

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

**Bruno Zanholo**

**PLATAFORMA DE GESTÃO E CONTROLE AMBIENTAL PARA MINERADORA**

Belo Horizonte  
2019

**Bruno Zanholo**

**PLATAFORMA DE GESTÃO E CONTROLE AMBIENTAL PARA MINERADORA**

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

Belo Horizonte

2019

*Dedico este trabalho a minha esposa e filha que mesmo nos  
momentos mais difíceis estão sempre ao meu lado, me  
animando, confortando e estimulando a seguir em frente.  
Amo muito vocês!*

## **AGRADECIMENTOS**

Como bom programador, agradeço a todas as entidades que nos proporcionaram recursos para a elaboração deste artefato. Agradeço também a todos os serviços externos que me forneceram dados e informações me ajudando tanto no crescimento pessoal quanto profissional.

## RESUMO

Este projeto aborda a criação de uma plataforma de gestão e controle ambiental voltada para atividades de negócio de uma empresa mineração.

Recentemente a população brasileira recebeu notícias de tragédias ocorridas em áreas de mineração. Estas tragédias poderiam ser evitadas com um melhor controle das atividades minerárias da Agência Nacional de Mineração do Brasil.

Para este cenário propõe-se o uso de tecnologia para o controle das atividades mineradoras, das consequências desta atividade e do controle e conformidade as normas ambientais para este fim, propostos pela Agencia Nacional de Mineração.

Com sensores, instrumentos computacionais e comunicação em tempo real podemos monitorar, prever e reagir mais rapidamente afim de evitar desastres.

Este projeto arquitetural aborda todos os aspectos da elaboração de uma arquitetura para suprir as necessidades deste monitoramento e a criação de uma prova de conceito para os requisitos considerados arquiteturalmente mais relevantes.

**Palavras-chave:** arquitetura de software, projeto arquitetural, requisitos arquiteturais, controle ambiental, mineração.

## SUMÁRIO

1. Objetivos do trabalho.....	8
<b>2. Descrição geral da solução .....</b>	<b>8</b>
2.1. Apresentação do problema.....	8
2.2. Descrição geral do software (Escopo) .....	9
<b>3. Definição conceitual da solução .....</b>	<b>10</b>
3.1. Requisitos Funcionais .....	10
3.2 Requisitos Não-Funcionais .....	13
3.3. Restrições Arquiteturais .....	19
3.4. Mecanismos Arquiteturais .....	20
<b>4. Modelagem e projeto arquitetural.....</b>	<b>21</b>
4.1. Modelo de casos de uso .....	21
4.2. Descrição resumida de casos de uso.....	26
4.3. Modelo de componentes .....	30
4.4. Modelo de implantação .....	34
4.5. Modelo de dados (opcional).....	36
<b>5. Prova de Conceito (POC) / protótipo arquitetural .....</b>	<b>37</b>
5.1. Implementação e Implantação .....	37
5.1.1. Requisitos não funcionais .....	37
5.1.2. Casos de uso.....	38
5.1.3. Tecnologias utilizadas .....	39
5.1.4. Implantação.....	40
5.2 Interfaces / APIs .....	40
<b>6. Avaliação da Arquitetura.....</b>	<b>42</b>
6.1. Análise das abordagens arquiteturais .....	42
6.2. Cenários .....	42
6.3. Avaliação .....	44
6.4. Resultado.....	59
7. Conclusão.....	61
<b>REFERÊNCIAS.....</b>	<b>62</b>
<b>APÊNDICES.....</b>	<b>63</b>



## 1. Objetivos do trabalho

O objetivo deste projeto é apresentar uma proposta arquitetural para uma plataforma de gestão e controle ambiental voltada para atividades de negócio de uma empresa mineração. O sistema tem como propósito oferecer mecanismos para o controle das operações minerárias com segurança e dentro das normas ambientais estabelecidas pela Agência Nacional de Mineração do Brasil. O sistema deve possuir uma arquitetura baseada em serviços, modular e distribuída. Deve ser altamente resiliente e disponível, pois lida com informações críticas e possível de ser hospedado em ambiente *cloud* ou *on-premise*.

Os objetivos específicos são:

- Criar um módulo para monitoramento das áreas de risco dentro da mineradora e nas áreas adjacentes. Manter monitoramento 24 x 7 destas áreas com a utilização de sensores e evidenciar incidentes que possam ocorrer com dados em tempo real e estatísticos.
- Criar um módulo de segurança e comunicação que deve analisar incidentes ocorridos, e com base em planos de ação configurados ativar alertas e notificações a todos os possíveis afetados.
- Criar as integrações necessárias com sistemas externos que serão utilizados para apoiar o sistema com dados de fornecedores, estatísticos, de controles processuais, ambientais e para notificação de envolvidos em todas as plataformas necessárias.

## 2. Descrição geral da solução

### 2.1. Apresentação do problema

Recentemente a população brasileira recebeu pelos veículos de informação a apresentação de tragédias ocorridas em áreas de mineração em solo nacional. Tragédias como a ocorrida na cidade de Mariana, no estado de Minas Gerais e a mais recente, na cidade de Brumadinho, também no estado de Minas Gerais, resultaram em um desastre de grande proporção humanitário e ambiental. Os desastres provocaram morte, ferimentos, desaparecimentos e desalojamento de centenas de pessoas, entre funcionários da empresa mineradora, pessoas que viviam ou trabalhavam nas áreas afetadas e até em municípios adjacentes.



Estas tragédias poderiam ser evitadas se houvesse um melhor controle das atividades minerárias pela própria mineradora e pela Agência Nacional de Mineração do Brasil.

Para este cenário propõe-se o uso de tecnologia para o controle das atividades mineradoras e das consequências desta atividade.

Os incidentes provocaram comoção pública e reação por parte das ANM – Agência Nacional de Mineração, que implantou novas normas para controle das áreas de mineração. A adequação e controle a esse conjunto de normas pode ser realizada por software de controle ambiental.

Ainda, com a utilização de sensores, comunicação em tempo real e de inteligência computacional seria possível monitorar e prever possíveis incidentes em áreas consideradas de risco. Em caso de incidentes seria possível alertar e notificar a todos que estão nas áreas de risco e sugerir as melhores formas de conter ou mitigar a situação. Em casos extremos, como nas tragédias citadas, seria possível antecipar os incidentes e tomar as melhores decisões para contenção ou evacuação da área que possa ser afetada.

## **2.2. Descrição geral do software (Escopo)**

A elaboração deste software tem por objetivo fornecer uma plataforma integrada de gestão e controle de riscos para uma empresa de mineração.

Os gestores poderão monitorar e controlar as atividades da mineradora. Ainda, poderão analisar os índices de produtividade através relatórios e dashboards, auxiliando na tomada de decisões.

Os técnicos e engenheiros poderão monitorar as operações da mineradora e garantir a conformidade com as normas ambientais.

O departamento jurídico poderá controlar o andamento e os prazos dos processos de outorga de mineração.

Será possível detectar e prever incidentes na área de mineração e nas áreas de risco. Em caso de incidentes, os planos de ação mais indicados serão automaticamente iniciados, enviando alertas e formas de contenção ou evacuação detalhadas.

A plataforma possuirá alta disponibilidade com comunicação em tempo real, segurança e resiliência. Fará o uso de diversas tecnologias, incluindo dispositivos IoT, como sensores, e estará disponível em ambiente web e mobile.

### **3. Definição conceitual da solução**

#### **3.1. Requisitos Funcionais**

- **Módulo de usuários**

- O sistema deve permitir o cadastro dos usuários e seus respectivos acessos no sistema de acordo com sua área de atuação.
- O sistema deve permitir apenas usuários identificados utilizarem o sistema.

- **Módulo cadastro de ativos**

- O sistema deve permitir que sejam cadastradas as zonas de mineração, o tipo de minério extraído, definida a forma de lavra e o tipo de beneficiamento do minério.
- O sistema deve permitir o cadastro de barragens.
- O sistema deve permitir o cadastro dos técnicos, engenheiros, operadores e os demais envolvidos na mineradora.
- O sistema deve permitir o cadastro de maquinário próprio e terceirizado e dos planos de manutenção dos maquinários.
- O sistema deve possuir acesso as APIs dos fornecedores de maquinários para que possam ser solicitadas manutenções, trocas e devoluções.
- O sistema deve permitir o cadastro de insumos (matérias-primas) para a atividade de mineração.
- O sistema deve permitir o cadastro dos minerais beneficiados, dos *prospects* e clientes da mineradora.

- **Módulo controle de processos minerários**

- O sistema deve permitir o controle de segurança dos operários da mineradora. Deve permitir o cadastro de vistorias e laudos das áreas de trabalho, dos maquinários e dos equipamentos de segurança dos operários.
- O sistema deve permitir o controle de todas as etapas do processo de mineração, beneficiamento, armazenamento, venda e transporte do minério. Para cada: tipo de exploração / minério / beneficiamento há um fluxo de processo diferente que pode ser modificado com base em consultorias externas integradas, que promovem a adoção das melhores práticas no processo de controle das atividades minerárias.
- O sistema deve permitir o controle dos rejeitos produzidos durante a atividade mineradora.
- O sistema deve controlar as vistorias e manutenções preventivas nas barragens.
- O sistema deve, através de serviços externos, controlar todas as etapas e prazos do processo de outorga de exploração mineral junto ao DNPM - Departamento Nacional de Produção Mineral - das áreas da mineradora.
- O sistema deve, através de serviços externos, manter-se atualizado e controlar os agendamentos de vistorias de técnicos e engenheiros do DNPM - Departamento Nacional de Produção Mineral.

- **Módulo de monitoramento**

- O sistema deve permitir o cadastro de áreas de risco dentro da mineradora e de áreas próximas.
- O sistema deve permitir o cadastro de operadores da mineradora e de pessoas que podem ser afetadas de alguma maneira dentro da área de risco.
- O sistema deve permitir o cadastro de sensores e as interfaces para a notificação dos dados do sensor. Os sensores poderão ser de diversos tipos.
- O sistema deve permitir o informe das atividades dos sensores. A cada novo informe o sistema deve analisar de acordo com o tipo e a intensidade um possível incidente.

- O sistema deve permitir o cadastro de laudos de monitoramentos resultantes das aferições da área minerada e das áreas de risco realizadas pelos técnicos e engenheiros da mineradora ou órgão regulamentador. A estes laudos deve ser possível anexar fotos e arquivos.
- O sistema deve evidenciar possíveis incidentes e iniciar o plano de ação para a contenção ou mitigação do incidente.
- Os incidentes devem ser classificados e o plano de ação deve depender da classificação do incidente. O sistema deve notificar as autoridades e os afetados da área.
- O sistema deve possuir um histórico de incidentes ocorridos nas áreas de risco como dado estatístico para prevenção de novos incidentes.

- **Módulo de segurança e comunicação**

- O sistema deve permitir o cadastro de planos de ação para incidentes na mineradora e nas áreas de risco.
- O sistema deve emitir alertas para as áreas com incidentes ou com risco de incidente. Todas as pessoas que podem ser afetadas por um incidente em uma área de risco devem ser notificadas do incidente.
- O sistema deve permitir o cadastro de planos de contenção e evacuação para áreas as áreas afetadas por incidentes.

- **Módulo de inteligência de negócio**

- O sistema deve realizar simulações com bases nos dados históricos de monitoramento das áreas de risco e de sistemas externos afim de evidenciar possíveis novos incidentes nas áreas de risco mapeadas.
- O sistema deve realizar simulações com base em dados estatísticos e geográficos das áreas de risco, afim de gerar planos de contenção e evacuação mais eficazes.
- O sistema deve apresentar dashboards das simulações de risco.

- O sistema deve apresentar dashboards gerenciais para auxílio na tomada de decisões.

- **Módulo de *compliance***

- O sistema deve alinhar o controle de rejeitos produzidos pela atividade mineradora e as normas que regulam o tratamento comparando os dados do monitoramento local com as especificações fornecidas por serviços externos.
- O sistema deve alinhar o controle de segurança dos operários a norma NR22.
- O sistema deve alinhar toda a atividade mineradora em relação as normas ambientais e de mineração fornecidas pelo DNPM - Departamento Nacional de Produção Mineral - através de serviços externos.

- **Módulo de relatórios de acompanhamento**

- O sistema deve emitir os laudos para todas as etapas do processo de outorga de exploração mineral junto ao DNPM - Departamento Nacional de Produção Mineral.
- O sistema deve emitir o relatório oficial anual de lavra.
- Relatórios de zonas de risco x nível de risco
- O sistema deve emitir relatórios gerenciais.

### 3.2 Requisitos Não-Funcionais

#### Usabilidade:

- O sistema deverá ser operável por usuários com pouca ou nenhuma necessidade de treinamento prévio.

<b>Estímulo</b>	Interação do usuário com a interface do sistema
-----------------	---

<b>Fonte do estímulo</b>	Usuário com pouca ou nenhuma experiência com o sistema
<b>Ambiente</b>	Produção, carga normal
<b>Artefato</b>	Qualquer módulo do sistema
<b>Resposta</b>	O sistema apresenta interface simples e opções de ajuda ao usuário na própria tela
<b>Medida de resposta</b>	O usuário opera o sistema sem nenhuma dificuldade

**Portabilidade:**

- O sistema deverá fazer uso de design responsivo em suas interfaces gráficas, comportando-se adequadamente em navegadores acessados via computador, smartphone ou tablet.

<b>Estímulo</b>	Uso do sistema pelo celular
<b>Fonte do estímulo</b>	Usuário acessa o sistema pelo aparelho celular
<b>Ambiente</b>	Produção, carga normal
<b>Artefato</b>	Qualquer módulo do sistema
<b>Resposta</b>	A usabilidade do sistema deve ser igual, independente do device utilizado pelo usuário do sistema
<b>Medida de resposta</b>	O usuário opera o sistema da mesma forma, tanto no computador quanto no celular ou tablet

**Interoperabilidade:**

- O sistema deve comunicar-se a sistemas externos e aos serviços distribuídos que compõe a arquitetura utilizando os protocolos HTTP e SOAP.

<b>Estímulo</b>	Acesso ao serviço de consultoria externa de processos minerários
<b>Fonte do estímulo</b>	Consulta a API de processos minerários externa
<b>Ambiente</b>	Produção, carga normal
<b>Artefato</b>	Módulo de controle de processos minerários, módulo de inteligência do negócio e módulo de <i>compliance</i>
<b>Resposta</b>	O serviço de consultoria respondeu com sucesso os processos minerários solicitados
<b>Medida de resposta</b>	Comunicação, transferência e recebimento de dados efetuados com sucesso

- O serviço deve comunicar-se com APIs de envio de e-mails e SMS para alerta dos afetados em caso de incidentes.

<b>Estímulo</b>	Envio de alertas por SMS ou e-mail para os afetados de uma área com incidente detectado.
<b>Fonte do estímulo</b>	Execução do plano de ação para incidentes
<b>Ambiente</b>	Produção, carga normal
<b>Artefato</b>	Módulo de segurança e comunicação
<b>Resposta</b>	As mensagens de alerta são enviadas
<b>Medida de resposta</b>	Comunicação e transferência de dados efetuada com sucesso

- Os sensores devem enviar dados de monitoramentos para o broker de comunicação de utilizando protocolo MQTT.

<b>Estímulo</b>	Envio de dados de monitoramentos pelos sensores
<b>Fonte do estímulo</b>	Sensores de monitoramento nas áreas de risco
<b>Ambiente</b>	Produção, carga normal
<b>Artefato</b>	Módulo de monitoramento
<b>Resposta</b>	O broker de comunicação recebe os dados do sensor
<b>Medida de resposta</b>	Comunicação e transferência de dados efetuada com sucesso

#### Segurança:

- Para acessar ter acesso ao sistema é necessário estar autenticado.

