

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

Guilherme Nunes Fontans

SISTEMA DE GESTÃO AMBIENTAL PARA EMPRESAS DE MINERAÇÃO

Porto Alegre
2020

Guilherme Nunes Fontans

SISTEMA DE GESTÃO AMBIENTAL PARA EMPRESAS DE MINERAÇÃO

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

Orientador(a): Prof. Pedro Alves de Oliveira

Porto Alegre

2020

A “dedicatória” é opcional.

Se quiser, pode escrevê-la sobre este texto.

*Se não, basta apagar o conteúdo desta página. (Formatação: fonte
Arial 12, com alinhamento à direita, espaçamento de 1,5 e itálico
opcional.)*

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

Nos últimos anos o Brasil e o mundo viu nos noticiários os acidentes ocorridos no rompimento de barragens pertencentes a empresas de mineração, sendo a maioria das vítimas pessoas das cidades de Mariana e Brumadinho, no qual veio a ocorrer rompimentos das barragens e vindo a deixar inúmeras vítimas fatais e milhares de pessoas desabrigadas. Tal catástrofe poderia ter sido evitada se houvesse um controle melhor das atividades minerárias dessas regiões, tanto pelas empresas mineradoras, quanto da Agência Nacional de Mineração do Brasil. E com base nessa necessidade foi criado esse projeto arquitetural, para que seja apresentado uma plataforma que seja capaz de auxiliar nos principais processos dessas empresas, e também auxiliar no monitoramento das barragens através de sensores para que possa ser tomado alguma ação preventiva caso seja encontrado alguma possível anomalia.

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

SUMÁRIO

1. Objetivos do trabalho	7
2. Descrição geral da solução	7
2.1. Apresentação do problema	7
2.2. Descrição geral do software (Escopo)	8
3. Definição conceitual da solução	8
3.1. Requisitos Funcionais	8
3.2 Requisitos Não-Funcionais	10
3.3. Restrições Arquiteturais	15
3.4. Mecanismos Arquiteturais	16
4. Modelagem e projeto arquitetural	17
4.1. Modelo de casos de uso	17
4.2. Descrição resumida dos casos de uso	22
4.3. Modelo de componentes	25
4.4. Modelo de implantação	27
5. Prova de Conceito (POC) / protótipo arquitetural	29
5.1. Implementação e Implantação	29
5.1.1. Requisitos não funcionais	29
5.1.2. Casos de uso	30
5.2. Interfaces/ APIs	30
6. Avaliação da Arquitetura	34
6.1. Análise das abordagens arquiteturais	34
6.2. Cenários	34
6.3. Avaliação	35
6.4. Resultado	45
7. Conclusão	47
REFERÊNCIAS	48
APÊNDICES	49
CHECKLIST PARA VALIDAÇÃO DOS ITENS E ARTEFATOS DO TRABALHO..	50

1. Objetivos do trabalho

O objetivo deste trabalho é apresentar a descrição do projeto arquitetural de uma aplicação que auxiliará nos processos de negócio de empresas de mineração. Para que esse propósito seja alcançado, foi levantado o requisito de sete módulos necessários para contemplar todo o escopo de processos dessas empresas, no qual cada requisito dos módulos será detalhado nesse projeto.

Os objetivos específicos são:

- Criar módulo de cadastro de ativos, permitindo que seja possível o cadastro de insumos das empresas, podendo ser possível também agendar as manutenções preventivas dos equipamentos
- Criar módulo de monitoramento que receberá dados de sensores adicionados nas barragens para que se possa saber como está o status de cada barragem cadastrada, e também que possa receber dados manuais pelos técnicos que efetuam as aferições das barragens. Dependendo da criticidade das métricas recebidas, o sistema de monitoramento deverá comunicar-se com o sistema de segurança e comunicação.
- Criar módulo de segurança e comunicação, no qual poderá ser utilizado para emitir alertas aos afetados, podendo ser disparados manualmente por um operador ou quando outro módulo do sistema solicitar.

2. Descrição geral da solução

2.1. Apresentação do problema

Nos últimos anos, ocorreu no Brasil duas grandes tragédias devido ao rompimento de barragens utilizadas por empresas de mineração, uma das tragédias foi na cidade de Mariana em 2015, e a outra na cidade de Brumadinho em 2019, ambas no estado de Minas Gerais. O rompimento dessas barragens veio a deixar centenas de mortos e milhares de pessoas desabrigadas, devido a destruição de suas casas pela lama trazida pelo rompimento.

Esses problemas ocorreram devido as empresas responsáveis não terem os controles necessários dos seus processos e acompanhamento de como estão os equipamentos utilizados

durante todo o processo de mineração, e também pela Agência Nacional de Mineração do Brasil por não fiscalizar melhor essas empresas.

Com um maior controle, também seria possível amenizar tais tragédias, como prever possíveis rompimentos de barragens devido a análise do comportamento das mesmas, podendo assim evacuar as pessoas que moram em áreas de risco e de funcionários que trabalham no local.

2.2. Descrição geral do software (Escopo)

A aplicação trata-se de um conjunto de módulos, no qual cada um possui uma determinada função dentro do sistema em si, que tem por finalidade auxiliar nos processos das mineradoras e também do seu acompanhamento, como um módulo para geração de relatórios, um módulo para cadastro dos ativos da empresa e que também possa ser controlado as manutenções desses ativos, um módulo de segurança e comunicação entre outros.

Com a utilização dos módulos propostos neste projeto, poderá ser possível emitir alertas para as pessoas que estão em áreas de risco de uma barragem, prever algum possível incidente nessas barragens através dos monitoramentos feitos através de sensores, e também de controlar as manutenções dos equipamentos utilizados durante todo o processo de mineração, além também de possibilitar a integração com sistemas de normas ambientais para que garanta que os processos da empresa não estejam de maneira irregular perante o meio ambiente.

3. Definição conceitual da solução

3.1. Requisitos Funcionais

Módulo de autenticação

- O sistema deve permitir o cadastro de novos usuários.
- O sistema deve permitir que um usuário efetue login.
- O sistema deve permitir que um usuário efetue o logoff.

Módulo de cadastros de ativos

- O sistema deve permitir o cadastro de equipamentos utilizados durante o processo de mineração.

- O sistema deve permitir o cadastro dos minérios recolhidos.
- O sistema deve permitir o cadastro de barragens.
- O sistema deve permitir o cadastro de pessoas que estão trabalhando em barragens cadastradas.

Módulo de controle de processos minerários

- O sistema deve permitir que seja registrado paradas e problemas com a produção diária.
- O sistema deve ter integração com o sistema da Agencia Nacional de Mineração para fins de consultar o status dos processos abertos.
- O sistema deve permitir o controle das vistorias das barragens.
- O sistema deve permitir o controle dos incidentes ocorridos durante o processo de mineração.
- O sistema deve controlar todo o processo de lavra do minério, desde a extração até o beneficiamento.

Módulo de monitoramento de barragens

- O sistema deve comunicar-se com sistemas externos como os sistemas das Defesa Civil municipais e estaduais envolvidos.
- O sistema deve permitir que seja cadastrado sensores que serão monitorados.
- O sistema deve permitir que seja cadastrado tipos de incidentes.
- O sistema deve permitir que seja cadastrado aferições feitas pelos consultores contratados.
- O sistema deve manter o histórico de incidentes ocorridos.
- O sistema deve interpretar os incidentes ocorridos, e de acordo com o tipo de incidente, comunicar o módulo de segurança e comunicação para que os devidos alertas sejam emitidos.

Módulo de segurança e comunicação

- O sistema deve permitir que seja cadastrados áreas que deverão receber alertas quando for necessário.
- O sistema deve permitir que seja cadastrado os contatos das pessoas que estão nas áreas cadastradas.

- O sistema deve emitir alertas para as áreas que devem ser notificadas.

Módulo de inteligência do negocio

- O sistema deve analisar os dados históricos gerados pelo módulo de monitoramento e gerar simulações de possíveis incidentes usando progressão linear.
- O sistema deve analisar os processos minerários cadastrados e gerar dashboards para que possa auxiliar a gerencia na tomada de decisões.
- O sistema deve analisar os insumos cadastrados e gerar simulações de futuros equipamentos que deverão ser substituídos ou serem revisados para que possa auxiliar a gerência na tomada de decisões.

Módulo de compliance

- O sistema deve possuir acesso a sistemas externos para fins de manter alinhados as suas atividades de acordo com as normas nacionais e internacionais do setor minerário assim como aprimorar os controles de governança.
- O sistema deve controlar a segurança dos funcionários.

Módulo de relatórios de acompanhamento

- O sistema deve permitir que seja gerado relatórios dos incidentes ocorridos.
- O sistema deve permitir que seja gerado relatório por período com gráficos de acordo com as métricas recebidas pelos sensores.
- O sistema deve permitir que seja gerado relatório por período de cada fase do processo de mineração.

3.2 Requisitos Não-Funcionais

- Usabilidade – O sistema deve prover boa usabilidade.

Estímulo	Usuário cadastrando um insumo no sistema
Fonte de estímulo	Usuário acessando a funcionalidade de cadastro de ativos do sistema
Ambiente	Produção, carga normal

Artefato	Módulo cadastro de ativos
Resposta	O frontend apresenta um layout simples e de navegação simples
Medida da resposta	Usuário consegue efetuar o cadastro do insumo em menos de cinco minutos

- Usabilidade – O sistema deve possuir interface responsiva, adequando o layout de acordo com o dispositivo que está sendo utilizado para o acesso.

Estímulo	Acesso ao sistema de um dispositivo móvel
Fonte de estímulo	Usuário acessando o sistema utilizando o seu celular
Ambiente	Produção, carga normal
Artefato	Qualquer módulo do sistema
Resposta	A usabilidade do sistema deve permanecer igual tanto no computador quanto no dispositivo móvel
Medida da resposta	Usuário consegue efetuar as mesmas operações com a mesma facilidade tanto no computador quanto no celular

- Segurança – O sistema não deve permitir o acesso aos módulos sem o usuário estar logado.

Estímulo	Acesso a uma página protegida do sistema sem estar logado
Fonte de estímulo	Usuário
Ambiente	Produção, carga normal
Artefato	Módulo de cadastro de ativos, módulo de controle de processos minerários, módulo de monitoramento de barragens, módulo de segurança e comunicação, módulo inteligência do negócio, módulo de compliance e módulo de relatórios de acompanhamento

Resposta	O sistema redireciona o usuário para a tela de login
Medida da resposta	O sistema não deve permitir que o usuário acesse alguma página desses módulos sem estar logado

- Manutenibilidade – O sistema deve ser fácil aplicar manutenções.

Estímulo	Ajuste em um módulo do sistema
Fonte de estímulo	Desenvolvedor aplicou um patch para corrigir um bug em dos módulos do sistema
Ambiente	Produção, carga normal
Artefato	Qualquer módulo do sistema
Resposta	Após o código com a correção for para a branch master, o gitlab cria uma nova imagem deste módulo
Medida da resposta	Caso os testes tenham passado, é gerado uma nova imagem docker do módulo testado, pronto para ir para deploy sem afetar os demais módulos.

- Manutenibilidade – O sistema deve ser fácil de testar.

Estímulo	Teste automatizado dos módulos do sistema
Fonte de estímulo	Pipeline de integração continua
Ambiente	Desenvolvimento
Artefato	Qualquer módulo do sistema
Resposta	Execução dos testes unitários do módulo testado
Medida da resposta	Caso os testes tenham passado, é gerado uma nova imagem docker do módulo testado

- Disponibilidade – O sistema deve operar mesmo após um dos seus nodos ficar fora.

Estímulo	Shutdown em um dos nodos da aplicação
Fonte de estímulo	Administrador do servidor de aplicação
Ambiente	Produção, carga normal
Artefato	Gerenciador de cluster
Resposta	Usuários que estão utilizando o sistema permanecem utilizando o mesmo sem nenhuma interferência
Medida da resposta	Todas as solicitações que estavam sendo feitas pela instância que sofreu shutdown devem ser processadas por outra instância e devolver a resposta para o cliente.

