

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

Welbert Pires Gomides

SISTEMA DE GESTÃO AMBIENTAL

Belo Horizonte
2019

Welbert Pires Gomides

SISTEMA DE GESTÃO AMBIENTAL

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

Belo Horizonte

2019

RESUMO

Este projeto aborda uma solução de gestão das atividades de negócio e controle do meio ambiente por meio da internet, utilizando dispositivos móveis ou *desktop*. Atualmente, algumas regiões do Brasil têm sido vítimas de eventos envolvendo a segurança de barragens e as consequências do rompimento de algumas delas.

O Sistema de Gestão Ambiental, também conhecido pela sigla SGA, é o conjunto de operações e/ou dispositivos destinados ao controle dos impactos negativos das intervenções físicas, efluentes líquidos, emissões atmosféricas e resíduos sólidos gerados pela atividade instalada, de modo a corrigir ou reduzir os seus impactos sobre a qualidade ambiental.

Apesar da mineração ser um ramo que causa muitos danos ao meio ambiente, ela é vital para o desenvolvimento e infraestrutura de um país, principalmente no cenário econômico. Isso explica a urgência em se aplicar um SGA na atividade de mineração para maior otimização dos recursos. Para solucionar este problema, o projeto propõe criar uma solução que traz forma e disciplina para alcançar melhorias contínuas no desempenho ambiental de uma organização sob vários indicadores.

O Projeto arquitetural aborda uma solução de microserviços que auxilia no gerenciamento de uma empresa minerária, automatizando os seus processos de manutenção de ativos e monitoramento de barragens.

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

SUMÁRIO

1. Objetivos do trabalho	5
2. Descrição geral da solução	5
2.1. Apresentação do problema	5
2.2. Descrição geral do software (Escopo)	6
3. Definição conceitual da solução	7
3.1. Requisitos Funcionais	7
3.2. Requisitos Não-Funcionais	10
3.3. Restrições Arquiteturais	17
3.4. Mecanismos Arquiteturais	17
4. Modelagem e projeto arquitetural	19
4.1. Modelo de componentes	19
4.2. Modelo de implantação	21
4.3. Modelo de dados	22
5. Prova de Conceito (POC) / protótipo arquitetural	23
5.1. Implementação e Implantação	23
5.2. Interfaces/ APIs	26
6. Avaliação da Arquitetura	29
6.1. Análise das abordagens arquiteturais	29
6.2. Cenários	29
6.3. Avaliação	31
6.4. Resultado	42
7. Conclusão	44
REFERÊNCIAS	45
APÊNDICES	46

1. Objetivos do trabalho

O objetivo geral deste projeto é apresentar uma proposta de arquitetura de um sistema que tem o intuito de otimizar a gestão dos processos envolvidos com a atividade minerária, permitindo a exploração das minas de maneira coordenada, eficiente e ambientalmente responsável.

Os objetivos específicos são:

1. Criar módulo de ativos, para cadastrar e gerir os ativos envolvidos com a atividade fim da empresa, tais como máquinas, equipamentos e outros insumos necessários.
2. Criar módulo de controle de processos minerários, para realizar a gestão dos processos envolvidos com a atividade minerária, permitindo a exploração das minas de maneira coordenada, eficiente e ambientalmente responsável.
3. Criar módulo de segurança e comunicação, permitindo o acionamento de alertas e procedimentos de evacuação quando necessário.
4. Criar módulo web/mobile de monitoramento de barragens, que integra sensores existentes nas barragens, envolvendo também o monitoramento de risco, aferido periodicamente por consultores contratados e que pode acionar módulo de segurança e comunicação.
5. Garantir a aderência da empresa às normas nacionais e internacionais do setor minerário assim como aprimorar os controles de governança.

2. Descrição geral da solução

2.1. Apresentação do problema

País de dimensões continentais e enormes recursos minerais, o Brasil tem explorado seu potencial de mineração por séculos, e apenas em 2018 o minério de ferro representou quase 17% das exportações brasileiras, arrecadando mais de US\$ 20 bilhões, mas dois desastres em três anos tingiram essa riqueza de sangue e lama.

A Vale, maior produtora mundial de minério de ferro, está no centro de uma tragédia que deixou pelo menos 99 mortos e 259 desaparecidos, após o rompimento de uma barragem em janeiro de 2019 na cidade de Brumadinho. Em novembro de 2015, a empresa já se viu envolvida em um desastre semelhante, que deixou 19 mortos e causou enormes danos ambientais perto de Mariana, a 120 km de distância. Essas duas catástrofes ocorreram no estado de Minas Gerais, onde a maior parte da atividade de mineração está concentrada.

Segundo a Agência Nacional de Águas (ANA), 204 das 790 barragens brasileiras apresentam altos riscos, seja para vidas humanas ou para o meio ambiente. Os impactos ambientais da mineração são diversos e apresentam-se em diversas escalas: desde problemas locais específicos até alterações biológicas, geomorfológicas, hídricas e atmosféricas de grandes proporções. Portanto, conhecer esses problemas causados e a minimização de seus efeitos é de grande necessidade para garantir a preservação dos ambientes naturais.

A contaminação por compostos químicos, com destaque para o mercúrio, também é um dos principais danos ambientais provocados pela mineração. Esses compostos são utilizados para a separação de misturas, retirada dos minerais e catalisação de reações. Após o processo, costumam ser descartados, o que ocorre muitas vezes de maneira indevida, principalmente em localidades de limitada fiscalização, ou até em minas ilegais, que, além de tudo isso, costumam empregar trabalho análogo ao escravo ou infantil.

2.2. Descrição geral do software (Escopo)

A solução se aplica em função de um melhor gerenciamento dos processos envolvidos com a atividade minerária, possibilitando com que manutenções preventivas e corretivas dos equipamentos sejam agendadas e realizadas segundo um cronograma estabelecido pelas áreas de engenharia e de manutenção da empresa.

O setor de engenharia da empresa também tem acesso às informações dos sensores existentes nas barragens, envolvendo o monitoramento de risco onde é

possível acionar alertas e procedimentos de evacuação quando necessário através de uma integração com os sistemas de Defesa Civil municipais e estaduais envolvidos, visando assim garantir a segurança das comunidades que vivem próximo às barragens.

O sistema conta com integração entre os sistemas externos de Normas Ambientais, Assessorias e Consultorias, visando garantir a aderência da empresa às normas nacionais e internacionais do setor minerário assim como aprimorar os controles de governança.

O sistema gera informações sobre insumos, produção, eventos anormais, onde será possível gerar relatórios e consultas sob demanda através de uma solução de mercado, permitindo o acompanhamento da real situação das atividades da empresa e auxiliar nas tomadas de decisão quando necessário.

3. Definição conceitual da solução

3.1. Requisitos Funcionais

Módulo Ativos

- Cadastro de ativos

O sistema deve permitir que o usuário cadastre máquinas, equipamentos e insumos relacionados com a atividade minerária, salvando todos os dados necessários para o gerenciamento feito pelas áreas de engenharia e de manutenção da empresa.

- Cadastro de cronogramas de manutenção

O sistema deve permitir a definição de cronogramas a partir de parâmetros de manutenção, tais como: frequência, melhor horário para realização, intervalo de horas de uso. Com isso, manutenções preventivas serão agendadas

automaticamente para os equipamentos que se enquadrarem nesses parâmetros.

- **Agendamento de manutenção**

O sistema deve permitir o cadastro manual de uma manutenção preventiva ou corretiva para um equipamento cadastrado e deve fornecer uma seção para informar a execução da mesma.

- **Aquisição de ativos**

Deverá ser contemplada uma integração com o sistema de aquisições, que irá permitir a solicitação da aquisição de um novo ativo e o cadastro do mesmo no sistema de forma automática.