<b>Estímulo</b>	Acesso aos cadastros do sistema
<b>Fonte do estímulo</b>	Usuário não autenticado
<b>Ambiente</b>	Produção, carga normal
<b>Artefato</b>	Módulo de cadastro de ativos
<b>Resposta</b>	O sistema redireciona o usuário para a tela de autenticação
<b>Medida de resposta</b>	O sistema não permite acesso ao sistema sem autenticação

#### Manutenibilidade:



- A manutenção no sistema deve ser fácil.

<b>Estímulo</b>	Modificação no layout de um modelo de relatório
<b>Fonte do estímulo</b>	Pipeline de integração contínua, <i>Deploy</i>
<b>Ambiente</b>	Desenvolvimento, Atualização
<b>Artefato</b>	Módulo de relatório de acompanhamentos
<b>Resposta</b>	Demais módulos do sistema permanecem em atividade
<b>Medida de resposta</b>	A atualização do módulo de relatórios de acompanhamentos não deve comprometer o uso do sistema

- O sistema deve ser testável.

<b>Estímulo</b>	Teste automatizado dos módulos do sistema
<b>Fonte do estímulo</b>	Pipeline de integração contínua
<b>Ambiente</b>	Desenvolvimento
<b>Artefato</b>	Qualquer módulo do sistema
<b>Resposta</b>	Demais módulos do sistema permanecem em atividade
<b>Medida de resposta</b>	Com a execução dos testes automatizados em conformidade o pipeline de integração continua deve realizar o <i>deploy</i> da nova versão do módulo.

#### Confiabilidade:

- O sistema deve ter alta disponibilidade, operando ininterruptamente.

<b>Estímulo</b>	Envio dos dados dos sensores ao broker IoT do sistema a cada segundo
<b>Fonte do estímulo</b>	Sensores
<b>Ambiente</b>	Produção, carga normal
<b>Artefato</b>	Módulo de monitoramento
<b>Resposta</b>	Os dados são recebidos e processados pelo módulo de monitoramento
<b>Medida de resposta</b>	Dados recebidos com sucesso

- O sistema deve recuperar-se de falhas.

<b>Estímulo</b>	Persistência de dados no banco de dados relacional do sistema
<b>Fonte do estímulo</b>	Usuário confirma a gravação de um dado no sistema
<b>Ambiente</b>	Produção, carga acima do normal
<b>Artefato</b>	Qualquer módulo do sistema
<b>Resposta</b>	A persistência dos dados é realizada mesmo após erro inicial
<b>Medida de resposta</b>	Ao detectar um deadlock ou timeout de conexão no banco de dados o sistema aplica políticas de <i>Retry</i> para confirmar a persistência de dados

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

<b>Estímulo</b>	Recuperação de dados
-----------------	----------------------

<b>Fonte do estímulo</b>	Usuário solicita ao sistema a mesma informação
<b>Ambiente</b>	Produção, carga normal
<b>Artefato</b>	Qualquer módulo do sistema
<b>Resposta</b>	O usuário recebe os dados solicitados a partir da segunda requisição mais rapidamente
<b>Medida de resposta</b>	Ao recuperar informações o sistema mantém estas em memória afim de otimizar novas consultas

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

<b>Estímulo</b>	Aumento do volume de requisições aos serviços que compõe o sistema devido a maior volume de usuário simultâneos.
<b>Fonte do estímulo</b>	Aumento na quantidade de usuários simultâneos
<b>Ambiente</b>	Produção, carga acima da média
<b>Artefato</b>	Qualquer módulo do sistema
<b>Resposta</b>	O usuário utiliza o sistema normalmente
<b>Medida de resposta</b>	O cluster de instâncias dos serviços detecta automaticamente o aumento de requisições ao serviço e ativa novas instâncias para os serviços afim de para compensar o aumento da demanda

### 3.3. Restrições Arquiteturais

- O sistema deve ser desenvolvido utilizando a plataforma .NET.
- O sistema deve ser modular e orientado a serviços afim de facilitar a implantação.

- O sistema deve realizar integrações com sistemas externos.
- O sistema deve ser implantado utilizando recursos de integração contínua.
- O sistema deve conter automação de testes no pipeline de integração contínua.
- O sistema deve possuir alta disponibilidade.
- O sistema deve ser possível de ser hospedado em ambiente *cloud* ou *on-premise*.

### 3.4. Mecanismos Arquiteturais

Mecanismo de análise	Mecanismo de design	Mecanismo de implementação
Front-end	Interface de interação com o usuário	ASP.NET MVC, Bootstrap, JQuery
Back-end	Regras de negócio	.NET Framework Core
Integrações entre módulos e outros sistemas	Interfaces de comunicação entre os módulos utilizando JSON, XML	API REST e Webservices SOAP
Persistência	Acesso a dados	Entity Framework Core
Persistência	Banco de dados relacional	Azure SQL
Cache	Acesso a dados em memória	Azure Cache for Redis
Segurança	Autenticação e autorização	Azure Directory
Host	Publicação do front-end	Azure AppServices
Host	Publicação do back-end	Azure AppService, Azure Functions, Azure Container

		Instances, Azure Api Management
IoT	Broker de comunicação com os sensores	Azure IoT Hub
Alta disponibilidade	Balanceamento de carga e monitoramento da aplicação	Azure Kubernetes Services, Azure LoadBalancer, Azure Monitor
Notificações	Notificação aos usuários e interessados via SMS, E-MAIL e WhatsApp	Sendgrid, Twilio
Relatórios	Gerador de relatórios	Crystal Reports
CI/CD	Pipeline de integração continua	GitLab
Automação de testes	Execução de testes automatizados	MSTest
Build	Geração das versões	MSBuild
Deploy	Configuração do deploy automatizado	GitLab
Versionamento	Controle do código fonte	Git / GitLab

## 4. Modelagem e projeto arquitetural

### 4.1. Modelo de casos de uso

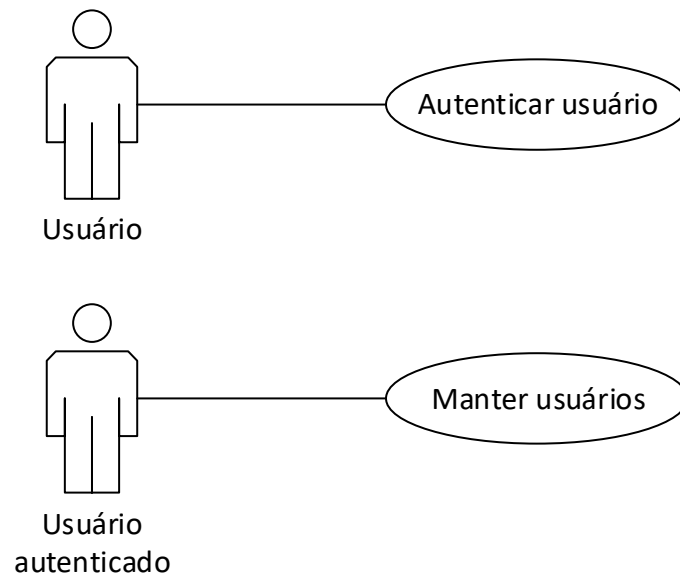


Figura 1 - Casos de uso do módulo de usuários

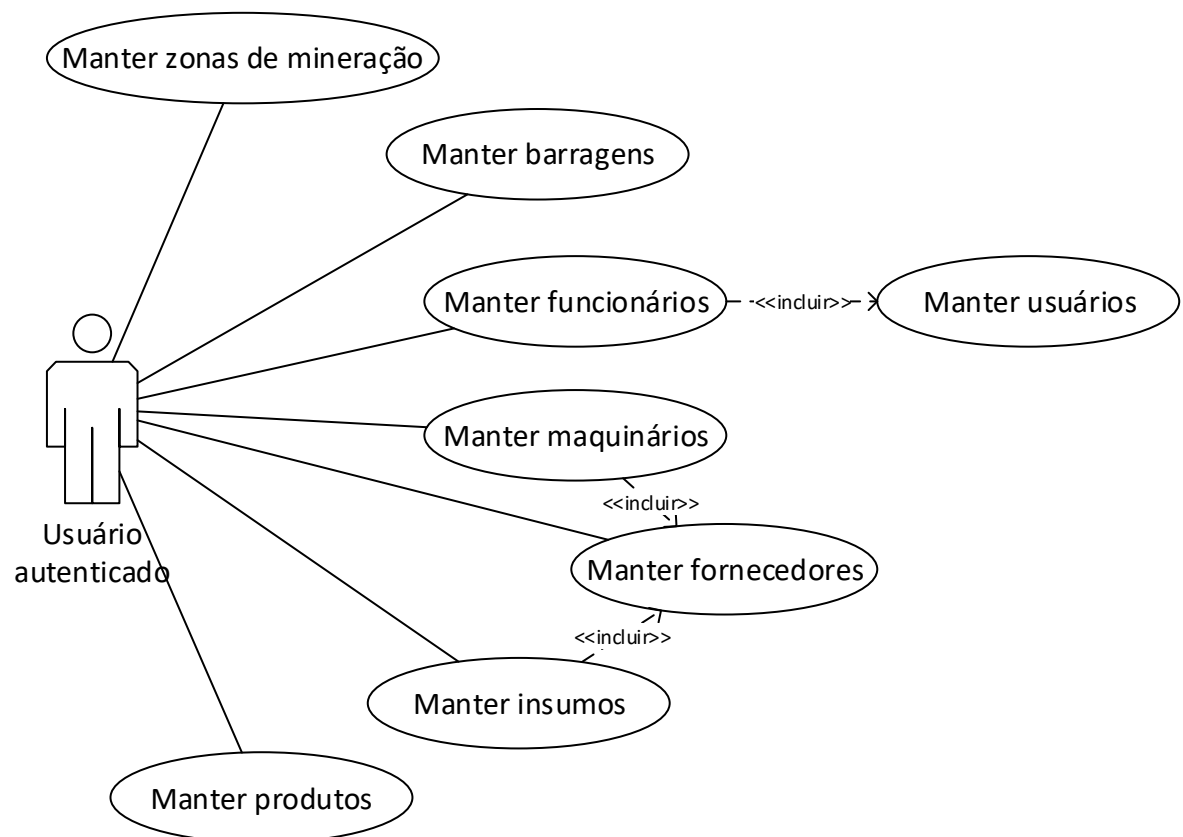
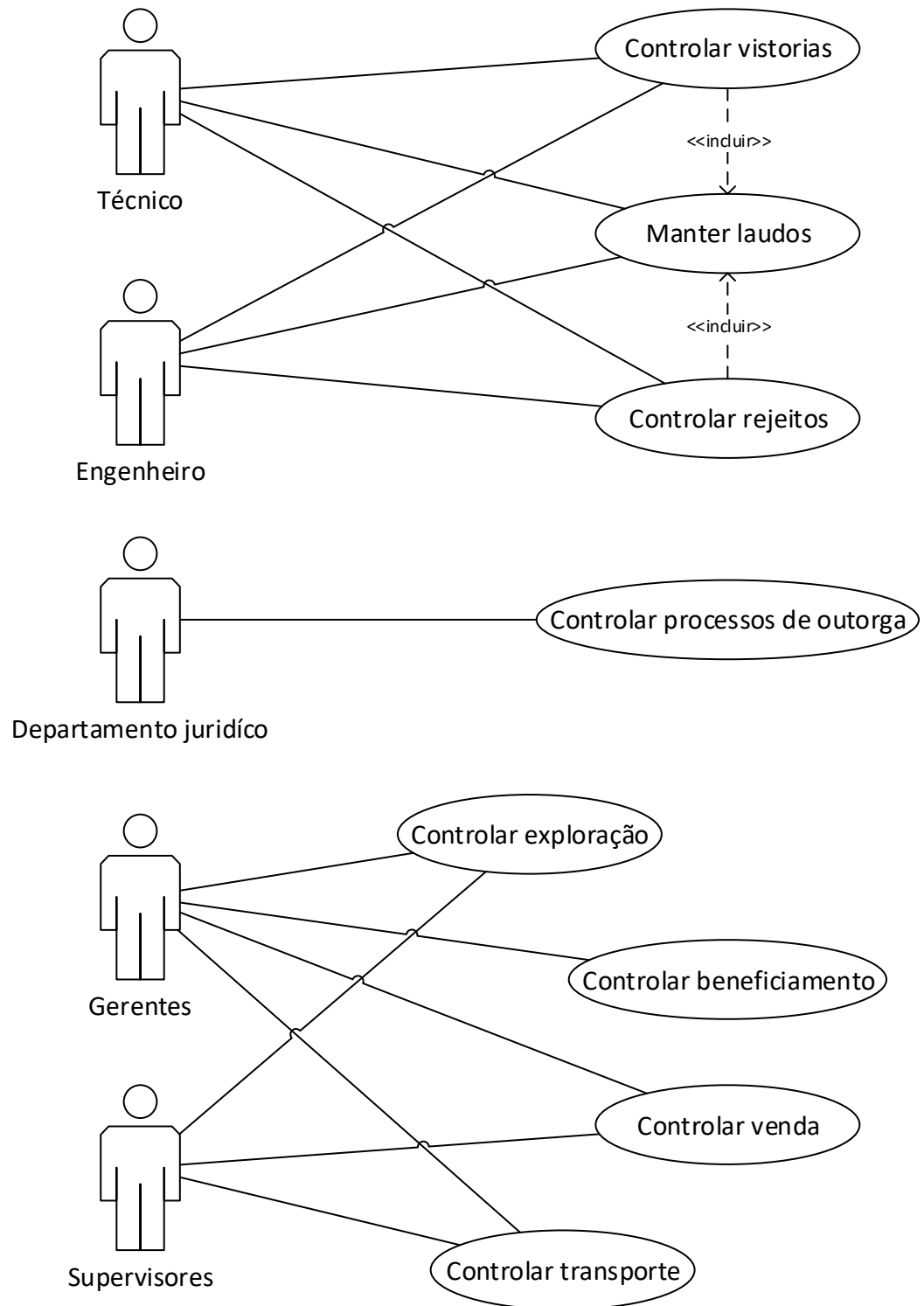
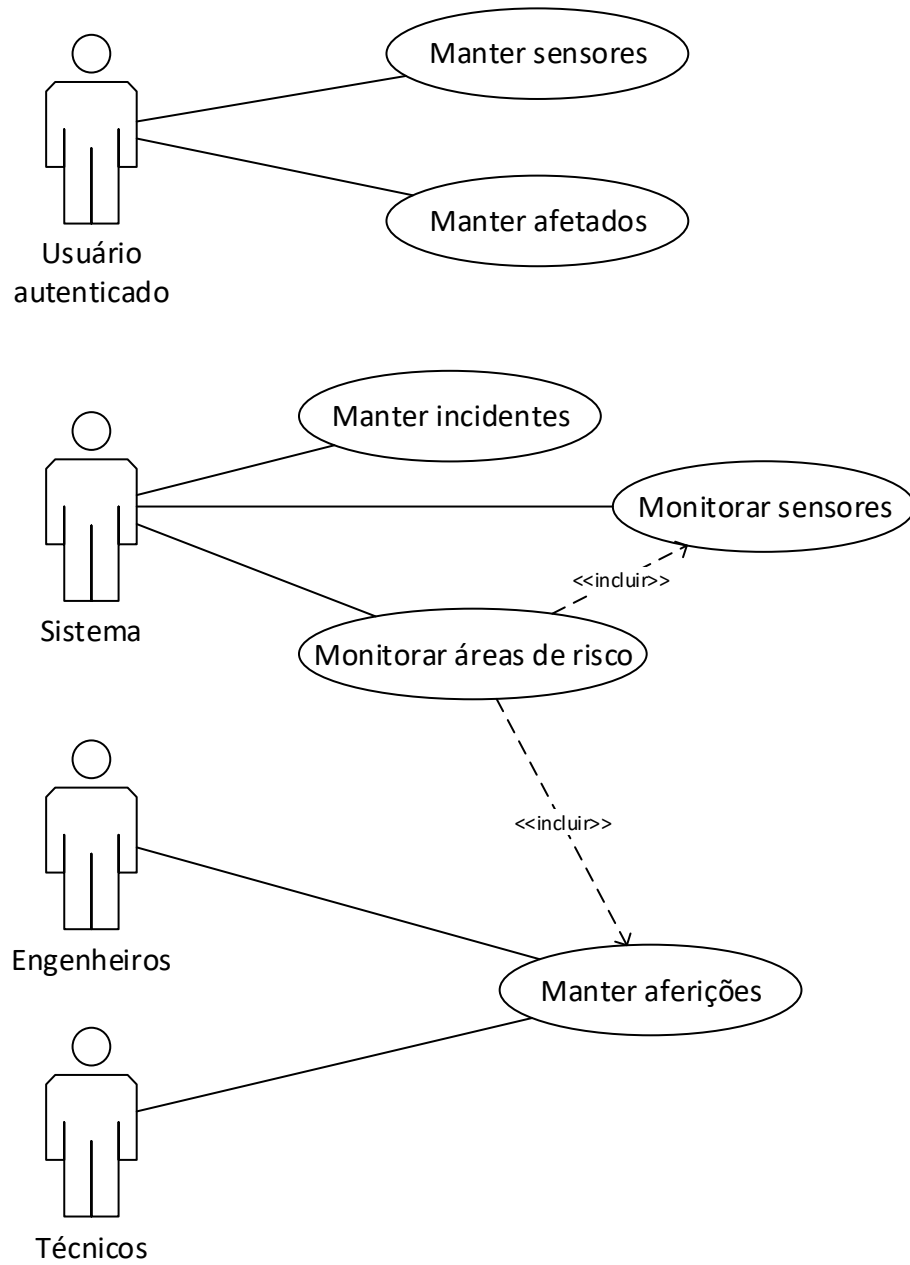