- Interoperabilidade – O sistema deve comunicar-se com sistemas externos através do uso de APIs.

Estímulo	Consulta a sistema externo de Gestão de Normas Ambientais
Fonte de estímulo	Sistema consultando dados de um sistema externo
Ambiente	Produção, carga normal
Artefato	Módulo de controle de processos minerários, módulo de inteligência do negócio, módulo de compliance e módulo de relatórios de acompanhamento
Resposta	O sistema externo consultado retorna os dados solicitados
Medida da resposta	Comunicação com o sistema externo efetuado com sucesso

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

Estímulo	Usuário abrindo uma tela do sistema que faz uso de dados armazenados em base de dados.
-----------------	--

Fonte de estímulo	Usuário utilizando o sistema
Ambiente	Produção, carga normal
Artefato	Módulo de cadastro de ativos, módulo de controle de processos minerários, módulo de monitoramento de barragens, módulo de segurança e comunicação, módulo inteligência do negócio, módulo de compliance e módulo de relatórios de acompanhamento
Resposta	Usuário recebe os dados normalmente no primeiro acesso e ao solicitar os mesmos dados as informações são trazidas mais rapidamente
Medida da resposta	O sistema cacheia por um minuto o resultado de uma requisição feita por um usuário, caso ele faça a mesma requisição as informações que ele irá receber irá vir do cache

- Interoperabilidade – Os sensores de monitoramento da barragem devem enviar os seus dados para o broker de comunicação.

Estímulo	Envio de dados dos sensores para o broker de comunicação.
Fonte de estímulo	Sensores de monitoramento das barragens
Ambiente	Produção, carga normal
Artefato	Broker de comunicação
Resposta	O broker recebe os dados do sensor
Medida da resposta	Comunicação do sensor com o broker efetuado com sucesso

- Interoperabilidade – Os Módulo de monitoramento deve comunicar-se com o módulo de comunicação e segurança quando há a necessidade de enviar alertas.

Estímulo	Módulo de monitoramento identifica um possível risco de rompimento de barragem
-----------------	--

Fonte de estímulo	Módulo de monitoramento
Ambiente	Produção, carga normal
Artefato	Modulo de monitoramento
Resposta	O módulo de monitoramento envia os dados necessários para que o módulo de segurança e comunicação dispare os alertas
Medida da resposta	Comunicação do módulo de monitoramento com o modulo de comunicação e segurança efetuado com sucesso.

- Interoperabilidade – O sistema deve comunicar com algum serviço externo para envio de alertas.

Estímulo	Módulo de segurança e comunicação precisa disparar um alerta
Fonte de estímulo	Módulo de segurança e comunicação
Ambiente	Produção, carga normal
Artefato	Modulo de segurança e comunicação
Resposta	O módulo de segurança e comunicação esse comunica com o serviço de Sendgrid passando os dados para o envio do email
Medida da resposta	Email é disparado pelo Sendgrid

3.3. Restrições Arquiteturais

- O sistema deve ser desenvolvido em módulos para facilitar a implantação.
- O sistema deve ser hospedado on-premise.
- O sistema deve ter sua arquitetura baseada em serviços.
- O sistema deve ter seu build feito através de integração continua.
- O sistema deve ter pipelines de teste em sua integração continua.
- O sistema deve possuir integrações com sistemas externos.

- O sistema deve possuir seus serviços acessados apenas pelas instancias que rodem o api gateway.
- O sistema deve possuir seu layout responsivo, para que possa ser utilizado com facilidade tanto em computadores como dispositivos moveis.

3.4. Mecanismos Arquiteturais

Mecanismo de análise	Mecanismo de design	Mecanismo de implementação
Integração entre outros módulos e/ou sistemas	Interfaces de comunicação entre os módulos utilizando JSON	APIs REST
Comunicação entre processos	Container web e aplicação	Docker
Frontend	Interface entre o usuário e a aplicação	React.js
Frontend	Dashboards com dados dos sensores	Kibana
Build	Ferramenta de build das imagens Docker	GitLab
ApiGateway	Camada intermediaria entre o frontend e os serviços da aplicação	Node.js, Express.js
Segurança	Autenticação e Autorização	JWT
Persistência	Base de dados NOSQL para autenticação e armazenamento de dos dados do sistema	Mongodb
Persistência	Base de dados NOSQL para persistência dos dados dos sensores	InfluxDB
Notificação	Envio de e-mails de alerta disparados pelo sistema	SendGrid
Versionamento	Versionamento do código fonte da aplicação	Git
Alta disponibilidade	Balanceamento de carga das requisições para os serviços,	Nginx, Kubernetes, Zabbix

	orquestração dos containers e monitoramento dos hosts	
Cache	Ferramenta de cache	Redis
Broker Sensores	Ferramenta responsável por receber dados dos sensores	Eclipse Mosquitto

