

# **SISTEMA DE GESTÃO DA QUALIDADE**

## **Uma proposta de arquitetura**

**Autor: Luiz Augusto Kill Bernardo**

**Orientadores: Luiz Alberto / Will Machado**



# SISTEMA DE GESTÃO DA QUALIDADE

## Proposta

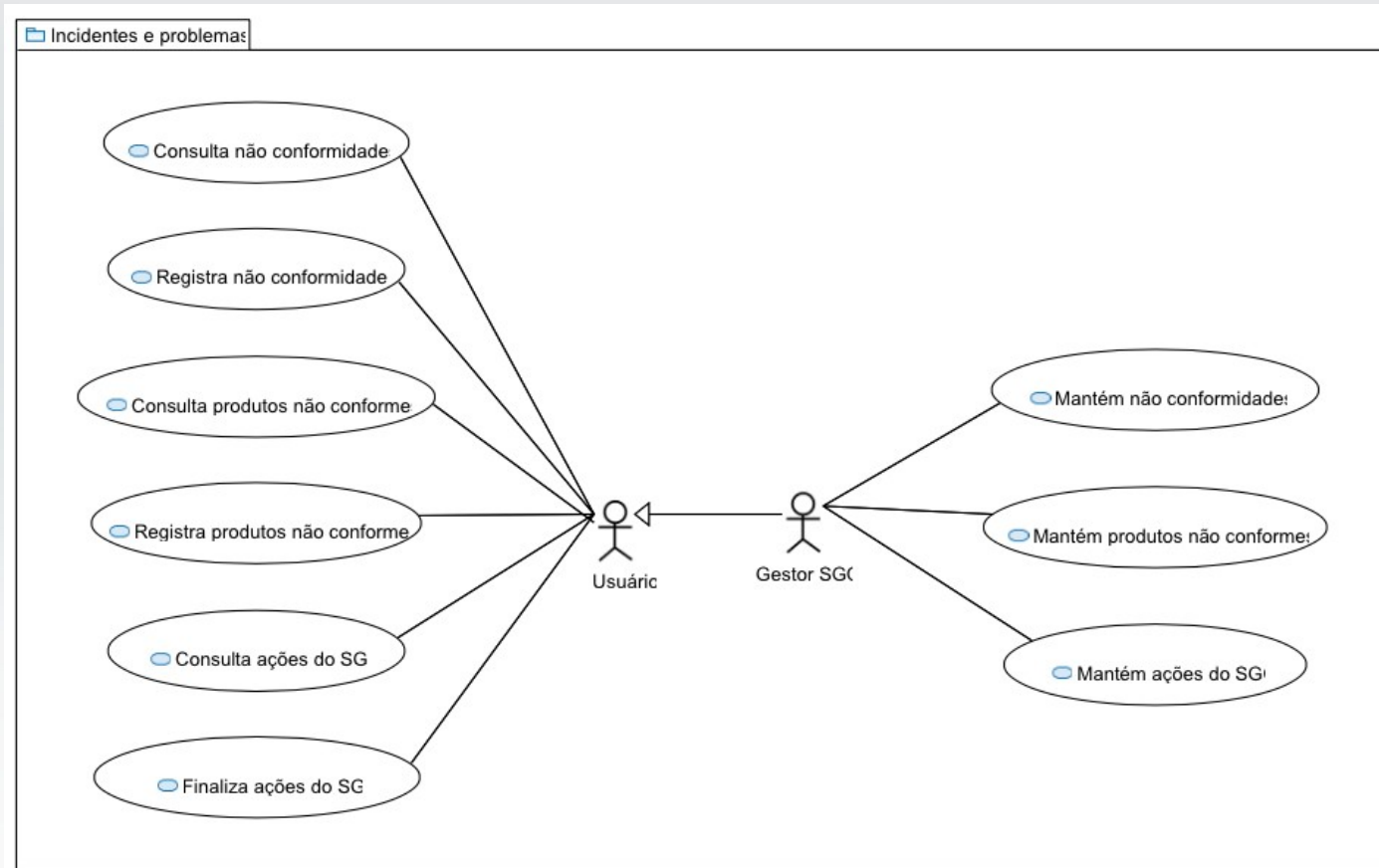
O objetivo geral deste projeto foi apresentar um protótipo arquitetural de uma aplicação de gestão da qualidade de uma empresa do setor automotivo. Seus objetivos foram:

- Construir módulos de incidentes e problemas / controle de processos / divulgação e transparência;
- *Descrever* módulos de BI e de relatórios operacionais;
- Construir integrações com catálogo de norma externo e sistema externo de consultoria;
- Disponibilizar ferramentas de monitoramento da solução.