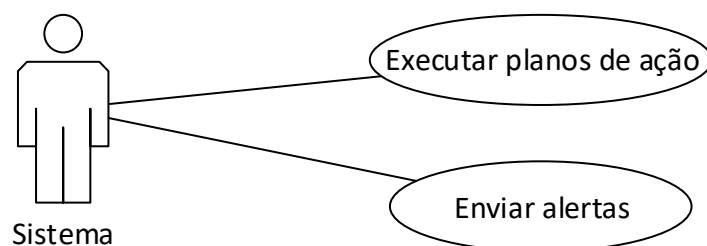
Figura 2 - Casos de uso do módulo de cadastro de ativos



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

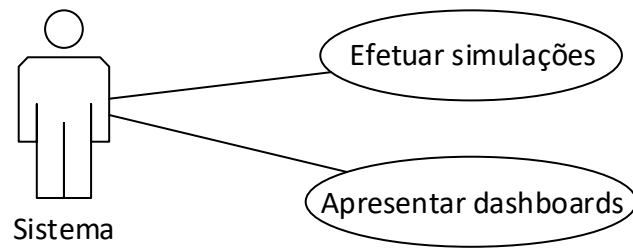


*Figura 4 - Casos de uso do módulo de monitoramento*

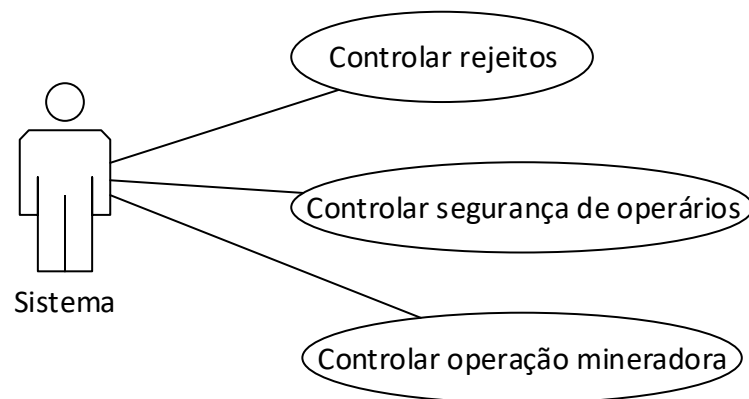


*Figura 5 - Casos de uso do módulo de segurança e comunicação*

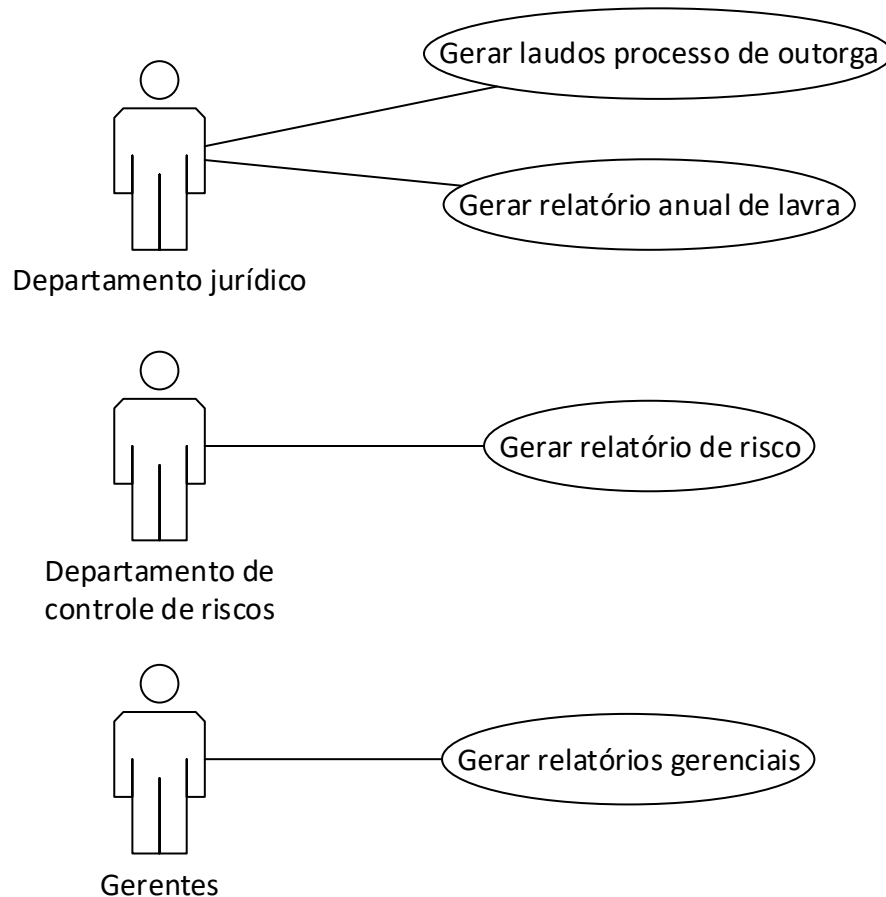




*Figura 6 - Casos de uso do módulo de inteligência de negócio*



*Figura 7 - Casos de uso do módulo de compliance*



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

#### 4.2. Descrição resumida de casos de uso

- Módulo de usuários
  - Caso de uso: **Manter usuários**
    - Descrição resumida: Este cadastro permite ao usuário cadastrar, alterar, visualizar e excluir demais usuários do sistema. Ainda, permite a definição dos departamentos e acessos permitidos ao usuário.
  - Caso de uso: **Autenticar usuário**
    - Descrição resumida: Permite que o usuário previamente cadastrado possa acessar os recursos do sistema mediante login com senha.
- Módulo de cadastro de ativos
  - Caso de uso: **Manter zonas de mineração**
    - Descrição resumida: Este cadastro permite ao usuário cadastrar, alterar, visualizar e excluir zonas de mineração em que a mineradora atua.

- Caso de uso: **Manter barragens**
  - Descrição resumida: Este cadastro permite ao usuário cadastrar, alterar, visualizar e excluir barragens de resíduos formadas nas zonas de mineração em que a mineradora atua. É possível determinar o tipo de barragem e sua classificação de acordo com o padrão estabelecido pela Superintendência de produção mineral.
- Caso de uso: **Manter funcionários**
  - Descrição resumida: Este cadastro permite ao usuário cadastrar, alterar, visualizar e excluir os funcionários da mineradora.
- Caso de uso: **Manter maquinários**
  - Descrição resumida: Este cadastro permite ao usuário cadastrar, alterar, visualizar e excluir maquinários próprios ou terceirizados utilizados na mineradora.
- Caso de uso: **Manter insumos**
  - Descrição resumida: Este cadastro permite ao usuário cadastrar, alterar, visualizar e os insumos necessários para a atividade de mineração e vinculá-los ao cadastro de fornecedores. É possível a determinação de estoques mínimos e máximos.
- Caso de uso: **Manter fornecedores**
  - Descrição resumida: Este cadastro permite ao usuário cadastrar, alterar, visualizar e excluir fornecedores de maquinários e insumos de mineração.
- Caso de uso: **Manter produtos**
  - Descrição resumida: Este cadastro permite ao usuário cadastrar, alterar, visualizar e excluir os produtos beneficiados pela mineradora.
- Módulo de controle de processos minerários
  - Caso de uso: **Controlar vistorias**
    - Descrição resumida: Este controle permite a gestão e agendamentos de vistorias nas áreas da mineradora pelos técnicos e engenheiros.
  - Caso de uso: **Manter laudos**
    - Descrição resumida: Este cadastro permite aos técnicos e engenheiros cadastrar, alterar, visualizar laudos resultantes da atividade de vistorias nas áreas da mineradora pelos técnicos e engenheiros.

- Caso de uso: **Controlar exploração**
  - Descrição resumida: Este controle permite a gestão das atividades diárias da exploração de minérios da mineradora pelos gerentes e supervisores.
- Caso de uso: **Controlar beneficiamento**
  - Descrição resumida: Este controle permite a gestão das atividades diárias de beneficiamento de minérios da mineradora pelos gerentes e supervisores.
- Caso de uso: **Controlar venda**
  - Descrição resumida: Este controle permite a gestão das atividades diárias de venda de minérios da mineradora pelos vendedores, gerentes e supervisores.
- Caso de uso: **Controlar transporte**
  - Descrição resumida: Este controle permite a gestão das atividades diárias de transporte de minérios da mineradora pelos gerentes e supervisores.
- Caso de uso: **Controlar rejeitos**
  - Descrição resumida: Este controle permite a gestão das atividades diárias de controle de rejeitos das barragens da mineradora pelos técnicos e engenheiros.
- Caso de uso: **Controlar processos de outorga**
  - Descrição resumida: Este controle permite a gestão de prazos das etapas dos processos de outorga para mineração pelo departamento jurídico.
- Módulo de monitoramento
  - Caso de uso: **Manter afetados**
    - Descrição resumida: Este cadastro permite ao usuário cadastrar, alterar, visualizar e excluir possíveis afetados em áreas de risco monitoradas pela mineradora.
  - Caso de uso: **Manter sensores**
    - Descrição resumida: Este cadastro permite ao usuário cadastrar, alterar, visualizar e excluir sensores que enviarão notificações de atividades nas áreas de risco.
  - Caso de uso: **Monitorar áreas de risco**

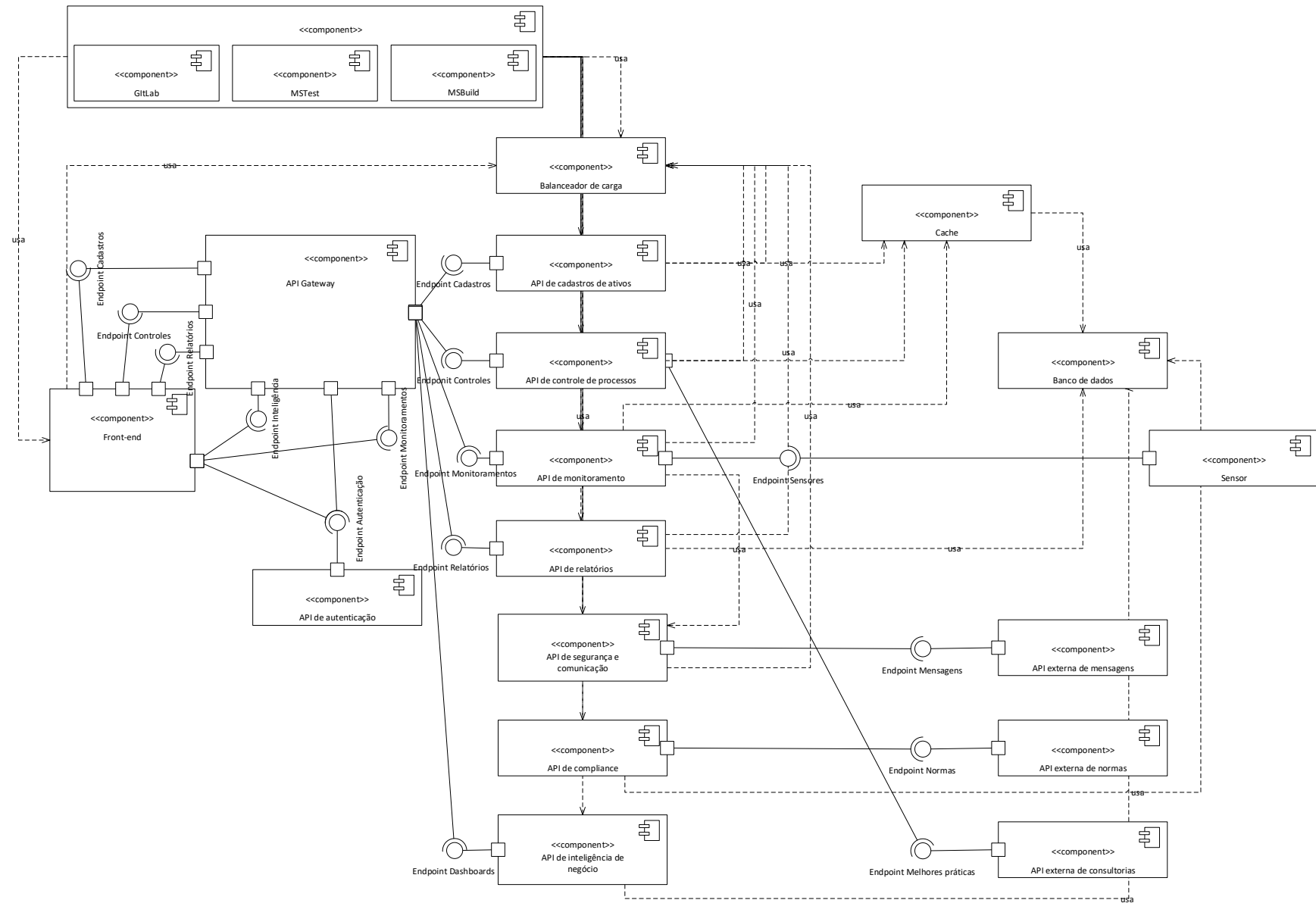
- Descrição resumida: Permite o monitoramento autônomo em tempo real das áreas de risco.
- Caso de uso: **Monitorar sensores**
  - Descrição resumida: Concentra o monitoramento dos sensores espalhados nas áreas de risco da mineradora. Recebe uma atividade do sensor e efetua o seu tratamento com base no tipo de evento informado pelo sensor e na intensidade, podendo ou não gerar um novo incidente.
- Caso de uso: **Manter aferições**
  - Descrição resumida: Este cadastro permite ao usuário cadastrar, alterar, visualizar e excluir dados das aferições realizadas por técnicos e engenheiros da mineradora nas áreas de risco.
- Caso de uso: **Manter incidentes**
  - Descrição resumida: Este cadastro permite ao usuário ou o sistema cadastrar, alterar, visualizar dados de possíveis incidentes e incidentes ocorridos nas áreas de risco.
- Módulo de segurança e comunicação
  - Caso de uso: **Executar planos de ação**
    - Descrição resumida: Após a evidência de um novo incidente, deve executar os planos de ação para a área do incidente, respeitando a classificação deste enviando alertas e mensagens aos afetados.
  - Caso de uso: **Enviar alertas**
    - Descrição resumida: Envia alertas e mensagens aos afetados por um plano de ação. Este recurso poderá enviar e-mails ou SMS aos afetados da área do incidente cadastrados.
- Módulo de inteligência de negócio
  - Caso de uso: **Efetuar simulações de incidentes**
    - Descrição resumida: Processo autônomo que realiza simulações de possíveis incidentes nas áreas de risco utilizando os dados encaminhados pelos sensores, dados externos e históricos.
  - Caso de uso: **Apresentar dashboards**
    - Descrição resumida: Apresenta dashboards de monitoramentos de risco e gerenciais para gestores.
- Módulo de *compliance*

- Caso de uso: **Controlar rejeitos**
  - Descrição resumida: Processo autônomo que monitora o controle de rejeitos produzidos pelo processo de mineração e analisa a conformidade junto as normas do DNPM - Departamento Nacional de Produção Mineral.
- Caso de uso: **Controlar segurança de operários**
  - Descrição resumida: Processo autônomo que monitora a segurança dos operários em relação a norma NR22 de trabalho.
- Caso de uso: **Controlar operação mineradora**
  - Descrição resumida: Processo autônomo que monitora as atividades de mineração e seus resultados em relação as normas ambientais e de mineração fornecidas pelo DNPM - Departamento Nacional de Produção Mineral.
- Módulo de relatórios de acompanhamento
  - Caso de uso: **Gerar laudos processos de outorga**
    - Descrição resumida: Gera os laudos necessários para os prazos jurídicos do processo de outorga em acordo ao DNPM - Departamento Nacional de Produção Mineral para o departamento jurídico.
  - Caso de uso: **Gerar relatório anual de lavra**
    - Descrição resumida: Gera o relatório anual de lavra, obrigatório de acordo com as normas do DNPM - Departamento Nacional de Produção Mineral para o departamento jurídico.
  - Caso de uso: **Gerar relatório de risco**
    - Descrição resumida: Gera o relatório de riscos baseados nos dados gerados pelo módulo de monitoramento para o departamento de gestão de risco.
  - Caso de uso: **Gerar relatórios gerenciais**
    - Descrição resumida: Gera os relatórios gerenciais para acompanhamento da produtividade da mineradora para os gerentes e supervisores.

