

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

**Pós-graduação *Lato Sensu* em Arquitetura de *Software* Distribuído**

Projeto Integrado

Relatório Técnico

Fleet Seguros

Ricardo Ferreira Ribeiro  
Thiago Ferreira de Aguiar

Belo Horizonte

Março de 2022.

## ● Projeto Integrado – Arquitetura de Software Distribuído

### *Sumário*

Projeto Integrado – Arquitetura de Software Distribuído	<b>3</b>
1. Introdução	4
2. Cronograma do Trabalho	5
3. Especificação Arquitetural da solução	6
3.1 Restrições Arquiteturais	6
3.2 Requisitos Funcionais	6
3.3 Requisitos Não-funcionais	7
3.4 Mecanismos Arquiteturais	8
4. Modelagem Arquitetural	8
4.1 Diagrama de Contexto	8
4.2 Diagrama de Container	9
4.3 Diagrama de Componentes	10
5. Prova de Conceito (PoC)	11
5.1 Integrações entre Componentes	12
5.2 Código da Aplicação	12
6. Avaliação da Arquitetura (ATAM)	15
6.1 Análise das abordagens arquiteturais	15
6.2 Cenários	15
6.3 Evidências da Avaliação	16
6.4 Resultados Obtidos	19
7. Avaliação Crítica dos Resultados	20
8. Conclusão	21
9. Referências	22

## **1. Introdução**

Com o aumento dos preços dos carros novos houve uma maior procura por carros usados, em uma época ainda muito influenciada pela pandemia da Covid-19, as empresas seguradoras de veículos começaram a oferecer melhores condições a seus clientes como preços mais atrativos, facilidades de pagamento e importantes descontos para novos clientes, iniciando assim uma acirrada competição entre as empresas. Outro ponto importante que tem chamado atenção para esse mercado de seguro veicular é o crescimento na locação de veículos devido principalmente a alta dos preços dos carros e passagens aéreas, fechamento de fábricas e diminuição na produção de peças e materiais, uma prova desse crescimento é que as ações das principais locadoras de veículos do país (Localiza, Movida e Unidas) tiveram uma valorização de 47,9%, 34,8% e 57,4%, respectivamente.

Neste contexto apresentamos o sistema Fleets Seguros que é responsável por fazer cotação de seguro veicular e apresenta alguns problemas críticos para os corretores que o utilizam, atualmente o Fleets Seguros faz uma única cotação por vez e o preenchimento dessa cotação é de forma manual o que tem levado a várias queixas quando existe um número significativo de cotações a se fazer (como de uma frota por exemplo), pois pode-se levar horas ou até dias para sua conclusão, sem falar também nos erros causados pelo preenchimento manual do usuário.

Esperamos refazer o sistema propondo uma nova arquitetura e novas tecnologias para que ele possa se tornar um software mais competitivo no mercado, beneficiando assim seu usuário com uma maior agilidade e facilidade no processo de cotação de um veículo individual ou de uma frota veicular e também iremos implementar uma nova interface mais simples, rápida e intuitiva a fim de diminuir possíveis erros manuais.

“O objetivo deste trabalho é apresentar a nova arquitetura do projeto Fleets Seguros e as vantagens obtidas com a mesma.”

“Os objetivos específicos propostos são:

- Diminuir o tempo gasto para se fazer uma cotação veicular;
- Tornar o software mais simples e intuitivo de se usar;
- Descrever a nova arquitetura de forma simples, concisa e objetiva;

## 2. Cronograma do Trabalho

A seguir é apresentado o cronograma proposto para as etapas deste trabalho.

Datas		Atividade / Tarefa	Produto / Resultado
De	Até		
08/03/2022	08/03/2022	1. Definição do escopo inicial do sistema a ser implementado.	Tecnologias definidas.
15/03/2022	15/03/2022	2. Finalidade dos sistemas e principais problemas enfrentados pelos usuários.	Arquitetura do sistema.
18/03/2022	18/03/2022	3. Definição da introdução e objetivos principais do trabalho.	Conclusão da introdução
19/03/2022	19/03/2022	4. Definição de Requisitos arquiteturais, funcionais e não funcionais	Conclusão dos requisitos
20/03/2022	20/03/2022	5. Criação do diagrama de contexto C4 models	Conclusão do diagrama
23/03/2022	23/03/2022	6. Criação dos mecanismos arquiteturais	Conclusão dos mecanismos arquiteturais
29/03/2022	29/03/2022	7. Revisão da primeira parte do trabalho	Trabalho revisado
10/04/2022	10/04/2022	8. Gravação do vídeo de apresentação	Vídeo gravado
13/04/2022	13/04/2022	9. Redefinição dos requisitos funcionais	Requisitos redefinidos
23/03/2022	23/03/2022	10. Criação dos mecanismos arquiteturais	Conclusão dos mecanismos arquiteturais
29/03/2022	29/03/2022	11. Revisão da primeira parte do trabalho	Trabalho revisado
17/05/2022	17/05/2022_	12. Criação do diagrama de Componente.	Diagrama de Componente criado.
24/05/2022	24/05/2022	13. Criação do diagrama de código da aplicação.	Diagrama de código da aplicação criada.
02/05/2022	02/05/2022	14. Definição do layout das telas	Layouts definidos.
10 /06 /2022	10 /06 /2022	15. Criando o banco de dados da aplicação	Banco de dados
14/06 /2022	18 /06 /2022	16. Implementando as classes da aplicação	BackEnd
22/07 /2022	30 /07 /2022	17. Adicionando o framework Spring Batch	Framework de manipulação de dados
08/09 /2022	09 /10 /2022	18. Finalização da parte 3	Parte 3 finalizada

### 3. Especificação Arquitetural da solução

Apresentaremos nessa seção o esboço inicial da arquitetura das funcionalidades a serem desenvolvidas, incluindo diagramas, restrições e requisitos definidos pelos autores, que permitem visualizar a macroarquitetura da solução.