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

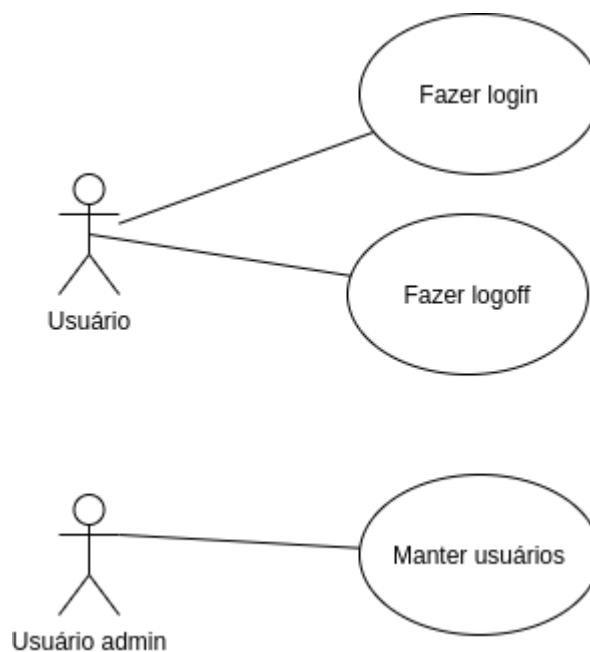


Figura 1 - Casos de uso do módulo de cadastro de ativos

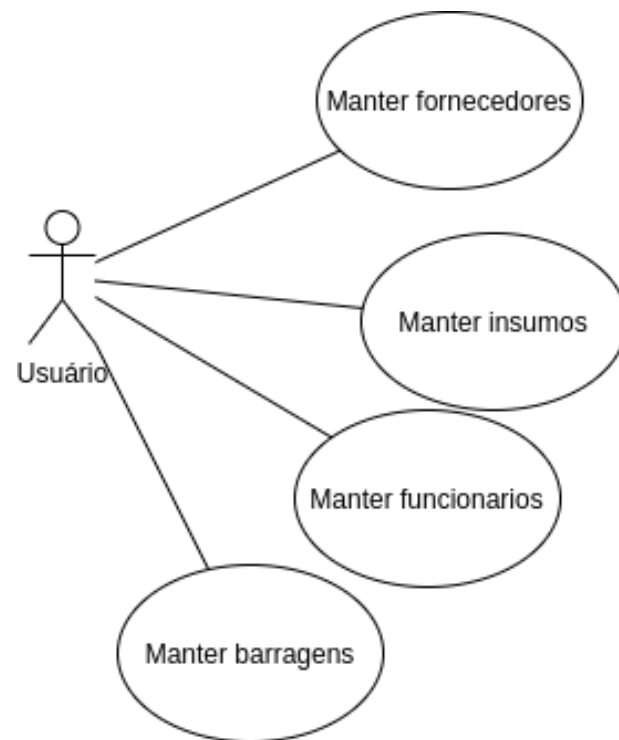


Figura 2 - Casos de uso do módulo de cadastro de ativos

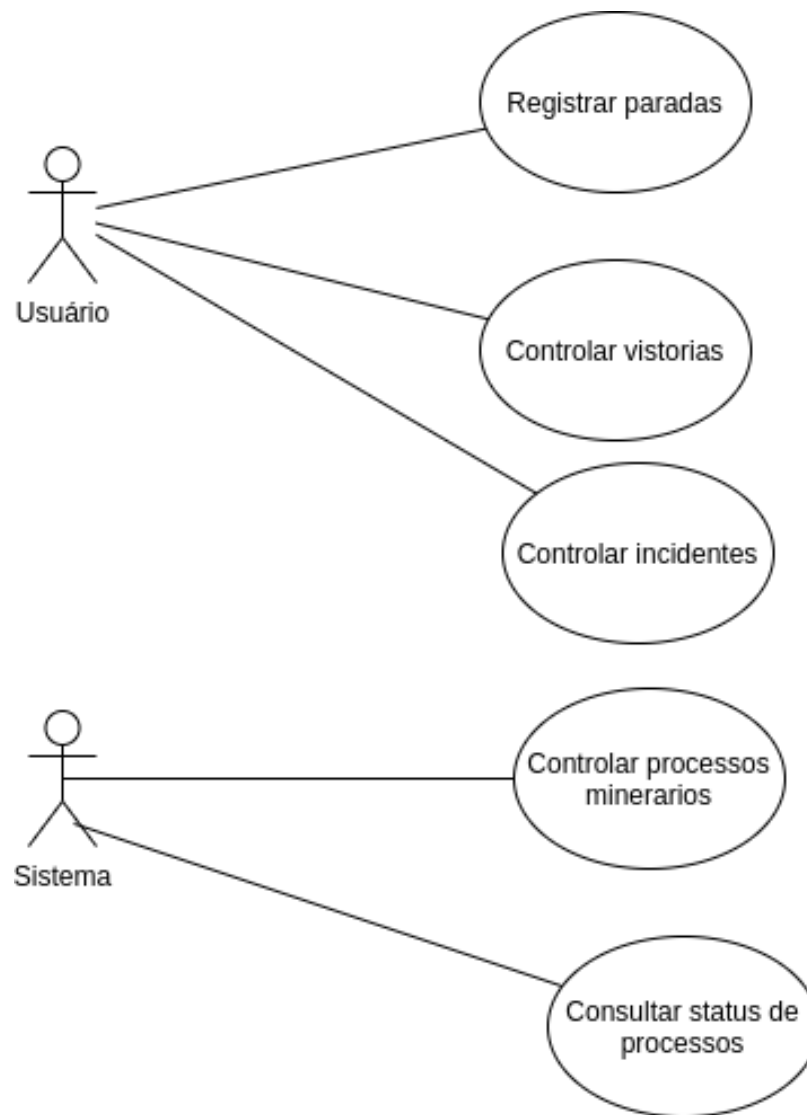


Figura 3 - Casos de uso do módulo de controle de processos minerários

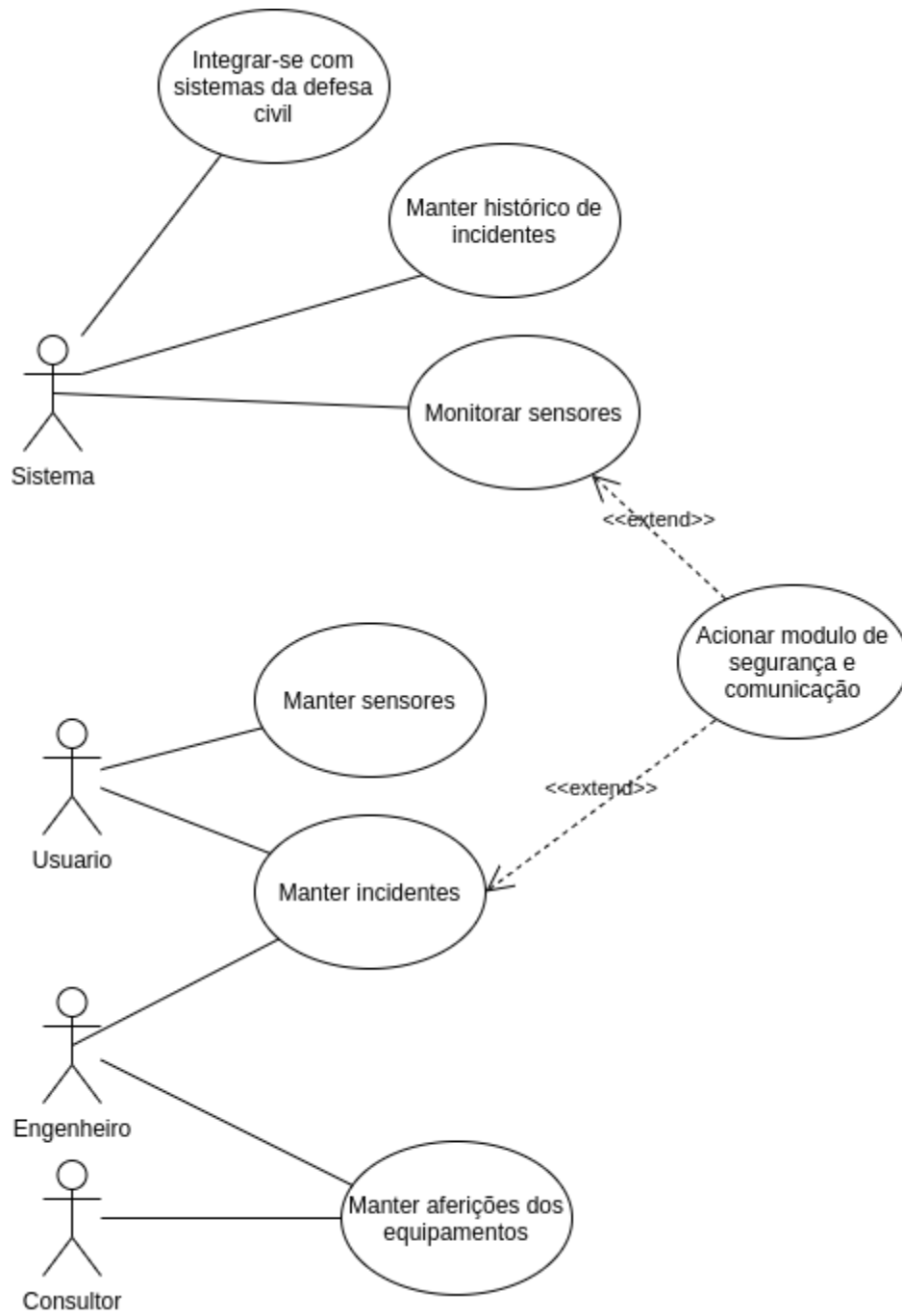


Figura 4 - Casos de uso do módulo de monitoramento

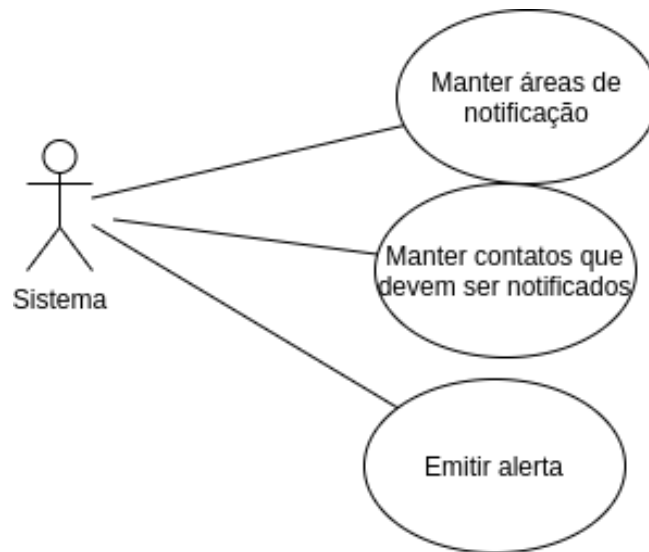


Figura 5 - Casos de uso do módulo de segurança e comunicação

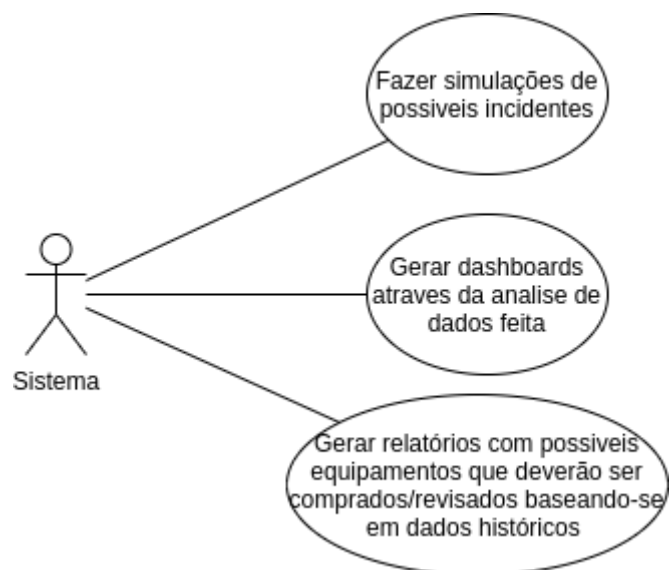


Figura 6 - Casos de uso do módulo de inteligência do negócio

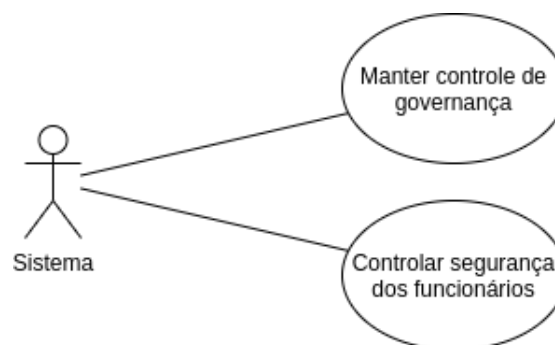


Figura 7 - Casos de uso do módulo de compliance

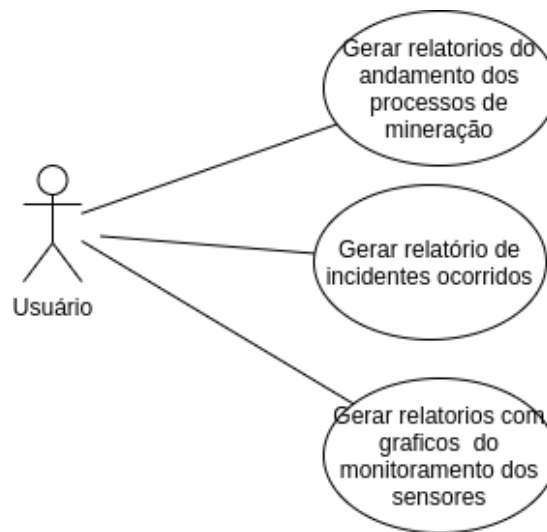


Figura 8 - Casos de uso do módulo de relatórios de acompanhamento

4.2. Descrição resumida dos casos de uso

- **Módulo de autenticação**

- Caso de uso: **Efetuar login**
 - Descrição resumida: Essa funcionalidade permite ao usuário obter o token de autenticação após informar seu usuário e senha.
- Caso de uso: **Efetuar logoff**
 - Descrição resumida: Essa funcionalidade permite ao usuário deslogar do sistema.
- Caso de uso: **Manter usuários**
 - Descrição resumida: Essa funcionalidade permite que o usuário administrador possa visualizar, cadastrar, atualizar e excluir os demais usuários do sistema.

- **Módulo de cadastro de ativos**

- Caso de uso: **Manter fornecedores**
 - Descrição resumida: Essa funcionalidade permite ao usuário visualizar, cadastrar, atualizar e excluir fornecedores.
- Caso de uso: **Manter insumos**
 - Descrição resumida: Essa funcionalidade permite ao usuário visualizar, cadastrar, atualizar e excluir os insumos da empresa, isso inclui também manter as vistorias desses insumos da empresa.
- Caso de uso: **Manter funcionários**
 - Descrição resumida: Essa funcionalidade permite que ao usuário visualizar, cadastrar, atualizar e excluir funcionários.
- Caso de uso: **Manter barragens**

- Descrição resumida: Essa funcionalidade permite que ao usuário visualizar, cadastrar, atualizar e excluir as barragens em que a empresa está trabalhando.
- **Módulo de controle de processos minerários**
 - Caso de uso: **Registrar parada**
 - Descrição resumida: Essa funcionalidade permite ao usuário registrar paradas que podem ocorrer no dia, podendo ser registrados por turno de trabalho.
 - Caso de uso: **Controlar vistorias**
 - Descrição resumida: Essa funcionalidade permite que o usuário faça a gestão das vistorias da empresa, desde a inclusão de uma nova vistoria até o encerramento da mesma.
 - Caso de uso: **Controlar incidentes**
 - Descrição resumida: Essa funcionalidade permite ao usuário registrar incidentes que podem ocorrer no dia, podendo ser registrados por turno de trabalho.
 - Caso de uso: **Controlar processos minerários**
 - Descrição resumida: Essa funcionalidade permite o controle dos processos de lavra da mineração, desde a extração até o beneficiamento.
 - Caso de uso: **Consultar status dos processos**
 - Descrição resumida: Essa funcionalidade permite a consulta dos processos abertos pela empresa via requerimento junto a Agência Nacional de Mineração.
- **Módulo de monitoramento**
 - Caso de uso: **Integrar-se com sistema da defesa civil**
 - Descrição resumida: Essa funcionalidade permite que o sistema também tenha acesso aos incidentes detectados pela defesa civil, e insira-os no sistema.
 - Caso de uso: **Manter histórico de incidentes**
 - Descrição resumida: Essa funcionalidade permite que os dados dos incidentes possam ser vistos posteriormente.
 - Caso de uso: **Monitorar sensores**
 - Descrição resumida: Essa funcionalidade permite que seja enviado dados dos sensores para o sistema, para que o sistema possa analisar o dado que recebeu e gerar ou não um incidente, caso gere um incidente, deve comunicar o módulo de segurança e comunicação.
 - Caso de uso: **Manter sensores**
 - Descrição resumida: Essa funcionalidade permite que o usuário cadastre, edite e exclua sensores no sistema.
 - Caso de uso: **Manter incidentes**
 - Descrição resumida: Essa funcionalidade permite que ao usuário criar, editar e excluir um incidente no sistema.
 - Caso de uso: **Manter aferições dos equipamentos**

- Descrição resumida: Essa funcionalidade permite que o usuário cadastre o resultado de um aferimento feito em um dos insumos da empresa.
- **Módulo de segurança e comunicação**
 - Caso de uso: **Manter áreas de notificação**
 - Descrição resumida: Essa funcionalidade permite ao usuário cadastrar, editar e excluir áreas para serem notificadas.
 - Caso de uso: **Manter contatos que devem ser notificados**
 - Descrição resumida: Essa funcionalidade permite ao usuário cadastrar, editar e excluir os contatos das pessoas que devem ser notificadas.
 - Caso de uso: **Emitir alerta**
 - Descrição resumida: Essa funcionalidade permite que o sistema dispare alertas quando solicitado manualmente, ou pelo módulo de monitoramento.
- **Módulo de inteligência do negócio**
 - Caso de uso: **Fazer simulações de possíveis incidentes**
 - Descrição resumida: Essa funcionalidade permite ao sistema fazer simulações através do cruzamento de informações dos dados históricos, do monitoramento atual e de serviços externos, para fins de tentar prever algum futuro incidente tanto em curto, médio e longo prazo.
 - Caso de uso: **Gerar dashboards através da análise de dados feita**
 - Descrição resumida: Essa funcionalidade permite que o usuário visualize dashboards com as previsões futuras feitos pelo módulo.
 - Caso de uso: **Gerar relatórios com possíveis equipamentos que deverão ser comprados/revisados baseando-se em dados históricos**
 - Descrição resumida: Essa funcionalidade permite ao sistema analisar os dados históricos dos equipamentos, vistorias e aferições já efetuadas, para fins de criar dados que possam ser utilizados pelo módulo de relatórios para fins de identificar possíveis gastos futuros com substituição de equipamentos ou desfalque temporário para revisão dos mesmos.
- **Módulo de compliance**
 - Caso de uso: **Manter controle de governança**
 - Descrição resumida: Essa funcionalidade permite ao sistema manter as suas normas sempre alinhadas com as normas especificadas pelos demais órgãos externos, como Agência Nacional De Mineração, Ministério do Trabalho.
 - Caso de uso: **Manter segurança dos funcionários**
 - Descrição resumida: Essa funcionalidade permite ao sistema gerenciar todas as medidas de segurança que devem ser aplicadas aos funcionários e nos locais onde os mesmos circulam.
- **Módulo de relatório de acompanhamento**
 - Caso de uso: **Gerar relatórios do andamento dos processos de mineração**