Módulo Controle de Processos Minerários

- **Cadastro de fluxo de trabalho**

O sistema deve permitir o cadastro de fluxogramas que representam o fluxo de trabalho dos setores diretamente ligados à exploração das minas, com as informações de suas atividades diárias, visando produzir e entregar o minério explorado.

- **Registro de paradas**

O sistema deve permitir o registro das paradas na produção diária por turno de trabalho, associando a mesma a um processo e se necessário a etapa em que foi feita.

- **Registro de problemas**

O sistema deve permitir o registro dos problemas na produção diária por turno de trabalho, associando o mesmo a um processo e se necessário a etapa em que ocorreu.

Módulo Monitoramento de Barragens

- Monitorar barragem

O sistema deve permitir que o usuário consultar em tempo real os sensores existentes nas barragens e fornecer os dados necessários para diagnosticar a presença de água nas estruturas e, desta forma, identificar precocemente a ocorrência de pontos de fraqueza na estrutura do barramento.

- Registrar alerta

O sistema deve acompanhar os resultados emitidos pelos sensores e registrar um alerta para o módulo de segurança e comunicação quando os mesmos ultrapassarem os limites pré configurados, e sobre a mesma condição será feita uma solicitação de reconhecimento de desastre para o sistema da Defesa Civil integrado.

Módulo Segurança e Comunicação

- Emitir alerta

O sistema deve permitir o envio de alertas para as comunidades que vivem próximas às barragens, e se necessário, acionar os processos de evacuação das mesmas.

Módulo Inteligência do Negócio

- Importar dados

O sistema deve periodicamente importar os dados de todos os módulos para uma base de dados separada, visando criar o insumo necessário aos serviços de *Business Intelligence* (BI).

- Visualizar insumos mais consumidos

Deve permitir a visualização dos insumos que foram mais consumidos nos últimos meses e separar quais apresentam um melhor resultado na produção e quais apresentaram mais problemas.

- Visualizar resultados dos fluxos de trabalho

Deve permitir a visualização dos resultados de cada fluxo de trabalho que foi aplicado, apontando gargalos e quais são as atividades que mais tem paradas ou problemas registrados.

- Visualizar equipamentos mais defeituosos

Deve permitir a visualização dos equipamentos que mais sofreram manutenções corretivas nos últimos meses e sugerir parâmetros para as manutenções preventivas dos mesmos.

Módulo Compliance

- Verificar aderência nas normas

O sistema deve realizar uma integração com os sistemas externos de Normas Ambientais e Assessorias e Consultorias, e apresentar quais são as aderências da empresa em relação às normas nacionais e internacionais do setor minerário, alertando também sobre quais são as pendências.

Módulo Relatórios

- Gerar relatório sob demanda

O sistema deve oferecer suporte para soluções de mercado de geração de relatórios, onde os usuários poderão gerar as saídas que desejarem, de acordo com seu perfil.

3.2 Requisitos Não-Funcionais

A seguir são apresentados os requisitos não funcionais do sistema:

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

Estímulo	Cadastro de ativo
Fonte do Estímulo	Usuário acessando o sistema de um smartphone
Ambiente	Funcionamento, carga normal
Artefato	Módulo Ativos
Resposta	A camada de apresentação se adaptou as resoluções e tamanhos das telas, mudando os componentes de posição de forma a melhorar a navegação do usuário
Medida da resposta	Identidade visual semelhante em todas as resoluções, com objetos redimensionados de acordo com a resolução e tamanho

- **Usabilidade - O sistema deve ser de fácil utilização.**

Estímulo	Usuário cadastrando ativo
Fonte do Estímulo	Usuário acessando uma funcionalidade de cadastro de ativos
Ambiente	Funcionamento, carga normal
Artefato	Módulo Ativos
Resposta	A camada de apresentação apresenta facilidades de interação e

	navegação
Medida da resposta	Usuário consegue cadastrar um ativo em até 5 minutos

- **Desempenho - O sistema deve ser rápido.**

Estímulo	Acessando a tela para acompanhar um fluxo de trabalho
Fonte do Estímulo	Usuário acompanhando um fluxo de trabalho
Ambiente	Funcionamento, carga normal
Artefato	Módulo Controle de Processos Minerários
Resposta	O sistema respondeu com os dados solicitados
Medida da resposta	O sistema respondeu em até 5 segundos

- **Manutenibilidade - O sistema deve ser de fácil manutenção.**

Estímulo	Um componente de negócio responsável por enviar os sinais de alerta precisa ser atualizado
Fonte do Estímulo	O processo de negócio para envio de alertas foi alterado

Ambiente	O módulo de Monitoramento de Barragens controla um componente de negócio, esse componente é responsável pelo envio de alertas ao Módulo de Segurança e Comunicação, uma falha nesse componente pode fazer com que um alerta seja emitido de forma desnecessária ou que um alerta importante deixe de ser emitido
Artefato	Módulo Monitoramento de Barragens, Módulo Segurança e Comunicação
Resposta	O módulo que cuida do processo de alertas deve ser atualizado para que os mesmos sejam emitidos corretamente
Medida da resposta	Os alertas devem ser emitidos corretamente

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

Estímulo	Execução de testes no sistema
Fonte do Estímulo	Analista desenvolvedor
Ambiente	Ambiente de Desenvolvimento
Artefato	Módulo Ativos

Resposta	O sistema testou todas as funcionalidades de inserção, visualização, edição e remoção de um ativo, incluindo a solicitação de compra.
Medida da resposta	O sistema deve possibilitar efetuar os testes com scripts automatizados executando com apenas um comando

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

Estímulo	Erro ao obter informações dos sensores nas barragens
Fonte do Estímulo	Problema nos sensores
Ambiente	Funcionamento, carga normal
Artefato	Módulo Monitoramento de Barragens
Resposta	O sistema deve prover uma opção de contingência para a requisição, retornando os últimos dados obtidos dos sensores
Medida da resposta	Os últimos dados registrados foram retornados

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

Estímulo	Teste de conexão
Fonte do Estímulo	Sistemas de Defesa Civil
Ambiente	Funcionamento, carga normal
Artefato	Módulo Monitoramento de Barragens e sistemas de Defesa Civil
Resposta	O WebService do sistema de Defesa Civil respondeu com sucesso a solicitação
Medida da resposta	Um arquivo JSON foi retornado

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

Estímulo	Acessar uma página privada pela url sem estar logado no sistema
Fonte do Estímulo	Qualquer usuário
Ambiente	Funcionamento, carga normal
Artefato	Módulo Ativos, Controle de Processos Minerários, Monitoramento de Barragens, Segurança e Comunicação, Relatórios

Resposta	O sistema deve redirecionar o usuário para a tela solicitando usuário e senha
Medida da resposta	O sistema não deve permitir o acesso a páginas privadas

- **Disponibilidade - O sistema deve operar 24 horas por dia nos sete dias da semana.**

Estímulo	Shutdown no cluster primário do servidor de aplicação
Fonte do Estímulo	Administrador do Servidor de Aplicação
Ambiente	Diversos usuários estão utilizando o sistema
Artefato	Gerenciador do cluster
Resposta	Todos os usuários logados na aplicação devem continuar utilizando o sistema sem perceber que houve uma queda de um dos nós do servidor de aplicação
Medida da resposta	Todas as solicitações dos usuários devem ser atendidas

- **Padrão - O sistema deve ser desenvolvido utilizando recursos de integração contínua.**

Estímulo	Nova alteração de código enviada
Fonte do Estímulo	Analista desenvolvedor
Ambiente	Ambiente de desenvolvimento
Artefato	Código Fonte, Servidor de Automação
Resposta	O servidor de automação deve detectar as alterações no código e iniciar um processo para compilar e testar as alterações enviadas
Medida da resposta	Um binário testado deve ser entregue ao final do processo