#### 4.3. Modelo de componentes

O diagrama de componentes abaixo apresenta a comunicação entre os componentes da arquitetura e as tecnologias de cada um. Os componentes estão organizados de forma que

possam ser reutilizados. Fornecem interfaces bem definidas de acordo com suas responsabilidades.





Consideramos para este sistema a divisão em módulos com uma separação bem definida entre front-end e back-end.

O front-end será responsável por toda interação com o usuário final, assim não possuirá nenhuma regra de negócio. Apenas validações de entradas serão permitidas. Será utilizado o framework ASP.NET MVC e o framework Bootstrap para confecção da interface. O Bootstrap possui inúmeros componentes de interface prontos, padronizados e responsivos, garantindo assim conformidade aos requisitos não funcionais de usabilidade e portabilidade.

O front-end será publicado no serviço Azure AppServices.

O acesso aos recursos do sistema será permitido mediante autenticação do usuário. O acesso será controlado por token gerado na autenticação do usuário. O token de acesso será gerado e validado pelo serviço Azure Active Directory, garantindo assim conformidade ao requisito não funcional de segurança.

O back-end será composto de vários módulos e cada módulo conterá ao menos um serviço. Estes módulos expõem endpoints para outro módulo do sistema ou para o front-end. A comunicação entre os módulos do back-end e do front-end com o back-end nunca será feita diretamente. Os módulos não conterão endpoints públicos. O acesso aos endpoints de cada módulo deve ser realizado através do gateway Azure API Management. Este serviço proverá um endpoint centralizado para acesso aos módulos, exigirá um token de autenticação para acesso aos recursos do módulo, garantindo assim conformidade ao requisito não funcional de segurança.

Os serviços de back-end serão publicados no Azure AppService, Azure Functions e Azure Container Instances. Os serviços poderão escalar automaticamente conforme o volume de requisições. Para orquestrar a escala automática serão utilizados: o Azure Kubernetes Services e Azure LoadBalancer, garantindo conformidade ao requisito não funcional de confiabilidade.

A aplicação conterá apenas um único banco de dados relacional. O banco de dados utilizado será o Azure SQL. Este serviço provê recursos de escala, replicação e backup automáticos, garantindo conformidade ao requisito funcional de confiabilidade.

Afim de otimizar a performance do sistema, será utilizado o recurso de cache, mantendo em memória dados frequentemente utilizados pelos serviços, economizando requisições ao banco de dados e tráfego de rede desnecessários. Para este cenário será utilizado o Azure Cache for Redis, garantindo assim, mais uma vez, conformidade ao requisito funcional de confiabilidade.

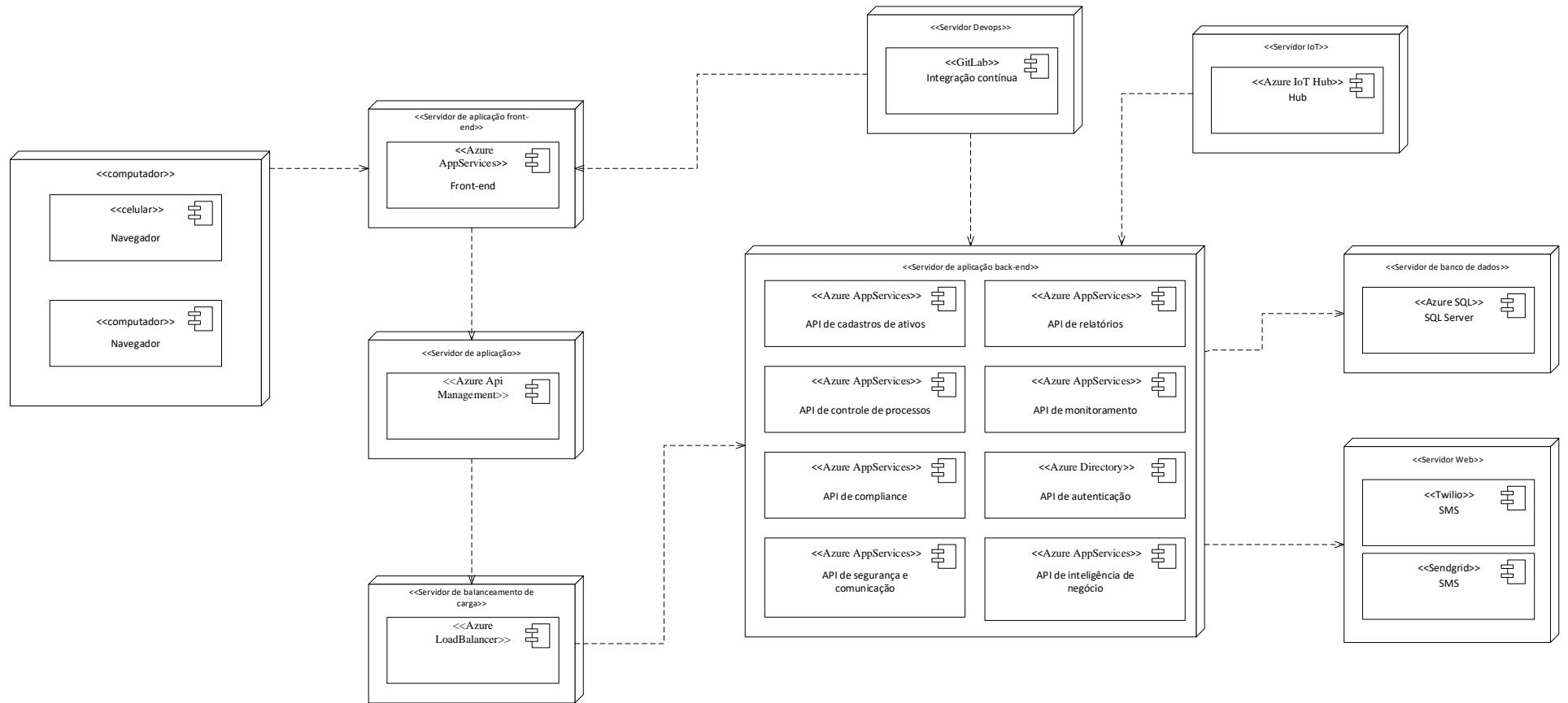
O sistema depende da utilização de dispositivos IoT (sensores) para obtenção de dados em tempo real das áreas de risco. Estes sensores utilizam protocolo MQTT (padrão para dispositivos IoT) para envio das informações. Será utilizado o broker Azure IoT Hub para receber os dados dos sensores. O sistema, também, prevê integrações com sistemas externos para recuperação das normas do DNPM, dos prazos de processo de outorga e das mensagens a serem enviadas aos afetados. Todas estas integrações respeitam o protocolo e interface de cada um destes sistemas. Para envio de alertas aos usuários serão utilizados os serviços Sendgrid e Twilio. Através destas integrações garantimos conformidade ao requisito funcional de interoperabilidade.

Todo o ambiente de produção será monitorado em tempo real. Será utilizado o serviço Azure Monitor para este monitoramento, garantindo conformidade ao requisito não funcional de confiabilidade.

Os artefatos executáveis produzidos serão construídos utilizando a plataforma .NET, serão versionados através do padrão Git na plataforma GitLab. Através desta plataforma, será construído o pipeline de integração contínua, incluindo a execução de testes automatizados com o MSTest, build de versão com o MSBuild e deploy em ambiente de produção. Com estes recursos Devops garantimos a conformidade ao requisito funcional de manutenibilidade.

#### **4.4. Modelo de implantação**

O modelo de implantação auxilia no entendimento de como os componentes de software estarão disponíveis fisicamente e como a comunicação entre deve ocorrer. Este modelo de implantação da arquitetura é apresentado abaixo.

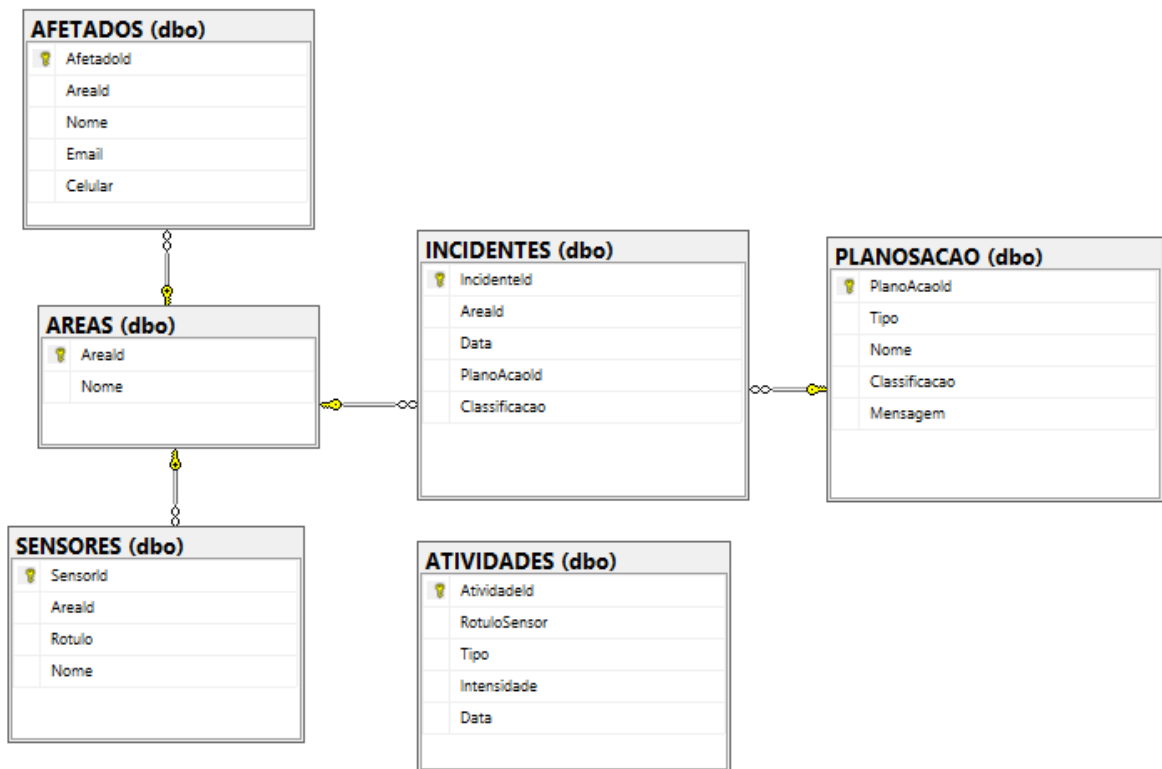


Os módulos do sistema poderiam ser implantados em uma plataforma on-premise também.

<b>Componente</b>	<b>Descrição</b>
Navegador	Representa os browsers utilizados para acesso das aplicações de front-end. Realizam a exibição do HTML para os usuários
Servidor de aplicação front-end	Servidor onde estará o front-end da plataforma.
Servidor de aplicação back-end	Servidor onde estarão todas as APIs da plataforma.
Servidor de balanceamento de carga	Servidor que realiza um balanceamento das requisições para o servidor de back-end.
Servidor Devops	Representa o servidor que armazenará o código fonte e realizará o pipeline de integração continua.
Servidor de banco de dados	Representa o servidor onde estará o repositório de dados da plataforma.
Servidor Web	Representa os servidores onde estão disponibilizados os serviços de terceiros utilizado pela plataforma.

#### **4.5. Modelo de dados (opcional)**

Como a aplicação utiliza um único banco de dados relacional, apenas um modelo de dados foi gerado.



## 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 tem como objetivo validar os aspectos importantes da arquitetura relativos aos requisitos não funcionais. Os requisitos não funcionais escolhidos estão listados abaixo.

- **Segurança:** Este requisito não funcional foi escolhido devido à criticidade e a preocupação em manter os dados seguros.
  - **Critérios de aceite:**
    - Impedir que usuários possam acessar páginas ou APIs privadas sem estar autenticação.
    - Redirecionar o usuário não autenticado para tela de autenticação.

- Permitir a livre navegação em telas públicas.
- **Interoperabilidade:** Este requisito não funcional foi escolhido pois a arquitetura do sistema é distribuída e a comunicação entre os módulos e sensores e com sistemas de terceiros são de extrema importância para o perfeito funcionamento sistema.
  - **Critérios de aceite:**
    - O sistema deve conseguir se comunicar com as mais diversas tecnologias de outros sistemas.
    - O sistema deve prover uma interface de comunicação para o broker de sensores nos padrões atuais de tecnologia.
- **Portabilidade:** Este requisito não funcional foi escolhido devido a necessidade de uso do sistema tanto no ambiente de escritório quanto pelos técnicos e engenheiros nas áreas da mineradora.
- **Critérios de aceite:**
  - As telas do sistema devem apresentar facilidade de navegação e os objetos devem se adaptar de acordo com a resolução identificada.
  - O sistema deve se manter com o mesmo padrão de cores e objetos independente do device utilizado para o acesso.
  - O sistema deve ser compatível com os principais browser do mercado como: Edge, Chrome e Firefox.

### 5.1.2. Casos de uso

Para esta POC foram implementados os casos de uso dos módulos de monitoramento e segurança e comunicação. Os casos de uso escolhidos estão listados abaixo.

Módulo	Caso de uso	Requisito não funcional

Usuários	Autenticar	Segurança / Portabilidade
Monitoramento	Manter sensores	Interoperabilidade / Segurança
	Monitorar sensores	Interoperabilidade
Segurança e comunicação	Executar plano de ação	Interoperabilidade

### 5.1.3. Tecnologias utilizadas

As tecnologias utilizadas na implementação da prova de conceito:

Caso de uso	Tecnologia
Autenticar	ASP NET MVC, ASP NET Web API, Bootstrap, JQuery e IdentityServer
Manter sensores	ASP NET Web API, IdentityServer, SQL Server
Monitorar sensores	ASP NET Web API, IdentityServer , SQL Server
Executar plano de ação	ASP NET Web API, IdentityServer , SQL Server, SendGrid, Twilio

#### 5.1.4. Implantação

Toda a prova de conceito desenvolvida foi implantada em ambiente cloud. O serviço escolhido foi o Azure.