# SISTEMA DE GESTÃO DA QUALIDADE

## Requisitos Funcionais - Diagrama de casos de uso ou tabela



# SISTEMA DE GESTÃO DA QUALIDADE

## Requisitos Não Funcionais

**Acessibilidade** – A aplicação deve ser responsiva

**Desempenho** – A aplicação deve apresentar performance elevada

**Disponibilidade** – A aplicação deve ser resiliente a falhas

**Interoperabilidade** – A aplicação deve consumir APIs

**Segurança** – A aplicação deve proteger dados e funcionalidades

**Disponibilidade** – A aplicação deve estar disponível 24/7

**Escalabilidade** – Deve possibilitar o escalonamento horizontal



# SISTEMA DE GESTÃO DA QUALIDADE

## Restrições de projeto

Angular	PWA
Spring Boot	MariaDB
Microserviços	Netflix OSS
Stateless	(Zuul, Hystrix, Ribbon, Eureka)
Responsiva	ELK
RESTful	MS Integration Services
JSON	IBM Cognos
JWT	MS SQL Reporting Services



# SISTEMA DE GESTÃO DA QUALIDADE

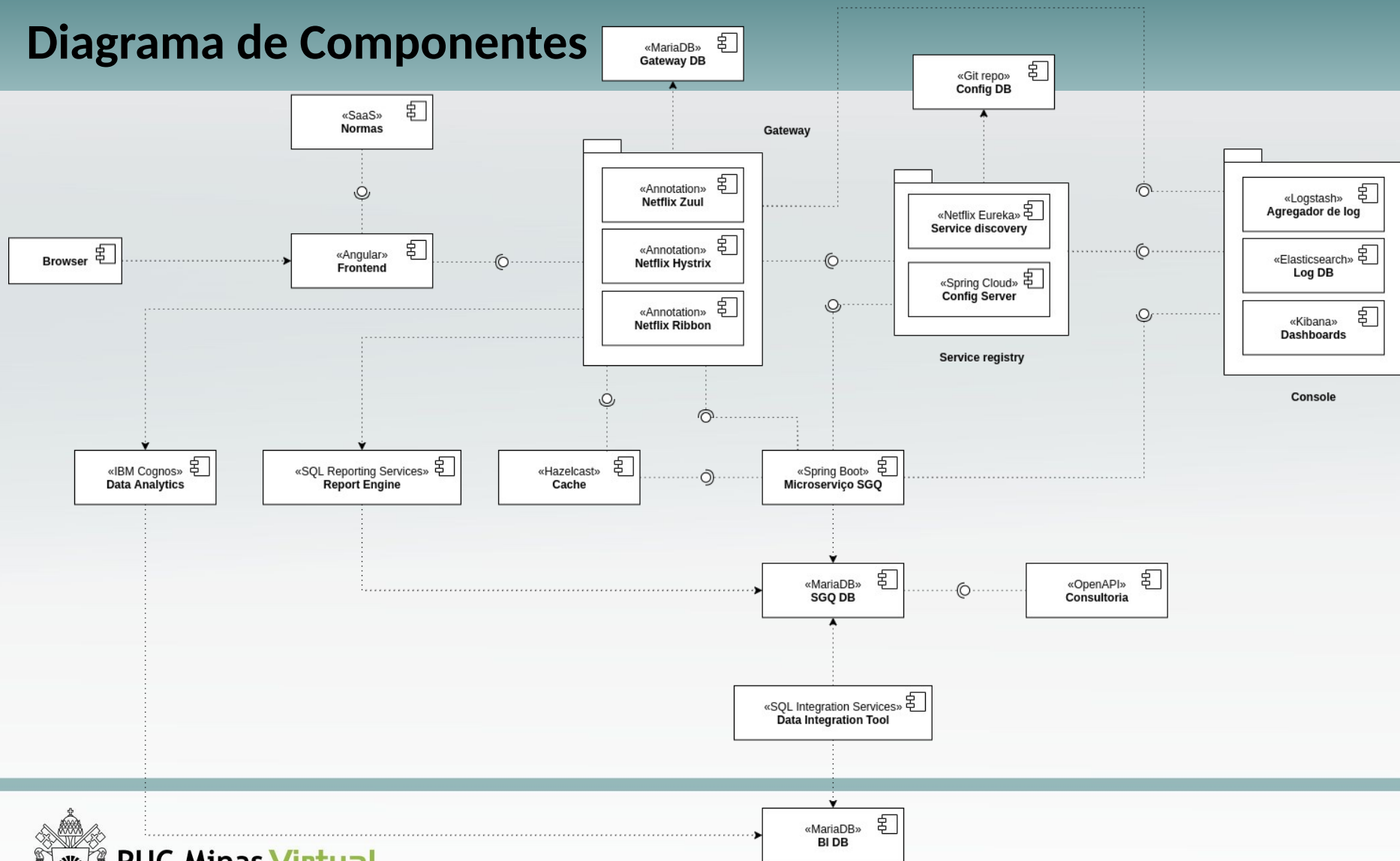
## Mecanismos arquiteturais

Mecanismo de análise	Mecanismo de implementação
Comunicação entre processos	Chamadas HTTP
Integrações com sistemas externos	APIs Restful JSON
Log	Logstash
Monitoramento	Elasticsearch + Kibana
Configuração	Spring Cloud Config
Cache	Hazelcast
Build	Maven, npm, Google jib
Deploy	Docker compose
Front end	Angular Typescript
Versionamento	Git
Autenticação e autorização	Tokens JWT
Alta disponibilidade	Netflix Ribbon (gateway)
Alta disponibilidade	Docker Swarm
Descoberta de serviços	Netflix Eureka
Documentação	Swagger (OpenAPI)
Controle de esquemas de banco	Liquibase
Sistema operacional	Debian