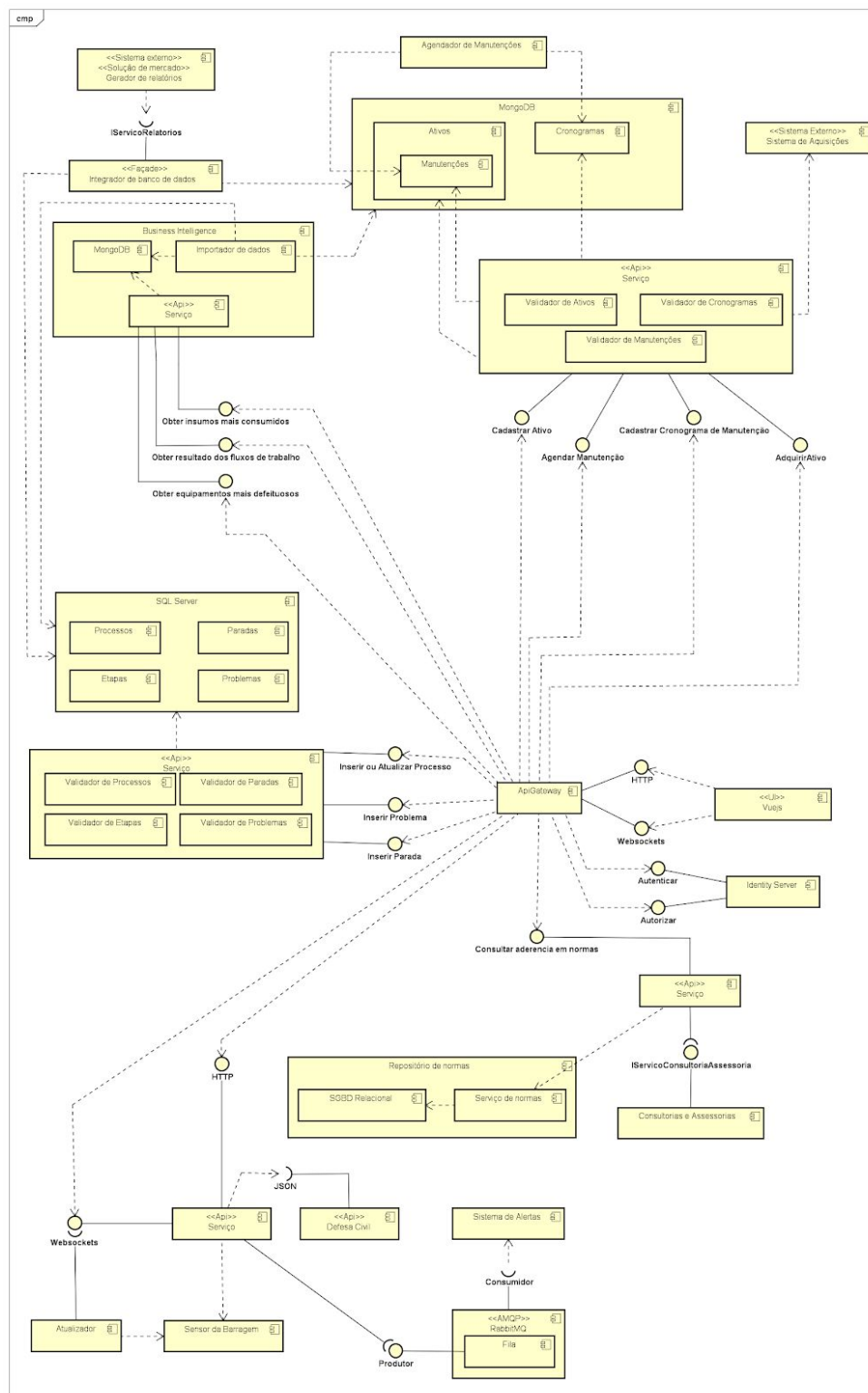
3.3. Restrições Arquiteturais

- Apresentar características de aplicações distribuídas, tais como abertura, portabilidade e uso extensivo de recursos de rede.
- Ser hospedado em nuvem e/ou 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

Mecanismo de Análise	Mecanismo de Design	Mecanismo de Implementação
Persistência	Banco de dados não relacional	MongoDB

Persistência	Banco de dados relacional	SQL Server
Comunicação entre processos	Contêiner Web e Aplicação	Docker
Integração com outros módulos	Interfaces utilizando JSON	WebApi
Exposição de serviços	Api Gateway	Ocelot
Log	Framework de log	Serilog
Build	Geração de artefato para servidor de aplicação	MSBuild e Nuget
Deploy	Deploy da aplicação no servidor e testes automatizados	Jenkins
Interface de usuário	Interface de comunicação com o usuário do sistema	Vue.js e Vuex
Versionamento	Versionamento do código fonte da aplicação	Git
Autenticação e Autorização	Verificação das credenciais e tentativas de conexão	Identity Server
Sistema Operacional	Sistema executado nos servidores	Ubuntu Server
Testes	Framework de testes de unidade	xUnit



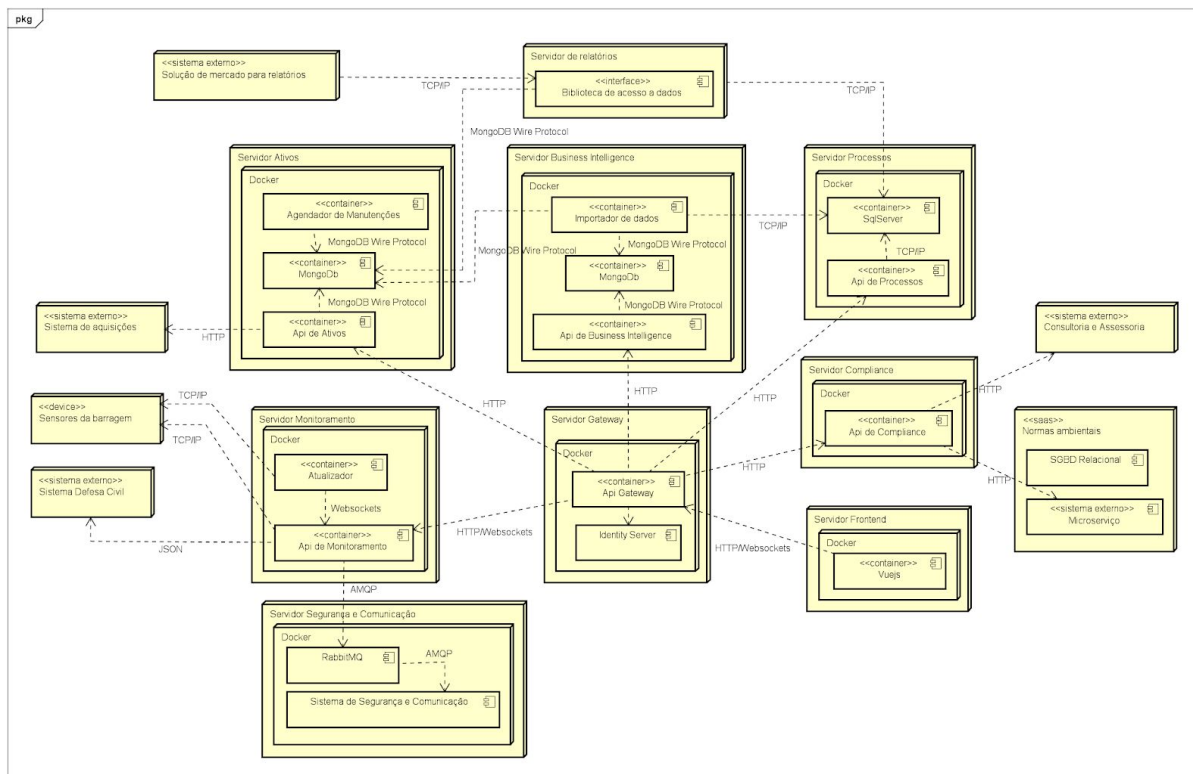
Nessa arquitetura devemos considerar a divisão do sistema em um aplicativo web desenvolvido em Vuejs, e uma série de microsserviços que estão disponibilizados através de um API Gateway, que irá fornecer todas as interfaces necessárias para manipulação dos dados.