<b>Implementação</b>	<b>Recurso de implantação</b>
Front-end	Azure AppServices
Back-end (todos as APIs)	Azure AppServices
Banco de dados	Azure SQL

Os links para acesso a plataforma encontram-se no apêndice desse documento.

#### 5.2 Interfaces / APIs

<b>API</b>	API REST para autenticação	
<b>URL</b>	<a href="https://api-tcc-autenticacao.azurewebsites.net/">https://api-tcc-autenticacao.azurewebsites.net/</a>	
<b>Método</b>	POST	
<b>Headers</b>	<b>Content-Type</b>	application/x-www-form-urlencoded
<b>Parâmetros</b>	<b>cliente_id</b>	tcc_auth_client
	<b>cliente_secret</b>	secret



	<b>grant_type</b>	client_credentials
<b>Exemplo</b>		
<p>POST /connect/token HTTP/1.1</p> <p>Host: https://api-tcc-autenticacao.azurewebsites.net</p> <p>Content-Type: application/x-www-form-urlencoded</p> <p>client_id=tcc_auth_client&amp;client_secret=secret&amp;grant_type=client_credentials</p>		

<b>API</b>	API REST para informe de atividade do sensor	
<b>URL</b>	https://api-tcc-monitoramento.azurewebsites.net/api/atividades	
<b>Método</b>	POST	
<b>Headers</b>	<b>Authorization</b>	Bearer {access_token}
<b>Parâmetros</b>	<b>rotuloSensor</b>	x10 / x11 / x12
	<b>tipo</b>	Tremor / ruido
	<b>intensidade</b>	{valor de 0 à 100}
<b>Exemplo</b>		
<p>POST /api/atividades HTTP/1.1</p> <p>Host: https://api-tcc-monitoramento.azurewebsites.net</p>		

```
Content-Type: application/x-www-form-urlencoded
```

```
Authorization: Bearer {access_token}
```

```
rotuloSensor=x12&tipo=tremor&intensidade=60
```

## 6. Avaliação da Arquitetura

### 6.1. Análise das abordagens arquiteturais

A arquitetura proposta contempla uma variedade de componentes. Cada componente possui um conjunto de tecnologias e características de implantação. Apesar da utilização de vários componentes proprietários e de uma infraestrutura em nuvem, toda a implementação foi feita de forma que seja o mais independente de plataforma possível. Espera-se que cada componente seja factível, necessário, priorizável, não ambíguo e verificável.

### 6.2. Cenários

Os atributos identificados estão relacionados aos requisitos listados na seção [5.1.1](#): segurança, interoperabilidade e portabilidade.

**Cenário 1:** Ao realizar o acesso a uma URL ou página do sistema protegida, o sistema deve garantir que apenas usuários autenticados mediante apresentação de login e senha, possam acessar este recurso. O sistema deve redirecionar o usuário para a tela de autenticação na tentativa de acesso às páginas protegidas sem autenticação prévia. O sistema garantirá que as páginas públicas possam ser acessadas sem necessidade de autenticação. Esta será a garantia de que o requisito não funcional de segurança foi satisfeito.

**Cenário 2:** Ao realizar o acesso a uma API protegida, o sistema deve garantir que apenas usuários autenticados mediante apresentação de token, possam acessar este recurso. O sistema deve retornar um erro "401 - Não autorizado" na tentativa de acesso as APIs protegidas sem autenticação prévia. Esta será a garantia de que o requisito não funcional de segurança foi satisfeito.

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

**Cenário 4:** O sistema deve conseguir comunicar-se entre seus módulos. No caso de autenticação, o sistema deverá conseguir acessar o serviço de autenticação e obter uma resposta válida. Essa comunicação bem-sucedida, juntamente com o cenário 5, são as garantias de que o requisito não-funcional de interoperabilidade e segurança foram atendidos.

**Cenário 5:** O sistema deve conseguir se comunicar com outros sistemas, inclusive de outras tecnologias. Essa comunicação bem-sucedida, juntamente com o cenário 4, são as garantias de que o requisito não funcional de interoperabilidade foi satisfeito.

Na priorização foi utilizado o método de árvore de utilidades reduzida com prioridades. Os atributos foram categorizados de acordo os requisitos a que estão relacionados, e classificados em função de sua importância e complexidade (considerando a percepção de negócio e arquitetura). As duas variáveis de priorização são "Importância" (IMP) e "Complexidade" (COM). As classificações possíveis são “Alta”, “Média” e “Baixa”.

<b>Categoria</b>	<b>Atributo de qualidade</b>	<b>Cenário</b>	<b>IMP</b>	<b>COM</b>
Confidencialidade	Segurança	Acesso a páginas protegidas mediante autenticação por login e senha.	Alta	Alta
		Acesso a APIs protegidas mediante autenticação por token.	Alta	Alta
Funcionalidade	Acessibilidade	O sistema deve suportar ambientes web responsivos e ambientes móveis.	Alta	Média

Compatibilidade	Interoperabilidade	O sistema deve comunicar-se entre seus módulos	Alta	Alta
		O sistema deve conseguir se comunicar com outros sistemas	Alta	Alta

### 6.3. Avaliação

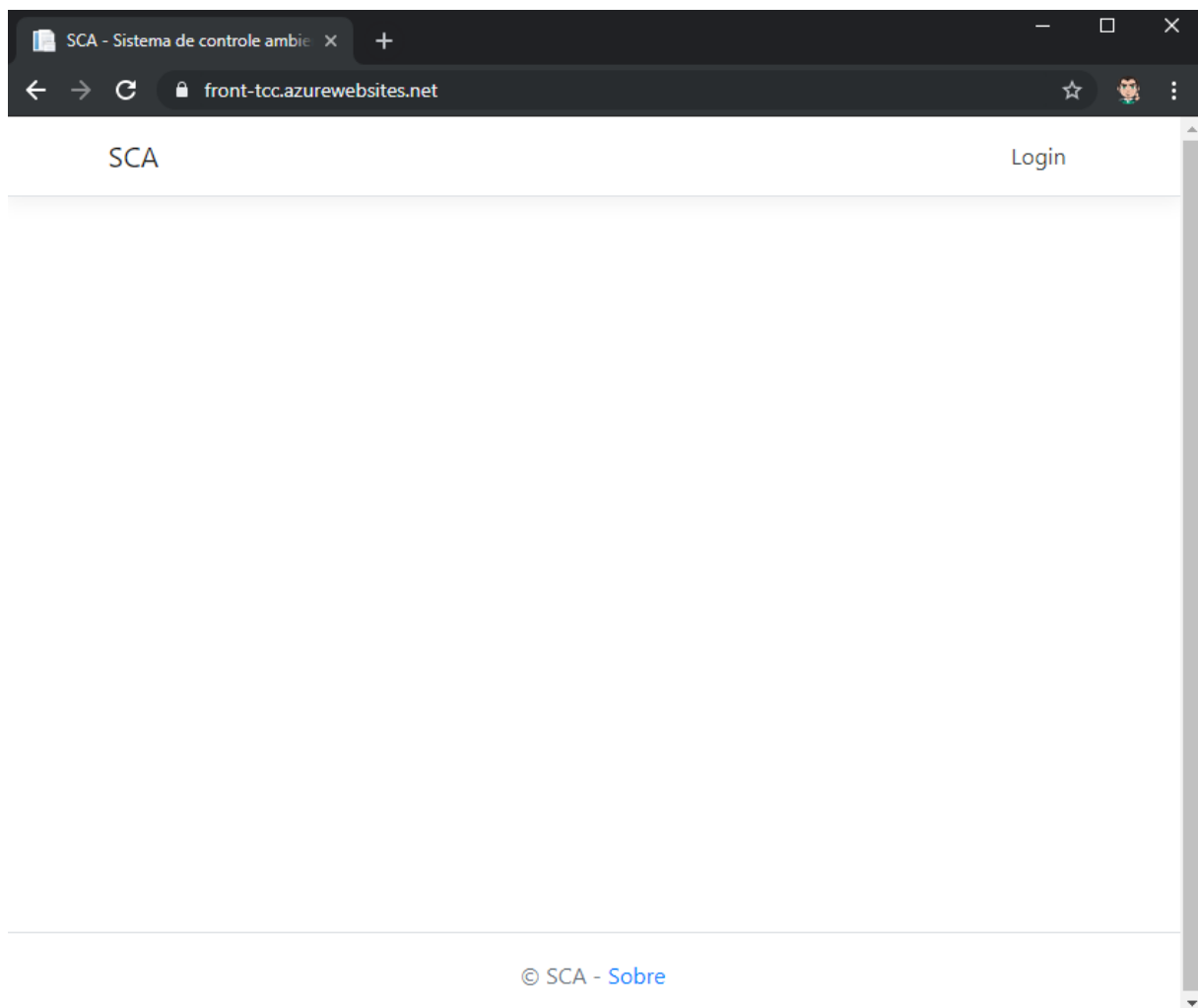
Processo de avaliação dos cenários identificados no item anterior. O objetivo aqui é determinar os riscos, não riscos, pontos de sensibilidade, trade-off's e evidenciar o atendimento aos requisitos de qualidade

#### Cenário 1, 2:

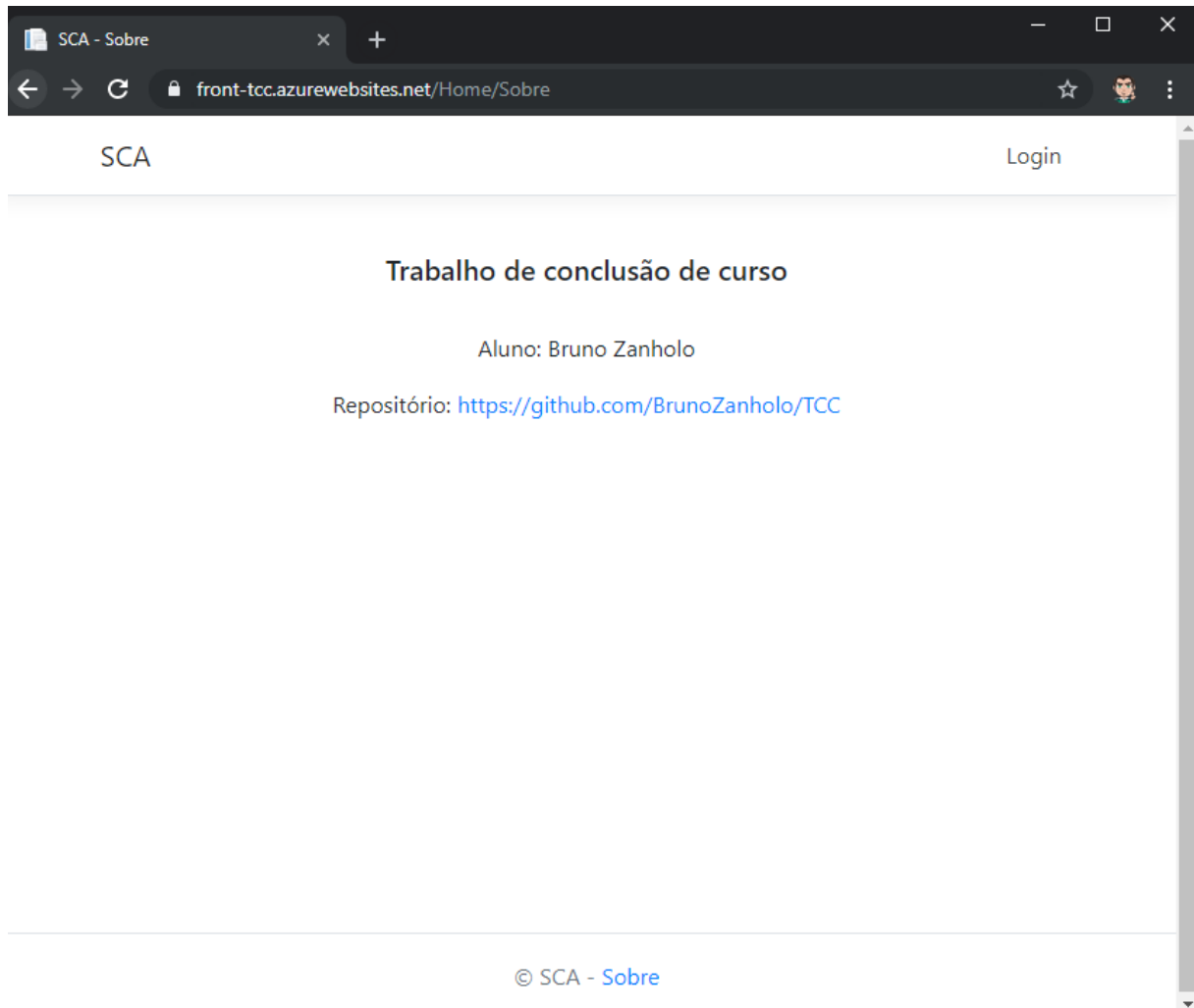
<b>Atributo de qualidade</b>	Segurança.
<b>Requisito de qualidade</b>	O sistema deve apresentar alto padrão de segurança.
<b>Preocupação</b>	
Impossibilitar o acesso a recursos protegidos do sistema.	
<b>Cenário(s)</b>	
Cenário 1 e Cenário 2.	
<b>Ambiente</b>	
Sistema com operação normal.	
<b>Estímulo</b>	
Usuário tentando acessar um recurso protegido do sistema sem autenticação prévia.	
<b>Mecanismo</b>	
Criar um mecanismo de autenticação centralizada para validação de credenciais (usuário/senha ou tokens) para acessar recursos protegidos.	
<b>Medida de Resposta</b>	
O usuário deve ser redirecionado para tela de autenticação caso o recurso seja o site ou deve receber a resposta “401 – Não autorizado” se for a API do sistema.	

Considerações sobre a arquitetura	
<b>Riscos</b>	O gerenciamento de autenticação e autorização são pontos críticos para a segurança. Falhas neste recurso poderiam provocar vazamento de credenciais e informações indevidas. A utilização de um recurso centralizado de autenticação é uma maneira simples e rápida de proteger todos os serviços que compõe o sistema.
<b>Pontos de sensibilidade</b>	Servidor de aplicação operando em modo HTTP.
<b>Trade off</b>	Não há.

