterações de incidentes. Esta é a garantia de que o requisito não funcional de interoperabilidade foi satisfeito (além de estarem em conformidade com o requisito não funcional de segurança).

Na priorização foi utilizado o método de Árvore de Utilidade reduzida e com prioridades. Foi categorizado de acordo os atributos de qualidade a que estão relacionados e então classificados em função de sua importância e complexidade, considerando a percepção de negócio e arquitetura. As duas variáveis de priorização "Importância" e "Complexidade", apresentadas nas colunas IMP. e COM. respectivamente forma classificadas em alta (A), média (M) e baixa (B) de acordo com as características do requisito.

<b>Categoria</b>	<b>Atributo de qualidade</b>	<b>Cenário</b>	<b>IMP</b>	<b>COM</b>
Confidencialidade	Segurança	1. O sistema deve apresentar altos padrões de segurança.	A	A
Funcionalidade	Usabilidade	2. O sistema deve prover boa usabilidade.	A	M
	Acessibilidade	3. O sistema deve suportar ambientes web responsivos e ambientes móveis.	M	A
Compatibilidade	Interoperabilidade	4. O sistema deve se comunicar com vários sistemas, com tecnologias heterogêneas.	A	A

### 6.3. Avaliação

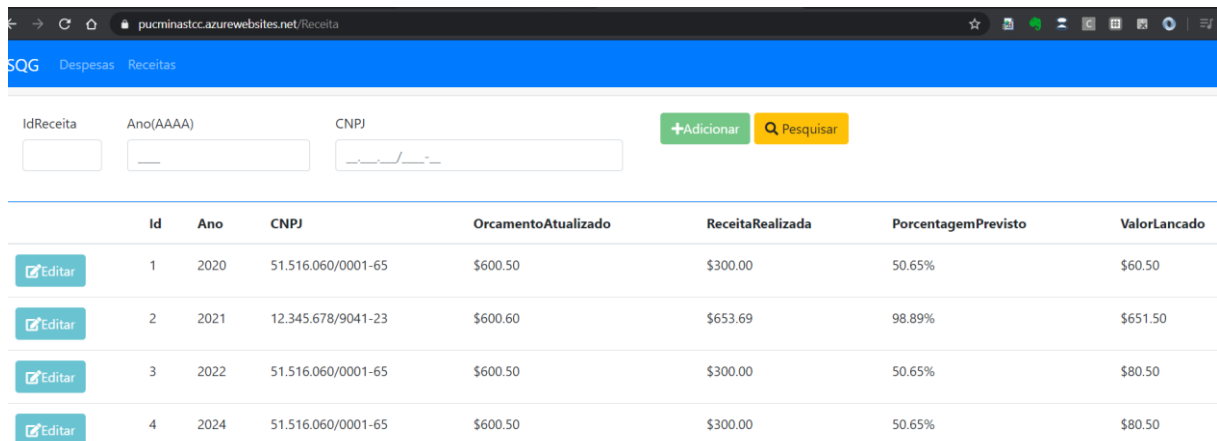
Processo de avaliação dos cenários identificados no item 6.2 são analisados. O objetivo é determinar os riscos, não riscos, pontos de sensibilidade e tradeoffs e as evidências mostrando o requisito de qualidade sendo atendido.





- Cenário 1:

Atributo de qualidade:	Segurança
Requisito de qualidade:	O sistema deve apresentar altos padrões de segurança
Preocupação:	
Impossibilitar o acesso a páginas privadas do sistema sem autenticação no sistema.	
Cenários(s):	
Cenário 1	
Ambiente:	
Sistema em operação normal	
Estímulo:	
Usuário tentando acessar uma página privada do sistema sem estar autenticado no sistema.	
Mecanismo:	
Criar um recurso de Filtro que possibilite o gerenciamento de todas as requisições HTTP do servidor, filtrando o endereço que está sendo acessado.	
Medida de Resposta:	
O usuário deve ser redirecionado para tela de autenticação. Considerações sobre	
Considerações sobre a arquitetura:	
Riscos	O gerenciamento de sessões e de autenticação apropriados são críticos para segurança web. Falhas nessa área frequentemente envolvem falha ao proteger credencias e sessões durante o ciclo de vida.
Pontos de Sensibilidade:	Servidor de aplicação operando em modo HTTPS
Tradeoff:	Não existe

- Evidências Cenário 1:

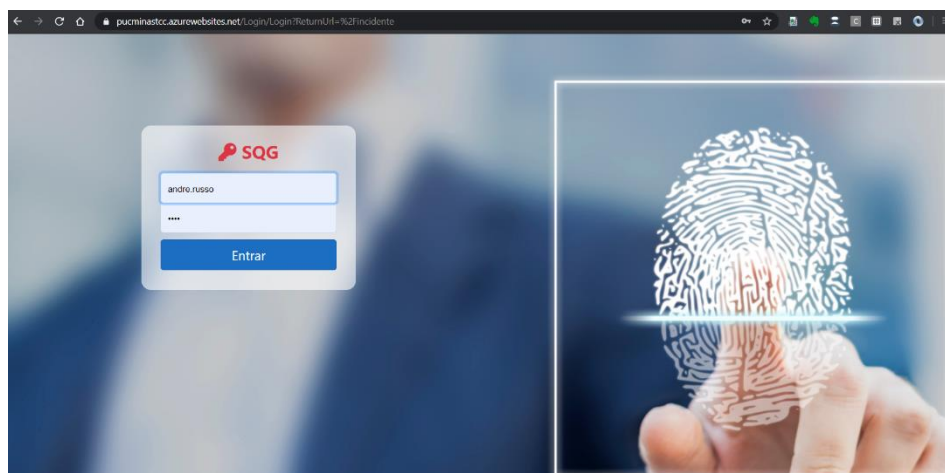
Usuário acessando página pública onde não é necessário autenticação, como o usuário está sem autenticação o menu somente disponibiliza as opções sem autenticação



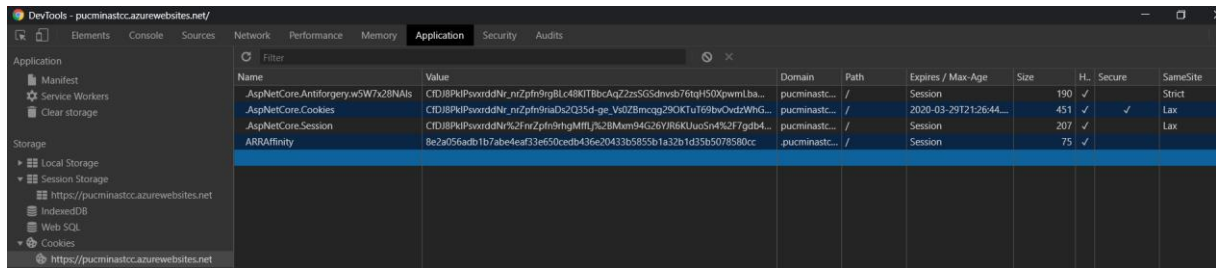
	Id	Ano	CNPJ	OrcamentoAtualizado	ReceitaRealizada	PorcentagemPrevisto	ValorLancado
	1	2020	51.516.060/0001-65	\$600.50	\$300.00	50.65%	\$60.50
	2	2021	12.345.678/9041-23	\$600.60	\$653.69	98.89%	\$651.50
	3	2022	51.516.060/0001-65	\$600.50	\$300.00	50.65%	\$80.50
	4	2024	51.516.060/0001-65	\$600.50	\$300.00	50.65%	\$80.50

**Figura 53 – Avaliação**

Caso o usuário tente o acesso via url para uma página privada, como exemplo: <https://pucminastcc.azurewebsites.net/incidente> , o sistema automaticamente direciona para a tela de login. Ao efetuar a autenticação o sistema gera um token (JWT), e gera uma sessão para que o usuário navegue entre as páginas privadas



**Figura 54 – Avaliação**








**Figura 55 – Avaliação**

- Cenário 2:

Atributo de qualidade:	Usabilidade
Requisito de qualidade:	Sistema deve apresentar desempenho satisfatório dentro dos limites aceitável.
Preocupação:	
Fornecer interfaces simples para agilizar a navegação e tornar a experiência do usuário bem rápida e objetiva.	
Cenários(s):	
Cenário 2	
Ambiente:	
Sistema em operação normal	
Estímulo:	
Usuário navegando no site e registrando um incidente, devendo apresentar objetividade e um registro rápido em no máximo 5 minutos.	
Mecanismo:	
Criação de telas simples e objetivas. Menus de navegação visíveis durante todo o tempo, possibilitando efetuar uma busca a qualquer momento. Carregamento de todo o conteúdo estático (HTML, CSS e Javascript) do site no momento do início da navegação, fazendo com que as requisições seguintes aos servidores de back-end tragam apenas dados em formato JSON, o que agiliza o carregamento das páginas	
Medida de Resposta:	
Caso o usuário não realize o registro do incidente em no máx 5 minutos, a sessão irá expirar e ele será redirecionado para a tela de login	
Considerações sobre a arquitetura:	