A segurança será realizada em um micro serviço que fica integrado ao API Gateway e permite a autenticação/autorização, com ele é possível gerenciar todos os acessos e perfis de forma isolada, onde nenhum impacto é gerado nos serviços que tratam do negócio da empresa.

O sistema de relatórios será adquirido como uma solução de mercado e integrado à estrutura através de uma biblioteca. Os bancos de dados do Módulo Controle de Processos Minerários e do Módulo Ativos serão compartilhados entre os serviços de Business Intelligence e a biblioteca de relatórios, que fornecerão as interfaces necessárias para consumir os dados visando a geração de inteligência ao negócio.

Serão feitas integrações do Módulo Compliance com o sistema de Consultoria e Assessoria, e com o sistema de Normas de Segurança. O Módulo Monitoramento de Barragens será integrado com o sistema da Defesa Civil através de troca de arquivos e com o Módulo Segurança e Comunicação através de um sistema de filas.

4.2. Modelo de implantação



Toda infraestrutura será distribuída utilizando-se de containers Docker, possibilitando uma fácil implantação do sistema, sendo essa dividida seguindo a divisão de módulos do sistema, de forma a permitir a implantação de acordo com a prioridade e necessidade da empresa.

O sistema de Normas ambientais é disponibilizado como um micro serviço SaaS que comunica com um SGBD relacional, juntamente com o sistema de Consultorias e Assessorias ambos são disponibilizados como serviços na nuvem e se comunicam através do protocolo HTTP. O sistema da Defesa Civil é disponibilizado na internet e se comunica através de arquivos JSON.

Os sensores das barragens fornecem uma interface TCP/IP para obtenção dos resultados, onde o Módulo de Monitoramento oferece suporte a comunicação HTTP para consultas eventuais(requisições) e Websockets para acompanhamento em tempo real.

A biblioteca de relatórios fornece uma interface TCP/IP para comunicação com a solução de mercado adquirida, onde irá trafegar os dados necessários para a geração dos relatórios.

4.3. Modelo de dados

Modelo de dados: Ativos

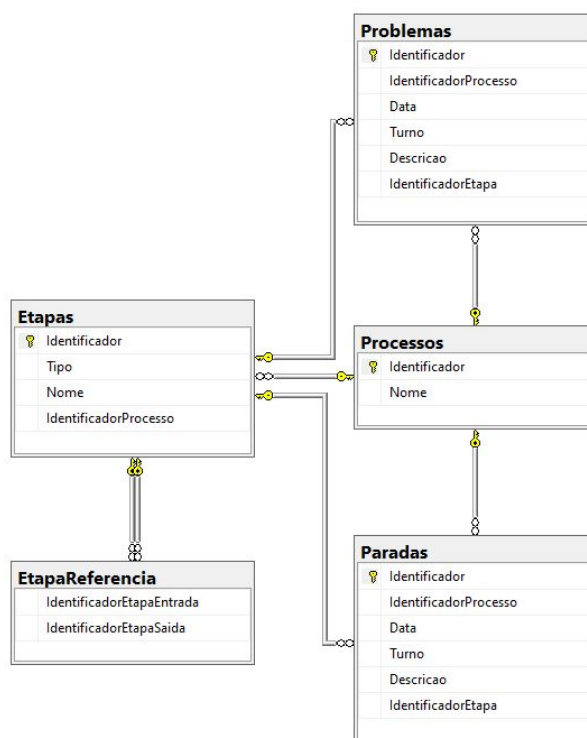
```
_id: ObjectId("5d6c6a5d6c6caf00012be7fb")
Nome: "Caminhão de mineração"
Patrimonio: "VL123456"
Tipo: 2
DataAquisicao: 2019-09-02T01:03:25.988+00:00
Observacoes: "Equipamento com número de série 543JJHGF8546HG, suporta até 363 tonela..."
MediaHorasUsoDiariamente: 4
~ Manutencoes: Array
  ~ 0: Object
    _id: ObjectId("5d6c6a806c6caf00012be7fd")
    DataHora: 2019-09-02T03:00:00.000+00:00
    DataHoraRealizada: null
    Tipo: 1
    Realizada: false
```

Modelo de dados: Cronogramas de Manutenção

```
_id: ObjectId("5d6c6a6b6c6caf00012be7fc")
Frequencia: 2
TipoAtivo: 2
IntervaloHorasUso: null
```

Para o Módulo de ativos será utilizado o MongoDB, um banco de dados com estrutura de documentos projetado para armazenar e consultar dados como documentos do tipo JSON, que irá permitir a manipulação de uma grande quantidade de ativos e suas manutenções, incluindo suporte total com índices, estilhamento e replicação, o que será necessário para a boa performance, robustez e escalabilidade do sistema.

Modelo de dados: Processos



O módulo de processos conta com um banco de dados relacional para garantir a integridade dos dados e suas relações, a baixa redundância e consistência de dados, que são elementos essenciais para um módulo trata diretamente dos processos da empresa.

5. Prova de Conceito (POC) / protótipo arquitetural

5.1. Implementação e Implantação

A prova de conceito desse projeto visa atender as necessidades dos requisitos dos módulos que controlam os ativos, processos e monitoramento de barragens presentes em uma mineradora. O objetivo deste protótipo é verificar se a arquitetura definida atenderá todas as necessidades do usuário em relação aos

requisitos de qualidade, minimizando os riscos, garantindo a eficiência no processo e atendendo a expectativa de valor a ser gerado com o sistema.

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

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

Esse RNF foi escolhido para garantir o atendimento de todas as exigências de um sistema responsivo que se adapte em celulares, tablets e desktops.

Os critérios de aceite são:

- A tela do sistema deve apresentar facilidade de navegação e os objetos da tela devem se adaptar de acordo com a resolução identificada, tanto para celulares como desktops.
- O sistema deve se manter com o mesmo padrão de cores e objetos.
- O sistema deve ser compatível com os principais browser do mercado como: Microsoft Edge, Chrome e Firefox.

- **Usabilidade - O sistema deve ser de fácil utilização**

Esse RNF foi escolhido devido à importância em se criar um sistema fácil de se utilizar, com uma navegação simples e objetiva.

Os critérios de aceite são:

- A tela do sistema deve apresentar facilidade de navegação.
- O usuário não pode demorar mais que 5 minutos para efetuar um agendamento.
- O acesso as funcionalidade devem apresentar objetividade e não serem confusos.

- **Manutenibilidade - O sistema deve ser de fácil manutenção**

Esse RNF foi escolhido para garantir a boa qualidade do código escrito e a facilidade em manutenções futuras.

Os critérios de aceite são:

- O sistema deve ter um baixo acoplamento entre seus módulos.
- Uma mudança não deve gerar impactos em outras partes do sistema.
- O código fonte deve estar organizado de acordo com um padrão.

- **Desempenho - O sistema deve ser rápido**

Esse RNF foi escolhido com o objetivo de garantir uma boa performance na aplicação, garantindo que o usuário não fique muito tempo esperando para obter informações do sistema.

Os critérios de aceite são:

- As telas devem ser bem otimizadas para que não demorem mais que 5 segundos a serem renderizadas ao usuário.
- Um comando do usuário deve ser processado em até 5 segundos.

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

Esse RNF foi escolhido com o intuito de que as funcionalidades do sistema possam ser testadas de forma rápida e eficiente, garantido a qualidade do sistema desde sua concepção até a fase de pós produção.