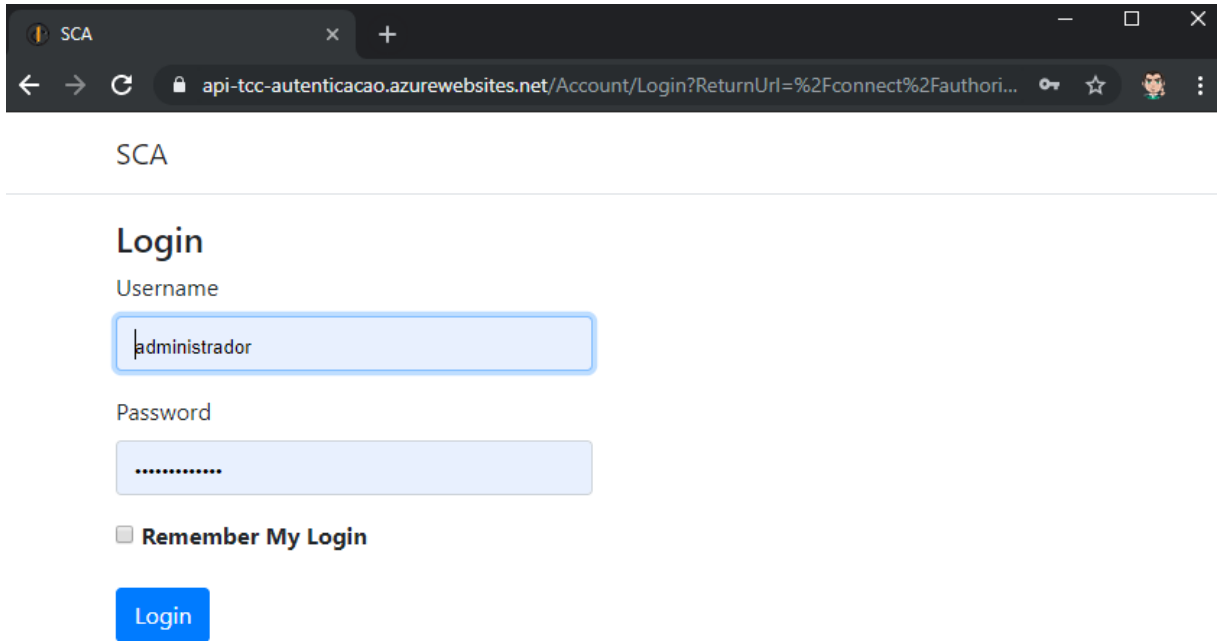
### Evidências do cenário 1:



*Figura 9 - Usuário não autenticado acessa a página inicial do sistema e nenhum menu é apresentado, exceto o menu "Login"*



*Figura 10 - Usuário não autenticado pode acessar páginas não protegidas (Sobre).*



The screenshot shows a web browser window with a single tab titled 'SCA'. The address bar displays the URL 'api-tcc-autenticacao.azurewebsites.net/Account/Login?ReturnUrl=%2Fconnect%2Fauthori...'. The page content includes the title 'SCA' followed by a horizontal line. Below this is the 'Login' section, which contains a 'Username' label, a text input field with 'administrador', a 'Password' label, a password input field with masked characters, a checkbox labeled 'Remember My Login', and a blue 'Login' button.

SCA

## Login

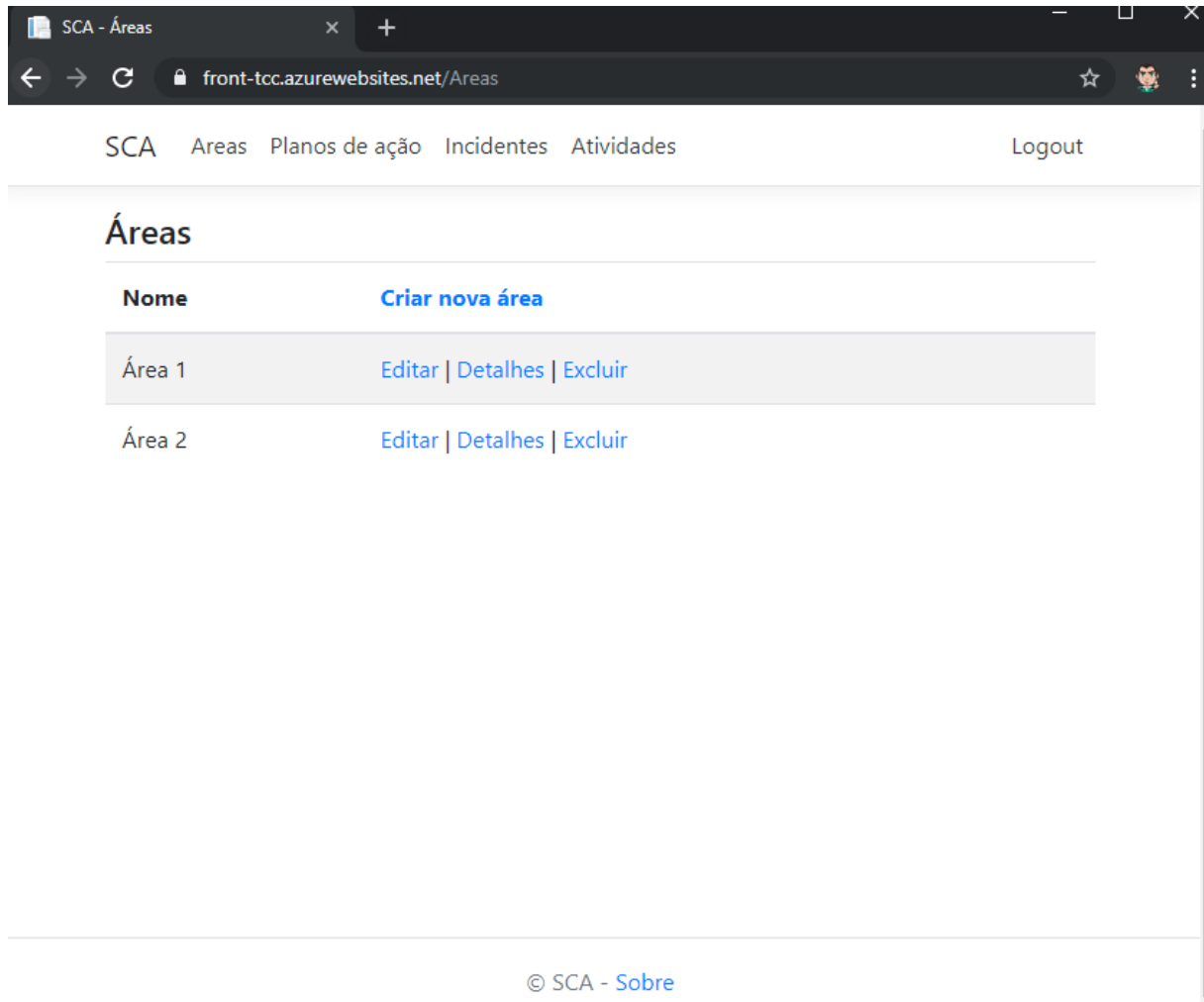
Username

Password

☐ Remember My Login

Login

*Figura 11 - Tentativa de acesso a página protegida "Áreas" direciona o usuário não autenticado para a tela de login.*



*Figura 12 - A autenticação possibilita o usuário acessar a página protegida (Áreas) e visualizar o menu para acesso aos demais recursos protegidos.*

## **Evidências do cenário 2:**



The screenshot shows the Postman interface with a POST request to `https://api-tcc-autenticacao.azurewebsites.net/connect/token`. The response is displayed in the 'JSON' tab, showing a successful token acquisition. The response body is a JSON object containing an 'access\_token' (a long alphanumeric string), 'token\_type' set to 'Bearer', and 'scope' set to 'openid profile tcc\_auth'.

*Figura 14 - Requisição de autenticação pela API de Autenticação retorna token de acesso.*

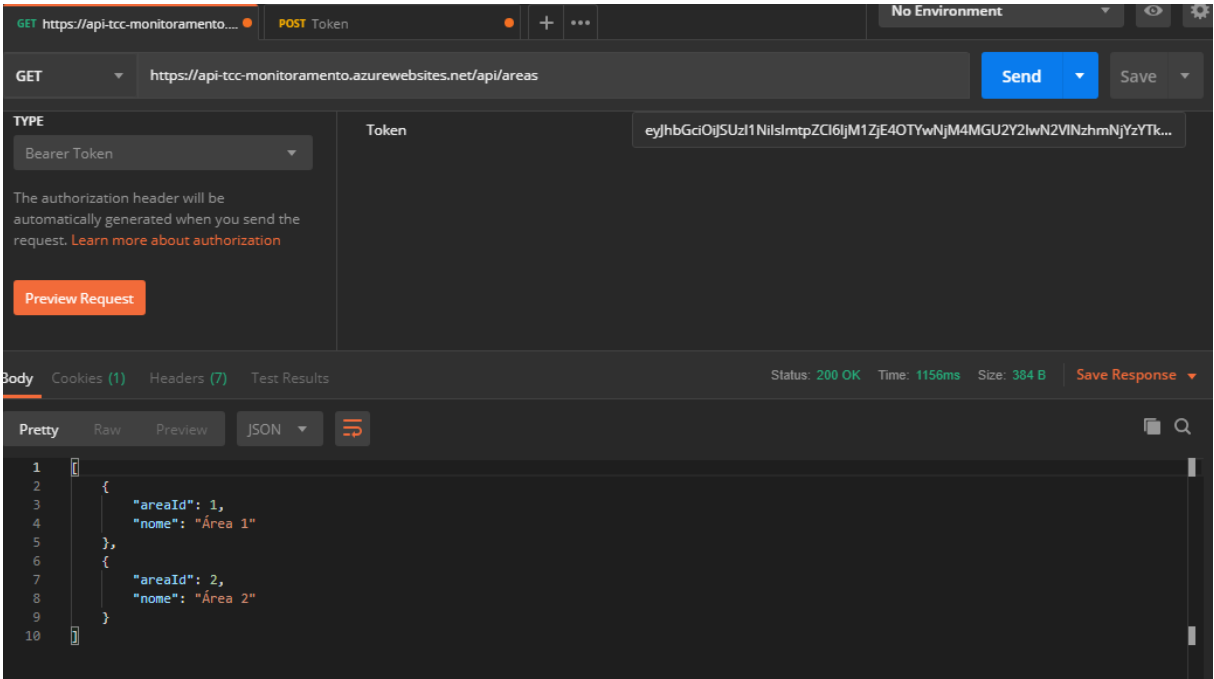


Figura 15 - Acesso a API de Monitoramento concedido com a apresentação do token de autenticação.

Cenário 3:

Atributo de qualidade	Portabilidade e usabilidade.
Requisito de qualidade	O sistema deve suportar qualquer device e ambientes web.
Preocupação	
O sistema deve se adaptar a interfaces de diversos devices e tamanhos sem perda de funcionalidade e sem causar impactos na qualidade de navegação.	
Cenário(s)	
Cenário 3.	
Ambiente	
Sistema com operação normal.	
Estímulo	
Usuário acessando tela de detalhes da área de risco.	
Mecanismo	
Criação de telas utilizando mecanismos de design responsivos e ajustáveis, movimentando os componentes para que caibam em dispositivos diferentes.	

Medida de Resposta	
O sistema deve se adaptar a resoluções de tela dos diversos dispositivos, sem perder funcionalidades.	
Considerações sobre a arquitetura	
<b>Riscos</b>	A experiência de navegação pode ser prejudicada pela qualidade da rede que está provendo o acesso. Além disso, resoluções extremamente pequenas ou dispositivos muito antigos poderão causar comportamentos indesejáveis de componentes (porém sem a perda das funcionalidades).
<b>Pontos de sensibilidade</b>	Não há.
<b>Trade off</b>	Não há.

### Evidências do cenário 3:

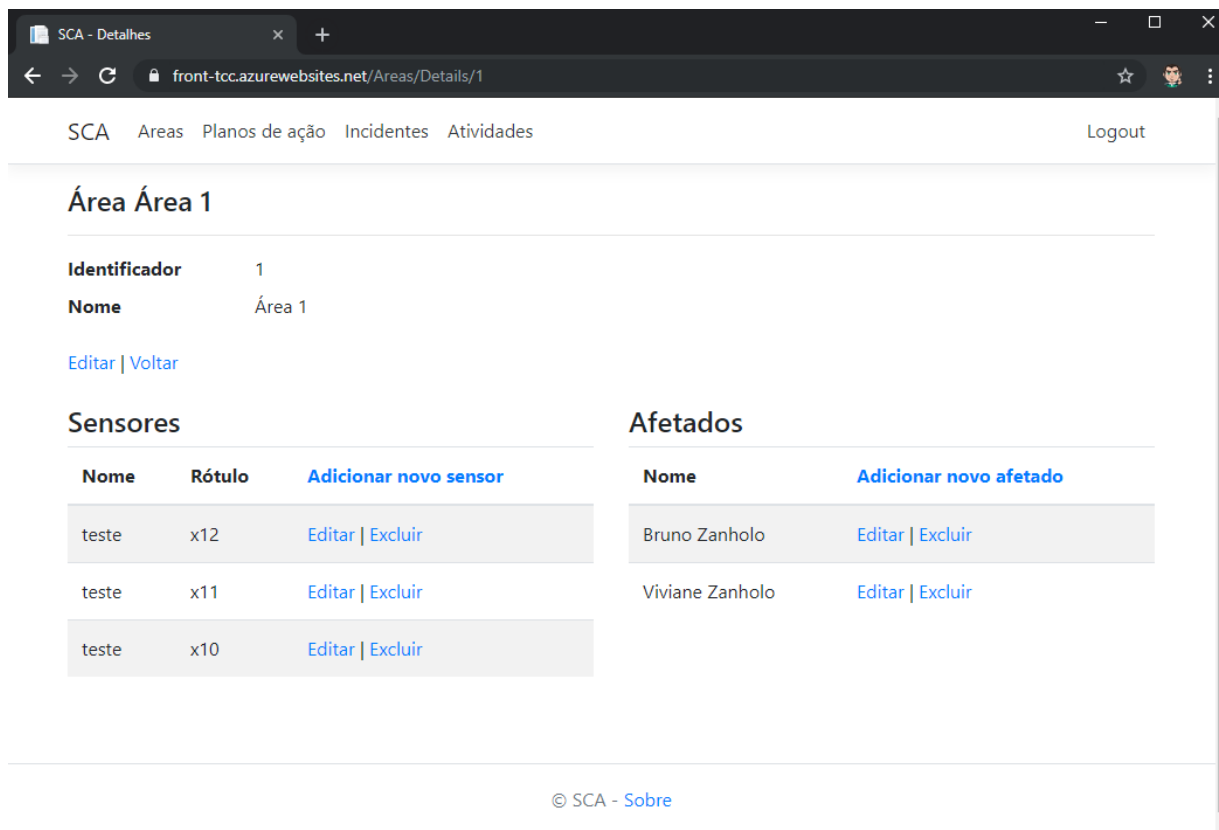
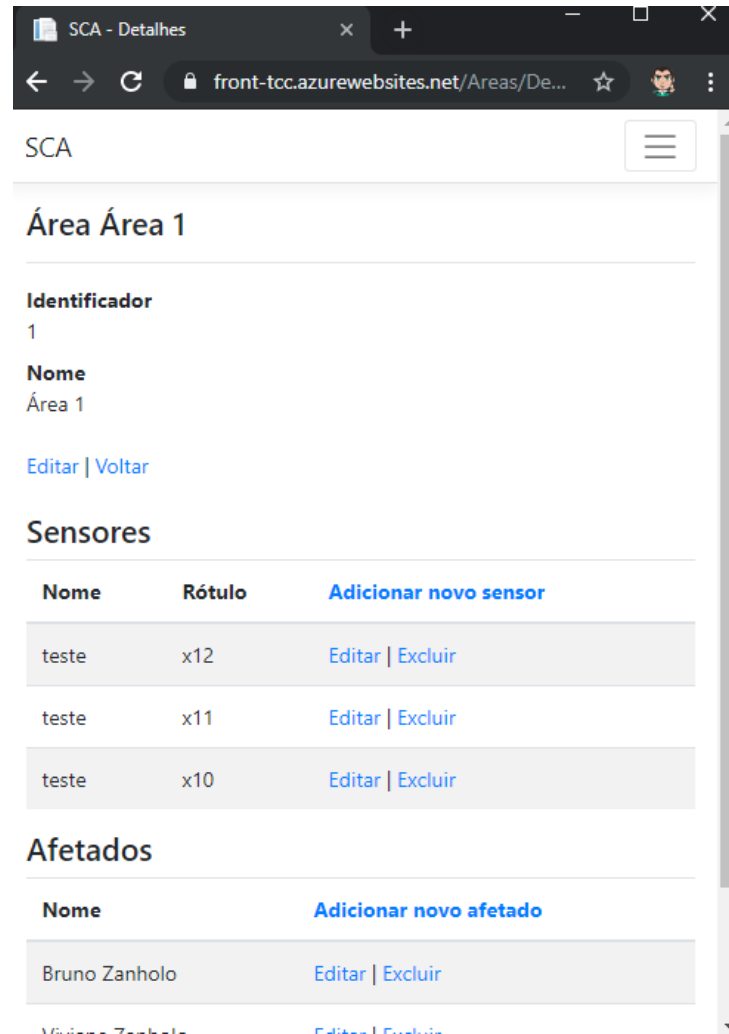


Figura 16 - Sistema com a apresentação de seus componentes na resolução de tela ideal.