Riscos	Pode ocorrer algum pico de memória no servidor ou um número de usuários muito grande ocasionando sobrecarga no servidor de aplicação, tornando os processamentos para obtenção de dados mais lentos por um período curto, prejudicando a experiência do usuário.
Pontos de Sensibilidade:	Balanceamento de carga ativo.
Tradeoff:	Não existe

O usuário autenticado, acessa o modulo de incidentes, ao clicar em adicionar é direcionado para a tela de registro de incidente.

SQG Incidentes Não Conformidades Despesas Receitas Usuários					ANDRE LOPES RUSSO	Sair
IdIncidente	Descricao	Não Conformidade	Estado	+Adicionar	Q Pesquisar	
		Todos	Todos			
Código	Descrição	Não Conformidade	Estado	Usuário Cadastro		
 12	O Produto saiu da empresa fora do conforme	Produto não embalado corretamente de acordo com PO	🟡Em Atendimento	VINICIUS LOPES RUSSO		
 13	Malote enviado ao cliente apresentou não conformidade	Cartão entregue para o cliente com foto trocada	🟢Aguardando Aceite	VINICIUS LOPES RUSSO		
 14	Falha ao realizar os testes	Teste de Software não foram realizados como é previsto no PO (Procedimento Operacional)	🟢Aberto	VINICIUS LOPES RUSSO		
 15	Descumprimento do PO	Teste de Software não foram realizados como é previsto no PO (Procedimento Operacional)	🔴Fechado	VINICIUS LOPES RUSSO		
 16	teste via api	Produto não embalado corretamente de acordo com PO	🟢Aberto	ANDRE LOPES RUSSO		

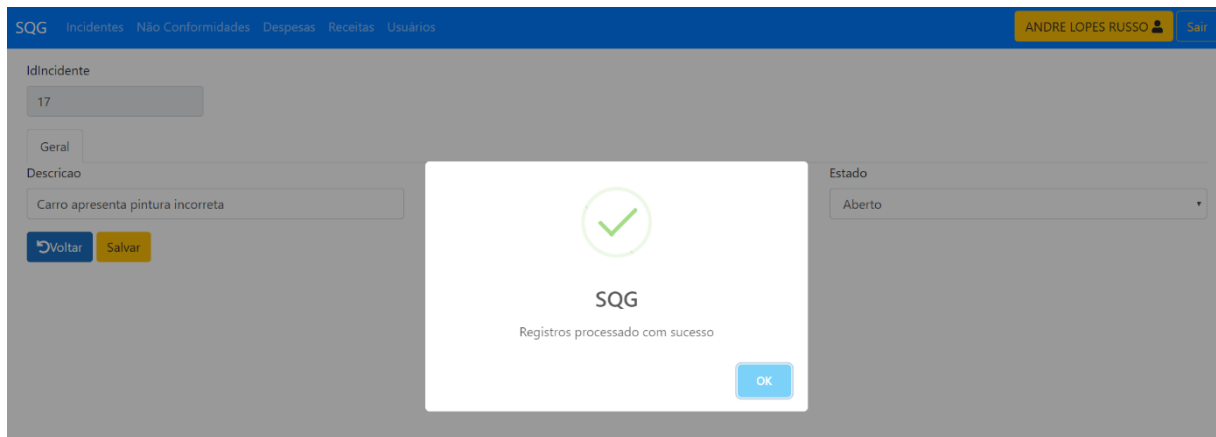
**Figura 56 – Avaliação**

Tela para registro do incidente.

SQG Incidentes Não Conformidades Despesas Receitas Usuários					ANDRE LOPES RUSSO	Sair
IdIncidente						
<input type="text"/>						
Geral						
Descricao		Não Conformidade		Estado		
<input type="text" value="Carro apresenta pintura incorreta"/>		<input type="text" value="Pintura do carro não está de acordo com padrão estabelecido"/>		<input type="text" value="Aberto"/>		
						