Os critérios de aceite são:

- As configurações do projeto, deve permitir que seja configurado e implementado algum framework de teste unitário, como por exemplo xUnit.
- O sistema deve efetuar testes regressivos para garantir que não houve nenhuma falha em alguma funcionalidade que já estava funcionando.
- Os testes devem ser automatizados e executados de forma simplificada com apenas um comando.

5.2 Interfaces/ APIs

Sessão 1: Métodos do sistema de aquisições

Os métodos e elementos necessários para utilização do Web Service serão descritos e exemplificados logo abaixo.

Método obterAdquiriveis()

Este método obtém a lista de itens que podem ser adquiridos no sistema de aquisições.

Assinatura do método:

object[] obterAdquiriveis()

Exemplo:

- Requisição

```

1  curl -X GET \
2  http://localhost:8080/api/ativos/adquirir \
3  -H 'Accept: */*' \
4  -H 'Accept-Encoding: gzip, deflate' \
5  -H 'Cache-Control: no-cache' \
6  -H 'Connection: keep-alive' \
7  -H 'Host: localhost:8080' \
8  -H 'Postman-Token: eda0f027-79bf-49dd-b976-a6ab04e56303,230486bc-95f3-4fd5-ae00-dd17834bf767' \
9  -H 'User-Agent: PostmanRuntime/7.16.3' \
10 -H 'cache-control: no-cache'
```

- Resposta

```

1  [
2      {
3          "identificador": 1,
4          "nome": "Caminhão de mineração",
5          "descricao": "Suporta até 363 toneladas de minério."
6      }
7  ]

```

Método adquirir()

Este método possibilita a aquisição de um item.

Assinatura do método:

void adquirir(id)

Exemplo:

- Requisição

```

1  curl -X POST \
2      http://localhost:8080/api/ativos/adquirir/1 \
3      -H 'Accept: */*' \
4      -H 'Accept-Encoding: gzip, deflate' \
5      -H 'Cache-Control: no-cache' \
6      -H 'Connection: keep-alive' \
7      -H 'Content-Length: ' \
8      -H 'Host: localhost:8080' \
9      -H 'Postman-Token: 0ef46257-9418-493b-a5bd-0aeb38dd6105,fcd0416f-b7da-436f-9e1d
10     -af64c132642a' \
11     -H 'User-Agent: PostmanRuntime/7.16.3' \
12     -H 'cache-control: no-cache'

```

- Resposta

```

1  {
2      "identificador": "5d704e559648c0001eeeb8d",
3      "nome": "Caminhão de mineração",
4      "patrimonio": "VL123456",
5      "tipo": 2,
6      "dataAquisicao": "2019-09-04T23:52:53.3407845+00:00",
7      "observacoes": "Equipamento com número de série 543JJHGF8546HG, suporta até 363 toneladas de minério.",
8      "mediaHorasUsoDiariamente": 4,
9      "manutencoes": []
10 }

```

Sessão 2: Interface dos sensores da barragem

Os métodos e elementos necessários para utilização do hardware dos sensores serão descritos e exemplificados logo abaixo.

Método obterResultados()

Este método consulta os sensores e retorna as métricas de resultado.

Assinatura do método:

object obterResultados()

Exemplo:

- Requisição

Não se aplica.

- Resposta:

```

1  {
2    "piezometro": {
3      "nivel": 0.96807339320335228,
4      "pressao": 0.0910778749226955
5    },
6    "deslocamento": {
7      "deslocamentoVertical": 0.80156282652242239,
8      "deslocamentoHorizontal": 0.67935923052921854
9    }
10 }
```

Sessão 3: Métodos do sistema da Defesa Civil

Os métodos e elementos necessários para utilização do Web Service serão descritos e exemplificados logo abaixo.

Método solicitarReconhecimentoDesastre()

Este método cria uma solicitação de reconhecimento de um possível desastre no sistema da Defesa Civil.

Assinatura do método:

void solicitarReconhecimentoDesastre(double latitude, double longitude, string observacoes)

Exemplo:

- Requisição:

```

1  curl -X POST \
2    http://localhost/api/monitoramento/reconhecimentoDesastre \
3    -H 'Content-Type: application/json' \
4    -H 'Postman-Token: c93579ef-7c74-4876-818c-2a5840324a9b' \
5    -H 'cache-control: no-cache' \
6    -d '{
7      "latitude": 38.8951,
8      "longitude": -77.0364,
9      "observacoes": "Alteração nos níveis de pressão da barragem de baixo"
10   }'
```

- Resposta

Não se aplica.

6. Avaliação da Arquitetura**6.1. Análise das abordagens arquiteturais**

Apresente um breve resumo das principais características da proposta arquitetural.

6.2. Cenários

Cenário 1: Uma manutenção no sistema deve gerar pouco ou nenhum impacto, ao mesmo tempo em que deve ser fácil corrigir defeitos, adequar-se a novos requisitos, aumentar a suportabilidade ou se adequar a um ambiente novo, de forma com que o

mesmo consiga evoluir sem prejudicar aquilo que já foi desenvolvido, atendendo assim um dos requisitos não funcionais.

Cenário 2: Ao navegar na tela, o sistema deve apresentar uma boa usabilidade, a navegação deve ser de fácil acesso e funcionalidades devem ser bem objetivas, o usuário deve conseguir efetuar o registro de um ativo em no máximo 5 minutos, garantindo assim a agilidade e usabilidade de acordo com o especificado nos requisitos não funcionais.

Cenário 3: Ao realizar o acesso a aplicação através de um dispositivo móvel ou desktop com resolução reduzida, utilizando os principais navegadores do mercado como Microsoft Edge, Chrome e Firefox, a tela do usuário deverá se adaptar automaticamente, redimensionando seus links, botões, textos e seções de acordo com a resolução, estando de acordo com acessibilidade necessária para atender um dos requisitos não funcionais.

Cenário 4: Ao realizar um acesso em alguma tela privada ou pública, o sistema deve ter um desempenho aceitável e responder em no máximo 5 segundos a renderização dos objetos na tela, assim como os comandos do usuário não devem demorar mais do que 5 segundos para serem concluídos, atendendo assim um dos 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 foram classificadas em alta (A), média (M) e baixa (B) de acordo com as características do requisito.

Atributos de Qualidade	Cenários	IMP.	COM.
Manutenibilidade	Cenário 1: O sistema deve ser de fácil	M	A

	manutenção		
Usabilidade	Cenário 2: O sistema deve ser de fácil utilização	M	B
Acessibilidade	Cenário 3: O sistema deve suportar ambientes web e móveis	M	A
Desempenho	Cenário 4: O sistema deve ser rápido	A	M

6.3. Avaliação

Processo de avaliação dos cenários identificados no item 6.1, o objetivo é determinar os riscos, não riscos, pontos de sensibilidade e trade offs e as evidências mostrando o requisito de qualidade a ser atendido.