#### 3.1 Restrições Arquiteturais

Requisitos arquiteturais são requisitos que de alguma forma afetam a arquitetura do sistema, podendo ser funcionais ou não funcionais. É função do arquiteto definir os requisitos arquiteturais dentre todos os requisitos levantados do sistema, a lista a seguir apresenta os requisitos arquiteturais que foram identificados para implementação inicial.

R1: O Back-end deve ser escrito em Java 1.8 usando o framework Spring Boot
R2: O Front-end deve ser escrito em angular 8 usando o framework Primeng
R3: O extensão do arquivo que será importado deve ser .xlsx
R4: As APIs devem seguir o padrão ReSTful
R5: Os dados da aplicação devem ser armazenados no Postgresql

**Obs:** acrescente quantas linhas forem necessárias.

#### 3.2 Requisitos Funcionais

Os requisitos funcionais descrevem as funcionalidades do sistema, a lista a seguir apresenta os requisitos funcionais identificados para o desenvolvimento inicial do sistema.

ID	Descrição Resumida	Dificuldade (B/M/A)*	Prioridade (B/M/A)*
RF01	O sistema deve permitir a importação de uma planilha no formato .xlsx	A	A
FR02	O sistema deve validar o template da planilha	A	A
RF03	O sistema deve permitir a listagem dos arquivos importados	B	M
RF04	O sistema deve permitir a exclusão de um arquivo importado	M	M
RF05	O sistema deve permitir a edição de um arquivo importado	M	M
RF06	O sistema deve validar a integridade dos dados da planilha	A	A
RF07	O sistema deve realizar a validação do login do usuário	A	A
RF08	O sistema deve realizar o cadastro de usuário	M	A
RF09	O sistema deve realizar a listagem de todos os usuários	M	M
RF10	O sistema deve realizar a edição do usuário	M	A

RF11	O sistema deve realizar a exclusão do usuário	M	A
RF12	O sistema deve realizar o cadastro de perfil	M	M
RF13	O sistema deve realizar a listagem de todos os perfis	M	M
RF14	O sistema deve realizar a edição do perfil	M	M
RF15	O sistema deve realizar a exclusão do perfil	M	M
RF16	O sistema deve realizar a emissão da cotação da apólice	A	A
RF17	O sistema deve realizar a exclusão da emissão da apólice	M	A
RF18	O sistema deve realizar a geração de PDF da apólice	A	M
RF19	O tamanho máximo da planilha importada é de 40MB	M	A
RF20	O sistema deve realizar o cadastro do cliente	M	A
RF21	O sistema deve realizar a listagem dos clientes	M	M
RF22	O sistema deve realizar a edição do cliente	M	M
RF23	O sistema deve realizar a exclusão do cliente	M	M
RF24	O sistema deve enviar o email com as cotações para o cliente	A	A
RF25	O sistema deve calcular o prêmio da apólice	A	A

\*B=Baixa, M=Média, A=Alta.

**Obs:** acrescente quantas linhas forem necessárias.

### 3.3 Requisitos Não-funcionais

Os requisitos não funcionais são aqueles que não interferem diretamente nas principais funcionalidades do sistema, eles descrevem aspectos relacionados a desempenho, usabilidade, confiabilidade, segurança e etc. A lista a seguir apresenta os requisitos não funcionais identificados.

ID	Descrição	Prioridade B/M/A
RNF01	O sistema deve ser assíncrono no importação dos dados da planilha	A
RNF02	O sistema deve funcionar no navegador Chrome na 99.0.4844.74 e superiores	A
RNF03	O sistema deve ser responsivo	M
RF04	O sistema deve permanecer disponível 5 X 13, das 07:00h às 20:00h de segunda a sexta	A
RF05	O tamanho da fonte deve facilitar a leitura para o usuário	B
RF06	As cores devem seguir padrões para que o usuário consiga identificar os elementos visuais	B

**Obs:** acrescente quantas linhas forem necessárias.

### 3.4 Mecanismos Arquiteturais

Esta seção apresenta os mecanismos arquiteturais do sistema que são conceitos técnicos aliados para uma solução, eles se baseiam em três estados: análise, design e implementação. Mecanismos de Análise são identificados no início do projeto e representam pontos para futuro desenvolvimento de software, em design definimos padrões tecnológicos baseados no mecanismo de análise, implementação é o desenvolvimento real do que foi definido nos dois mecanismos anteriores.