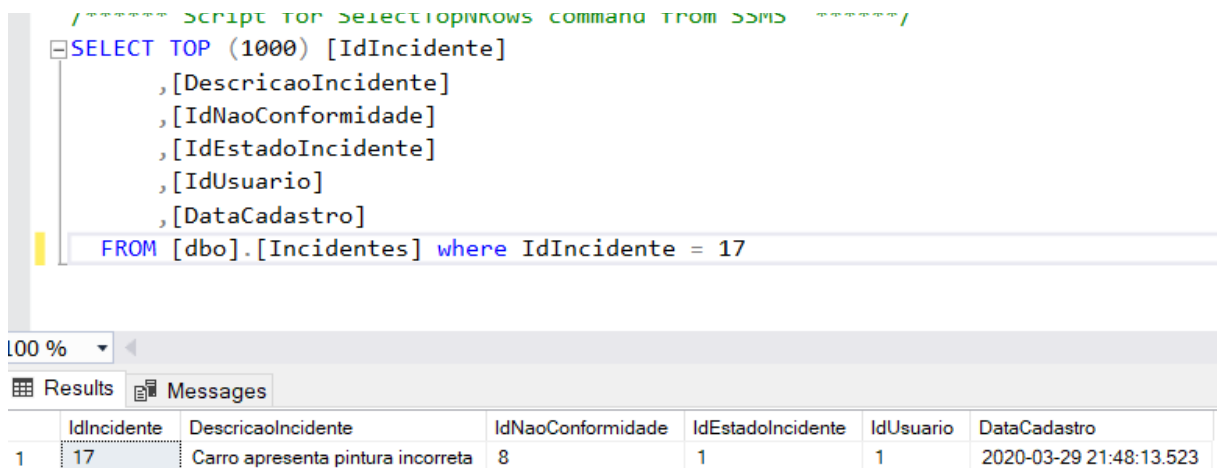
**Figura 57 – Avaliação**

Confirmação do registro do incidente, com o incidente preenchido



**Figura 58 – Avaliação**

Evidência do cadastro na base.



**Figura 59 – Avaliação**

- Cenário 3:

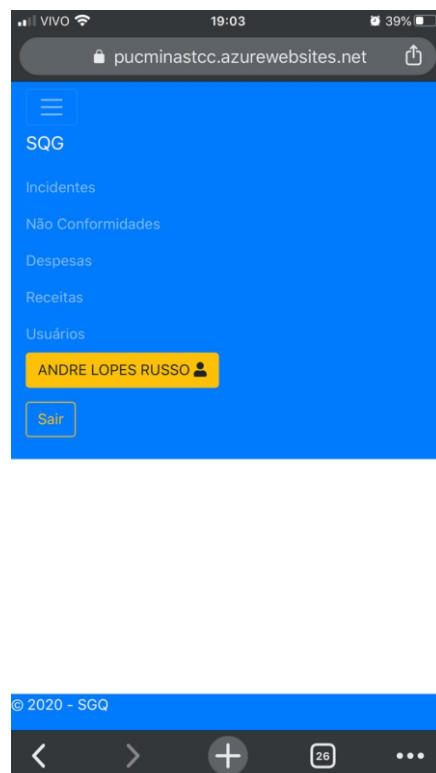
Atributo de qualidade:	Acessibilidade
Requisito de qualidade:	O sistema deve suportar ambiente web responsivos e ambientes móveis.
Preocupação:	
O sistema deve redimensionar seus objetos de acordo com o tamanho da tela.	
Cenários(s):	
Cenário 3	

Ambiente:	
Sistema em operação normal	
Estímulo:	
Usuário se autentica no sistema, e registra um incidente através de um ambiente móvel	
Mecanismo:	
Criação de telas utilizando mecanismos de design responsivos e ajustáveis, movimentando os componentes para que caibam em dispositivos diferentes.	
Medida de Resposta:	
O sistema deve se adaptar a resoluções de tela dos diversos dispositivos, sem perder funcionalidades.	
Considerações sobre a arquitetura:	
Riscos	A experiência de navegação pode ser altamente prejudicada pela qualidade da rede em que se está tentando o acesso. Além disso, resoluções extremamente pequenas de dispositivos muito antigos poderão causar alguns deslocamentos indesejáveis de componentes (porém sem a perda das funcionalidades).
Pontos de Sensibilidade:	Não existe
Tradeoff:	Não existe

O usuário autenticado, acessa o módulo de incidentes em um ambiente móvel.



**Figura 60 – Avaliação**



**Figura 61 – Avaliação**



VIVO 19:03 39%

pucminastcc.azurewebsites.net

IdIncidente

Descricao

Não Conformidade

Todos

Estado

Todos

+Adicionar Pesquisar

	Código	Descrição	Não Conformidade
Editar	12	O Produto saiu da empresa fora do conforme	Produto não embalado corretamente de acordo com

© 2020 - SGQ

< > + 26 ...