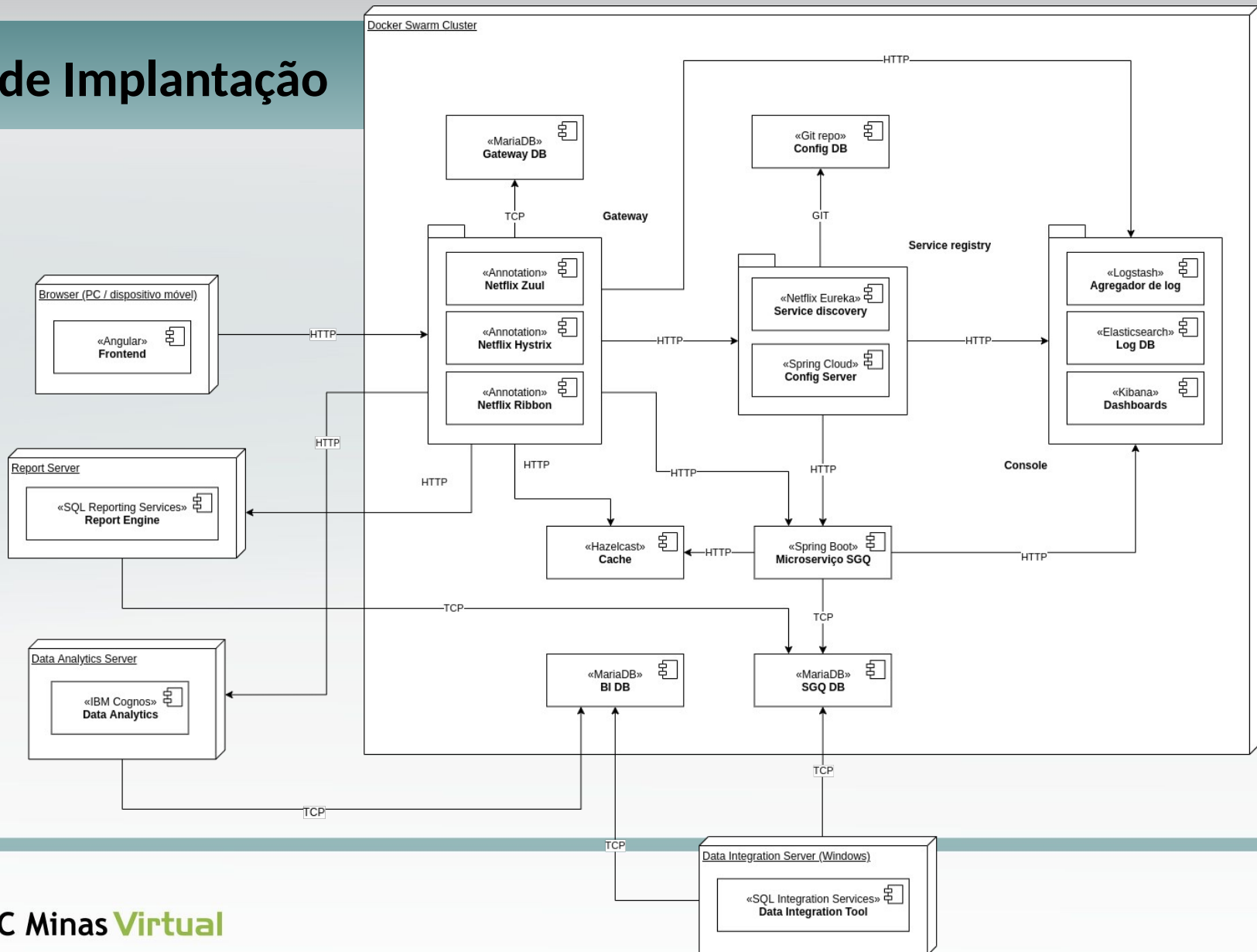
# SISTEMA DE GESTÃO DA QUALIDADE

## Diagrama de Componentes



# SISTEMA DE GESTÃO DA QUALIDADE

## Diagr. de Implantação





# SISTEMA DE GESTÃO DA QUALIDADE

## Apresentação do Protótipo Arquitetural

Vídeo (*screencast*) de apresentação da aplicação web.

Sugestão de gravador de tela: <http://www.screenr.com>

Duração: 2'30"



# SISTEMA DE GESTÃO DA QUALIDADE

## Avaliação da Arquitetura

**Cenário 1:** ao acessar a aplicação a partir de um dispositivo móvel com dimensões de tela reduzidas os componentes do frontend devem se ajustar da melhor forma para que a experiência do usuário não seja prejudicada. Eles devem ter suas proporções redimensionadas, posicionamentos recalculados, textos omitidos e etc. A aplicação deve manter sua identidade visual quando exibe layouts ajustados.

**Cenário 2:** ao executar os casos de uso da aplicação o usuário deve ter uma experiência fluida e intuitiva, de modo que a organização dos componentes e os fluxos das telas não sejam entraves para o seu trabalho. O usuário deve ser capaz de inserir um registro de não conformidade em até 3 minutos.

**Cenário 3:** ao acessar os casos de uso da aplicação o usuário deve ter uma experiência vívida, sem travamentos ou lentidão. O usuário deve visualizar a primeira página da consulta de eventos operacionais em no máximo 3 segundos. Este tempo deve incluir a chamada ao backend, o processamento no microserviço, o tráfego dos dados de retorno, o processamento pelo frontend e a renderização da lista.

**Cenário 4:** ao tentar acessar uma tela privada da aplicação sem autenticação prévia o usuário deve ser redirecionado ao formulário de login. Entretanto, o acesso a telas públicas deve ser permitido sem autenticação. Se uma chamada a uma rota privada do microserviço de SGQ for realizada um erro apropriado deve ser retornado. Chamadas a rotas públicas devem ser permitidas.