Análise	Design	Implementação
Front end	Single Page Application	Angular 8
Front end	Estilização	Primeng
Back end	Java	JDK 1.8
Back end	Java	Spring Boot
Back end	Java	Spring Batch
Persistência	Banco de dados relacional	Postgresql
Persistência	ORM	Hibernate
Persistência	ORM	JPA
TDD	Teste unitário	JUnit 4
Container	Virtualização	Docker
Controle de versão	Git	GitHub
Distribuição	Integração contínua	Jenkins
Plataforma Cloud	Provedor de serviço	AWS

**Obs:** acrescente quantas linhas forem necessárias.

## 4. Modelagem Arquitetural

Apresentamos nessa seção a modelagem arquitetural do sistema, permitindo assim um total entendimento visando à implementação da prova de conceito.

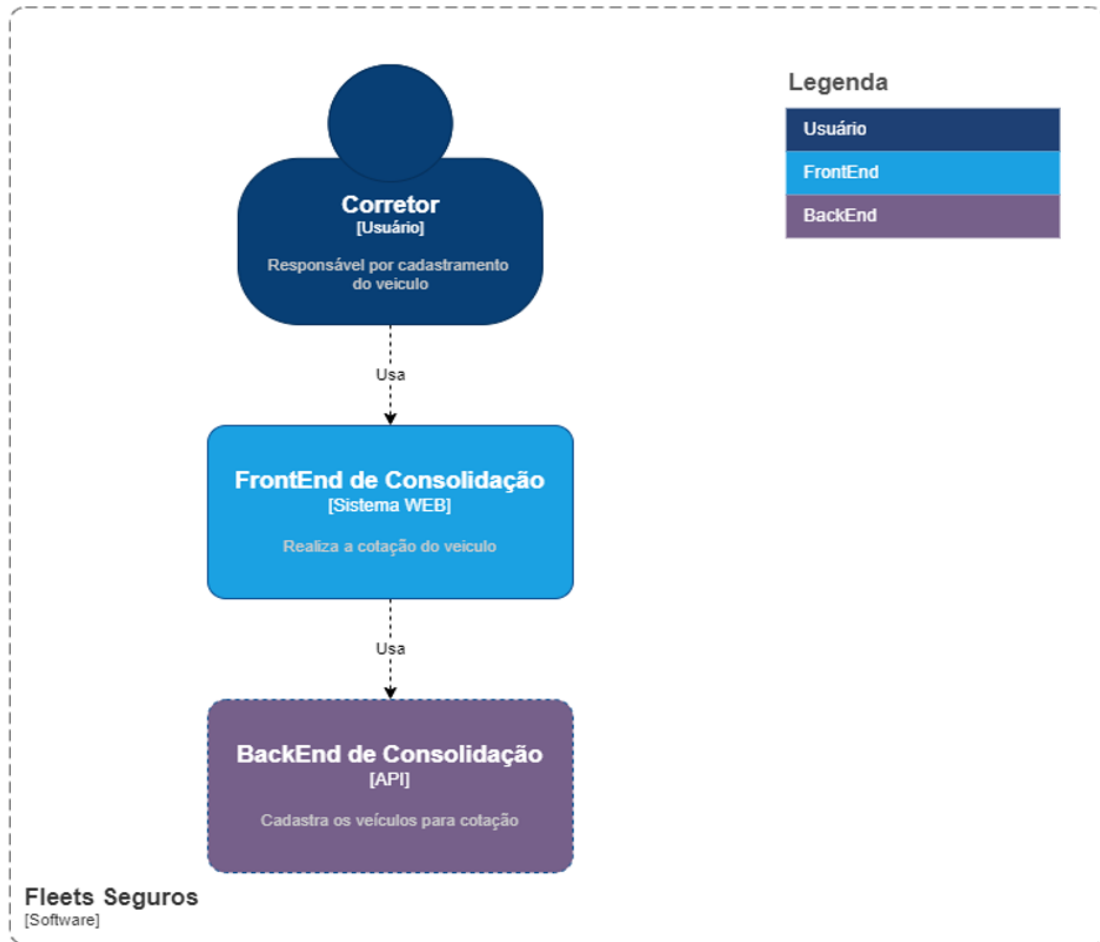
Optamos por utilizar o modelo C4 para a documentação de arquitetura de software desta modelagem arquitetural, serão apresentados aqui três dos quatro níveis que compõem o modelo C4, são eles: diagrama de contexto, o diagrama de container e o diagrama de componentes.

### 4.1 Diagrama de Contexto

Apresentamos na figura abaixo o diagrama de contexto da solução proposta, neste diagrama é possível observar que os corretores usam o Fleet Seguros para realizar cotações de seguros veiculares e cadastros necessários. O corretor irá interagir com o front-end que por sua vez envia dados para o back-end que processa e armazena as