**Figura 62 – Avaliação**

VIVO 19:03 39%

pucminastcc.azurewebsites.net

IdIncidente

Geral

Descricao

Não Conformidade

Todos

Estado

Aberto

Voltar

Salvar

© 2020 - SGQ

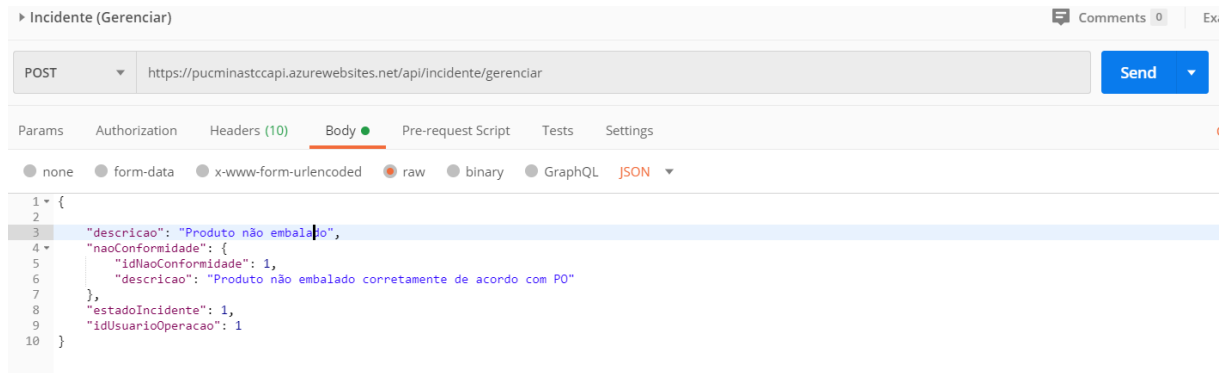
< > + 26 ...

**Figura 63 – Avaliação**

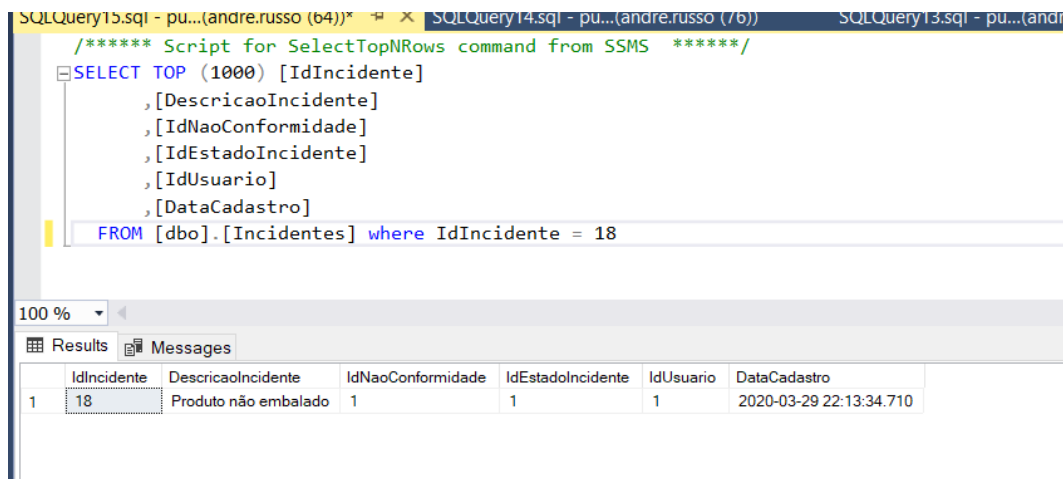
- Cenário 4:

Atributo de qualidade:	Interoperabilidade
Requisito de qualidade:	O sistema deve se comunicar com vários sistemas, com tecnologias heterogêneas.
Preocupação:	
Possibilitar que o sistema realize integrações com sistemas terceiros	
Cenários(s):	
Cenário 4	
Ambiente:	
Sistema em operação normal	
Estímulo:	
Sistema executando registro de incidente vindo de um sistema terceiro	
Mecanismo:	
Criar um mecanismo de integração utilizando a tecnologia REST recebendo um Json para registrar o incidente provindo de um sistema de incidente terceiro.	
Medida de Resposta:	
O sistema conseguiu registrar o incidente provindo de um sistema terceiro	
Considerações sobre a arquitetura:	
Riscos	Comunicações via rede estão sujeitas à instabilidade e indisponibilidade do ambiente a que se deseja acessar. Se não houver um mecanismo de retry para chamadas sem sucesso, pode-se acabar um processamento com os dados em um estado inconsistente.
Pontos de Sensibilidade:	Não existe
Tradeoff:	Não existe