**Cenário 5:** ao acessar uma página da aplicação que obtém dados de um serviço externo, esta deve exibir uma mensagem de erro caso o serviço esteja offline. Isso deve ser feito sem que a aplicação como um todo seja impactada. Tão logo o serviço externo seja reestabelecido a página deve exibir os dados requisitados.

**Cenário 6:** a arquitetura da aplicação deve permitir o escalonamento horizontal de qualquer componente quando for percebido um aumento significativo do volume de requisições de usuários. Isso deve ser feito de maneira clara e simples, sem a necessidade de configuração manual de componentes de rede ou de quaisquer outro tipo.



# SISTEMA DE GESTÃO DA QUALIDADE

## Avaliação da Arquitetura

- A empresa deve entender o projeto como uma jornada e não apenas como um cronograma associado a um produto. Deve haver apoio total dos níveis hierárquicos superiores. Gestores devem estar cientes dos riscos, dos desafios e dos ganhos dessa jornada;
- Deve haver no projeto uma equipe madura nas habilidades de devops, pois esta arquitetura demanda muita automação, monitoramento e melhoria contínua. Caso não exista, ela deve ser formada;
- O escopo do projeto deve ser definido cuidadosamente, bem como a equipe que estará envolvida nele. Escopos muito extensos aumentam as chances de insucesso . Indivíduos multiplicadores devem ser selecionados e capacitados.

# SISTEMA DE GESTÃO DA QUALIDADE

## Conclusões

A arquitetura escolhida, baseada em microserviços, mostrou-se plenamente capaz de satisfazer os requisitos não funcionais da aplicação, conforme evidenciado anteriormente. Pôde-se verificar os pontos fortes e os pontos de atenção inerentes a esta arquitetura, bem como as características de várias ferramentas utilizadas na implementação. Em resumo, consideramos que os objetivos do projeto foram atingidos.