- Descrição resumida: Essa funcionalidade permite que o sistema consulte o status dos processos que foram abertos junto a Agência Nacional de Mineração e gere relatórios dos processos solicitados.
- Caso de uso: **Gerar relatório de incidentes ocorridos**
 - Descrição resumida: Essa funcionalidade permite ao sistema gerar relatório com todos os incidentes que ocorreram por período.
- Caso de uso: **Gerar relatórios com gráficos do monitoramento dos sensores**
 - Descrição resumida: Essa funcionalidade permite que o sistema gere relatórios com gráficos referentes as atividades dos sensores localizados nas barragens, esses relatórios podem possuir mais de uma métrica, como pressão, umidade, geolocalização, entre outros.

4.3. Modelo de componentes

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

Abaixo segue a imagem de como os componentes estão organizados dentro da arquitetura proposta.

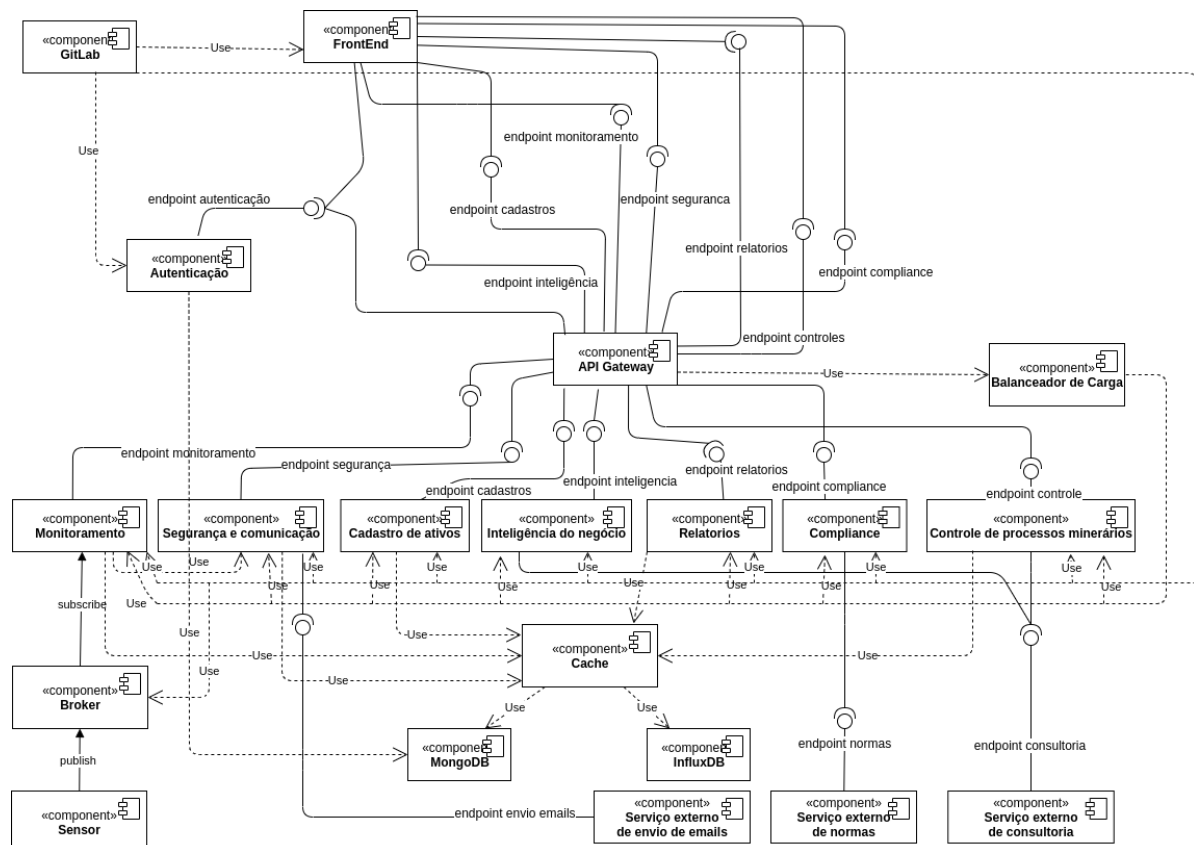


Figura 9 – Diagrama de componentes

Na imagem acima está sendo apresentado os componentes do sistema e como os mesmos estão distribuídos.

Para fins de manter a manutenibilidade do sistema fácil, o mesmo foi desenvolvido tendo o frontend completamente separado do backend, este último tendo cada módulo desenvolvido na arquitetura de micro serviço, fazendo com que a comunicação entre frontend e backend seja feita através de chamadas de API, trafegando os seus dados no formato json.

Todo o sistema foi projetado para rodar em ambiente *on premise*, sendo cada componente representado na figura acima um container Docker. Para a integração continua do projeto, foi utilizado os jobs do GitLab, facilitando a aplicação de todos os testes unitários e de integração após algum módulo do sistema sofrer alguma alteração, assim como criar uma nova imagem Docker para o módulo alterado, garantindo assim os requisitos não funcionais de testabilidade e manutenibilidade.

O sistema possui as APIs dos seus módulos privados, podendo apenas serem acessados a partir do componente de API gateway que fica entre o frontend e o backend, vindo a mascarar os endpoints reais da aplicação, recebendo as requisições que vem do frontend e repassa-las para os módulos solicitados utilizando o componente de load balance, no qual a tecnologia utilizada é o NGINX, para que verifique qual instância do serviço está mais liberada, garantindo assim o requisito não funcional de escalabilidade. Ainda em relação as APIs do sistema, as mesmas só são disponíveis desde que a requisição que está solicitando o serviço possua um token válido no cabeçalho, este token é obtido através da tela de login e senha que o frontend possui, no qual a mesma faz uma requisição para o endpoint de login do componente de autenticação, que caso o usuário e senha seja valido, é retornado o token que deverá ser mandado no cabeçalho das próximas requisições, o usuário não conseguirá acessar uma tela ou um recurso sem o token ou com um token inválido. Essa funcionalidade garante o requisito não funcional de segurança.

Para o módulo de monitoramento é utilizado sensores, que se trata de dispositivos que utilizam IOT, nos quais os mesmos encontram-se próximos das barragens gerando dados sobre o status das mesmas. Esses dispositivos utilizam o protocolo MQTT para a transmissão dos dados, essa transmissão é feita através de publish e subscribe, ou seja, o sensor publica a sua mensagem e o subscribe a ouve. Para que essa comunicação possa ser feita, é necessário a utilização de um broker para receber esses dados e repassá-los para os subscribers, o broker que será utilizado para esse fim é a ferramenta chamada Eclipse Mosquitto. Os subscribers serão os clients que ficarão ouvindo os dados que chegam dos sensores, nos quais irão salvar esses dados em uma base de dados para que possam ser analisados posteriormente e ou gerar gráficos de capacidade e performance para monitoramento de cada métrica monitorada nas barragens. Esses clients também serão os responsáveis por analisar os dados que chegam dos sensores e caso necessário, baseando-se em valores pré definidos para cada métrica, acionar o módulo de segurança e comunicação. Devido haver alta demanda de dados chegando pelos sensores e a

necessidade de ter esses dados em tempo real, a tecnologia selecionada para o armazenamento desses dados é o banco de dados temporal InfluxDB.

O sistema também prevê a integração com serviços externos, para a consulta dos processos minerários, envio de alertas por e-mail e consulta de normas ambientais, para isso o sistema respeita a interface exposta de cada um desses serviços, garantindo assim o requisito não funcional de interoperabilidade.

Com a finalidade de melhorar a performance no acesso aos dados do sistema foi adicionado a ferramenta Redis para uso do cache, sendo configurado para cachear as informações que são muito solicitadas, fazendo com que seja diminuído o tráfego de rede desnecessário e a carga imposta nas bases de dados, fazendo com que a navegação pelo sistema fique mais performática, reforçando o requisito funcional de desempenho.

Para fins de garantir o requisito funcional de confiabilidade, os componentes do sistema serão monitorados pela ferramenta Zabbix, no qual poderá emitir alertas para a equipe de TI responsável pelo monitoramento do sistema caso haja alguma anormalidade. Para reforçar o requisito não funcional de escalabilidade, será utilizado o Kubernetes para efetuar a orquestração dos módulos do sistema, o mesmo estará configurado para subir novos containers para atender a alta demanda quando necessário.

4.4. Modelo de implantação

Para uma melhor representação de como ficam alocados fisicamente os componentes citados na sessão anterior, segue abaixo o diagrama de implantação do sistema, apresentando em qual hardware os componentes estarão alocados para a implantação de um ambiente *on premise*. Também está sendo mostrado o protocolo utilizado para a comunicação entre um hardware e outro.