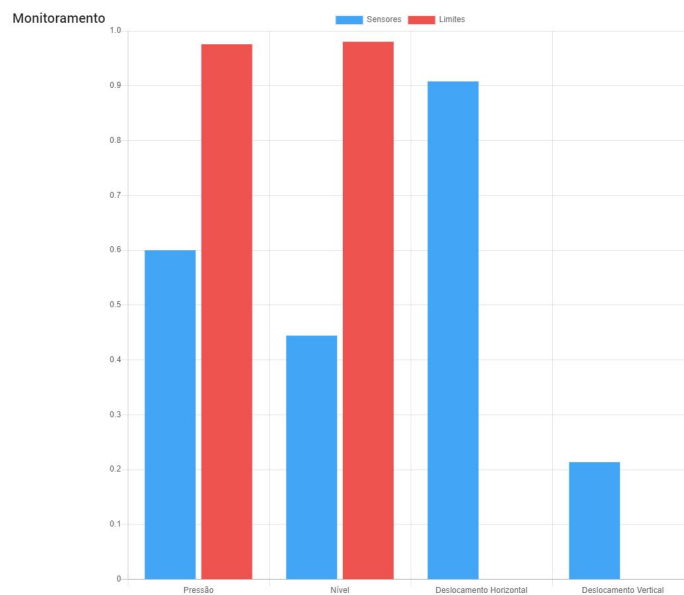
- **Cenário 1**

Atributo de Qualidade:	Manutenibilidade
Requisito de Qualidade:	O sistema deve ser de fácil manutenção
Preocupação:	
Realizar uma alteração no sistema com pouco esforço e nenhum impacto	
Cenário(s):	
Cenário 1	
Ambiente:	
O módulo de Monitoramento de Barragens controla um componente de negócio, esse componente é responsável pelo envio de alertas ao Módulo de Segurança e Comunicação	

Estímulo:	
Um componente de negócio responsável por enviar os sinais de alerta precisa ser atualizado	
Mecanismo:	
Alterar a configuração de limites para que o sistema respeite uma nova regra	
Medida de resposta:	
Sistema emite os alertas corretamente conforme a nova regra	
Considerações sobre a arquitetura:	
Riscos:	Uma falha nesse componente pode fazer com que um alerta seja emitido de forma desnecessária ou que um alerta importante deixe de ser emitido
Pontos de Sensibilidade:	Módulo de Segurança e Comunicação, Módulo de Monitoramento de Barragens
Tradeoff:	Não se aplica

- **Evidências do Cenário 1**

A página de Monitoramento mostra os limites configurados em vermelho.



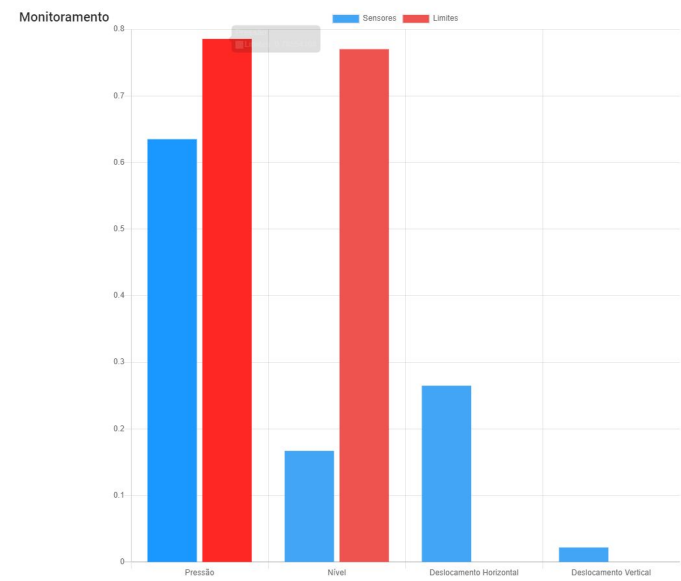
Uma alteração é feita no JSON de configurações dos limites (Monitoramento.Api/appsettings.json).

```
{
  "RabbitMQ": {
    "Host": "rabbitmq",
    "Fila": "alertas"
  },
  "Limites": {
    "NivelMaximoPermitido": "0.98014213",
    "PressaoMaximaPermitida": "0.97554168"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

→

```
{
  "RabbitMQ": {
    "Host": "rabbitmq",
    "Fila": "alertas"
  },
  "Limites": {
    "NivelMaximoPermitido": "0.77014213",
    "PressaoMaximaPermitida": "0.78554168"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

O serviço de monitoramento é reiniciado e a página de Monitoramento atualizada. Os novos limites estão configurados.



- Cenário 2**

Atributo de Qualidade:	Usabilidade
Requisito de Qualidade:	O sistema deve ser de fácil utilização
Preocupação:	
	O sistema deve ser simples, intuitivo e apresentar um desempenho satisfatório
Cenário(s):	
	Cenário 2
Ambiente:	
	Sistema em operação normal
Estímulo:	
	Usuário navegando no site e realizando um cadastro de ativo
Mecanismo:	
	Telas simples e objetivas sem muitos componentes a serem carregados possibilitando o navegador renderizar de forma rápida os objetos na tela do

usuário	
Medida de resposta:	
O Usuário não deve demorar mais que 5 minutos para realizar o cadastro e os comandos devem gerar respostas intuitivas	
Considerações sobre a arquitetura:	
Riscos:	Pode ocorrer alguma instabilidade no servidor que refletiria em uma indisponibilidade ou lentidão temporária da API ou da própria página
Pontos de Sensibilidade:	API Gateway, Banco de Dados
Tradeoff:	Não se aplica

- **Evidências do Cenário 2**

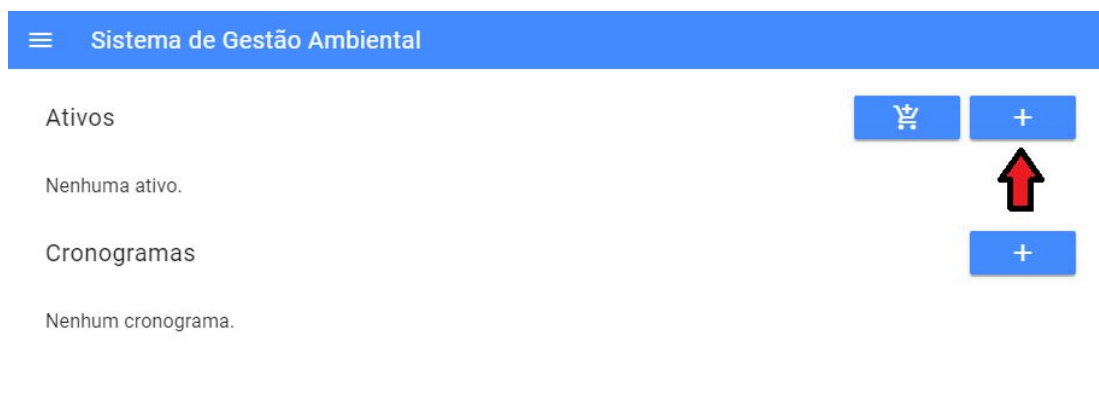
Usuário abre a tela inicial e recebe a orientação para usar o menu lateral.



Usuário navega para a seção de ativos.



Usuário clica no botão para criar um novo ativo.



Usuário preenche o formulário e clica em salvar.

The screenshot shows the 'Sistema de Gestão Ambiental' interface. On the left, there are sections for 'Ativos' (empty) and 'Cronogramas' (empty). A modal window titled 'Criar ativo' is open in the center. The form fields are as follows:

- Patrimônio: TTP244CE
- Nome: Máquina de Teste
- Tipo: Máquina (dropdown menu)
- Horas de uso(Diário): 4
- Data de aquisição: 2017-10-10 (calendar icon)
- Observações: Comprada em leilão.

A blue 'SALVAR' button is at the bottom right of the modal.

O novo ativo é registrado no sistema.

The screenshot shows the 'Sistema de Gestão Ambiental' interface after saving the new asset. The 'Ativos' section now displays the following information:

- Máquina de Teste
- TTP244CE
- Máquina
- 10/10/2017
- Comprada em leilão.

The 'Cronogramas' section remains empty. There are blue buttons with a shopping cart icon and a '+' sign next to the 'Ativos' section, and a blue button with a '+' sign next to the 'Cronogramas' section.