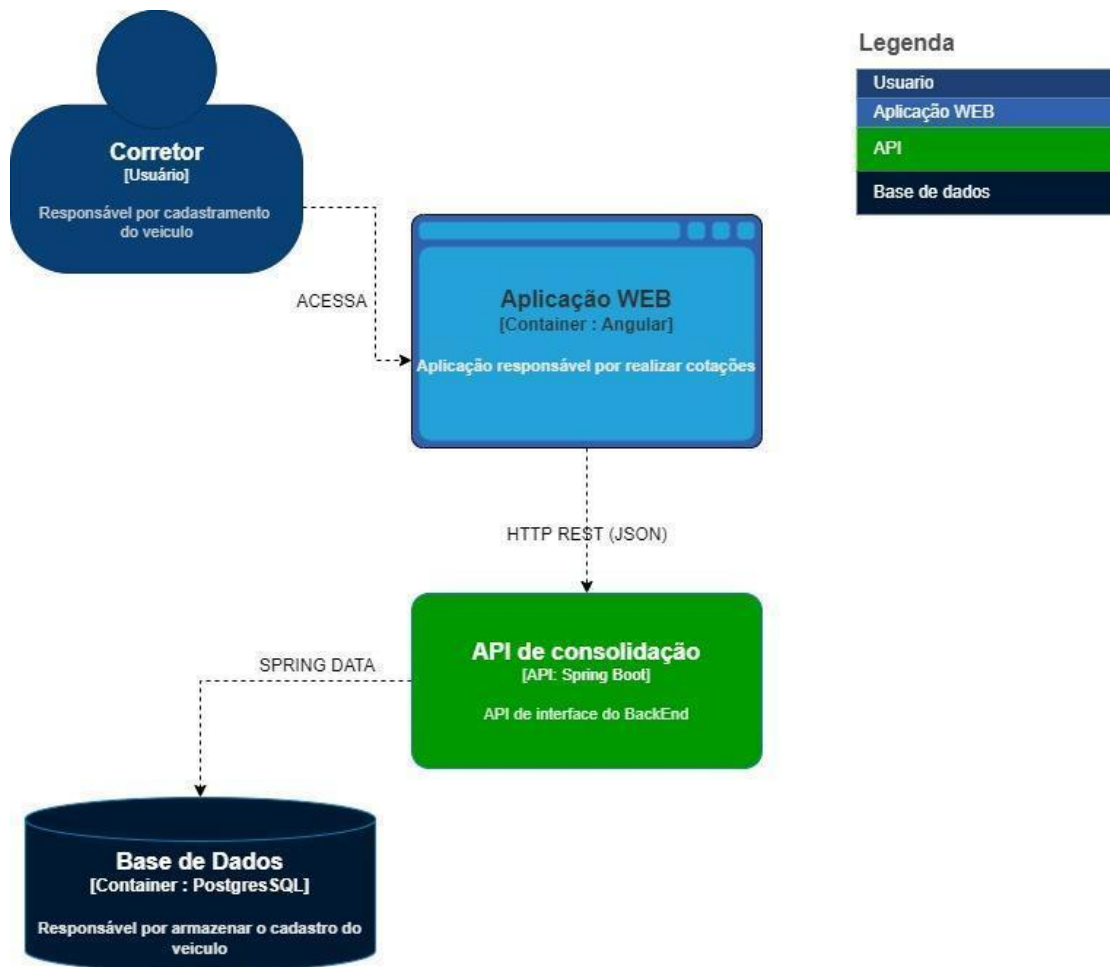
informações. O código de cores no diagrama distingue o usuário (caixa azul escuro) do front-end (caixa azul clara) e do back-end (caixa roxa).



**Figura 1 - Visão Geral da Solução.** Fonte: <https://www.infoq.com/br/articles/C4-architecture-model/>

## 4.2 Diagrama de Container

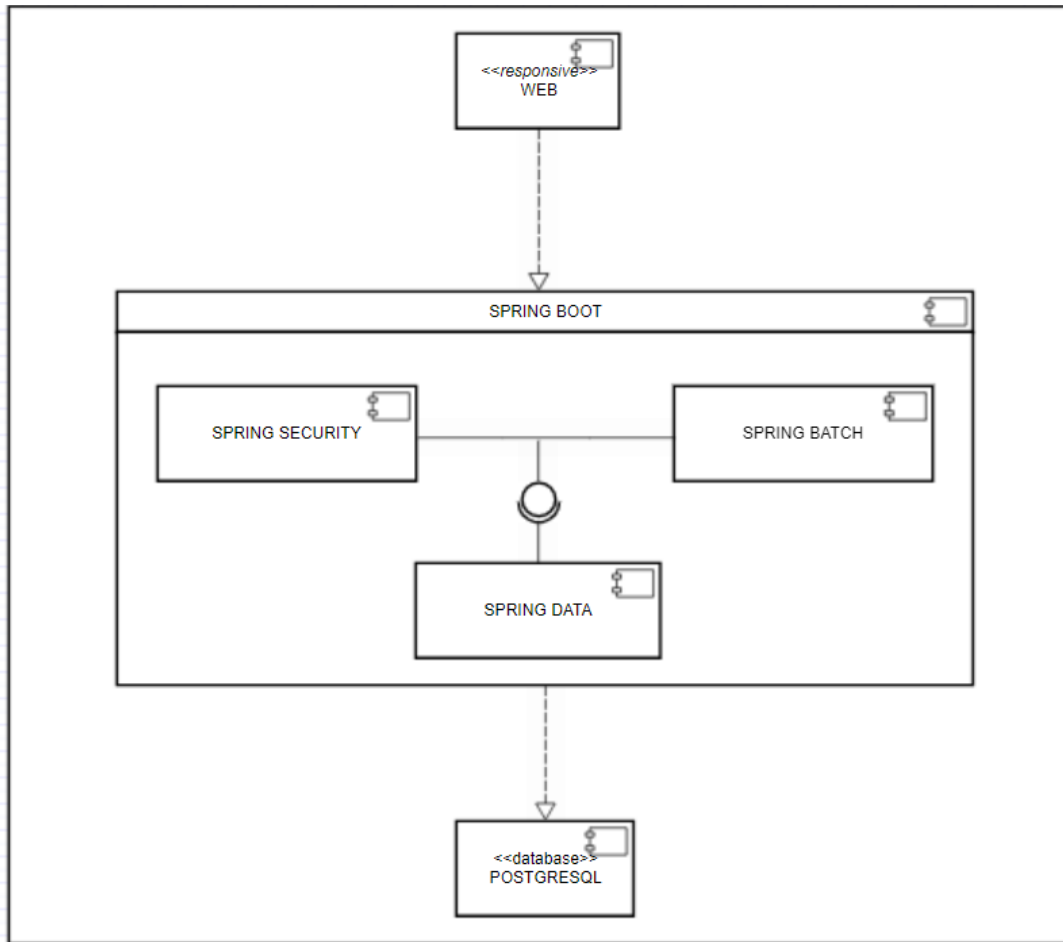
Apresentamos na figura abaixo o diagrama de Container da solução proposta, neste diagrama é possível observar como os componentes dos sistema estão distribuídos e organizados. Podemos observar que o corretor acessa a aplicação WEB e são realizadas chamadas do tipo REST para o back-end da aplicação que acessa o banco de dados para inserir ou recuperar informações.