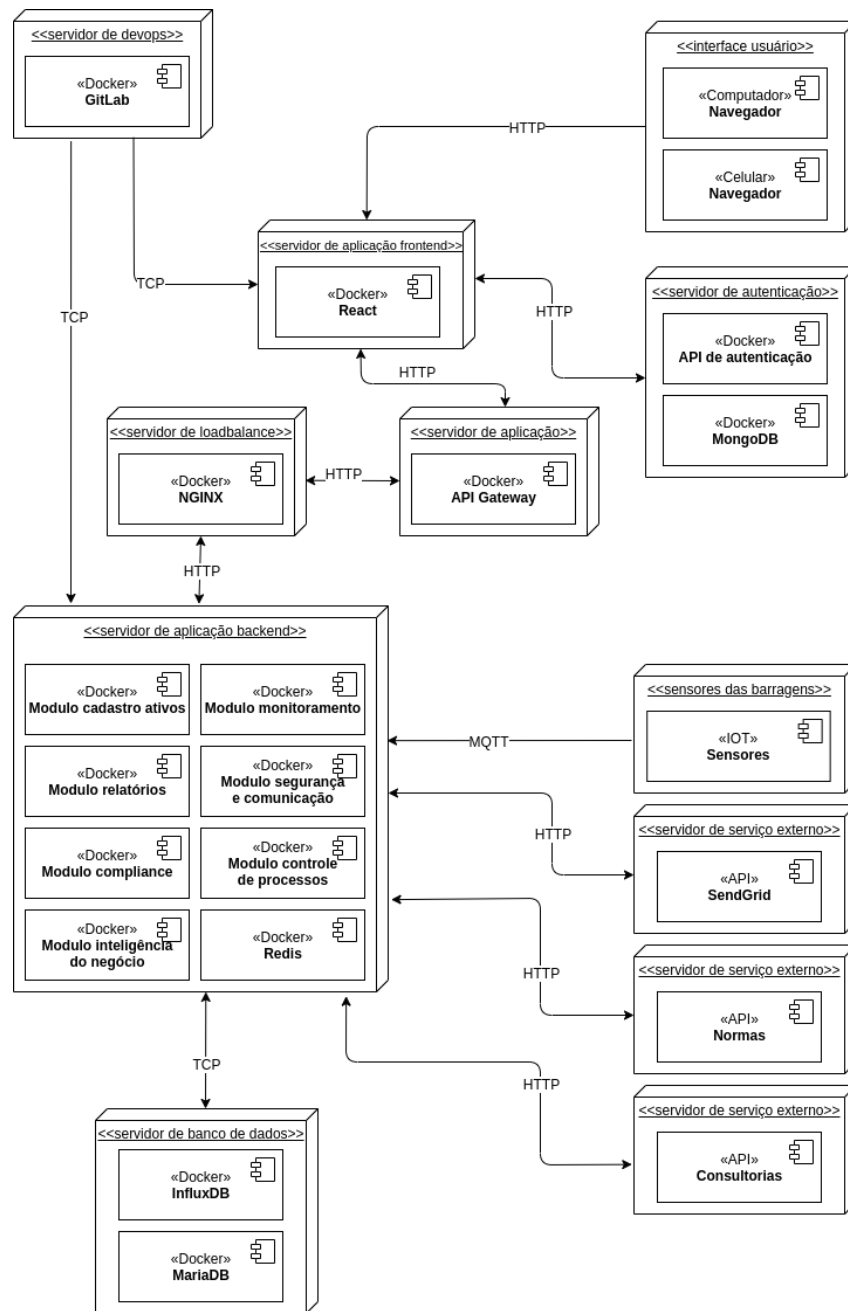


Figura 10 – Diagrama de implantação

5. Prova de Conceito (POC) / protótipo arquitetural

5.1. Implementação e Implantação

5.1.1. Requisitos não funcionais

A prova de conceito deste projeto arquitetural teve por objetivo validar se os requisitos não funcionais considerados mais importantes foram contemplados, abaixo é listado os requisitos funcionais selecionados:

- **Segurança:** O sistema não deve permitir o acesso aos módulos sem o usuário estar logado.

Esse requisito não funcional foi escolhido devido haver preocupação com a segurança da informação dos dados do sistema

Os critérios de aceite são:

- Na interface web, redirecionar o usuário para a tela de login caso o mesmo tente acessar alguma tela privada do sistema sem estar logado.
- Nas APIs do sistema, retornar o código 401(unauthorized) caso seja feita uma requisição para um dos endpoints privados do sistema sem passar no cabeçalho o x-access-token válido, no qual é obtido através do endpoint de autenticação.
- **Interoperabilidade** – O sistema deve comunicar com algum serviço externo para envio de alertas.

Esse requisito não funcional foi escolhido para fins de garantir que a comunicação com um dos serviços externos mais importante do sistema esteja operacional, pois ele é o responsável por enviar os alertas para os afetados.

Os critérios de aceite são:

- Após o cadastro de um incidente, deve ser disparado o e-mail de alerta para os afetados da área que ocorreu o incidente.
- **Usabilidade:** O sistema deve possuir interface responsiva, adequando o layout de acordo com o dispositivo que está sendo utilizado para o acesso.

Esse requisito não funcional foi escolhido devido haver preocupação com a experiência do usuário, pois como nem sempre os mesmos estarão utilizando um computador, é de extrema importância que seja possível efetuar as mesmas operações no celular, sem haver dificuldades.

Os critérios de aceite são:

- O sistema deve manter a facilidade de execução de suas ações quando o usuário estiver usando dispositivos móveis.
- Os componentes devem se adaptar de acordo com a resolução do dispositivo que o usuário está utilizando.

5.1.2. Casos de uso

Os casos de uso que foram escolhidos para a implementação dessa poc estão descritos na tabela abaixo:

Caso de uso	Módulo	Requisito não funcional
Logar no sistema	Autenticação	Segurança
Cadastrar incidentes	Monitoramento	Interoperabilidade / Usabilidade
Disparar alertas	Segurança e comunicação	Interoperabilidade
Cadastrar afetados	Segurança e comunicação	Usabilidade

5.2. Interfaces/ APIs

API	API REST para autenticação	
URL	http://localhost:5000/login	
Método	POST	
Headers	x-access-token	token
Parâmetros	username	login do usuário

	password	senha
--	----------	-------

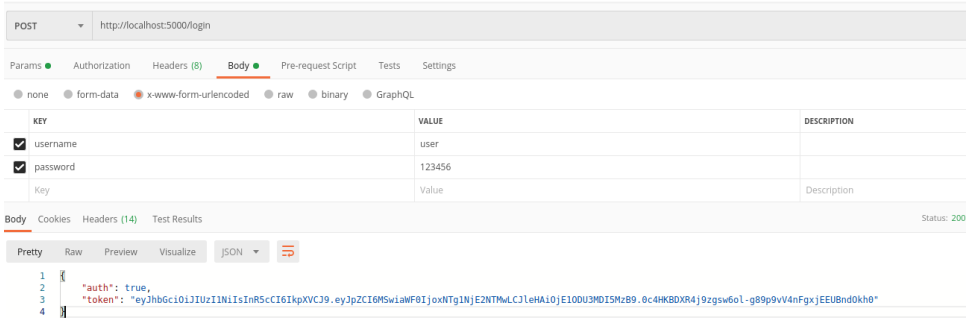


Figura 11 – exemplo de request e response da api de login

API	API REST para cadastro de incidentes	
URL	http://localhost:5000//incidentes	
Método	POST	
Headers	x-access-token	token
Parâmetros	data	data do incidente
	tipo	tipo de incidente
	area_id	id da área do incidente
	grauRisco	gravidade do incidente

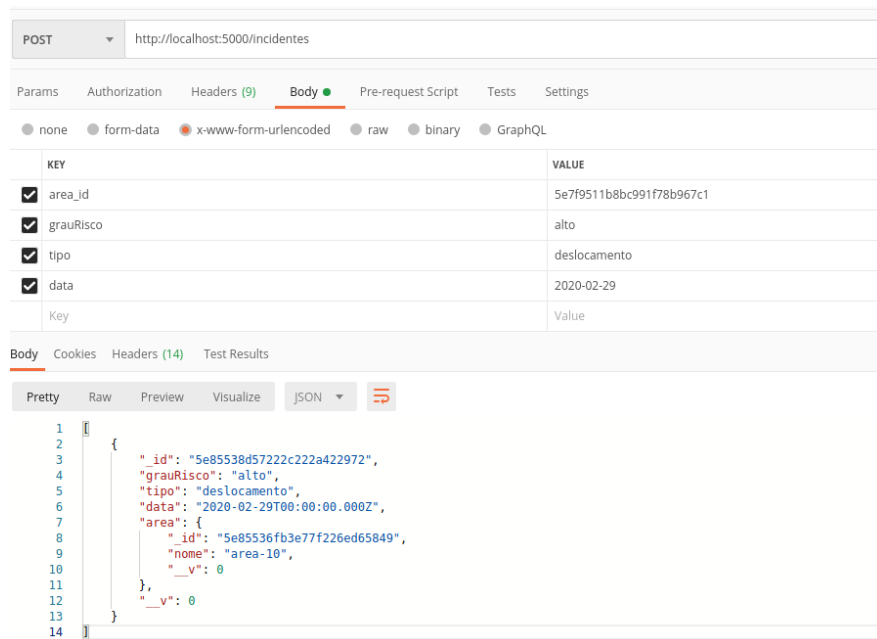


Figura 12 – exemplo de request e response da api de cadastro de incidentes

API	API REST para busca de áreas	
URL	http://localhost:5000//areas	
Método	GET	
Headers	x-access-token	token

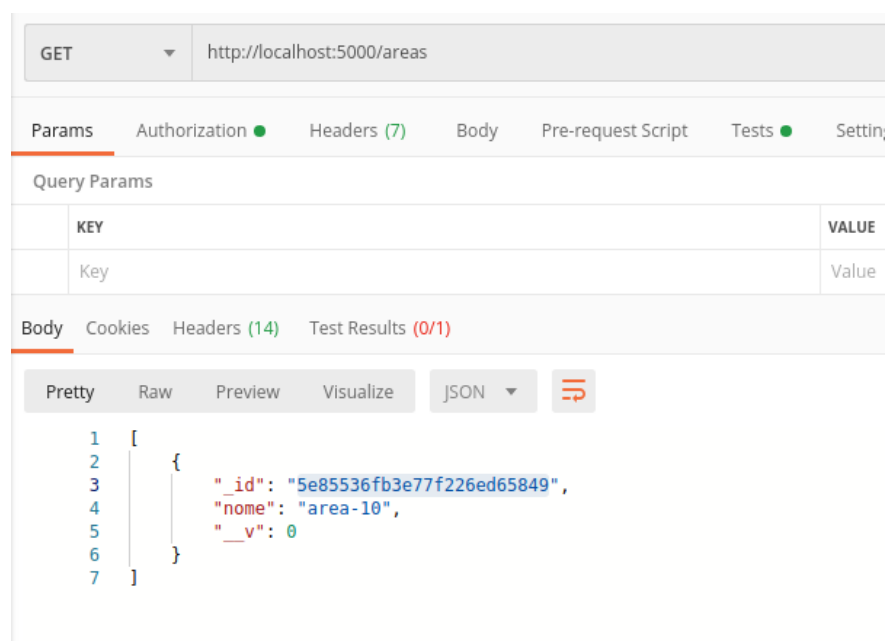
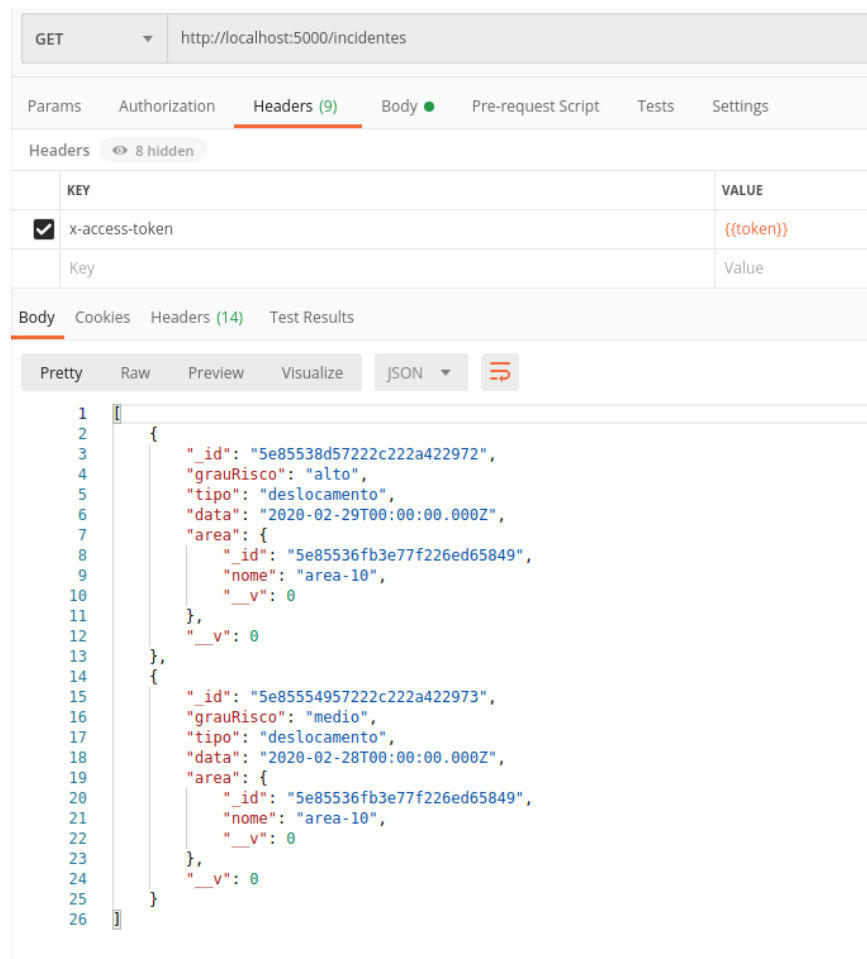


Figura 13 – exemplo de request e response da api de busca de áreas

API	API REST para busca de incidentes	
URL	http://localhost:5000//incidentes	
Método	GET	
Headers	x-access-token	token



The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** http://localhost:5000/incidentes
- Headers (9):** x-access-token (checked), Key, Value
- Body:** JSON (selected)
- Response (JSON):**

```

1  [
2    {
3      "_id": "5e85538d57222c222a422972",
4      "grauRisco": "alto",
5      "tipo": "deslocamento",
6      "data": "2020-02-29T00:00:00.000Z",
7      "area": {
8        "_id": "5e85536fb3e77f226ed65849",
9        "nome": "area-10",
10       "__v": 0
11      },
12     "__v": 0
13   },
14   {
15     "_id": "5e8554957222c222a422973",
16     "grauRisco": "medio",
17     "tipo": "deslocamento",
18     "data": "2020-02-28T00:00:00.000Z",
19     "area": {
20       "_id": "5e85536fb3e77f226ed65849",
21       "nome": "area-10",
22       "__v": 0
23     },
24     "__v": 0
25   }
26 ]

```

Figura 14 – exemplo de request e response da api de busca de incidentes

6. Avaliação da Arquitetura

6.1. Análise das abordagens arquiteturais

A arquitetura proposta foi projetada utilizando tecnologias open source e planejada para rodar em ambiente on-premise, na sua POC já está disponível as configurações para poder subir toda a aplicação utilizando o docker-compose, vindo a deixar a implementação menos dependente da plataforma que será utilizada.