*Figura 17 - Sistema apresentado em tamanho de tela reduzido sem comprometimento da usabilidade por meio da responsabilidade dos componentes.*



Figura 18 - Sistema apresentado em outro device (celular) sem comprometimento da usabilidade por meio da responsabilidade dos componentes.

#### Cenário 4 e 5:

<b>Atributo de qualidade</b>	Interoperabilidade
<b>Requisito de qualidade</b>	O sistema deve conseguir se comunicar com seus serviços (APIs) e com outros sistemas.
<b>Preocupação</b>	
O sistema deve conseguir se comunicar com suas APIs e com outros sistemas, inclusive de outras tecnologias.	
<b>Cenário(s)</b>	
Cenário 4 e cenário 5.	
<b>Ambiente</b>	
Sistema com operação normal.	

<b>Estímulo</b>	
API de monitoramento recebendo uma atividade de um sensor.	
<b>Mecanismo</b>	
Criação de API REST para notificação da atividade do sensor por um mecanismo terceiro (broker IoT).	
<b>Medida de Resposta</b>	
O sistema deve aceitar a requisição, processá-la e responder com sucesso o requisitor.	
<b>Considerações sobre a arquitetura</b>	
<b>Riscos</b>	Comunicações via rede estão sujeitas à instabilidade e indisponibilidade do ambiente a que se deseja acessar.
<b>Pontos de sensibilidade</b>	Não há.
<b>Trade off</b>	Não há.

#### Evidências do cenário 4:

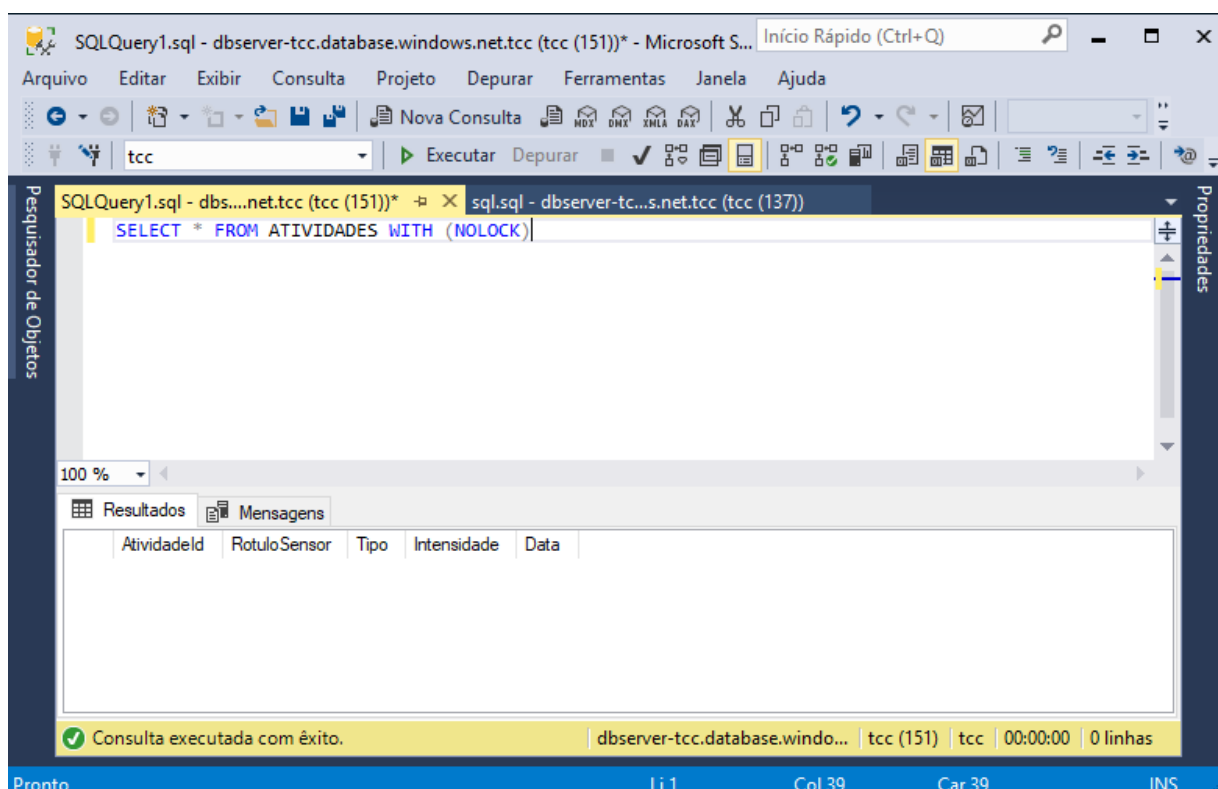


Figura 19 - Nenhuma atividade de sensor no banco de dados do sistema.

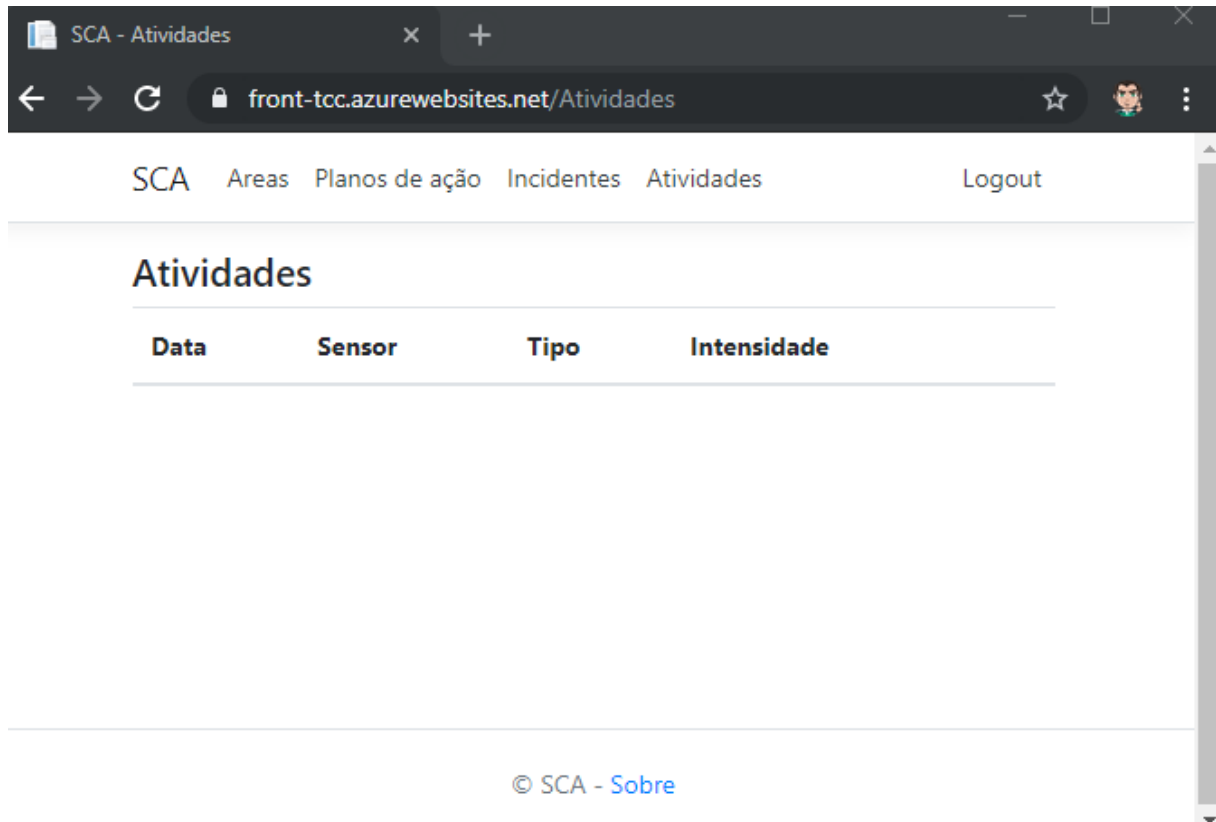


Figura 20 - Nenhuma atividade de sensor apresentada pelo sistema.

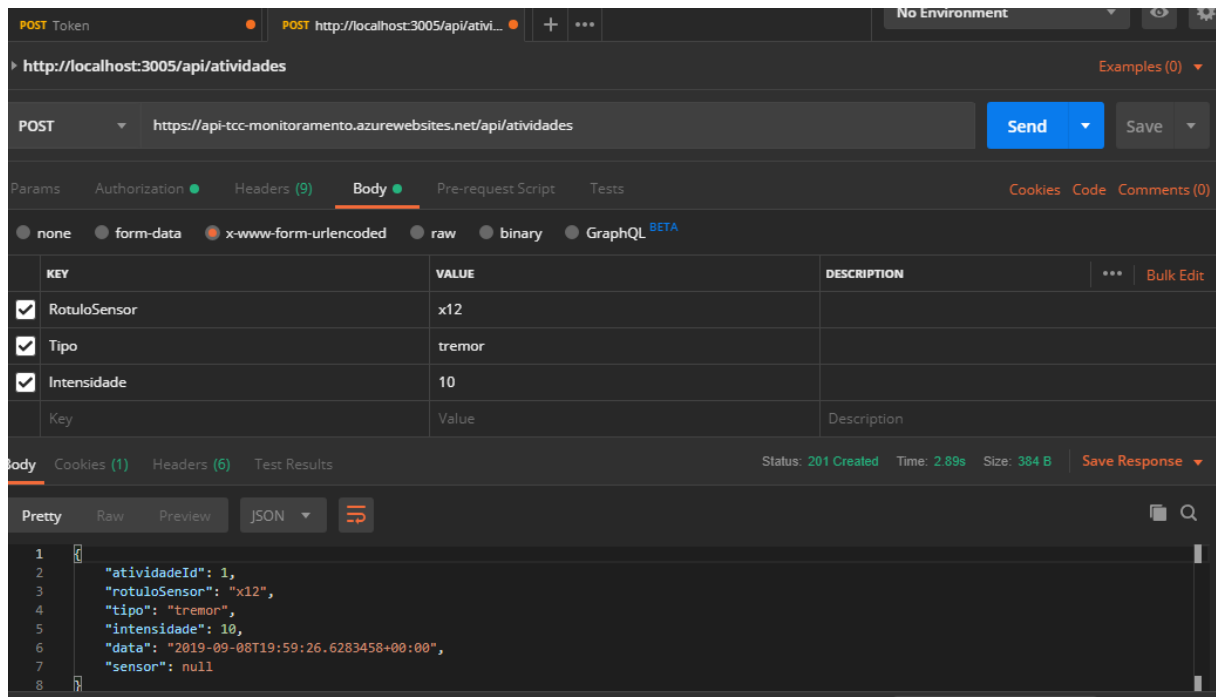
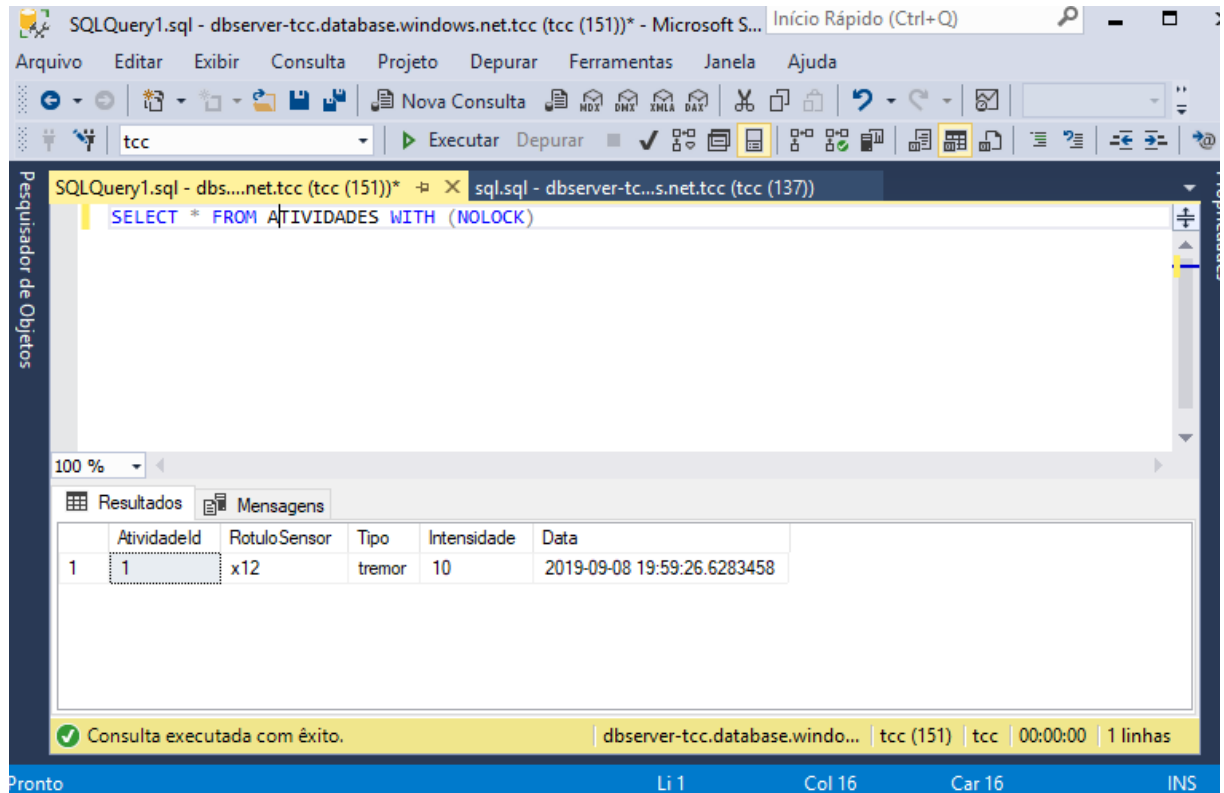
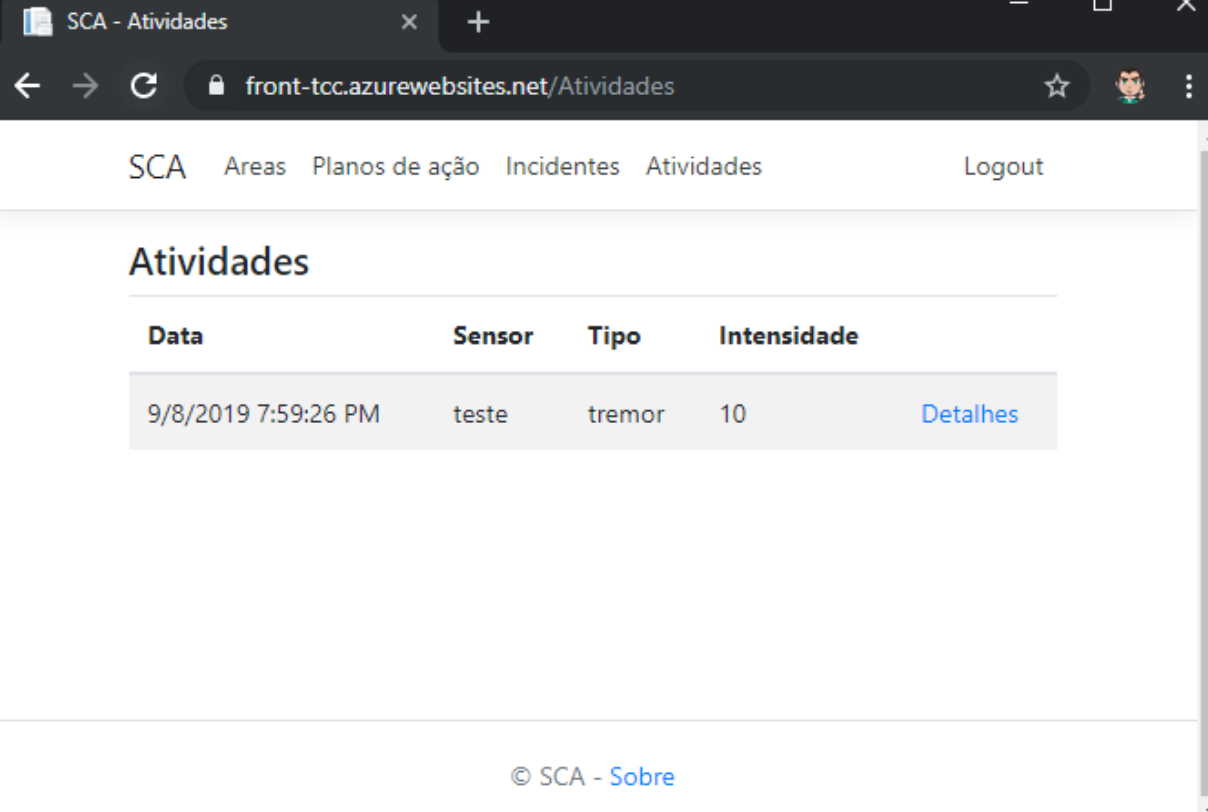


Figura 21 - Notificação da atividade do sensor pela API de monitoramento.