**Figura 2 – Diagrama de container.** Fonte: <https://www.infoq.com/br/articles/C4-architecture-model/>

### 4.3 Diagrama de Componentes

Observamos na figura abaixo o diagrama de Componentes da aplicação, podemos observar que foi utilizado o padrão MVC para a construção do software, tudo começa com a interação do usuário no front-end da aplicação (camada View), uma requisição do tipo REST chega no Controller é transmitida para o Service que é responsável por aplicar as regras de negócio, em seguida conectando ao banco de dados onde realizamos a gestão das informações. Todas as requisições transmitidas a partir do front-end possuem um token de segurança e em todos os Controllers é realizada a autenticação de acesso.



*Figura 3 – Diagrama de Componentes.*

## **5. Prova de Conceito (PoC)**

Será apresentado como prova de conceito arquitetural três funcionalidades desenvolvidas ao longo desta disciplina: o cadastro de usuários do sistema e também o cadastro automático e manual de cotações. Foi desenvolvido um protótipo navegável onde podemos apresentar as telas dos requisitos citados acima e será detalhada também a forma de comunicação entre os componentes criados.

## 5.1 Integrações entre Componentes

Foi desenvolvido um protótipo navegável do sistema onde o usuário pode interagir nos três requisitos prioritários selecionados, o Fleets foi dividido em duas partes, front-end que possui as telas da aplicação (desenvolvido em Angular 8) e Back-end (desenvolvido em Java 8) onde estão todas as regras de negócio do sistema e acesso ao banco de dados. A comunicação entre essas duas partes se dá através de APIs REST onde existe um método HTTP específico para cada chamada feita ao servidor, no Header (Cabeçalho) de cada requisição enviamos um token de segurança que o usuário recebe ao acessar corretamente o sistema, com isso todas as chamadas realizadas são validadas no Back-end e só são processadas se possuírem um token válido.

## 5.2 Código da Aplicação

Temos abaixo o padrão arquitetural C4 da estrutura de código do Fleets. Observamos que as chamadas realizadas no Front-end chegam numa classe do tipo Controller que é responsável pelo processamento das requisições e por gerar respostas, ela funciona como uma camada intermediária entre a camada de apresentação (View) e a camada de negócios (Model). A classe Controller se comunica com uma classe do tipo Service responsável pela lógica de negócio da sua aplicação, além de se comunicar com as camadas de dados do software (Repository).

A estrutura da aplicação mostrada na Figura 4 apresenta os componentes de código e suas funções no software implementado, o repositório do código do Front-end do Fleets pode ser visualizado no link:

<https://github.com/thiagoferreiraaguilar/fleets/tree/master/fontes/frontend-fleets>

E o repositório do código fonte do Back-end pode ser visto no link:

<https://github.com/rickfer8/fleets/tree/main/fontes/backend-fleets>

- Listagem de cotações

Exibimos abaixo a tela que lista as cotações cadastradas, as cotações sempre são exibidas do maior número para o menor.

Número	Placa	Chassi	Cod. Fipe	Marca/Modelo	Cidade/Estado
8	BNZ-1489	1SGAAS656XEMZ2029	023174-6	Nissan/VERSA Exclusive 1.6 16V Flex Aut	Belo Horizonte/MG
7	XCG-7643	7MUMPADA2BSPL6320	023179-7	Nissan/KICKS Sense 1.6 16V Flex Mec	Belo Horizonte/MG
6	LKM-1456	78G0Y5BEEMXMU9065	029055-6	Volvo/S80 T-6 Executive 2.8	Belo Horizonte/MG
5	IQX-8646	4YTW5ZAAL3EAH7191	005345-7	VW/Gol (novo) 1.0 Mi Total Flex 8V 2p	Belo Horizonte/MG
4	EBT-1239	1UGDHYX2ANGTL0747	001488-5	Fiat/ARGO HGT 1.8 16V Flex Aut	Belo Horizonte/MG
3	GBD-0749	6CGZ6U2U8E48F2318	003281-6	Ford/Focus 1.6 S/SE/SE Plus Flex 8V/16V 5p	Belo Horizonte/MG

**Figura 4 – Tela Listagem de Cotações .**

## - Importação automática de cotações

Abaixo é exibida a tela de cadastro automático de cotações, o usuário seleciona a planilha com as cotações desejadas e o sistema as insere automaticamente, sendo então exibidas na tela de listagem.