6.2. Cenários

Os cenários descritos abaixo são os utilizados para testar os requisitos não funcionais listados na seção 5.1.1.

Cenário 1: Na tentativa de acesso a uma página privada o usuário deve ser redirecionado para a página de login caso não esteja autenticado. O usuário não pode conseguir acessar uma página privada do sistema sem estar logado no mesmo.

Cenário 2: Na tentativa de consumir algum serviço do sistema através da API, o usuário é obrigado a passar um x-access-token válido no cabeçalho, caso o token não seja mais válido ou não exista, então deverá ser retornado o código 401 (unauthorized).

Cenário 3: Quando o modulo de monitoramento receber uma atividade que está classificada para emitir alertas, o modulo de monitoramento deve comunicar-se com o módulo de segurança e comunicação através da api rest para que o respectivo módulo envie os alertas para os e-mails já cadastrados.

Cenário 4: Ao realizar o acesso na aplicação através de um dispositivo móvel, os componentes da interface deverão se adequar a resolução do dispositivo para ficar com uma boa acessibilidade para fins de contemplar os requisitos não funcionais.

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.

Categoria	Atributos de Qualidade	Cenários	IMP.	COM.
Confidencialidade	Segurança	Cenário 1: Acesso protegido as páginas privadas mediante login.	A	A
		Cenário 1: Acesso protegido aos serviços privados mediante token.	A	A
Compatibilidade	Interoperabilidade	Cenário 3: Comunicação entre os módulos sendo feita através das APIs	A	A
Funcionalidade	Usabilidade	Cenário 4: Adaptação das telas	A	M

6.3. Avaliação

Processo de avaliação dos cenários identificados no item anterior. O objetivo aqui é determinar os riscos, não riscos, pontos de sensibilidade, trade-off's e evidenciar o atendimento aos requisitos de qualidade.

- **Cenário 1:**

Atributo de Qualidade	Segurança
Requisito de Qualidade	Acesso protegido as páginas privadas mediante login.
Preocupação	
O usuário estar autenticado.	
Cenário	

Cenário 1.	
Ambiente	
Produção, carga normal.	
Estímulo	
Usuário tentando acessar uma página privada do sistema.	
Mecanismo	
Criar um mecanismo de autenticação que verifique a cada requisição se está sendo enviado o token esperado pelo backend.	
Medida de resposta	
O usuário deve ser redirecionado para tela de login caso não envie o token ou o mesmo seja inválido.	
Considerações sobre a arquitetura	
Riscos	O gerenciamento de autenticação é um ponto crítico para a segurança. Uma falha neste recurso poderia provocar vazamento de informações que não deveriam ser acessadas por pessoas indevidas.
Pontos de Sensibilidade	Servidor de aplicação operando em modo HTTP
Tradeoff	Não existe

- Evidências do cenário 1:

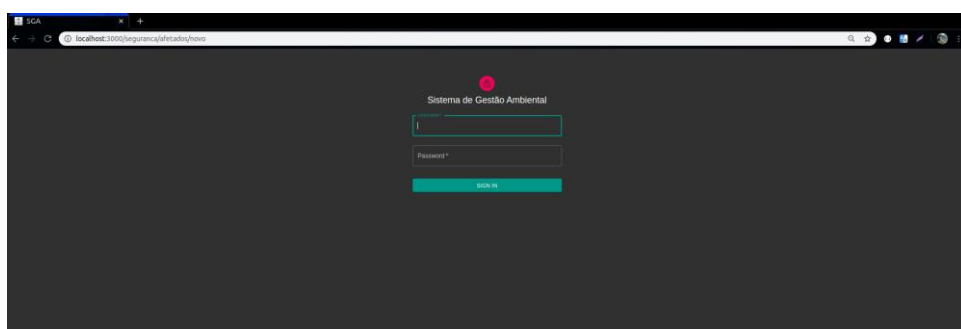


Figura 15 – Usuário tentando acessar uma tela do sistema sem estar logado e sendo redirecionado para a tela de login.

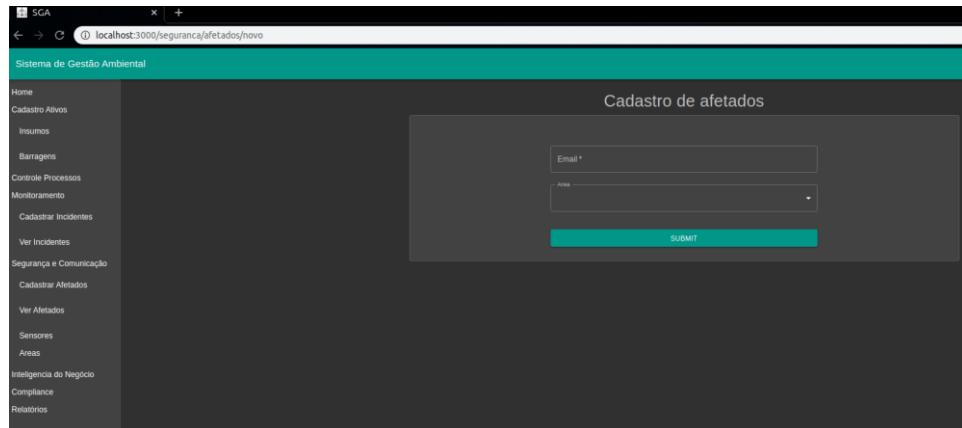


Figura 16 – Usuário acessando o sistema após ter feito login.

- **Cenário 2:**

Atributo de Qualidade	Segurança
Requisito de Qualidade	Acesso protegido aos serviços privados mediante token.
Preocupação	
O usuário ter passado um token válido.	
Cenário	
Cenário 2.	
Ambiente	
Produção, carga normal.	
Estímulo	
Usuário tentando acessar um recurso através da API rest do sistema sem informar o token x-access-token no cabeçalho da requisição.	
Mecanismo	
Criar um mecanismo de autenticação que verifique se o token foi passado e se o mesmo é válido.	

Medida de resposta	
O usuário receber o código http "401 - unauthorized" como resposta.	
Considerações sobre a arquitetura	
Riscos	O gerenciamento de autenticação é um ponto crítico para a segurança e crucial para a API do sistema. Ter uma API com autenticação por token garante o acesso a sistemas ou usuários que realmente deveriam ter acessos a um determinado recurso.
Pontos de Sensibilidade	Servidor de aplicação operando em modo HTTP
Tradeoff	Não existe

- Evidências do cenário 2:**

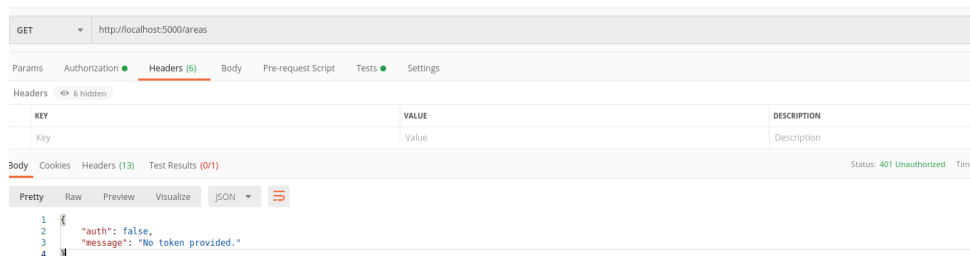


Figura 17 – Usuário consultando um recurso do sistema sem passar o token.

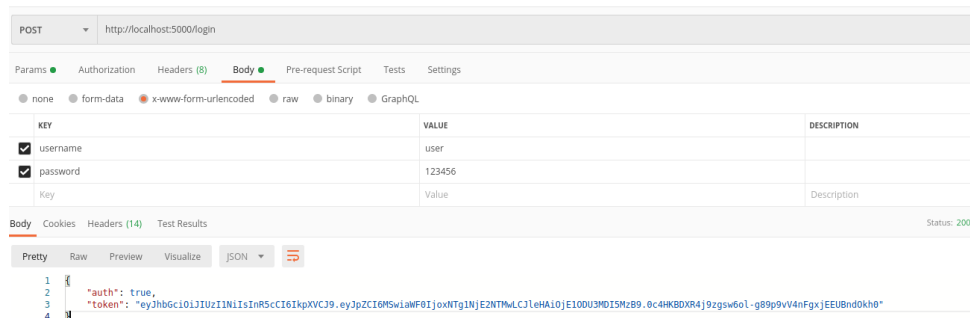


Figura 18 – Usuário buscando um token passando suas credenciais de acesso.

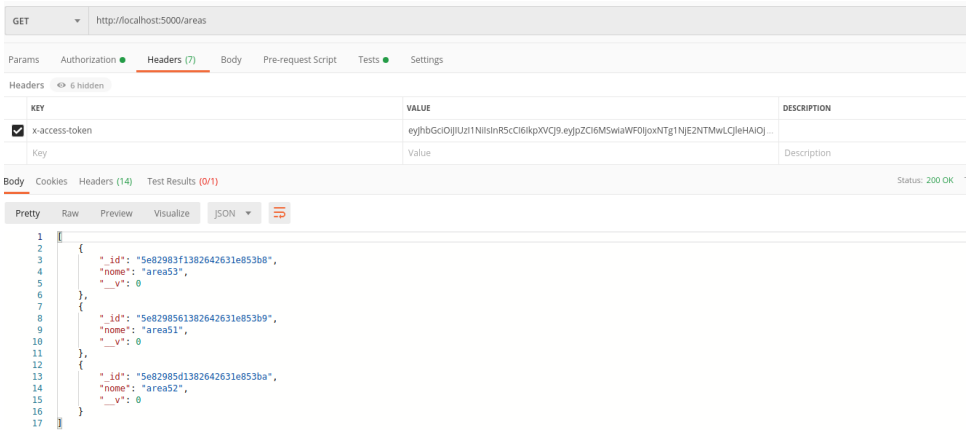


Figura 19 – Usuário consultando um recurso do sistema passando o token obtido.

• Cenário 3:

Atributo de Qualidade	Interoperabilidade
Requisito de Qualidade	Comunicação entre os módulos sendo feita através das APIs
Preocupação	
O sistema comunicar-se com outros módulos	
Cenário	
Cenário 3.	
Ambiente	
Produção, carga normal.	
Estímulo	
Sistema recebe um dado de um sensor que caracteriza-se ser um incidente, modulo de monitoramento então aciona módulo de segurança e comunicação.	
Mecanismo	
Modulo de monitoramento faz um request para a API de segurança e comunicação.	
Medida de resposta	
O usuário que está cadastro na área afetada receber o e-mail de alerta.	