Sistema terceiro envia via API Rest JSON o registro de um incidente



**Figura 64 – Avaliação**



**Figura 65 – Avaliação**

## 6.4. Resultado

Levando em consideração os requisitos funcionais e não funcionais, podemos observar que a arquitetura proposta atende as necessidades do projeto, a avaliação mostrou resultados positivos.

Requisitos não funcionais	Testado	Homol
Segurança – o sistema deve apresentar altos padrões de segurança	Sim	Sim
Usabilidade – o sistema deve prover boa usabilidade	Sim	Sim
Acessibilidade - o sistema deve ser suportar ambientes web responsivos e ambientes móveis	Sim	Sim

Interoperabilidade - o sistema deve se comunicar com vários sistemas, com tecnologias heterogêneas	Sim	Sim
--	-----	-----

O sistema foi desenvolvido na linguagem C# utilizando como tecnologia o ASP.NET Core 3.0, suporta ambientes web e móveis, por meio de páginas HTML, JavaScript e BootStrap, apresentando um layout simples intuitivo e fácil utilização. Podemos observar um ótimo desempenho na velocidade da todas as operações, com a arquitetura proposta de contêiner em nuvem híbrida, o sistema se recupera na ocorrência de qualquer possível falha, e aloca mais recursos conforme a demanda de utilização, permitindo assim a disponibilidade 24 horas.

O desenvolvimento proposto em contêineres e módulos, facilitou a manutenção e reduziu o acoplamento do sistema, permitindo cada parte do sistema possuir suas próprias versões e ser corrigido separadamente. Durante o desenvolvimento utilizamos o Github que possibilitou a integração continua para o DevOps.

## 7. Conclusão

Este trabalho apresentou um projeto arquitetural para uma plataforma de controle de qualidade voltado para a indústria automotiva. Utilizamos diversas tecnologias, a maioria delas escolhidas de forma a otimizar ao máximo o custo x benefício, lembrando é claro, que as escolhas foram limitadas de acordo com o conhecimento da equipe de desenvolvimento. Foram apresentadas sugestões de implantações em nuvem containerizada, porém não há qualquer necessidade, podendo ser implantado da forma como desejado. Foi proposta a utilização do Microsoft Azure para a criação da nuvem híbrida para resolver o problema de picos de estresse no sistema e problemas no servidor local, os componentes são independentes e modulares de forma a facilitar a manutenção e escalabilidade. Por se tratar de um projeto com o tema complexo, encontramos dificuldades e percebemos a importância de uma boa arquitetura que atenda aos requisitos, e também a necessidade de não ficar preso a apenas uma tecnologia, quanto maior o leque de opções mais assertiva será a arquitetura. Para conclusão podemos ver que a arquitetura proposta atendeu todos os requisitos funcionais e não-funcionais, embora haja muita margem para melhorias e uma infinidade de tecnologias não exploradas. Para uma próxima versão ou um próximo projeto, levando em conta toda a experiência adquirida a cada projeto, a arquitetura será otimizada tendo em vista a margem para melhora.

## REFERÊNCIAS

Microsoft. Guia do .NET Core. < <https://docs.microsoft.com/pt-br/dotnet/core/> >, 2019.  
Docker. Docker Documentation. < <https://docs.docker.com/> >, 2019.  
Microsoft. Documentação do Azure. < <https://docs.microsoft.com/pt-br/azure/?product=featured> >, 2019.  
Governo Federal. Portal da Transparência. < <http://www.portaltransparencia.gov.br/despesas/orgao?ordenarPor=orgaoSuperior&direcao=asc> >, 2019.  
Qualiex. Ferramentas. < <https://qualiex.com/#ferramentas> >, 2019.

## APÊNDICES

Código Fonte: <https://github.com/andreirusso/PUCMinasTCC>

Url do vídeo de apresentação da POC: <https://youtu.be/rmuYihNb25s>

URL do site hospedado na nuvem: <https://pucminastcc.azurewebsites.net/>

Url da Api hospetada na nuvem: <https://pucminastccapi.azurewebsites.net/>