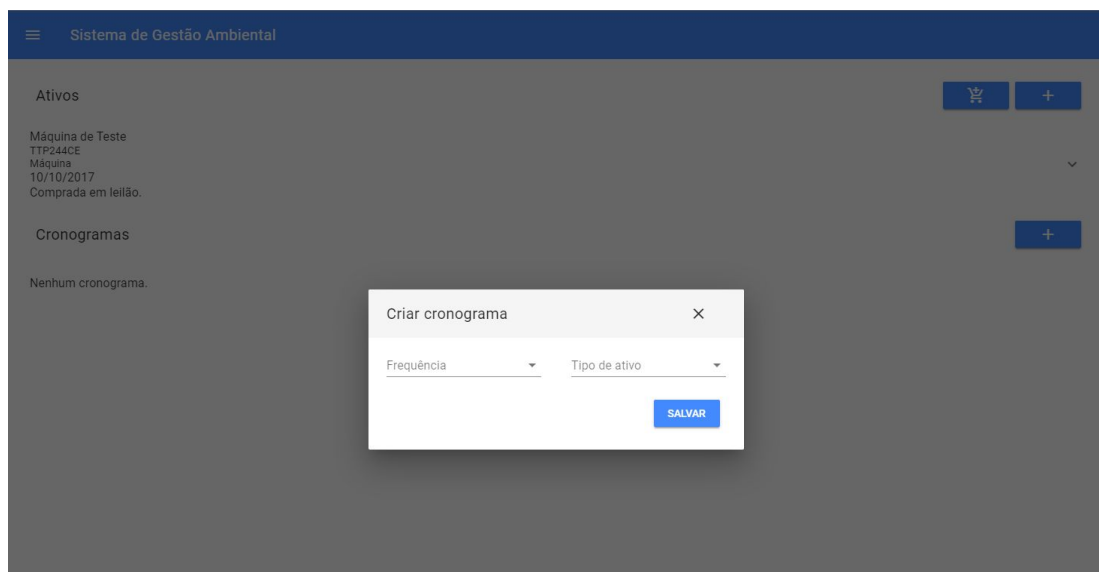
- **Cenário 3**

Atributo de Qualidade:	Acessibilidade
Requisito de Qualidade:	O sistema deve suportar ambientes web e móveis
Preocupação:	
O sistema deve redimensionar seus objetos de acordo com o tamanho da tela	
Cenário(s):	
Cenário 3	
Ambiente:	
Sistema em operação normal	
Estímulo:	
Usuário navegando no site e realizando um cadastro de cronograma de manutenção	
Mecanismo:	
Telas criadas utilizando tecnologias(CSS Media Queries) que permitam o dimensionamento dos componentes de acordo com a resolução atual do usuário	
Medida de resposta:	
O sistema deve se adaptar a resoluções de tela do dispositivo móvel, desktop ou tablet	
Considerações sobre a arquitetura:	
Riscos:	Uma pequena porcentagem dos usuários acessam o sistema por meio de redes móveis 2G, estes usuários poderão enfrentar lentidão devido a baixa velocidade destas redes

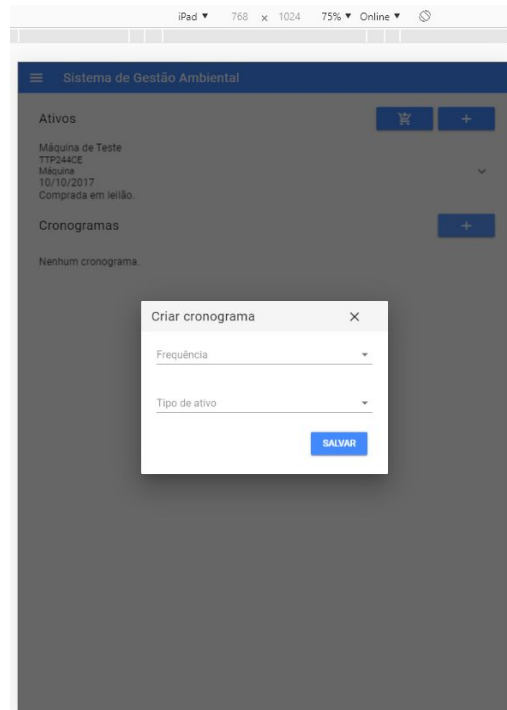
Pontos de Sensibilidade:	Não se aplica
Tradeoff:	Não se aplica

- **Evidências do Cenário 3**

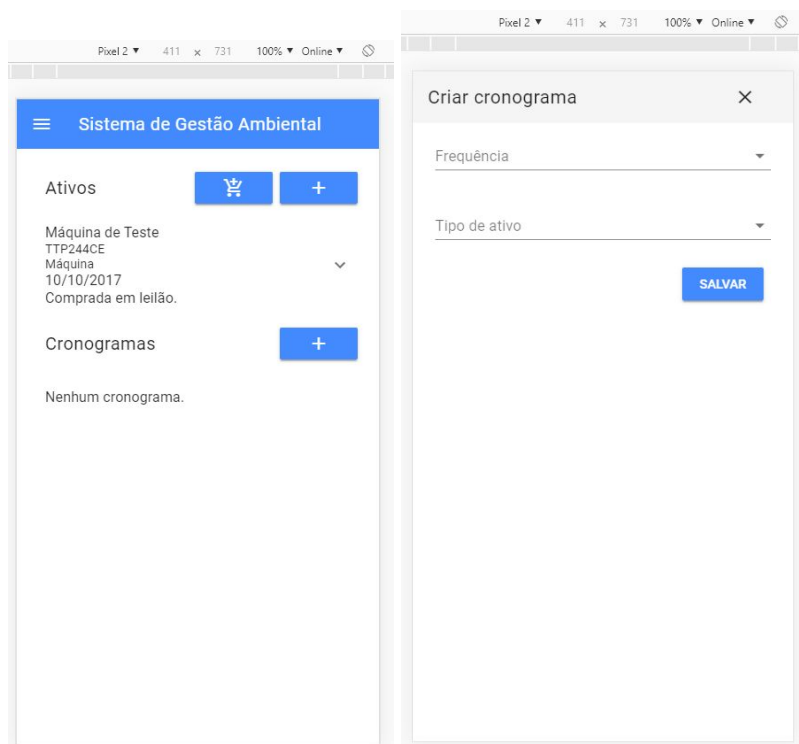
Usuário acessa o sistema com um notebook.



Usuário acessa o sistema com um tablet.



Usuário acessa o sistema com um smartphone.