Considerações sobre a arquitetura	
Riscos	Atraso na comunicação devida a quantidade de alertas que devem ser emitidos.
Pontos de Sensibilidade	Carga excessiva de requests.
Tradeoff	Poderia ser utilizado mensageria para a comunicação dos módulos.

- **Evidências do cenário 3:**

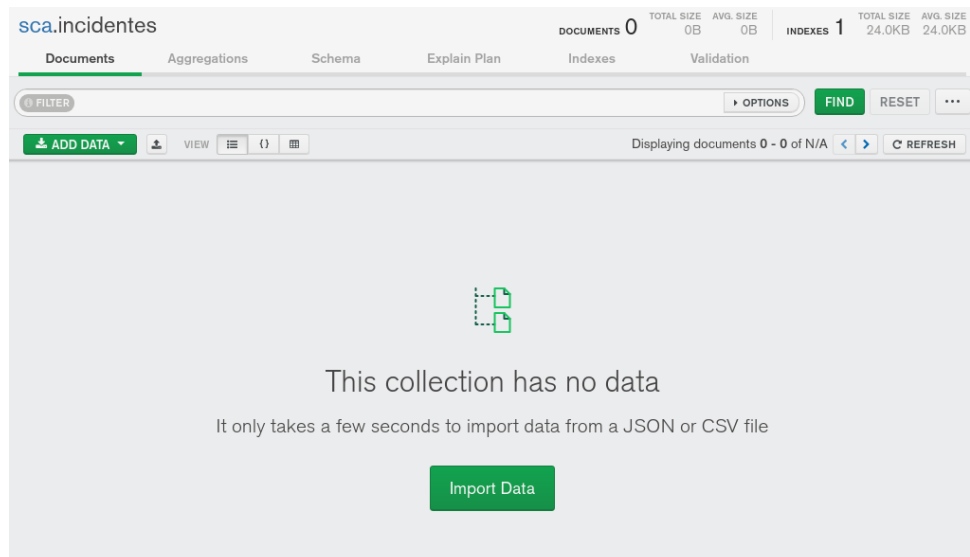


Figura 20 – Consulta na base para verificar que não há incidentes registrados.

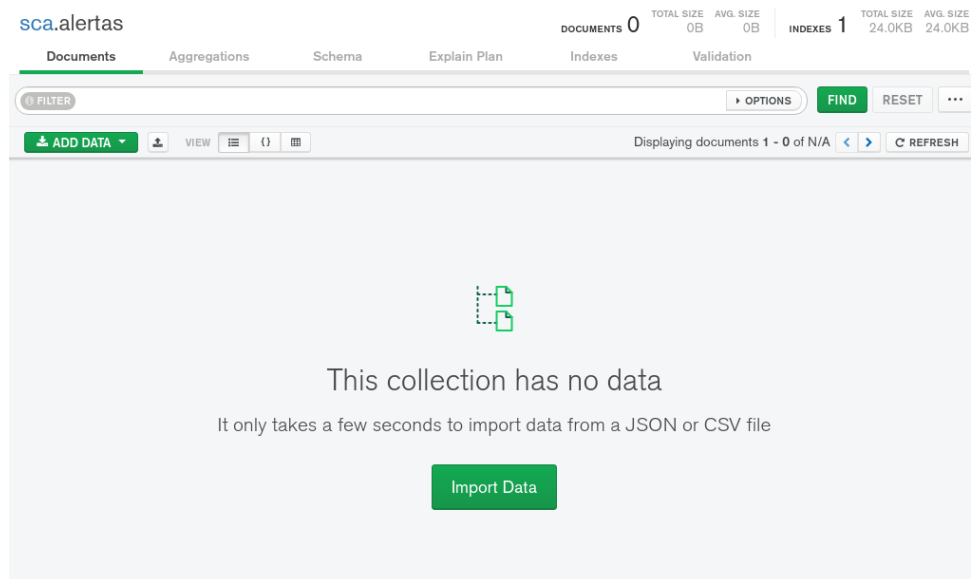


Figura 21 – Consulta na base para verificar que não há alertas registrados.

The screenshot shows a web form titled 'Cadastro de Incidentes'. At the top, there is a green success message: 'Incidente cadastrado com sucesso!!'. Below the message, there are four input fields: 'Data' with the value '2020-05-18', 'Tipo de Incidente' with the value 'Deslocamento', 'Area' with the value 'area53', and 'Grau de Risco' with the value 'Alto'. At the bottom of the form is a green button labeled 'SUBMIT'.

Figura 22 – Cadastro de um incidente.

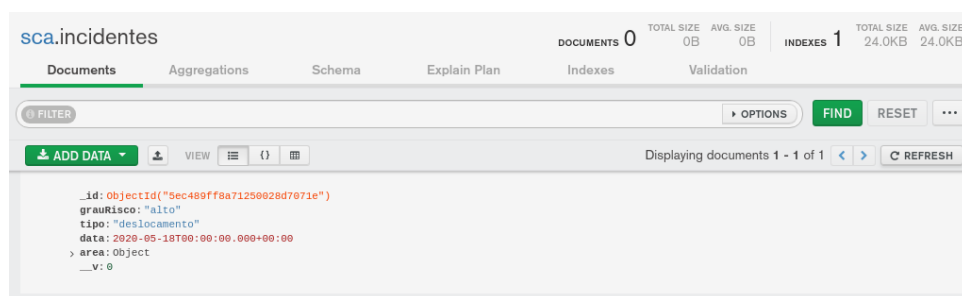


Figura 23 – Consulta na base para verificar se foi salvo o incidente.

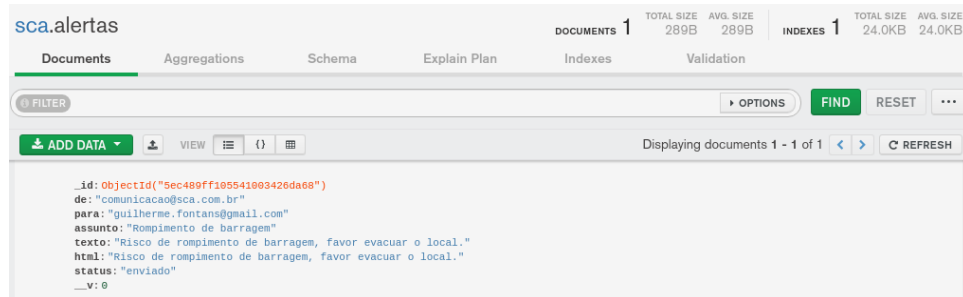


Figura 24 – Consulta na base para verificar se foi salvo o alerta.

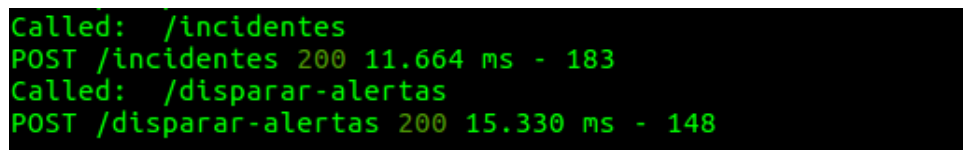


Figura 25 – Log do modulo de api gateway para evidenciar que o recurso de disparar alertas foi acionado após a criação do incidente.

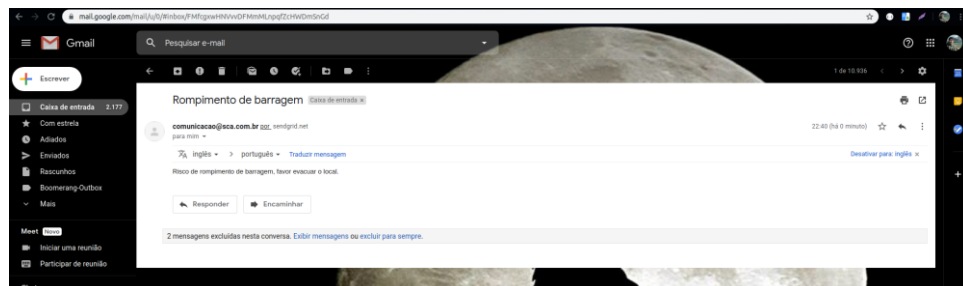


Figura 26 – Email recebido através da integração da aplicação com a ferramenta externa Sendgrid.

- **Cenário 4:**

Atributo de Qualidade	Interoperabilidade
Requisito de Qualidade	Comunicação entre os módulos sendo feita através das APIs.
Preocupação	
O sistema deve prover uma boa interface para o usuário, não permitindo que alguma funcionalidade seja perdida devida ao acesso estar sendo feito através de um dispositivo móvel de menor resolução que um <u>desktop</u> convencional.	

Cenário	
Cenário 4.	
Ambiente	
Produção.	
Estímulo	
Usuário acessando o sistema a partir de um dispositivo móvel.	
Mecanismo	
Criar um layout responsivo que seja capaz de adequar os componentes diminuindo os mesmos ou redirecionando-os na tela para garantir uma melhor experiência ao usuário.	
Medida de resposta	
O sistema deve adaptar os seus componentes de acordo com a resolução que o mesmo está sendo acessado, sem perder as suas funcionalidades.	
Considerações sobre a arquitetura	
Riscos	Experiência de navegação prejudicada devido à baixa resolução de dispositivos móveis mais antigos.
Pontos de Sensibilidade	Não existe
Tradeoff	Não existe

- **Evidências do cenário 4:**

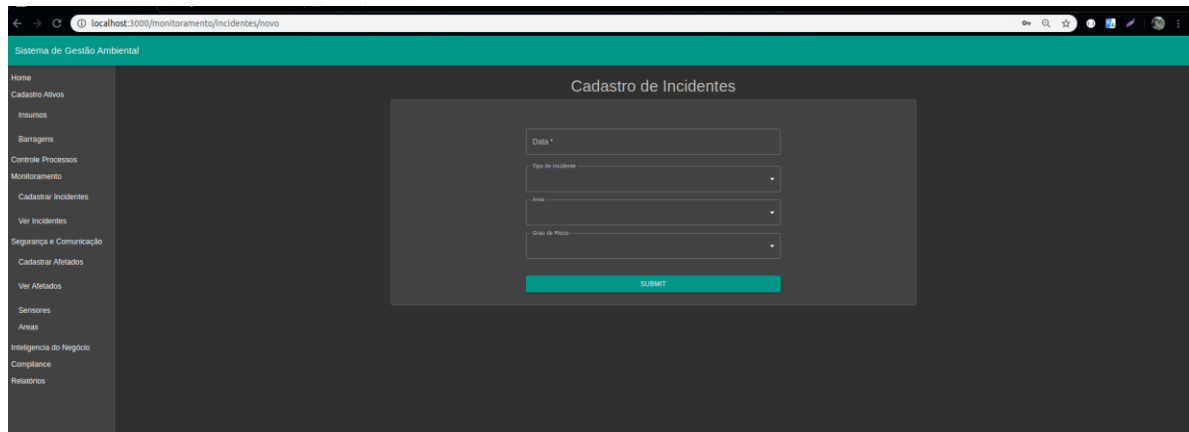


Figura 27– Tela de cadastro de incidentes sendo apresentada em uma resolução 1920x1080.