**Figura 5 – Tela Cadastro Automático de Cotações .**

## - Importação manual de cotações

Abaixo é apresentado a tela de cadastro manual de cotações, a tela é dividida em três abas (Casco, Cobertura e LMI) e somente é possível cadastrar uma cotação por vez.

Home Acesso Cadastro Sair

**Cotação**

☐ Cadastro Automático ☒ Cadastro Manual

Casco Cobertura LMI

Vigência inicial Vigência final Placa

Chassi Código Fipe Marca

Modelo Ano de fabricação Ano do Modelo

Combustível Classe Bônus Tipo

Cadastrar

**Figura 6 – Tela de Cadastro Manual - Aba Casco .**

Home Acesso Cadastro Sair

**Cotação**

☐ Cadastro Automático ☒ Cadastro Manual

Casco Cobertura LMI

Uf Cidade Cobertura

24H Carro Reserva Cobertura de Vidros

Extensão Novo RCTR Claus Tipo franquia

Valor Franquia Comissão Danos Morais Terceiros

Cadastrar

**Figura 7 – Tela de Cadastro Manual - Aba Cobertura.**

Home Acesso Cadastro Sair

**Cotação**

☐ Cadastro Automático ☒ Cadastro Manual

Casco Cobertura LMI

LMI casco LMI acessório LMI equipamentos

LMI blindagem LMI kit gás LMI aparelhos

LMI app morte LMI danos morais LMI terceiros

Cadastrar

**Figura 8 – Tela de Cadastro Manual - Aba LMI.**

## 6. Avaliação da Arquitetura (ATAM)

Nesta seção abordaremos a avaliação arquitetural desenvolvida que atenda às solicitações do cliente. Utilizaremos os métodos ATAM para melhor entendimento.

### 6.1 Análise das abordagens arquiteturais

Apresentaremos aqui um resumo das características mais relevantes da proposta arquitetural e iremos aplicar o método Architecture Tradeoff Analysis Method (ATAM) onde detalharemos os cenários dessa análise.

Atributos de Qualidade	Cenários	Importância	Complexidade
Desempenho	Cenário 1: O sistema deve manter alto desempenho independente da quantidade de registros das planilhas de entrada.	A	A
Escalável	Cenário 2: A infra estrutura deve disponibilizar recursos dinâmicos.	A	B
Usabilidade	Cenário 3: O sistema deve prover boa usabilidade, facilitando assim a experiência dos usuários.	A	M
Segurança	Cenário 4: O sistema deve garantir a segurança dos dados do usuário e das empresas (LGPD)	M	A

### 6.2 Cenários

Nesta seção apresentaremos os cenários utilizados nos testes da aplicação que satisfazem os requisitos não funcionais do sistema.

*Cenário 1 - Desempenho: Ao importar os dados da planilha para o cadastro automático o sistema deve manter um desempenho satisfatório no processamento das informações.*

*Cenário 2 - Escalável: A infraestrutura do projeto deve possuir a capacidade de gerenciar os recursos de memória e processamento automaticamente conforme a demanda solicitada.*

*Cenário 3 - Usabilidade: O sistema deve possibilitar simplicidade na funcionalidade de cadastro automático, apresentando uma estrutura organizada para realização de uma cotação mais eficiente com menos riscos de erro humano.*

### 6.3 Evidências da Avaliação

Apresente as medidas registradas na coleta de dados. Para o que não for possível quantificar apresente uma justificativa baseada em evidências qualitativas que suportem o atendimento ao requisito não-funcional.

Atributo de Qualidade:	Desempenho
Requisito de Qualidade:	O desempenho deve ser satisfatório independente da quantidade de registros na planilha.
Preocupação:	
O sistema deve processar planilhas com quantidade de registros distintas	
Cenário(s):	
Cenário 1	
Ambiente:	
Sistema em operação normal	
Estímulo:	
O usuário importa a planilha e é enviada uma requisição REST para o processamento dos dados.	
Mecanismo:	
Criar um serviço REST para atender às requisições do cadastro automático.	
Medida de resposta:	
O sistema deve cadastrar os registros no banco de dados.	
Considerações sobre a arquitetura:	
Riscos:	O sistema não conseguir inserir os dados.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