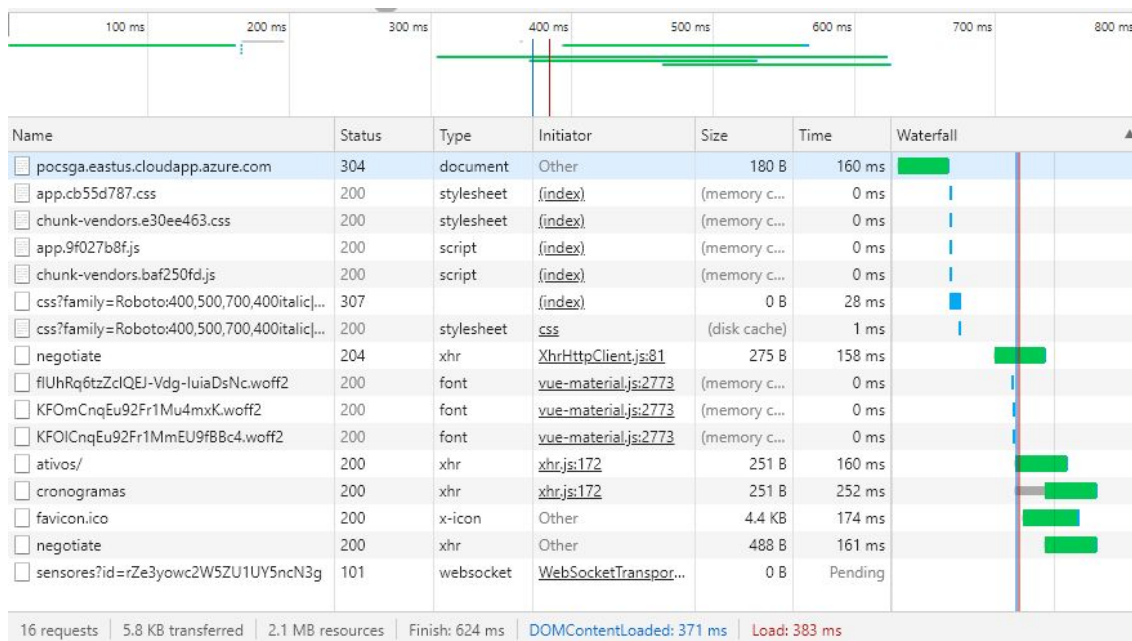


- **Cenário 4**

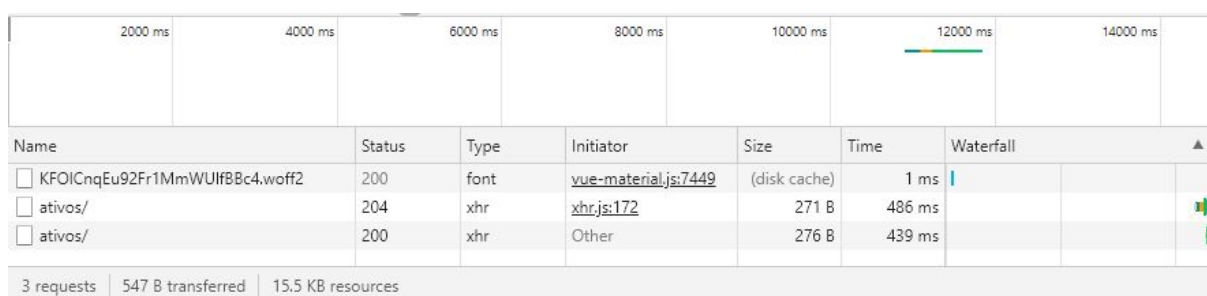
Atributo de Qualidade:	Desempenho
Requisito de Qualidade:	O sistema deve ser rápido
Preocupação:	
Sistema deve apresentar desempenho satisfatório dentro dos limites aceitáveis	
Cenário(s):	
Cenário 4	
Ambiente:	
Sistema em operação normal	
Estímulo:	
Usuário navegando no site e criando um ativo	
Mecanismo:	
Serviços pequenos que fazem poucas coisas e distribuição dos dados em bases diferentes para que haja pouca concorrência entre os recursos no servidor	
Medida de resposta:	
As páginas e comandos do usuário devem apresentar rapidez e bom desempenho	
Considerações sobre a arquitetura:	
Riscos:	Pode ocorrer uma sobrecarga no servidor devido a um alto volume de requisições simultâneas, fazendo com que os processamentos fiquem mais lentos.
Pontos de Sensibilidade:	Servidor de Aplicação
Tradeoff:	Não se aplica

- Evidências do Cenário 4

Usuário navegando para a seção de ativos. Carregamento inicial da página com o sistema hospedado na nuvem com ótima performance.



O tempo de resposta do cadastro de um ativo também foi muito baixo.



6.4. Resultado

Considerando os atributos de qualidade propostos para a POC, a validação arquitetural teve como objetivo analisar se os mesmos foram atendidos, definindo de forma mais objetiva os testes, cenários, pontos fortes e pontos fracos da solução. Abaixo um quadro com os resultados dessa avaliação:

Requisitos Não Funcionais	Testado	Homologado
O sistema deve ser de fácil manutenção	SIM	SIM
O sistema deve ser de fácil utilização	SIM	SIM
O sistema deve suportar ambientes web e móveis	SIM	SIM
O sistema deve ser rápido	SIM	SIM

A arquitetura dividiu o sistema em módulos tanto na sua concepção quanto em sua implementação, todos os serviços foram criados de forma independente com sua lógica dividida em blocos que tratam cada parte do fluxo de forma isolada, seguindo uma divisão de software orientada a domínio, além disso, foi utilizado um software de mensageria de código aberto que implementou originalmente o protocolo *Advanced Message Queuing Protocol* para criar a comunicação entre os módulos do sistema, o que faz com que ele fique ainda mais independente, tudo isso somado faz com que o sistema tenha uma fácil manutenção e sua evolução seja possível sem grandes complicações.

A sua interface foi desenvolvida como um *Single Page Application* utilizando um framework moderno, o que facilitou a criação de uma interface simples e responsiva utilizando os componentes desenvolvidos por sua comunidade, e garantiu um bom desempenho do sistema para o usuário pois a página é carregada somente uma vez e todas as demais requisições seguem como em um sistema *desktop*.

Com toda a divisão realizada, o sistema ganhou muita performance, pois até os bancos de dados são divididos de acordo com os módulos, por outro lado, isso cria uma complexidade alta na infraestrutura do projeto, o que faz necessário o uso de ferramentas como o *Docker* para criar imagens da aplicação. Os testes foram desenvolvidos para permitir que ferramentas de integração contínua e entrega contínua fossem integradas à solução, e apesar da *POC* não utilizar delas, essas

seriam essenciais em um ambiente de produção para garantir o build e deploy da aplicação.

Todas as integrações com sistemas externos foram feitas com Objetos Mock, o que implica que são apenas simulações de uma comunicação com um sistema real, ou seja, em um cenário real essas comunicações seriam diferentes e seria necessário uma adaptação desse software para com as mesmas.

7. Conclusão

Este trabalho apresentou um protótipo arquitetural para um sistema de gestão ambiental de uma mineradora. Com base nos seus resultados, entende-se que a solução atenderia plenamente os seus objetivos com uma arquitetura robusta, escalável e de fácil manutenção.

As maiores dificuldades ficam por conta das várias integrações que devem ser feitas e que criam pontos de falha no sistema, pontos esses que se devidamente tratados (com uso de orquestradores e ferramentas para resiliência) não gerarão problemas para a solução.

REFERÊNCIAS

AFP. **Mineração do Brasil:** Tragédias e receitas enormes. [S. l.]: Estado de Minas, 31 jan. 2019. Disponível em: https://www.em.com.br/app/noticia/internacional/2019/01/31/interna_internacional,1026459/mineracao-do-brasil-tragedias-e-receitas-enormes.shtml. Acesso em: 7 set. 2019.

PENA, Rodolfo F. Alves. **Impactos Ambientais Da Mineração.** [S. l.], 8 mar. 2016. Disponível em: <https://mundoeducacao.bol.uol.com.br/geografia/impactos-ambientais-mineracao.htm>. Acesso em: 7 set. 2019.

SABINO, Rodrigo O.; AGRA, Richardson V.; TOMI, Giorgio De. **DESAFIOS NA GESTÃO DE ATIVOS EM PROJETOS DE MINERAÇÃO DE PEQUENO PORTE: EXEMPLO PRÁTICO.** In: SABINO, Rodrigo O.; AGRA, Richardson V.; TOMI, Giorgio De. **DESAFIOS NA GESTÃO DE ATIVOS EM PROJETOS DE MINERAÇÃO DE PEQUENO PORTE: EXEMPLO PRÁTICO.** São Paulo: Universidade de São Paulo, [entre 2008 e 2019]. Disponível em: http://www.ufrgs.br/rede-carvao/Sess%C3%B5es_A4_A5_A6/A6_ARTIGO_03.pdf. Acesso em: 7 set. 2019.

TORRE, Cesar de la; WAGNER, Bill; ROUSOS, Mike. **.NET Microservices: Architecture for Containerized .NET Applications.** Redmond, Washington: Microsoft Corporation, 2019. Disponível em: <https://docs.microsoft.com/pt-br/dotnet/architecture/microservices/index>. Acesso em: 7 set. 2019.

APÊNDICES

URL do sistema implantado no Azure: <http://pocsga.eastus.cloudapp.azure.com>

URL do GitHub: <https://github.com/welbertwpg/poc-sga>

URL da apresentação da POC no Youtube: <https://youtu.be/TimqShUwRXI>