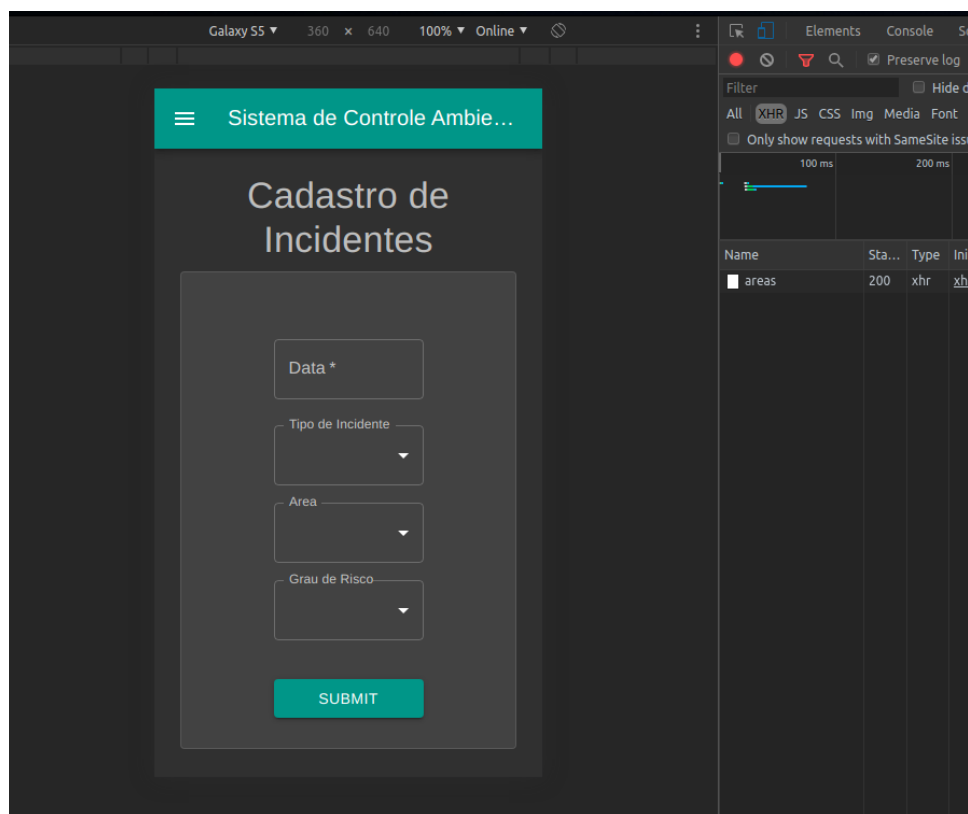


Figura 28 – Tela de cadastro de incidentes sendo apresentada em uma resolução que simula o dispositivo móvel GalaxyS5.

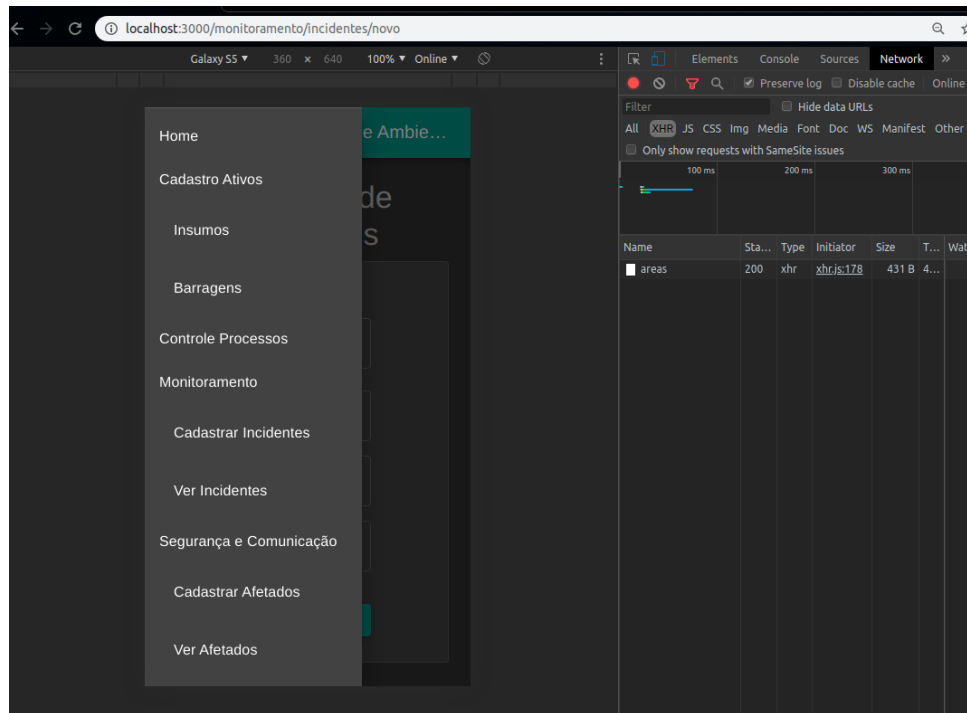


Figura 29 – Menu lateral do sistema sendo apresentado em uma resolução que simula o dispositivo móvel GalaxyS5.

6.4. Resultado

Considerando os atributos de qualidade e tendo sido realizado a validação arquitetura através da criação da POC, nota-se que as necessidades propostas foram atendidas e há espaço para possíveis melhorias.

A avaliação permitiu que fosse possível concretizar de forma mais objetiva os testes e cenários para definir pontos fortes e pontos fracos nessa avaliação. Nessa avaliação foi considerado os seguintes requisitos de qualidades no quadro abaixo.

Requisitos não funcionais	Testado	Homologado
Segurança: O sistema não deve permitir o acesso aos módulos sem o usuário estar logado.	SIM	SIM
Usabilidade: O sistema deve possuir interface responsiva, adequando o layout de acordo com o dispositivo que está sendo utilizado para o acesso	SIM	SIM

Operabilidade: O sistema deve comunicar com algum serviço externo para envio de alertas.	SIM	SIM
--	-----	-----

Avaliando a arquitetura proposta nesse projeto para a interação do usuário com a aplicação, foi possível identificar alguns pontos fortes, nos quais um deles é a separação total entre o desenvolvimento do frontend e do backend, o que facilita no futuro uma melhoria no layout da aplicação, podendo ser aplicado um design modelado através de equipes de UX e desenvolvido por programadores especializados em frontend quando em paralelo a equipe de backend trabalha em outras melhorias. Outro ponto forte também foi a utilização da biblioteca Material-UI, no qual já possui diversos componentes prontos e que trabalham de maneira responsiva. Também foi identificado pontos a serem melhorados, como uma melhoria na interação com o usuário, nos quais poderiam ser aplicadas no futuro após a prototipação do layout através de consultorias com equipes de UX.

Avaliando também a camada de backend, também se nota pontos fortes como a facilidade de aplicar manutenção caso seja necessário, isso devido a divisão de cada módulo do sistema em um projeto separado, expondo apenas a sua interface para que módulo de api gateway consiga se comunicar com os seus respectivos endpoints. Também apresenta uma boa integração entre a linguagem utilizada para a aplicação e a base de dados utilizada para armazenamento de configurações, o que facilita a manutenção tendo em vista que NodeJS já possui componentes para a integração com MongoDB. Avaliando também a alta criticidade e performance necessárias para tratar os dados que são enviados pelos sensores presentes nas barragens, temos uma boa integração entre o InfluxDB que é uma base de dados focada em monitoramento de desempenho e que seu maior uso envolve armazenar dados quase em tempo real, muito contrário de SGBDs relacionais geralmente utilizados para armazenamento de configurações, e em conjunto com esse SGBD é utilizado os clients desenvolvidos em Python, que também é uma linguagem utilizada para tratamento de dados em alta escala. Como ponto de melhoria foi analisado que mesmo tendo uma alta integração entre NodeJS e MongoDB, poderia ter sido utilizado um SGBD relacional, como MySQL ou SQLServer que atendem bem a requisitos de armazenamento de configurações e possuem uma sintaxe de busca de seus dados mais simples para este requisito.

Analisando a integração feita entre os componentes e serviços externos vimos que a comunicação entre os artefatos são bem sucedidas, porém pode haver melhoria na comunicação

entre do módulo de monitoramento com o módulo de segurança e comunicação, podendo ser feita através de mensageria no lugar de uma chamada http de dentro do módulo de monitoramento.

Do ponto de vista da implantação, a solução em Docker atende as necessidades para a montagem do ambiente, diminuindo a dependência de determinado sistema operacional para que o projeto opere, além de também poder escalar sob demanda os containers com os micro-serviços, desde que seja utilizado alguma tecnologia para fazer a orquestração dos containers, como o Kubernetes e Docker Swarm, tendo assim a mesma eficiência de uma solução em nuvem, no qual também poderia ser migrado com facilidade devido ao Docker.

7. Conclusão

Este trabalho apresentou um protótipo arquitetural para uma plataforma de gestão para empresas do ramo de mineração. Entende-se que os objetivos propostos foram atingidos e que há margem para melhorias.

REFERÊNCIAS

Como um projeto da arquitetura de uma aplicação não requer revisão bibliográfica a inclusão das referências não é obrigatória, embora seja recomendada. Caso você deseje incluir referências relacionadas às tecnologias, padrões, ou metodologias que utilizadas no seu trabalho relacione-as de acordo com o modelo a seguir. Siga as normas ABNT, disponíveis em www.pucminas.br, na página da biblioteca.

Exemplo:

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.

APÊNDICES

URL do repositório no GitHub: <https://github.com/guilhermefontans/tcc-poc-sga>

URL do vídeo de apresentação da POC:

CHECKLIST PARA VALIDAÇÃO DOS ITENS E ARTEFATOS DO TRABALHO

Nº	Item a ser cumprido	Sim	Não	Não se aplica
Completeza do documento				
1	Todos os elementos iniciais do documento (capa, contracapa, resumo, sumário...) foram definidos?			
2	Os objetivos do trabalho (objetivos gerais e pelo menos três específicos) foram especificados?			
3	Os requisitos funcionais foram listados e priorizados?			
4	Os requisitos não funcionais foram listados identificados usando o estilo estímulo-resposta?			
5	As restrições arquiteturais foram definidas?			
6	Os mecanismos arquiteturais foram identificados?			
7	O diagrama de caso de uso foi apresentado junto com uma breve descrição de cada caso de uso?			
8	O modelo de componente e uma breve descrição de cada componente foi apresentada?			
9	O modelo de implantação e uma breve descrição de cada elemento de hardware foi apresentada?			
10	Prova de conceito: uma descrição da implementação foi feita?			
11	Prova de conceito: as tecnologias usadas foram listadas?			
12	Prova de conceito: os casos de uso e os requisitos não funcionais usados para validar a arquitetura foram listados?			
13	Prova de conceito: os detalhes da implementação dos casos de uso (telas, características, etc) foram apresentadas?			
14	Prova de conceito: foi feita a implantação da aplicação e indicado como foi feita e onde está disponível?			
15	As interfaces e/ou APIs foram descritas de acordo com um modelo padrão?			
16	Avaliação da arquitetura: foi feita uma breve descrição das características das abordagens da proposta arquitetural?			
17	Avaliação da arquitetura: Os atributos de qualidade e os cenários onde eles seriam validados foram apresentados?			
18	Avaliação da arquitetura: a avaliação com as evidências dos testes foi apresentada?			
19	Os resultados e a conclusão foi apresentada?			
20	As referências bibliográficas foram listadas?			
21	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				
1	Todos os requisitos funcionais foram mapeados para casos de uso?			
2	Todos os casos de uso estão contemplados na lista de requisitos funcionais?			
3	Os requisitos não funcionais, mecanismos arquiteturais e restrições c arquiteturais estão coerentes com os modelos de componentes e implantação?			
4	Os modelos de componentes e implantação estão coerentes com os requisitos não funcionais, mecanismos arquiteturais e restrições arquiteturais?			
5	As tecnologias listadas na implementação estão coerentes com os requisitos não funcionais, mecanismos arquiteturais e restrições arquiteturais?			
6	Os casos de uso e os requisitos não funcionais listados na implementação estão coerentes com o que foi listado nas seções anteriores?			
7	Os atributos de qualidade usados na avaliação estão coerentes com os requisitos não funcionais na cessão três?			
8	Os cenários definidos estão no contexto dos casos de uso implementados?			
9	O apresentado no item resultado está coerente com o que foi mostrado no item avaliação?			