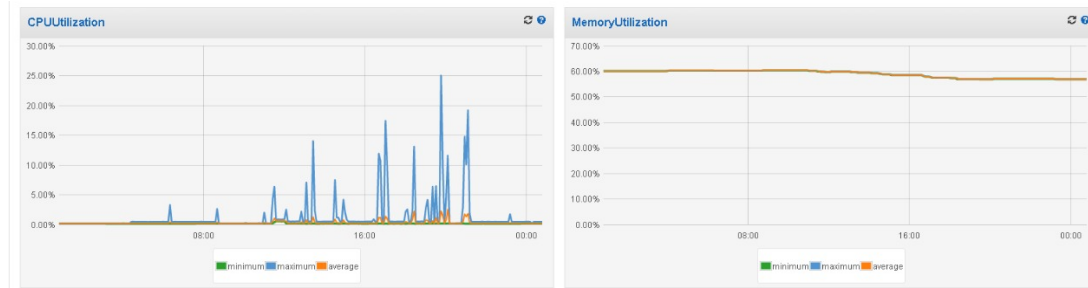


id	vigencia_inicial	vigencia_final	placa	chassi	codigo_fipe	marca	modelo	ano_fabricacao	ano_modelo	combustivel
1	2022-06-07	2022-06-07	ABX-8744	9BGRD08X04G117974	002154-7	Toyota	ETIOS X 1.3 Flex 16V 5p Aut	2022-06-07	2022-06-07	Gasolina
2	2022-06-07	2022-06-07	UYT-9876	7VTRA7KA07HW49452	003451-7	Ford	EcoSport STORM 2.0 4WD 16V Flex 5p Aut	2022-06-07	2022-06-07	Gasolina
3	2022-06-07	2022-06-07	GBD-0749	6CGZ6U2U8E48F2318	003281-6	Ford	Focus 1.6 S/SE/SE Plus Flex 8V/16V 5p	2022-06-07	2022-06-07	Gasolina
4	2022-06-07	2022-06-07	EBT-1239	1UGDHYX2ANGT0747	001488-5	Fiat	ARGO HGT 1.8 16V Flex Aut	2022-06-07	2022-06-07	Gasolina
5	2022-06-07	2022-06-07	IQX-8646	4YTW5ZAA13EAH7191	005345-7	VW	Gol (novo) 1.0 Mi Total Flex 8V 2p	2022-06-07	2022-06-07	Gasolina
6	2022-06-07	2022-06-07	LKM-1456	78G0Y58EEMXU9065	029055-6	Volvo	S80 T-6 Executive 2.8	2022-06-07	2022-06-07	Gasolina
7	2022-06-07	2022-06-07	XCG-7643	7MUMPAD28SPL6320	023179-7	Nissan	KICKS Sense 1.6 16V Flex Mec	2022-06-07	2022-06-07	Gasolina
8	2022-06-07	2022-06-07	BNZ-1489	1SGAA5656XEMZ2029	023174-6	Nissan	VERSA Exclusive 1.6 16V Flex Aut	2022-06-07	2022-06-07	Gasolina

**Figura 9 – Tabela de cotação do banco de dados.**

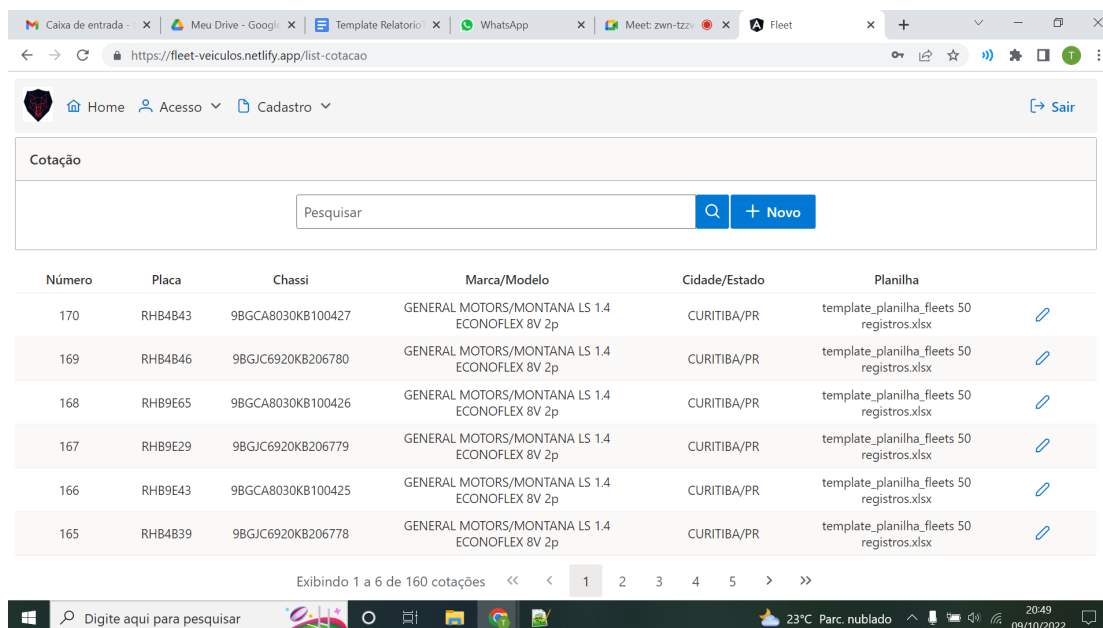
Atributo de Qualidade:	Escalável
Requisito de Qualidade:	O sistema deve continuar funcionando mesmo que sofra alteração no contexto em volume e tamanho para atender as necessidades do usuário.
Preocupação:	
O sistema não deve parar de funcionar por falta de memória ou processador.	
Cenário(s):	
Cenário 2	
Ambiente:	
Sistema em operação normal	
Estímulo:	
O usuário deve importar uma planilha com grande volume de dados.	
Mecanismo:	
Criar uma configuração para atender esse volume de dados	
Medida de resposta:	
O sistema não deve ficar lento ou permanecer offline.	
Considerações sobre a arquitetura:	

Riscos:	O sistema não conseguir processar os dados.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há