*Figura 22 - Registro da atividade no banco de dados após a inclusão pela API de monitoramento.*





The screenshot shows a web browser window with the address bar displaying 'front-tcc.azurewebsites.net/Atividades'. The page has a navigation bar with links: SCA, Areas, Planos de ação, Incidentes, Atividades, and a Logout button. Below the navigation bar, the title 'Atividades' is displayed. A table with four columns (Data, Sensor, Tipo, Intensidade) contains one row of data. A 'Detalhes' link is present next to the data row. The footer shows '© SCA - Sobre'.

Data	Sensor	Tipo	Intensidade
9/8/2019 7:59:26 PM	teste	tremor	10

*Figura 23 - Registro da atividade no sistema após a inclusão pela API de monitoramento.*

**Evidencias do cenário 5:**

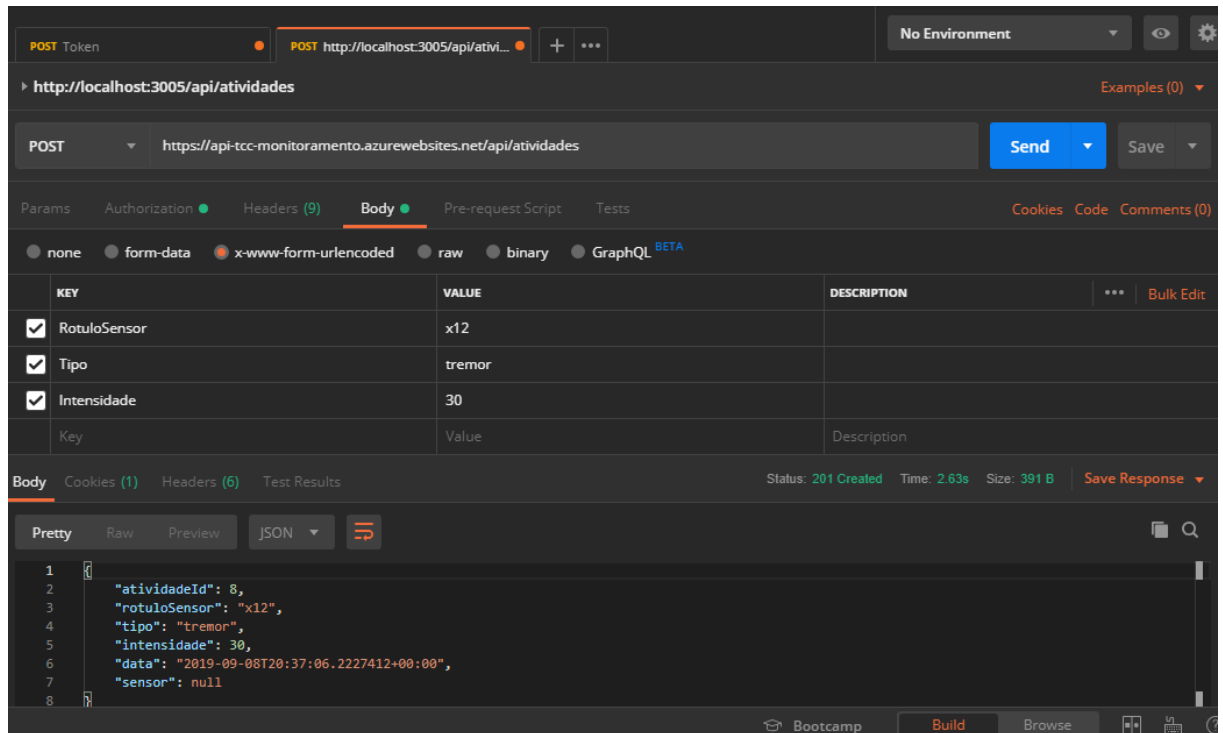


Figura 24 - Atividade gerada por sensor enviada pela API de Monitoramento.

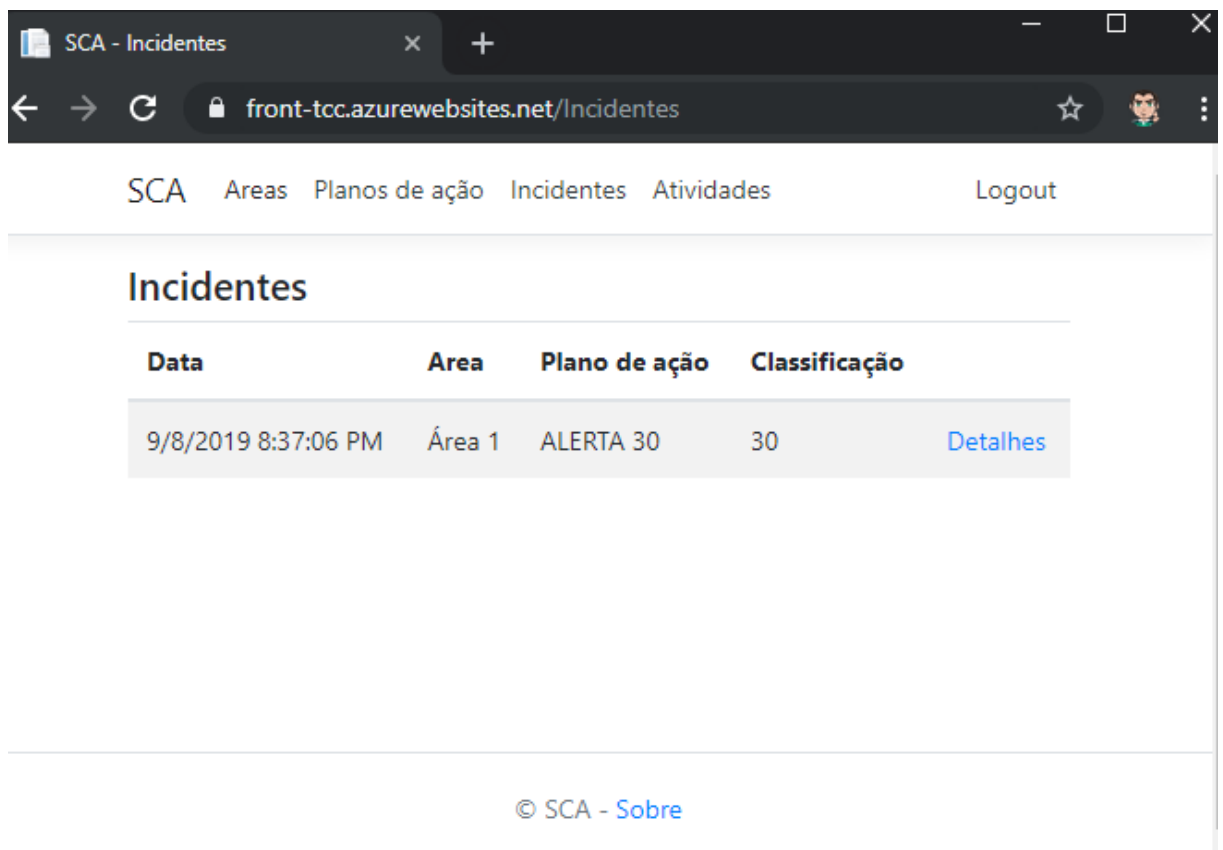
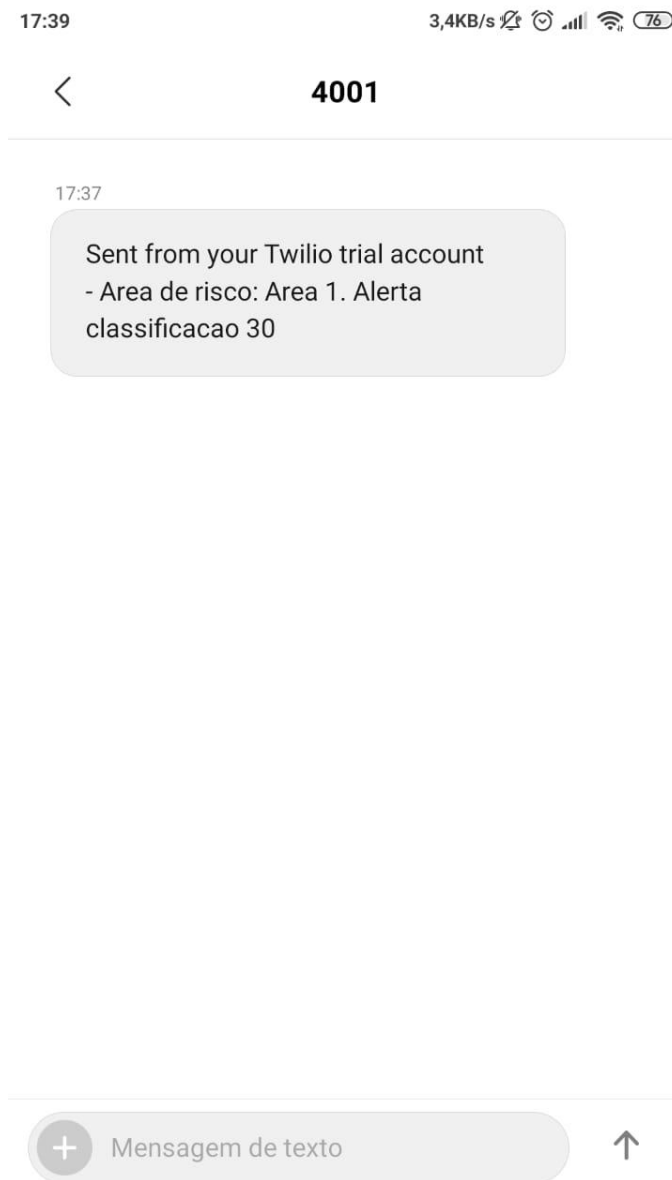


Figura 25 - Incidente gerado automaticamente pela atividade do sensor.



*Figura 26 - SMS enviado pelo Twilio ao afetado cadastrado pelo plano de ação ativado pelo incidente.*

#### 6.4. Resultado

Dados os atributos de qualidade e realizada a validação arquitetural, nota-se que as necessidades propostas foram atendidas, porém há margem para melhora. A validação permitiu executar cenários para definir os pontos fortes e fracos da proposta. Os seguintes requisitos de qualidade foram validados:

<b>Requisitos não funcionais</b>	<b>Testado</b>	<b>Homologado</b>
Segurança - O sistema deve apresentar alto padrão de segurança.	SIM	SIM
Portabilidade - O sistema deve suportar qualquer device e ambientes web.	SIM	SIM
Interoperabilidade - O sistema deve conseguir se comunicar com seus serviços (APIs) e com outros sistemas independente da tecnologia utilizada pelo serviço.	SIM	SIM

Do ponto de vista da construção de código, os componentes foram divididos em camadas. O front-end em ASP NET MVC Core com a utilização do Bootstrap como framework CSS agiliza o desenvolvimento de interfaces amigáveis responsivas, o que poupa bastante tempo em prototipações e interfaces mais simples. Para recursos visuais mais complexos, uma equipe especializada em UX/UI e desenvolvimento de front-end atingirá melhores resultados.

As API's em .NET Core são divididas em camadas lógicas, separando o que é exposição de dados, regras de negócio e acesso a repositórios de dados (sejam bancos de dados ou serviços de terceiros). Em casos que a velocidade no desenvolvimento e a facilidade de mudança é necessária, o ORM Entity Framework Core cumpre bem as necessidades. Porém, quando houver necessidade de performance, há de se preferir a utilização de ADO.NET para acesso a banco.

Do ponto de vista das integrações, as comunicações realizadas com sistemas terceiros são bem-sucedidas, independente da tecnologia. Porém, a comunicação é muito dependente da implementação realizada em .NET. Há uma margem para melhora incluindo um ESB – Enterprise Service Bus - para gerir essas comunicações, removendo da camada .NET a responsabilidade por gerir essa miscelânea de tecnologias de comunicação.

No que diz respeito a utilização de componentes prontos, fornecidos pela Azure, existem também os prós e os contras. A principal vantagem é utilizar componentes

amplamente testados e validados, e no caso da implantação na nuvem da própria empresa há um ganho na facilidade da realização de integrações com outros serviços. Há também ganho no tempo de desenvolvimento por utilizar soluções integráveis simples. Há também a desvantagem de estar preso a um fornecedor, visto que as chamadas às API's de notificação por e-mail e SMS são feitas por um serviço proprietário (Twilio e Sendgrid). A mudança para outro produto implicaria na reescrita de uma parte do código.

Do ponto de vista da implantação, utilizar a solução em nuvem da Azure para prover os serviços traz uma facilidade de configuração, escalabilidade e integração entre componentes que é bastante interessante. A maioria das configurações podem ser feitas por interface gráfica, sendo bem intuitivas e com uma excelente documentação. Além disso, não há a necessidade de contar com uma infraestrutura robusta, adquirir servidores, links de internet, firewalls e todas as equipes para cuidar desse tipo de coisa. O contra nesse caso é o custo. Há ainda uma margem para evolução com a utilização de contêineres para implantação das API's.

Provisionar um ambiente de DevOps nessa arquitetura é bem interessante, pois agiliza e facilita o processo de validação de novas implementações, auxiliando no cumprimento dos requisitos não funcionais de testabilidade e manutenibilidade.

## **7. Conclusão**

Este trabalho apresentou um projeto arquitetural para uma plataforma de gestão e controle ambiental voltada para atividades de negócio de uma empresa mineração. Entende-se que os objetivos foram atingidos, há margem para melhora se houvesse mais tempo para a análise, todavia mudanças poderão ser adicionadas ao projeto sem grande dificuldade.

## REFERÊNCIAS

SOMMERVILLE, Ian. Engenharia de Software. 9ª edição. São Paulo: Pearson, 2011.

Wikipedia, Rompimento de barragem em Brumadinho. Disponível em:

<[https://pt.wikipedia.org/wiki/Rompimento\\_de\\_barragem\\_em\\_Brumadinho](https://pt.wikipedia.org/wiki/Rompimento_de_barragem_em_Brumadinho)>. Acesso em: 26 de julho de 2019.

Bootstrap. Bootstrap Documentation. Disponível em:

<<https://getbootstrap.com/docs/4.3/getting-started/introduction/>>. Acesso em: 18 de agosto de 2019.

Azure, Azure Documentation. Disponível em: <<https://docs.microsoft.com/pt-br/azure/>>.

Acesso em: 19 de agosto de 2019.

## APÊNDICES

URL do front-end na Azure: <https://front-tcc.azurewebsites.net/>

URL da API back-end de autenticação: <https://api-tcc-autenticacao.azurewebsites.net/>

URL da API back-end de monitoramento no Azure: <https://api-tcc-monitoramento.azurewebsites.net/>

URL da API back-end de segurança e comunicação no Azure: <https://api-tcc-seguranca.azurewebsites.net/>

URL de apresentação da POC no Youtube:

URL do repositório no GitHub: <https://github.com/BrunoZanholo/TCC>