*Figura 10 – Gráfico de métricas de consumo (memória e processamento).*

Atributo de Qualidade:	Usabilidade
Requisito de Qualidade:	O sistema deve possuir funcionalidades simples e fáceis de usar.
Preocupação:	
O sistema não deve ter um layout confuso e de difícil manuseio.	
Cenário(s):	
Cenário 3	
Ambiente:	
Sistema em operação normal	
Estímulo:	
O usuário deve conseguir realizar o cadastro automático.	
Mecanismo:	
Utilizar componentes reutilizáveis e de simples interação.	
Medida de resposta:	
A tela deve ter mensagens de feedback das operações realizadas.	
Considerações sobre a arquitetura:	
Riscos:	O usuário não conseguir interagir com a tela.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há



**Figura 11 – Tela de listagem das cotações importadas.**

## 6.4 Resultados Obtidos

Apresente os resultados da arquitetura produzida, indicando seus pontos fortes e suas limitações. A título de sugestão construa uma tabela apresentando esses resultados, como no exemplo que segue:

Requisitos Não Funcionais	Teste	Homologação
RNF01: O sistema deve processar planilhas com mais de 5 MB	OK	OK
RNF02: O sistema deve ser utilizado em diferentes dispositivos.	OK	OK
RNF03: O sistema deve permanecer disponível 24 x 7.	OK	OK
RNF04: O sistema possui fonte legível.	OK	OK
RNF05: O sistema deve ser utilizável no Chrome.	OK	OK
RNF06: O sistema contém cores que facilitam os elementos visuais.	OK	OK

## 7. Avaliação Crítica dos Resultados

Apresente aqui, de forma resumida, os principais pontos positivos e negativos da arquitetura proposta. Adote uma postura crítica que permita entender as limitações arquiteturais, incluindo os prós e contras das tecnologias. Você pode utilizar o formato textual ou produzir um quadro resumo.

Ponto avaliado	Descrição
Processamento Custo	Ao utilizarmos uma planilha com volume grande de dados o ambiente não suportou processamento e reiniciou, foi preciso aumentar as configurações de memória e processamento aumentando assim o custo do projeto.
Tempo de Execução	Ao utilizarmos uma planilha com volume grande de dados o tempo total para a execução dos dados foi superior a 10 minutos, foi preciso desenvolver uma estratégia para essa execução não travar a tela e o usuário conseguir continuar usando o sistema.
Desacoplamento	- O projeto foi desenvolvido desacoplando o frontend do backend obtendo assim uma fácil atualização, otimização do código fonte aumentando a escalabilidade.

## **8. Conclusão**

Durante o teste da aplicação observamos que algumas melhorias poderiam ser feitas para aprimorar o processo de importação dos dados das planilhas, quando vários usuários realizam upload do arquivo, com grande volume de dados, o sistema fica lento. Podemos resolver esse problema utilizando um serviço de mensageria como o Kafka para orquestrar as solicitações enviadas para o backend.

Durante o desenvolvimento do projeto observamos que trabalhar com processamento de dados não é uma tarefa simples, algumas questões não foram planejadas durante a discussão da arquitetura do software como custo da infraestrutura da nuvem, otimização do processamento de dados e o consumo de memória.

Em alguns estudos que realizamos descobrimos que existem frameworks específicos para trabalhar com processamento de dados, podemos citar como uma melhoria futura a mudança da linguagem do backend de Java para o Python, pois o mesmo é uma linguagem mais apropriada para trabalhar com grande volumes de dados.

## 9. Referências

AGUIAR, Thiago, RIBEIRO, Ricardo. **Projeto Integrado relatório tecnico Fleet Seguros**. Belo Horizonte, 2022. Disponível em:  
<https://github.com/rickfer8/fleets/blob/main/Apresenta%C3%A7%C3%A3o%20do%20Trabalho%20Fleets%201%20parte.mp4>

BOSA, Gabriel. **Supervalorização de carros usados faz seguro subir até 20%**. São Paulo, 2022. Disponível em:  
<https://jovempan.com.br/noticias/economia/supervalorizacao-de-carros-usados-faz-seguro-subir-ate-20.html>. Acesso em: 05 de Março de 2022.

BROWN, Simon. **O modelo C4 de documentação para Arquitetura de Software**. 01 de agosto de 2018. Disponível em:  
<https://www.infoq.com/articles/C4-architecture-model/>. Acesso em: 01 de Março de 2021.

MALAR, João Pedro. **Inflação global é impulsionada por alta demanda e problemas em cadeias produtivas**. São Paulo, CNN Brasil Business, 2021. Disponível em:  
<https://www.cnnbrasil.com.br/business/inflacao-global-e-impulsionada-por-alta-demanda-e-problemas-em-cadeias-produtivas/>. Acesso em: 15 de Fevereiro de 2022.

OLIVEIRA, Gabriel. **O que está acontecendo com o preço dos carros no Brasil**. Vitória, Agazeta, 2021. Disponível em:  
<https://www.agazeta.com.br/colunas/gabriel-de-oliveira/o-que-esta-acontecendo-com-o-preco-dos-carros-no-brasil-1021>. Acesso em: 18 de Fevereiro de 2022.

SOARES, Rebeca. **Cenário para locadoras de veículos é positivo no segundo semestre**. São Paulo, Estadão, 2021. Disponível em:  
<https://investidor.estadao.com.br/mercado/acoes-locadoras-veiculos-2021>. Acesso em: 02 de Janeiro de 2022.